



---

## **REEL Demo – Romande Energie ELectric network in local balance Demonstrator**

Deliverable: 3d1 Algorithms and models for the  
prediction of the day ahead energy demand of  
households and the distribution grid (24hours)

Demo site: Chapelle

---

Developed by

Lorenzo Nespoli, ISAAC, SUPSI

Vasco Medici, ISAAC, SUPSI

Roman Rudel, ISAAC, SUPSI

[Canobbio, 17.01.2019]

# 1 Description of deliverable and goal

## 1.1 Executive summary

Distributed control techniques that are used to coordinate a set of agents electrical loads or batteries rely on the forecasts of each agents electrical consumption/production. In order to counteract fast change of power induced by renewable distributed energy resources (DERs) we are interested in coordinating agents with a timestep in the range of minutes. Moreover, since both residential (and industrial) power consumption and DERs power production profiles have a daily seasonality, a 24 hours ahead planning is typically used. The high number of timesteps, the frequency at which the problem must be solved, the number of agents to be coordinated (in the range of hundreds) and the limited computational power of the devices on which the distributed control problem is solved (smart meters), require a careful selection of both optimization strategy and forecasting algorithm. In order to reduce the computational burden we have investigated the impact of considering non-uniform stepsize in model predictive control (MPC) algorithms, by means of randomly generated power profiles. The solution based on the reduced number of steps achieve a similar performance in terms of objective function, when compared to the solution achieved with the uniform timestep discretization. We have then compared the performances of different multi-step-ahead forecasters for the composite residential power measurements, up to 24 hours. We did this by means of synthetic datasets representative of swiss residential power production and consumption. Secondly, we have investigated the influence of modeling PV power plants on the forecast accuracy. Even if the modeling does increase prediction accuracy (estimating power production from meteorological conditions), this does not always results in an increase of forecasted generated power, which depends on the accuracy of the numerical weather prediction forecasts. At last, we proposed a distributed algorithm to perform hierarchical reconciliation of power forecasts, which can be used to preserve privacy of the users while obtaining sum-consistent forecasts.

## 1.2 Research question

In section 1.4.1 we investigate the possibility of using non-uniform stepsize for the control algorithm of a battery with a multi-objective optimization (minimizing cost while performing peak shaving), which would result in a reduction of computational time for both forecasting and control. Assessing accuracy in forecasting requires a representative dataset of the signals that we want to predict. Datasets of energy consumption data at a good resolution and for a high number of households are scarce. Moreover the location of the houses is often unavailable, in order to preserve privacy. In order to test various forecasters on a representative dataset consisting in power consumption for uncontrolled devices, heat pumps, electric boilers and PV power production, we generated synthetic data by means of thermo-electric simula-

tions of households. In section 4.1 we present a detailed description of the python simulation environment, along with heating system control logic, that we have used to generate the residential net power profiles. The latter are then used in section 1.5 to investigate the accuracy of various forecasters. All the selected forecasters produces conditional probability distributions, which are compared in cross validation by means of quantile skill score. We have additionally compared them by means of required computational time. As a growing number of roof-mounted PV system is being installed, in section 1.5.2 we investigate the effect of exploiting physics-based PV models in order to forecast their power production. Forecasting PV power is of great importance for the future electrical grid, as more accurate power predictions allows to better handle abrupt rump up in regional power flow, due to change in cloud configuration, and ultimately permit to increase the number of PV plants which can be hosted in the distribution grid.

The last part of the report is dedicated to hierarchical forecasting. When forecasting power consumption in the electrical grid, we are typically interested in different level of aggregation. For example, when we aim at controlling an HVAC system of a residential building for minimizing future expenses, we are interested in single household level forecasts. On the other hand, when we want to provide a peak-shaving service to a local DSO, we are most interested in the forecast of the power consumption at the medium voltage transformer level. The most complex case is the one in which we would like to use demand side management to control a group of prosumers and sell their flexibility to a third party. In this case we are interested in system-level objectives (e.g. peak shaving of the power flow at the LV/MV transformer), and in respecting grid constraints. Since distribution grid topology is usually radial, they can be described using a rooted tree. We can therefore say that we would like to have accurate prediction at all the levels of the grid hierarchy, while assuring consistency among forecasts. This means that forecasts at the bottom level of the hierarchy should sum up to the forecast of the aggregate, which is not guaranteed when forecasting them separately. Moreover, single house electrical consumption is usually difficult to predict, due to the high variability of the signal. Overall forecast accuracy could be increased performing a forecast of the aggregated signal and using hierarchical forecasting techniques to reconcile the measurements. In section 1.5.6 we introduce a distributed algorithm for hierarchical forecasting, which can be used to reconcile household forecasts while preserving privacy of the prosumers (since forecast data is available at upper level of the hierarchy only by means of aggregated profiles) that was tested using real data of 2200 households coming from a UK smart meter trial.

### **1.3 Novelty of the proposed solutions compared to the state-of-art**

Many research in the past decades focus on the reduction of MPC computational time, mainly due to the need of performing real-time control on embedded platforms. Historically, two methods has become widely adopted to speed up MPC com-

putation. The first one relies on the approach used in linear quadratic regulators, which is based on the Bellman's principle of optimality. Basically, in an equality constrained linear quadratic dynamic problem, the objective function at time  $t_2$  can be reformulated as a function of the system state at time  $t_1$  only. This, together with the fact that the optimal at the end of the control horizon can be approximated, allows one to find an optimal control policy by backward recursion solving a Riccati equation, and then apply it sequentially to the system's state. This techniques can be modified to include exogenous inputs, for example allowing for the solution of tracking problems. Moreover, Riccati recursion is able to exploit peculiar properties of the problem's Hessian deriving from the regularity of the condensed form of the dynamic matrices [1, 2], allowing to achieve a computation cost which is linear in the number of time-steps [1], namely  $\mathcal{O}(N(n_x + n_u)^3)$ , where  $n_x$  and  $n_u$  are the number of states and control variables, while  $N$  is the length of the control horizon. However, when dealing with inequality constraints, this approach must be coupled with [3] other methods such as the Mehrotra predictor-corrector interior-point algorithm [4], losing part of the computational advantage. The second method which has gained a lot of popularity for real time control is the explicit MPC formulation, which relies on multiparametric programming [5]. The explicit formulation allows to partition the state space in terms of feedback control laws, that is, the optimal control policy can be provided in in terms of look-up tables, depending on the state of the system. This is convenient as long as the state space has a low number of partitions. Unfortunately this number increases fast with the number of states and control horizon, so that this method can become practically infeasible. Furthermore, explicit MPC cannot handle cases in which system matrices, cost function or constraints are time-dependent [6]. Since we want to keep the formulation general we investigate the possibility of reducing the computational burden by means of considering non-uniform step-size control problems, in which disturbances (the predicted power production/generation) are mediated over increasingly large time-steps. Indeed, a battery minimizing energy cost at an household's main must solve a problem with time-varying exogenous variables (prices and forecasted power), inequality constraints and discontinuous price function (the slope of objective function with respect to the exchanged power at the household's main changes while the prosumer switches his role from consumer to seller). These requirements lead us to consider a very general formulation. Demand side management algorithms use forecasts of the overall power consumed or generated at the point of common coupling with the grid in order to properly plan an optimal control signal. They typically need up to 24 hours ahead forecasts, since residential power profiles have a strong daily seasonality. Forecasting a signal several steps ahead in the future poses technical and conceptual challenges, depending on the adopted technique. Some simple methods, like state space methods, ARMAX and exponential smoothing, are usually used to perform multi-step ahead forecasting through a recursive technique, in which the 1 step ahead forecast is used as an input to the same model to predict

the next step ahead [7]. Although this procedure is very simple, since it does not change the model at each time step, it can be prone to instabilities: since errors tend to accumulate and cannot be compensated, the recursive technique strongly depends on the quality of future exogenous inputs, that are usually forecasted using some other methods. Another possible way to apply a general purpose univariate interpolators to multivariate regression is to perform some sort of embedding of the time step ahead one is trying to predict, and pass it as an additional input to the interpolator. The so called direct way is to fit  $T$  different interpolators where  $T$  is the number of steps ahead to be predicted, keeping the set of regressors fixed, while changing the target variable to be the  $t_{th}$  step ahead target. The latter one can show better accuracy when compared to multivariate regression, also known as multiple input multiple output (MIMO) regression, but it lacks of interdependency modeling between time-steps, which is needed in the case we are interested in producing scenarios for stochastic control out of the predictions [8]. Anyway, the interdependency structure can always be superimposed later, through time dependent copulas, for example with the dual-ensemble copula-coupling approach [9]. A comparison of several multi-step ahead strategies for forecasting, applied to very simple univariate time series of the NN5 competition [10], using non-probabilistic methods, is presented in [11]. Since we are interested in forecasters (which have to be periodically retrained) which can run on embedded chips with limited computational resources, we are also interested in comparing training times. Moreover, we are interested in forecasters which are able to retrieve the conditional probability distribution on the whole horizon. Therefore, we proposed two different methods which compare favorably in terms of training time with respect to quantile random forests, namely an double seasonality Holt-Winter exploiting exogenous inputs by means of linear de-trending, and an ensemble of extreme learning machines.

## 1.4 Description

### 1.4.1 Multi-step-ahead forecast for control with non-uniform step size

Systems with an internal buffer, such as batteries, boilers and the thermodynamics systems composed by an heatpump and a building, can increase their performance when shifting from a reactive control to a predictive one. Furthermore, when the dynamics of the controlled system can be modeled accurately, we can use the system model to generate prediction on the system state into the future, as it's done in Model Predictive Control (MPC). In any case, the predictive control requires a multi-step-ahead forecast of the disturbances when the typical rate of change of the system rate due to control actions and disturbances is meaningful effect we get from controlling the system, has a typical time scale which is smaller than the rate of change of the system's state (e.g. the state of charge of a battery) when subject to a typical control action. Typically, is more difficult to predict disturbances at several steps ahead into the future than predicting a single step forward. Fortunately, when dealing

with the aforementioned system, we can to some extent disregard to use the same time-step when forecasting distance events into the future, since these events have a lower impact on the optimal control action we must apply to the system at the present time, with respect to the disturbance of the next timestep. This means that, instead of predicting one day ahead with 1 minute timesteps, we can instead predict averaged values of the system's disturbance on wider timesteps during the end of the day. A similar approach is used in reinforcement learning, where a discount factor is applied to the reward of future actions.

The second advantage of having a variable stepsize in the forecasting horizon is the reduction of computational time needed for the training and prediction phase of the forecasters and for the solution of the control problem. Even when considering an optimistic computational time for the optimization problem of  $\mathcal{O}(n)$ , where  $n$  is the number of steps, passing from an horizon of 1440 steps (one minute intervals) to one of 14 steps reduces the computational time by a factor of 100.

#### 1.4.2 Estimation of sub-optimality of non uniform step-size

In order to estimate the loss in the objective function of a typical receding-horizon optimal control problem, we compared the solution of two different control problems. The first one is obtained with a fixed 10 minutes step, while the second uses logarithmically spaced steps. We limit the investigation to the case in which a residential user would like to controlling a single state electrical battery to minimize its electricity costs and perform peak shaving. We consider the battery's control law to be the solution of the following optimization problem:

$$u^* = \underset{x \in \mathcal{X}, u \in \mathcal{U}}{\operatorname{argmin}} \alpha \|\hat{P} + Su\|_2 + \sum_{t=1}^T c(\hat{P} + Su) \quad (1)$$

where  $x \in \mathbb{R}^T$  and  $u \in \mathbb{R}^{2T}$  are the state of charge of the battery and the vector of control actions (charging and discharging power) of the battery,  $S \in \mathbb{R}^{T \times 2T}$  is a summation matrix, summing charging and discharging battery operations with appropriate signs,  $\mathcal{X}$  and  $\mathcal{U}$  are the constraints sets containing the battery dynamics and operational limits,  $\hat{P} \in \mathbb{R}^T$  is the forecasted uncontrolled power at the electrical main of the user, and  $c(\cdot)$  is the energy cost function defined as:

$$c(z_t) = \begin{cases} p_{b,t}z_t, & \text{if } z_t \geq 0 \\ p_{s,t}z_t, & \text{otherwise} \end{cases} \quad (2)$$

where  $p_{b,t}$  and  $p_{s,t}$  are the buying and selling tariffs, respectively, at time  $t$ .

We generated 100 power profiles using detrended random walks whose variance was drawn from a Gaussian distribution. The random walks are obtained as:

$$x_{i,t+1} = x_{i,t} + w_{i,t}\sigma_i \quad (3)$$

where  $x_{i,t}$  is the  $i_{th}$  random walk at the  $t$  timestep,  $w_{i,t} \sim \mathcal{N}(1,1)$  and  $\sigma_i \sim \mathcal{N}(1,1)$  are random variables drawn from the normal distribution. We generate 3-days long scenarios, with an initial timestep of 10 minutes, resulting in 100 scenarios of length 432 steps. The signal was then detrended using a moving average with a window of 36 steps, equal to 6 hours. In Fig.1 a sample of 10 random walks and their detrended signals are shown.

We used logarithmically spaced bins for the reduction of the control horizon. The desired number of control steps was fixed a-priori equal to 15. We then found  $k$  such that:

$$\begin{cases} \text{step}_{i+1} = k \text{ step}_i & \forall i \\ \sum_{i=1}^{15} \text{step}_i = 144 \end{cases} \quad (4)$$

The resulting step lengths were also used to obtain the final synthetic forecasted signals, through bin averages. We then solved the problem in a receding horizon fashion, where only the solution of the optimal control action of the first time step is actually applied. This means that for each scenario we solved 2-days long receding horizon problems, which means that for each scenarios we solved 288 optimization problems, retrieving an equal number of optimal control actions  $u_{i,t}^*, t \in [1, 288]$ .

In Fig. 2 the estimated density function of the relative change in the objective function (1) when solving it with variable length rather than fixed time-steps is shown. Half of the observed relative changes lie in the interval  $[-1.4e - 3, 2.3e - 4]$  while 90% of them lie in the interval  $[-7.7e - 3, 1.6e - 2]$ . As can be seen for Fig. 2, the maximum relative deviation (represented by three outliers) is in the range of 10%. Fig. 3 shows the boxplots of the scenario-average computational time obtained following the two strategies on a IntelCorei7 – 5500UCPU@2.4GHz. The median computational time for the variable step-size strategy was  $1.4e^{-2}$  seconds, while for the constant step-size strategy was  $1.2e^{-1}$  seconds.

## 1.5 Evaluation of multi step ahead forecasters for net power prediction

Forecasting multi step-ahead power profiles for residential households is an extremely complex task, mainly due to the high dimensionality of the multivariate output, the even higher dimensionality of the inputs, which make us incur in curse of dimensionality [12] related problems, the volatility of the target and its low signal to noise ratio. The last two points doom any point forecasting method to return large errors when compared to the task of forecasting more predictable time series (like the power flow at a medium voltage transformer). Furthermore, a decision maker or a control algorithm is, generally speaking, not interested in point forecasts, but rather in probabilistic ones, that is, the a-posteriori probability distribution of the target given the present conditions. As such, we only focused on methods able to cope with those aspects, and evaluated them also based on the predicted quantile of the conditional distributions. For 3 of them we directly predict time aggregates of the target variable, as motivated in subsection 1.4.2, in the attempt of mitigating the

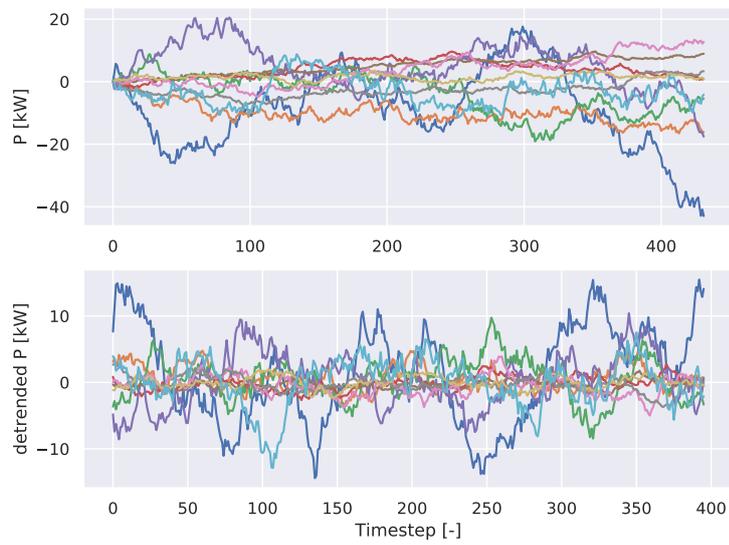


Figure 1: 10 out of the 100 generated profiles. Upper: original random walks. Lower: detrended profiles

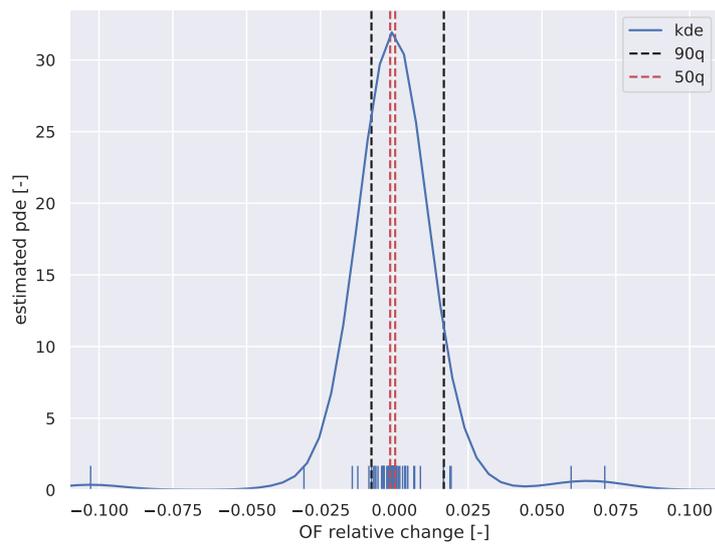


Figure 2: Estimated probability density function of the relative change in the objective of the receding-horizon control

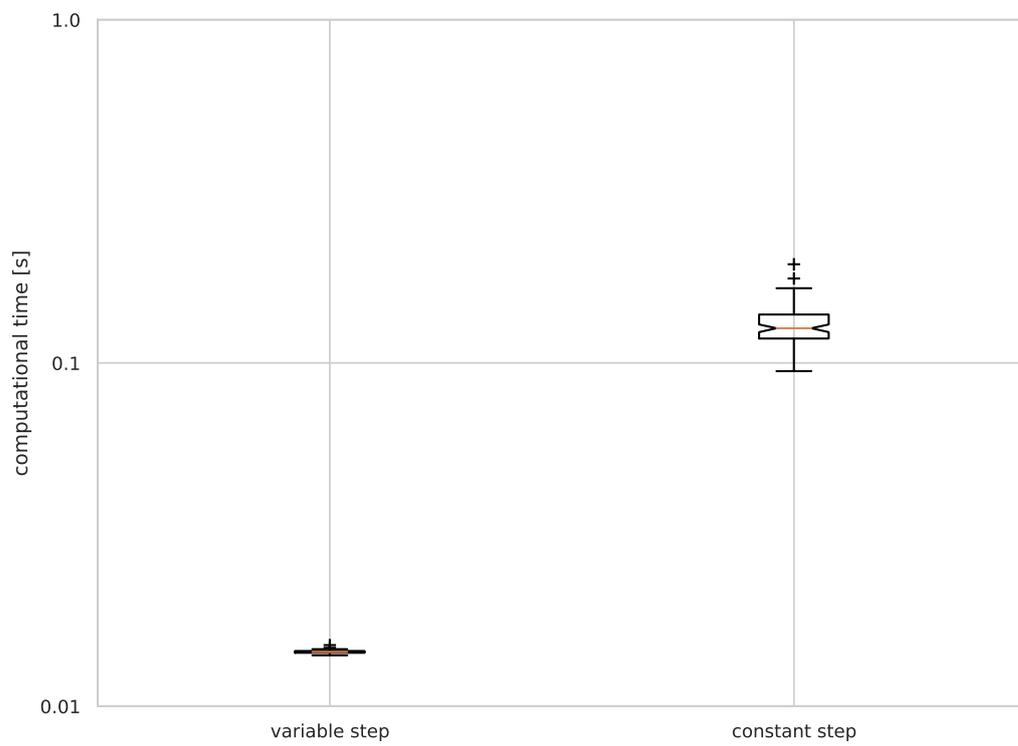


Figure 3: Boxplots of the computational time for the variable stepsize resolution method and for the constant step method. Each point in the boxplot refers to the mean computational time for the  $i_{th}$  scenario, for each of which problem (1) is solved 288 times.

difficulties of predicting an output with high dimensionality.

In the following we describe the four methods we employed for this analysis. Each method belongs to a different class of multi-step ahead methodology introduced before.

**Quantile regression forest and recursive quantile regression forest** Random forests are basically an ensemble of (high variance and low bias) decision trees, trained with random subsets (independently chosen) of the explanatory variables and samples of the original training dataset. It is known that aggregating multiple forecasters leads to variance reduction of the prediction, when base learners (the to-be aggregated forecasters) are trained using random samples of the dataset. This technique is known as bagging or bootstrap aggregation [13]. Each tree is trained following a greedy strategy, choosing which variable to split and at which level in a sequential way, minimizing a loss function (usually squared error). In order to get the quantiles out of a random forest, one could simply estimate the quantiles of the prediction of the  $N$  trees, assuming each tree has been independently optimized. Anyway, a better alternative exists, which consists of keeping track of all the data in the trees' leaves, and estimating empirical quantiles out of them. This algorithm is known as quantile regression forest (QRF) [14]. The first method based on QRF uses a direct approach, which uses past variables up to time  $t - T$  to predict the next  $T$  outputs, namely:

$$\hat{y}_{t+k|t} = f(x_{[t-T,t]}) \quad \forall k \in [1, T] \quad (5)$$

where  $x_{[t-T,t]}$  is the set of explanatory variables from timestep  $t - T$  up to  $t$ ,  $\hat{y}_{t+k|t}$  is the forecasted variable at time  $t + k$  having knowledge of  $X$  up to time  $t$ . Formally, given a datasets with  $N$  observations, we obtain the Hankel matrix of targets  $Y \in \mathbb{R}^{(N-T) \setminus T \times T}$ , where  $\setminus$  stands for integer division, and the Hankel matrix of the explanatory variables,  $X \in \mathbb{R}^{(N-T) \setminus T \times n_x \times T}$ , where  $n_x$  is the number of explanatory variables. Verbosely,  $X$  and  $Y$  are the following:

$$X = \begin{bmatrix} x_{1,t-T} & x_{1,t-T+1} & \dots & x_{1,t} & x_{2,t-T} & x_{2,t-T+1} & \dots & x_{n_x,t} \\ & & & \dots & & & & \\ x_{1,t-T+1} & x_{1,t-T+2} & \dots & x_{1,t+1} & x_{2,t-T+1} & x_{2,t-T+2} & \dots & x_{n_x,t+1} \\ x_{1,N-2T} & x_{1,N-2T+1} & \dots & x_{1,N-T} & x_{2,N-2T} & x_{2,N-2T+1} & \dots & x_{n_x,N-T} \end{bmatrix} \quad (6)$$

$$Y = \begin{bmatrix} y_{t+1} & y_{t+2} & \dots & y_{1,t+T} \\ & & & \dots \\ y_{N-T+1} & y_{N-T+2} & \dots & y_N \end{bmatrix} \quad (7)$$

where  $x_{1,t}$  stands for the first explanatory variable at time  $t$ . In order to ulteriorly reduce the dimensionality of the problem, we applied time aggregation, with bins symmetric to the time aggregation for the output. That is, referring to table 2, the first 31 (465 divided by 15) columns of matrix  $X$ , which are the farthest in time with respect to the one step ahead prediction, are averaged together, and so on for all

the columns. This reduces the size of  $X$  from  $n_x(N - T)$  to  $10(N - T)$ . This strategy assumes that, since we are requiring forecasts with a loss of temporal resolution as we move forward in time from  $t$ , we can as well get rid of the high temporal resolution of historical values, as we move backward from it.

The second method still relies on QRF, but uses a recursive-like logic to generate the  $t + k$  step ahead. The first step ahead is simply generated taking the full Henkel matrix  $X$  as regressors, that is, the one described by equation (6) and with 15 minutes sampling. The second step ahead is generated replacing the first column of  $X$  with the prediction of the first step ahead,  $\hat{y}_{t+1|t}$ . The procedure is iterated since 10th step ahead, for which all the columns but the last 465 have been removed by the original  $X$ . That is, the explanatory variables matrix for predicting  $\hat{y}_{t+T|t}$  will consist of 475 columns, the first 9 being the output of the QRF of the first 9 timesteps.

**Tree bagging Extreme learning machines** This method uses a MIMO regressor, the so called extreme learning machine (ELM) [15], to predict all the 10 step ahead at once. Since ELMs were originally thought as interpolators, we perform an ensemble through bagging in order to obtain the conditional distribution of the prediction. The ELM is a 2-layers perceptron, in which the first matrix of weights linking the inputs to the layers of neurons, and the neurons biases, are randomly initialized. As such, the only weights that need to be learned are the ones linking the neurons with the (multivariate) output. This can be done efficiently by means of linear regression exploiting the Moore-Penrose inverse. The main advantage of the ELM is that, since there is no need of training the first matrix of weights and biases, we can bypass backpropagation (gradient descent) or other types of iterative optimizers, which are usually needed due to the presence of the nonlinear activation function, and just retrieve the optimal values of the second matrix of weights. In practice, this method firstly augment the space of explanatory variables by applying random nonlinear transformations, and retrieves their optimal combination by linear regression. Since the dimensionality of the explanatory variables is highly increased, we can penalize the weights adding a Ridge punishment in order to regularize the output and decrease the forecaster variance. Formally, the ELM is described by:

$$\begin{aligned} Y^T &= \theta O + \epsilon \\ O &= \sigma(WX^T + b) \end{aligned} \tag{8}$$

where  $X$  and  $Y$  are the previously described Henkel matrices,  $\sigma$  is the activation function (which is usually a sigmoid),  $W \in \mathbb{R}^{(N-T) \setminus T \times n_n}$  and  $b \in \mathbb{R}^{n_n}$  are the randomly initialized matrix of weights and biases of the first neuron layer, where  $n_n$  is the number of neurons.  $O \in \mathbb{R}^{n_n \times (N-T) \setminus T}$  is the output of the first layer and  $\theta \in \mathbb{R}^{n_n \times T}$  is the matrix of final weights which connects the output of the first layer  $O$  to the matrix of targets  $Y$ . As in normal linear regression, assuming Gaussian noise  $\epsilon$  and

applying regularization, we retrieve the analytical solution for  $\theta$ :

$$\theta^* = (OO^T + \lambda I)^\dagger (OY) \quad (9)$$

where  $\lambda$  is the Ridge regularization parameter,  $I$  is the identity matrix of appropriate dimensions and  $\dagger$  stands for the pseudoinverse. Due to the presence of the sigmoid activation function, inputs must be normalized to be centered in the interval  $[-1, 1]$ . We normalized  $X$  to unit variance; informally:

$$X_n = \hat{\sigma}_s^{-1}(X - \hat{\mu}_s) \quad (10)$$

where  $\hat{\sigma}_s$  is the vector containing the estimated sample standard deviation of the columns of  $X$ , and  $\hat{\mu}_s$  is the vector of the estimated sample means of the columns of  $X$ . In order to speed up the activation function computation, we have used a piecewise linear function instead of the analytic sigmoid. Notice that in our case the nonlinearity reduces to an upper and lower threshold. For the tests we have adopted 400 neurons ELM, with  $\lambda$  equal to  $1e - 1$ . To estimate the a-posteriori quantile distribution and reduce the variance of the predictions, we have applied bagging to 100 ELMs. For each ELM we selected 70% of variables and 70% of the original observations, at random.

**Detrended multiple Holt-Winters** The Holt-Winters (HW) method [16] is a special class of the exponential smoothing [17], which consists of three smoothing equation, such that the final prediction is a combination of the level  $a$ , trend  $b$  and seasonality  $s$ . We tested different flavors of the HW families and based on performance we adopted a double seasonality additive HW:

$$\begin{aligned} \hat{y}_{t+h} &= (a_t + hb_t) + s_{1,t-p(1)+1+(h-1)\setminus p_1} + s_{2,t-p_2+1+(h-1)\setminus p_2} \\ a_t &= \alpha(y_t - s_{1,t-p_1} - s_{2,t-p_2}) + (1 - \alpha)(a_{t-1} + b_{t-1}) \\ b_t &= \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1} \\ s_{1,t} &= \gamma_1(y_t - a_t - s(2, t - p_2)) + (1 - \gamma_1)s_{1,t-p_1} \\ s_{2,t} &= \gamma_2(y_t - a_t - s(1, t - p_1)) + (1 - \gamma_2)s_{2,t-p_2} \end{aligned} \quad (11)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma_1$  and  $\gamma_2$  are parameters to be learned from data, while  $p_1 = 96$  and  $p_2 = 672$  are the periods of the seasonalities. The model (11), and HW in general, do not include exogenous inputs. Since quantities like external temperature and irradiance are important explanatory variables in load forecasting, we included them with an a-priori linear detrend, such that the new target is  $y = y - X\beta_d$ , where  $X$  is a three column matrix containing  $GHI$ ,  $T$  and the unit vector (for the intercept), and  $\beta_d$  is the vector of linear coefficients. Moreover, instead of keeping a single set of  $\alpha$ ,  $\beta$ ,  $\gamma_1$  and  $\gamma_2$ , we fitted 96 sets of these values, based on the step ahead. To identify them, we used synthetic generated power profiles, using the methodology described in section 4.1. Due to the linear detrend we applied to the target, the fitted  $\beta$  values

where close to 0 for all the steps ahead, and thus we decided to fix this parameter to 0. The identified parameters are shown in Fig. 4. For the HW the conditional probability distribution of the prediction was obtained a-priori, using data from the training set. That is, for each fold, we obtained the distribution of the error on the training set, based on the step ahead and the hour of the day:

$$\hat{q}_{\alpha_i, h, k} = \text{quantile}_{\alpha_i}(e_{h, k}) \quad (12)$$

where  $e_{h, sa}$  is the set of training errors related to the hour  $h$  and to the  $k_{th}$  step ahead and  $\alpha_i$  is the level of the quantile.

### 1.5.1 Results from synthetic dataset

We compared the performance of the four different forecasters in predicting the power demand/production of a group of 100 prosumers, at different levels of aggregation, using one year synthetic data generated as described in appendix 4.1. We used 15 minutes-sampled time series, but we did not predict 96 timesteps. Rather, we predicted 10 steps ahead with different levels of aggregations, using logarithmically spaced bins, as explained in section 1.4.2. The length of each bin is reported in Table 1.

Table 1: Number of minutes for each step ahead.

step	1	2	3	4	5	6	7	8	9	10
minutes	15	15	30	45	75	105	150	225	315	465

Each user power consumption is obtained combining 3 different appliances: an heat pump (HP), a roof-mounted PV and uncontrolled loads (UL). The simulated prosumers are composed as described in Table 2. The forecasters are evaluated in cross validation, using 6 folds of 2 months each. We adopt the same cross validation approach later described in section 1.5.4, in which the training set in each of the 6 folds is composed by groups of 3 consecutive days (see Fig. 6). As explanatory variables we use the historical values of the power, GHI and T. Additionally, we gave the regressor the perfect forecast of GHI and T. This was motivated firstly by the lack of NWP for the typical meteorological year we have used to generate the synthetic load, and secondly because we wanted to evaluate the best possible performance of the forecaster, without dealing with NWP accuracy. In fact, NWP services provide forecast with different accuracy, based on the presence of on-ground measurements used for prediction calibration. The evaluation metrics we used for the forecast-

Table 2: Number of houses per type of appliances.

PV+HP+UL	PV+UL	HP+UL	UL
20	20	10	60

ers are the root mean squared error (RMSE), the mean absolute error (MAE) and a

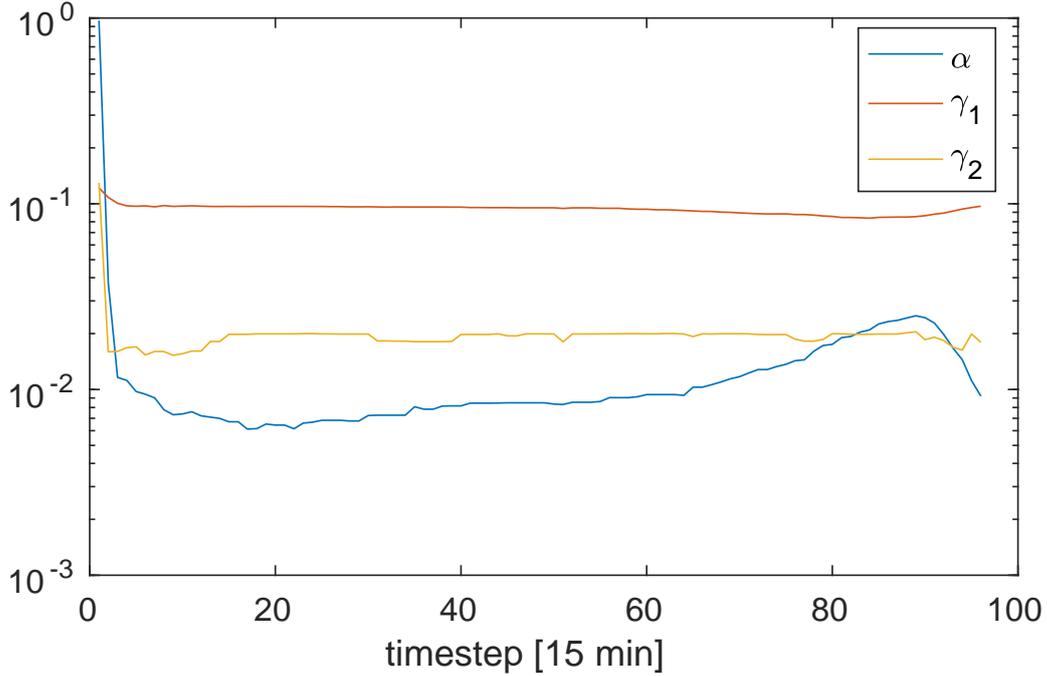


Figure 4: Fitted parameters for the HW model, as a function of the step ahead

quantile skill score  $S$  which compares the quality of the predicted quantiles of the a-posteriori probability distribution returned by the forecasters, with the empirical one, estimated on the test set. Formally,  $S$  is defined as:

$$S(\hat{y}_{t+k|t}, y_{t+k}) = - \sum_{i=1}^m \left( \mathbb{1}_{\{y_{t+k} \leq \hat{q}_{t+k|t}^{\alpha_i}\}} - \alpha_i \right) \left( y_{t+k} - \hat{q}_{t+k|t}^{\alpha_i} \right) \quad (13)$$

where  $\hat{q}_{t+k|t}^{\alpha_i}$  is the predicted  $\alpha_i$  level quantile for  $t+k$  at time  $t$  and  $\mathbb{1}_x$  is the indicator function on the condition  $x$ . The skill score (13) is a proper skill score [18], it is always positive, and lower values of  $S$  means higher quantile prediction accuracy. The results for the individual agents are shown in Fig. 5, by means of boxplots. Each boxplot contains the observations for the 100 agents, already mediated across the 6 folds. QRF and the RQRF clearly obtain better performance in all the indicators, with the exception of the one step ahead prediction, in which the HW is consistently the best forecaster. Besides the accuracy of the methods, we reports also the agent computational time for each forecasters, obtained on a Intel Core i7-4790K @ 4.00GHz with 32 GB of RAM. The values in the boxplots are, once again, already mediated across the folds, and refers to the training period on each fold, which was roughly 2 months. The median computational time for each forecaster is reported in table 3. While the QRF and RQRF achieve greater accuracy in the forecasts, they are hardly embeddable in a smart meter due to both the high number of parameters and the high computational time. On the other hand, both the ELM and the detrended HW only requires matrix inversions in the training phase, and simple algebraic multiplication for the test phase. While the ELM requires  $\mathcal{O}(n_n)$  parameters, where  $n_n$

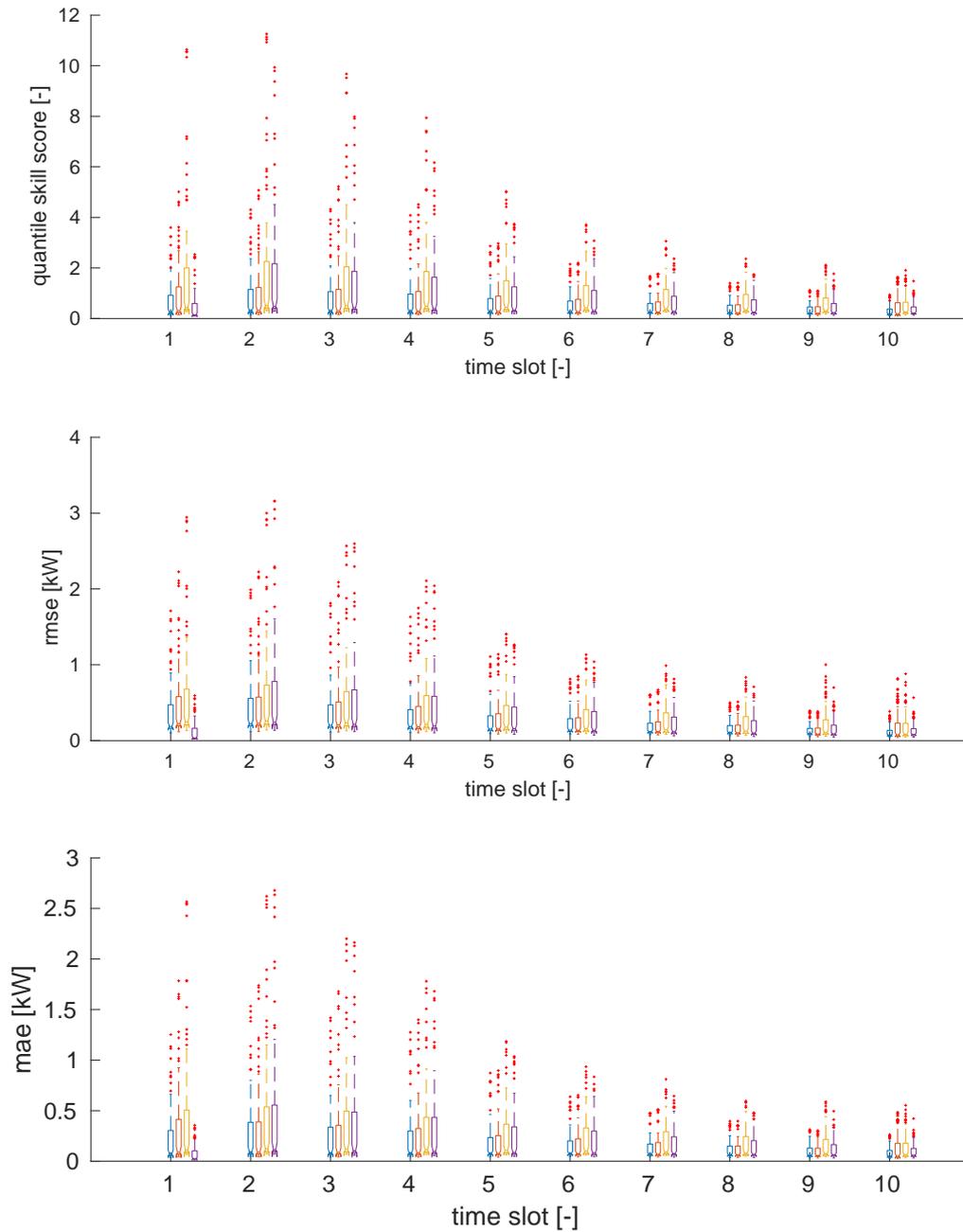


Figure 5: Evaluation of different regressors for multiple step ahead forecasting. Each boxplot contains 100 points, which are the results for each agent, mediated across the CV folds. Blue: QRF, direct. Red: RQRF, recursive. Yellow: bagging of ELM. Violet: detrended HW

is the number of neurons, the HW only requires  $3 \times 96$  parameters, which can be reduced to 12 if the parameters in figure 4 are approximated with piecewise linear functions. More importantly, the HW does not require to keep historical values for the training dataset, being an adaptive state-space method. This means that all the information needed for performing a forecast are stored in the model parameters. This is also true for the detrend of the exogenous variables when an adaptive linear fit is applied, as it was the case in this study. This means that the HW method can be easily embedded in a microcontroller.

Table 3: Agent median computational time for 2 months of data, for each forecaster.

forecaster	QRF	RQRF	ELM	HW
training time [s]	45.7	241.1	6.3	3.5

### 1.5.2 Influence of PV modeling on forecasting

Several studies have tried to include physics based model in the forecasting task [19], while a plethora of methods relying on econometrics and machine learning have been proposed to forecast PV using a data-driven approach [20], [21]. However in these studies the authors tend to focus on one of the two approaches, and as such is hard to assess the contribution of physics-based models to PV power forecasting.

We split the analysis in two, starting from estimating the effect of PV models in predicting PV AC power, given the global horizontal irradiance ( $GHI$ ) and ambient temperature  $T$  and the current time  $t$ . That is, we want to find a regressor

$$\hat{P}_{pv,t} = g(GHI_t, T_t, t) \quad (14)$$

which maps the aforementioned variables to the estimated PV production, and see if the reconstruction accuracy increases when considering a physics-based model for the PV. We then investigate the effect of the same models when trying to forecast the PV power for the next 24 hours, that is, we want to find a regressor

$$\hat{P}_{pv,[t,t+T]|t} = f\left(GHI_{[t-T,t]}, T_{[t-T,t]}, G\hat{H}I_{[t,t+T]|t}, \hat{T}_{[t,t+T]|t}\right) \quad (15)$$

where  $x_{[t,t+T]|t}$  means all the values of variable  $x$  from  $t$  up to  $t + T$ , given the information available at time  $t$ ,  $G\hat{H}I_{[t,t+T]|t}$  and  $\hat{T}_{[t,t+T]|t}$  being the forecasts at time  $t$  for irradiation and ambient temperature provided by a numerical weather prediction service, up to time  $t + T$ . The rationale behind this is that, if we possess accurate enough  $GHI$  and  $T$  forecasts for the next day, we can combine them with an estimated PV model to increase the forecast accuracy. For this task we use real data coming from 4 PV roof-mounted power plants located in Biel-Benken, Switzerland. The PV power plants are composed by differently oriented folds; their description and metadata can be found in [22].

### 1.5.3 Evaluation of PV modeling for prediction

Since we want to estimate the effect of a physics-based model on the accuracy of  $g$ , we chose a-priori a family of regressors and keep it fixed during the analysis. We chose to use a random forest regressor due to its ability in modeling nonlinearities, its high resiliency to overfitting and its low prediction variance [13]. We compared the performance of  $g$  against physics-based models, and then we try to increase its predictive power including the models as regressors in  $g$ . The overall estimated models are the following:

1. We just use the random forest to map  $GHI$  and  $T$  to the PV generated power. Instead of including time as a regressor, we use the solar azimuth instead.

$${}^1\hat{P}_{pv,t} = g(GHI_t, T_t, Az_t) \quad (16)$$

2. The PV power is estimated using only a physic based model, which is described by a linear regression making use of a vector of proxies for the power produced by the identified PV panel. Briefly speaking, each proxy is an esteem of the AC power produced by a typical PV panel with a given spatial orientation. We used 21 proxies, uniformly distributed on the unit sphere, disregarding the one facing north. The  $\theta$  coefficients linking proxies with the actual PV power production are then retrieved by means of a robust linear regression. An accurate description of this method can be found in [22]. The PV model is identified starting from AC PV power measurements and  $GHI$ .

$${}^2\hat{P}_{pv,t} = Pr_t \hat{\theta} \quad (17)$$

where  $Pr_t \in \mathbb{R}^{n_p}$  is the vector of proxies at time  $t$ , with  $n_p$  the number of proxies.

3. The PV power is still estimated using only a physic based model, but this time the model is identified blindly, based only on the aggregated power measurements at the main of a residential building. Firstly, we use a disaggregation technique to estimate the generated AC power production of the PV, which is then used to identify the PV model, as described in [23]

$${}^3\hat{P}_{pv,t} = Pr_t \hat{\theta}_{bl} \quad (18)$$

4. In the fourth model the PV power estimated with the second model is passed as a regressor to the random forest  $g$ , along with the same inputs used for the first model. The idea here is to investigate if the non-trivial nonlinear transformation which project  $GHI$  on the plane of array, which were obtained with a physics-based approach and which are contained in the matrix of proxy signals, can help the random forest in predicting  $P_{pv}$

$${}^4\hat{P}_{pv,t} = g\left(GHI_t, T_t, Az, {}^2\hat{P}_{pv,t}\right) \quad (19)$$

5. This model is equal to the previous one, but uses the prediction of model 3, which were obtained through the blind identification

$${}^1\hat{P}_{pv,t} = g\left(GHI_t, T_t, Az, {}^3\hat{P}_{pv,t}\right) \quad (20)$$

#### 1.5.4 Results

We evaluated the methods through k-fold cross validation on a dataset of increasing size. In particular, used 10 datasets, spanning from a minimum of 40 days up to a maximum of 400 days. For each of these datasets we estimated the performance of the methods using a 10-folds cross validation. The cross validation is done in the following way: each dataset is divided in 10 folds; for each fold we extract a training and a test set. The training set is obtained taking sequences of 3 days out of 4, for all the length of the fold. The remaining data constitutes the test set. This fold selection was done to exclude seasonality effects from the analysis. In fact, should be noticed that this methodology is more realistic compared to simply taking the first period of the data in each fold as the training test and the last part as the testing set, as the test sets can be as long as 100 days. This would lead to test the model on period of the years which are significantly different from the data seen by  $g$  in the training set. On the other hand the aforementioned methodology mimic the behavior of estimating a model for the PV power plant once each three days, which is reasonable. Fig. 6 shows as an example the division of the cross validation folds for the 80 days dataset.

As performance metrics we use the root mean squared error (RMSE) and the mean absolute error (MAE), without any normalization. The results, in forms of boxplot containing all the cross validation results for the four different PV power plants (that is, each boxplot contains 40 points), are shown in Fig.7 for RMSE and MAE respectively.

From the results, we can draw the following conclusions: first, the accuracy of the identification does not show significant changes with the dataset size, for datasets larger than 120 days, for all the methods but the blind identification  ${}^3\hat{P}_{pv}$ . Secondly,

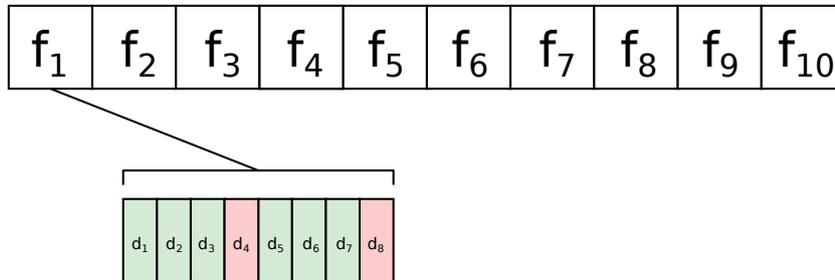


Figure 6: Example of 10 folds cross validation on the 80 days dataset. The dataset is divided in 10 folds, each of 8 days. For each fold, the training set (green) is composed by 3 consecutive days each 4 days, while the test set (red) is composed by the remaining days.

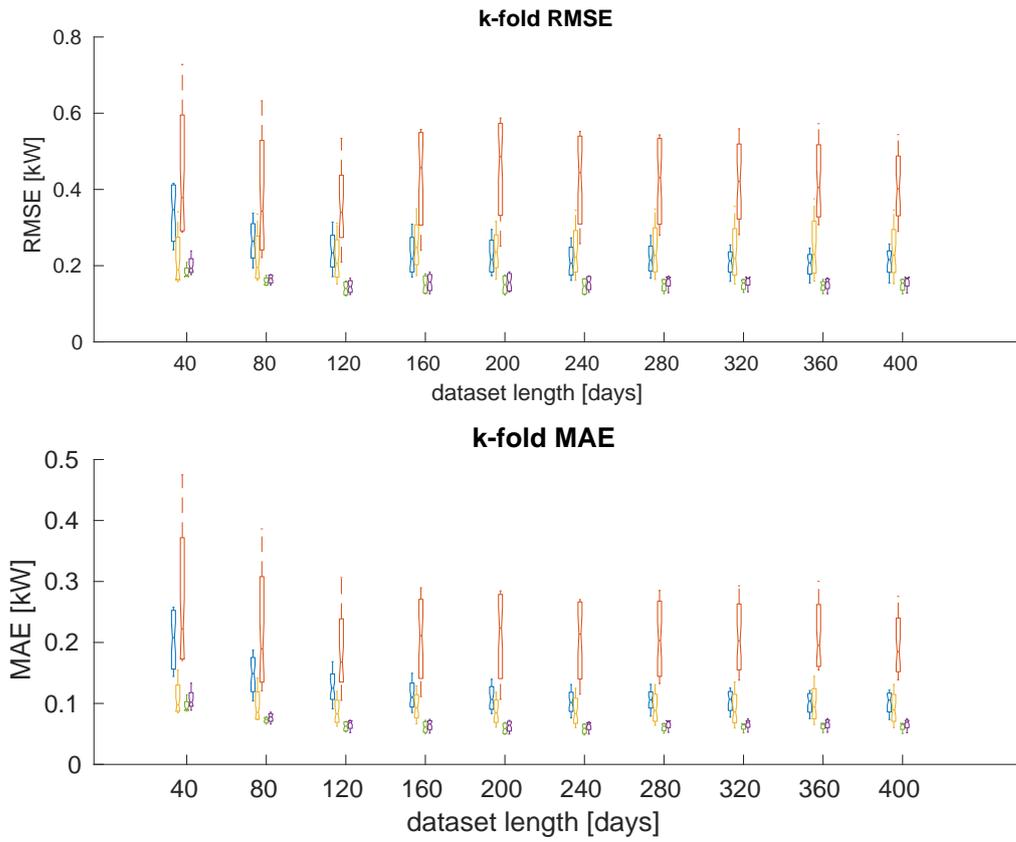


Figure 7: Boxplots of the RMSE and MAE based on the number of days of the dataset and on the method of . Blue:  $^1\hat{P}_{pv}$ , yellow:  $^2\hat{P}_{pv}$ , red:  $^3\hat{P}_{pv}$ , green:  $^4\hat{P}_{pv}$ , violet:  $^5\hat{P}_{pv}$ .

while the random forest estimation  ${}^1\hat{P}_{pv}$  has similar performance to the proxy based identification  ${}^2\hat{P}_{pv}$  in terms of RMSE, the latter is slightly better for all the dataset periods when considering the MAE. Last, the most accurate methods for estimating  $P_{pv}$  are clearly  ${}^4\hat{P}_{pv}$  and  ${}^5\hat{P}_{pv}$ , which combines the physics-based methods with the random forest regression. This could be explained by the fact that the proxies used for the physics based regressions include the projection of the GHI signal onto differently oriented planes. These projections are non-trivial, since they include the split of GHI in the diffuse and direct irradiance seen by these planes. At the same time, the physics behind these splits and projection is well-known. It seems reasonable that the random forest predictive accuracy increases when the information of these projections is (indirectly) included as a regressor.

In Fig. 8 the average values of the identified  $\theta$  coefficient, mediated across the folds, for  ${}^2\hat{P}_{pv}$  and  ${}^3\hat{P}_{pv}$  are shown, based on the total number of training days. Each line of the figure refers to one of the four households hosting the PV power plants. We can see that the  ${}^2\hat{P}_{pv}$  method, which only relies on the aggregated power profile, presents a dense pattern in the  $\theta$  value, meaning that the true orientation of the PV folds is not accurately identified. On the other hand, the  ${}^3\hat{P}_{pv}$  method, which makes use of values of measured  $GHI$ , present a sparse and consistent pattern in the values of  $\theta$  across the datasets, meaning that the dataset size is less relevant to the identified values.

### 1.5.5 Evaluation of PV modeling for forecasting

As mentioned previously, predicting the value of PV power output and forecasting PV production are two different task. As we have seen in the previous section, using a physics based model for PV increases the prediction accuracy when we try to estimate PV power production starting from known values of GHI, but whether this results in an increase of forecasting accuracy mainly depends on the NWP quality and resolution, as we argument in this section. Once again we used cross validation, this time using folds of 120 days, since no gain in accuracy was seen in 1.5.4 when selecting higher dataset size in PV power prediction. In order to estimate the influence of the PV modeling on forecasting, we evaluated the performance of a QRF, which was the forecaster with the higher accuracy among the one evaluated for the power forecast in the analysis presented in 1.5.1, on the Biel-Benken dataset. We used two variants of the  ${}^4\hat{P}_{pv,t}$  model (equation 19) in order to increase the accuracy of the forecast. Namely, in each fold we estimated a physics based model, using the proxy technique, as explained before. Secondly, with the same data, we trained a random forest regressor to learn the map from  $GHI$   $T$  and the solar azimuth to the power production. We then fed this model with the NWP forecast for  $GHI$  and  $T$ , and then we pass the result as an explanatory variable to the QRF. Formally, the

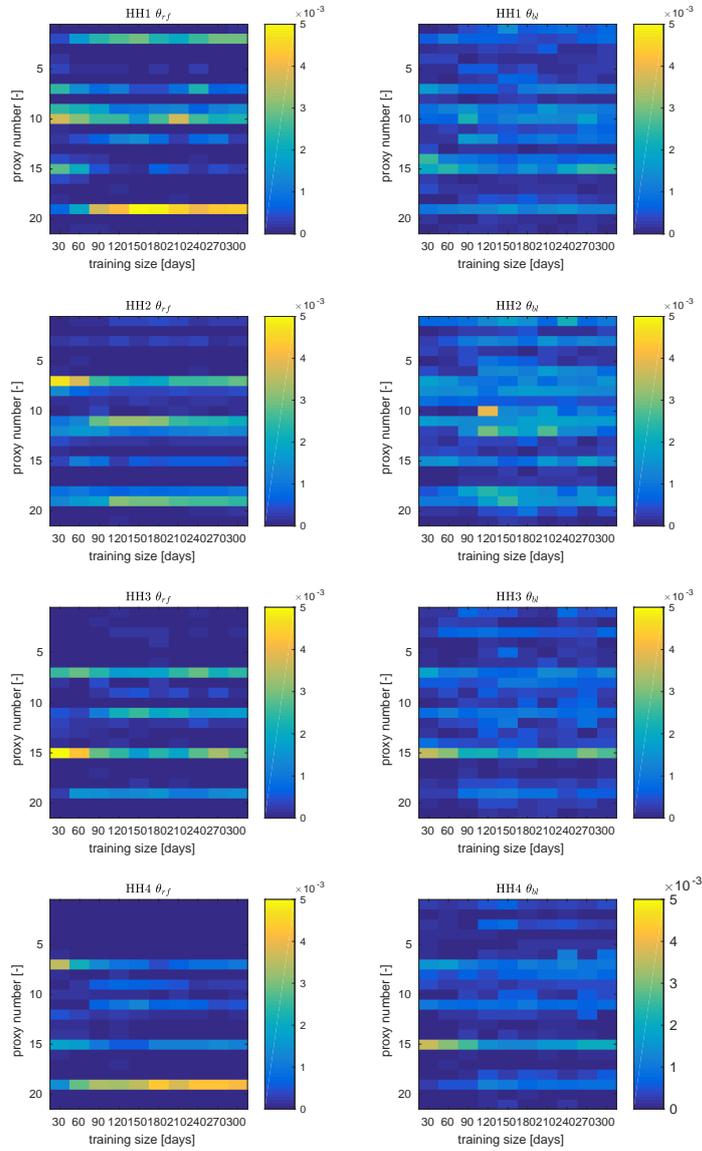


Figure 8: Values of the identified  $\theta$  coefficient for all the households, for the robust fit regression (left) and for the blind identification (right), mediated over the cross validation folds, based on the number of training days.

forecaster can be described as:

$$\begin{aligned} \hat{P}_{pv,[t,t+T]|t} &= f(X) \\ X &= \left[ GHI_{[t-T,t]|t}, T_{[t-T,t]|t}, {}^4\hat{P}_{pv,t} \left( \hat{\theta}, G\hat{H}I_{[t,t+T]|t}, \hat{T}_{[t,t+T]|t} \right) \right] \end{aligned} \quad (21)$$

where  ${}^4\hat{P}_{pv,t}$  is the best model for the PV power prediction 19. The difference from the prediction task is that now the model makes use of the NWP forecast  $G\hat{H}I_{[t,t+T]|t}$  and  $\hat{T}_{[t,t+T]|t}$ . We used MeteoBlue as NWP service, which provides local forecasts for  $GHI$  and  $T$  at hourly resolution for the next 36 hours. The forecasts are updated twice per day. As such the accuracy of the NWP forecasts is not constant with the step ahead, but is also dependent from the time of the day. Some authors use Kalman filters to reduce the forecast error, but this method need to retrieve a dynamic model for the error, which in this case has discontinuities (twice per day, at the moment of the update), thus it is hard to model with a simple autoregressive model. To perform the correction we decided once again to use a random forest which we fed with the perfect forecasts from historical values of  $GHI$  and  $T$ , and the hour of the day. We stress that the RFs for the NWP correction and the models for the PV prediction ( ${}^4\hat{P}_{pv,t}$ ) were trained and identified in each fold, in order to keep the results from different folds statistically independent.

In the second variant we assumed the lack of a pyranometer for the measurement of the local value of  $GHI$ . As such, we identified the PV model,  $\hat{\theta}$ , directly from the PV AC power measurements, using only observations from clear sky periods and exploiting robust regression. In this case, in order to correct the NWP forecasts, we reconstructed the local  $GHI$  seen by the PV panel starting from the identified model, as described in [22].

We used  $MAE$  and  $RMSE$  as KPIs, where the error is previously normalized using the mean of non-zero values for each step ahead, that is:

$$nRMSE_{sa} = \left[ \frac{1}{n} \sum_{t=1}^n \left( \frac{y_{t,sa} - \hat{y}_{t,sa}}{\bar{y}_{sa}} \right)^2 \right]^{\frac{1}{2}} \quad (22)$$

$$nMAE_{sa} = \frac{1}{n} \sum_{t=1}^n \frac{y_{t,sa} - \hat{y}_{t,sa}}{\bar{y}_{sa}} \quad (23)$$

where  $\bar{y}_{sa}$  is the mean of non zero values of  $y$  for the current fold, at the  $sa$  step ahead. Note that this normalization does not requires to re-weight for the different length of the step ahead bins.

We started assessing the effect of the estimated PV models using perfect forecasts for  $GHI$  and  $T$  in order to have a lower bound for our evaluations. In Fig. 9 and Fig. 10 are shown the boxplots containing the  $nRMSE$  and  $nMAE$  for all the households and all the folds, as a function of the step-ahead. The blue boxplots refers to the base case predictions, where no PV models are estimated, while the red and yellow boxplots refer to the first and second variants of PV model estimations. We can

see that the improvement in forecast accuracy due to the estimation of PV models are significant both in  $nRMSE$  and  $nMAE$ , especially for the most aggregated time-steps. Additionally, the effect of estimating  ${}^4\hat{P}_{pv,t}$  without knowing  $GHI$  is marginal. We additionally investigate the loss of accuracy due to the resolution of the forecasts: since the NWP are available with a sampling time of one hour, we are interested in the loss of accuracy when the perfect forecasts are downsampled using the same resolution. In Fig. 11 the  $nRMSE$  for the base case forecast, in which no PV models are used, are shown. The blue boxplots refer to the perfect forecasts, while the green and bordeaux refer to the 1 hour downsampled perfect forecasts and to the real forecasts, respectively. We can see that in the first timesteps, the distribution of the  $nRMSE$  of the real forecasts is close to the distribution of  $nRMSE$  of the downsampled perfect forecasts. This means that there is little or no bias in the corrected NWP forecasts. Despite this, PV modeling does not significantly affects the forecasts accuracy in any step ahead, as can be seen in Fig. 12, where the  $nRMSE$  boxplots are shown in the case in which the corrected NWP forecasts for  $GHI$  and  $T$  are used. This can be explained additionally considering the effect of PV modeling when using the 1 hour downsampled perfect forecasts, as shown in Fig. 13. The effect of modeling PV when using the 1 hour sampling time resolution is negligible for the first 3 step ahead, while increasing for the last steps ahead. We can conclude that the NWP forecasts accuracy and temporal resolution for  $GHI$  and  $T$  are not accurate enough to induce a decrease in the forecast error when using a PV model. Anyway, we can see from Fig. 11 that the accuracy of the real forecasts is already close to the 1 hour mediated perfect forecasts. This means that the (anyway modest) increase of accuracy that can be seen in Fig. 9 for the first step ahead is mainly due to the perfect knowledge of  $GHI$ . It seems not reasonable that the  $NWP$  accuracy can be improved for the first step ahead. This is mainly due to the fact that  $GHI$  signal has a typically high variance during overcast days. Any low-variance forecaster which uses a quadratic loss will tend to smooth out the high frequency components of the signal to be predicted, since under least squares error minimization criterion the conditional expected value of the signal is the best minimizer. On the other hand, we can see from Fig. 11 that PV modeling decreases the error starting from the 7<sup>th</sup> step ahead (which corresponds to 2 hours ahead) up to 24 hours ahead, even when using 1 hour sampling time. At the same time we can see from Fig 11 that the last step ahead NWP forecast accuracy is distant from the perfect forecasts downsampled signal. This means that if the NWP forecasts accuracy increases for the last steps ahead, this will results in an increase of accuracy in PV power prediction when using PV modeling.

At last, we repeated the same analysis using only clear day samples. The clear days are identified as the 10% of days which shows the lowest error between the NWP

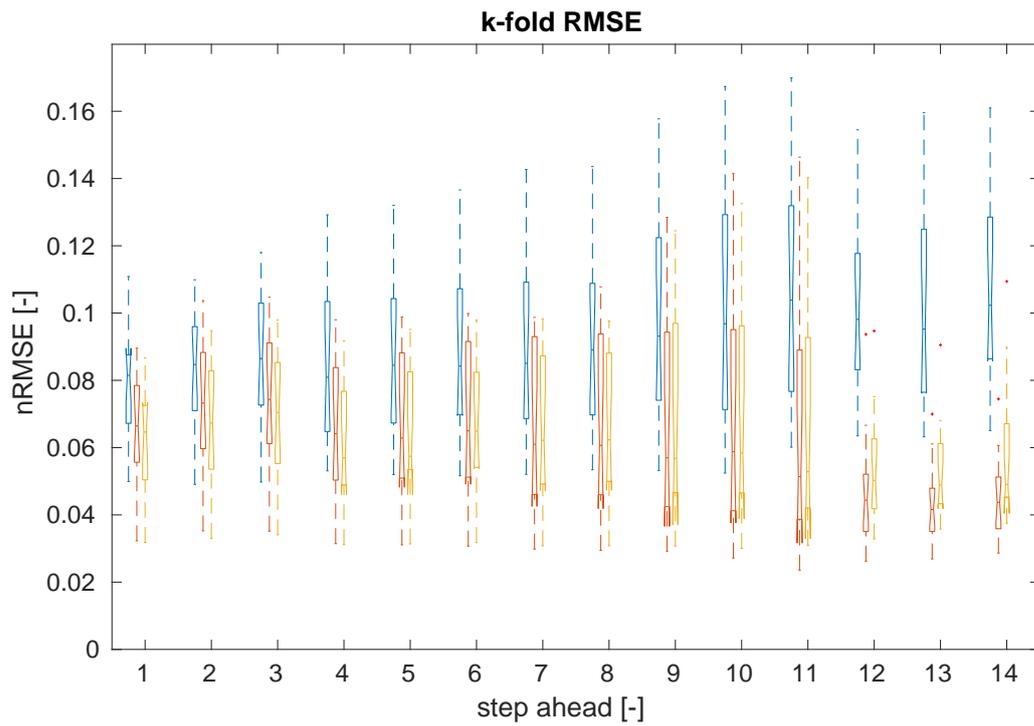


Figure 9: nRMSE as a function of step ahead for perfect forecasts. Blue: base case. Red: with PV model. Yellow: with PV model estimated without *GHI*.

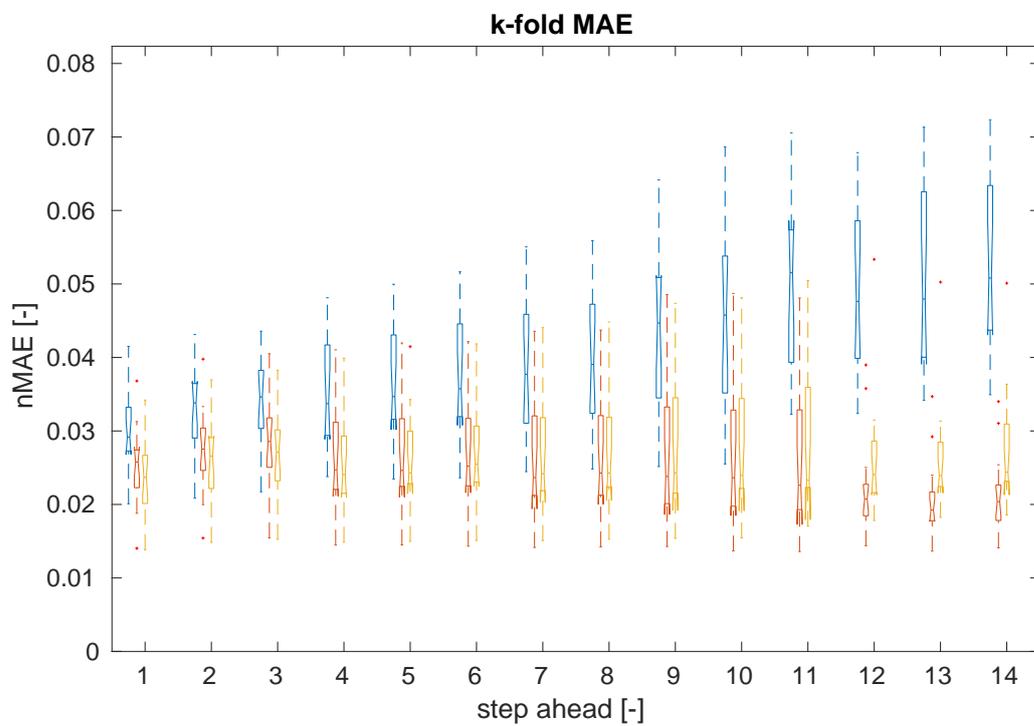


Figure 10: nMAE as a function of step ahead for perfect forecasts. Blue: base case. Red: with PV model. Yellow: with PV model estimated without *GHI*.

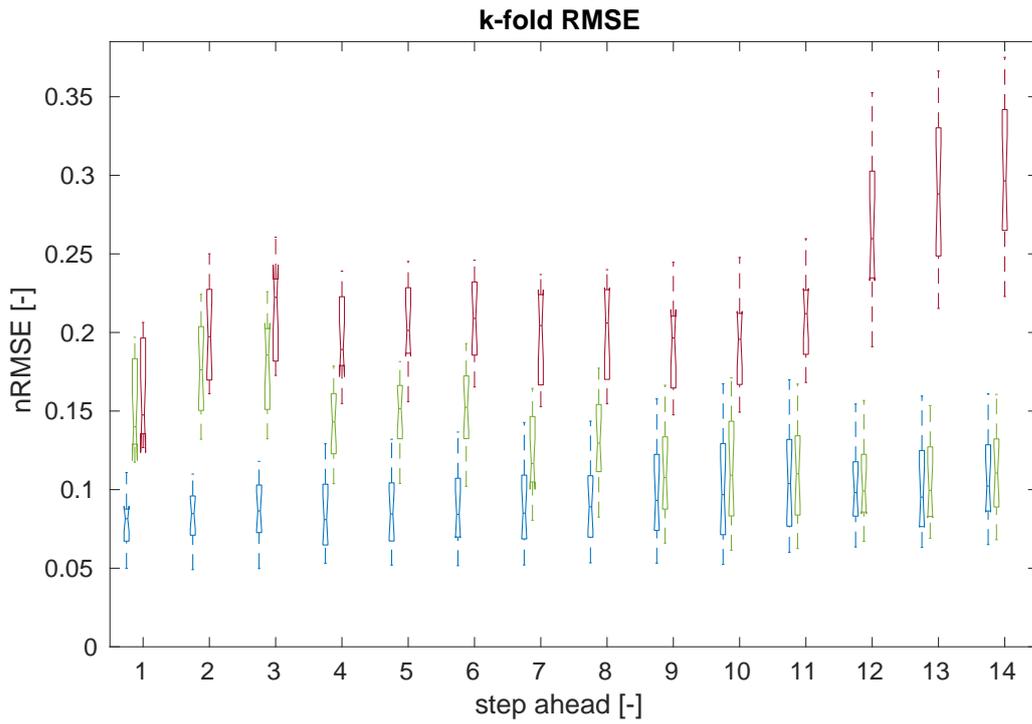


Figure 11: nRMSE for perfect forecasts, perfect forecasts downsampled and real forecasts, for the base case (no PV models)

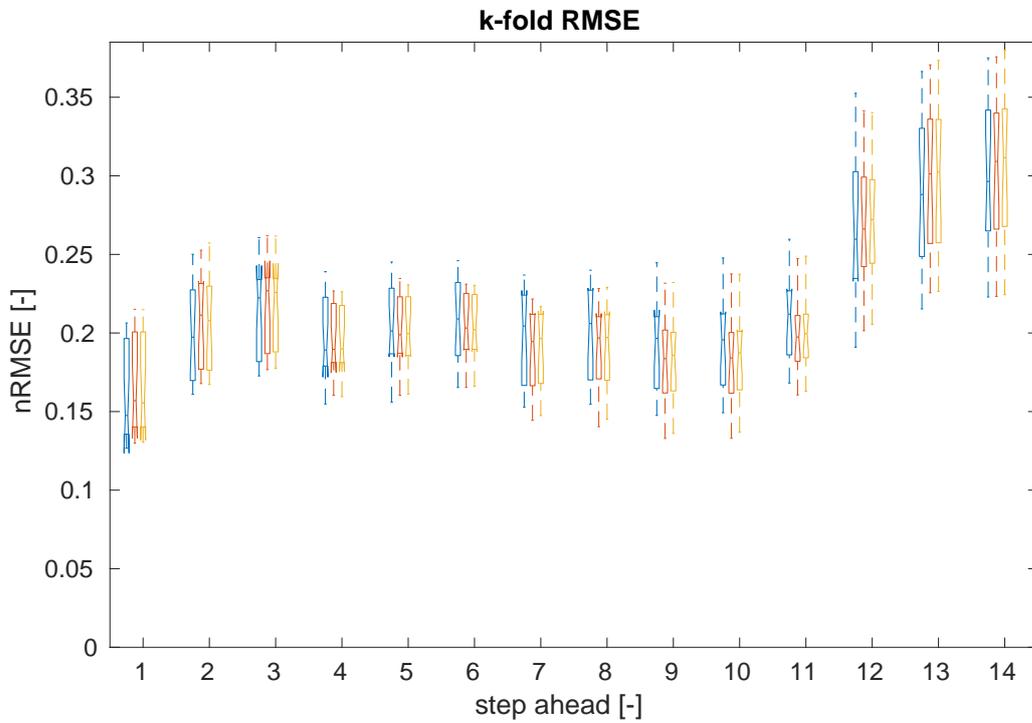


Figure 12: Normalized MAE as a

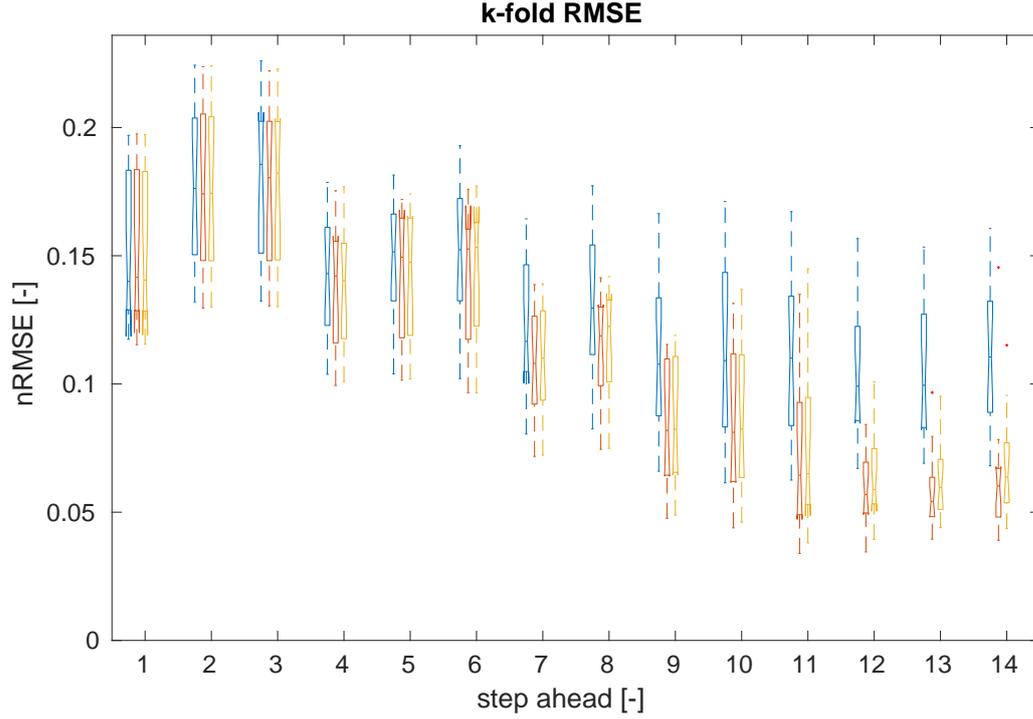


Figure 13: nRMSE as a function of step ahead for 1 hour downsampled perfect forecasts. Blue: base case. Red: with PV model. Yellow: with PV model estimated without  $GHI$ . The effect of modeling PV is negligible for the first 3 steps ahead.

forecasts and the expected extra-terrestrial irradiance:

$$\epsilon_{cl,t} = \frac{\frac{1}{n} \sum_{k=t}^{t+H} G\hat{H}I_{k|t}}{\frac{1}{n} \sum_{k=t}^{t+H} E_k} \quad (24)$$

$$s_t = \begin{cases} 1 & \text{if } \epsilon_{cl} < q_{0.1}(\epsilon_{cl}) \\ 0 & \text{otherwise} \end{cases}$$

where  $s_t$  is the indicator for clear day selection,  $E_k$  is the extraterrestrial irradiance, which is known for a given time and geographical location,  $G\hat{H}I_{k|t}$  is the NWP forecasted of  $GHI$  available at time  $t$  for time-step  $k$ ,  $\epsilon_{cl,t}$  is the normalized error at time  $t$ ,  $\epsilon_{cl}$  is the vector of all the normalized error for all the dataset,  $H$  is the number of step-ahead and  $q_\alpha$  stands for the quantile of level  $\alpha$ . Note that both signals are known in advance, so that this filter can be actually implemented to switch between different forecasting models. In Fig.14 the empirical cumulative distribution function (ECDF) of the nRMSE for the whole forecasting horizon, for the case of perfect forecasts, is shown. Formally, we plotted the ECDF of

$$nRMSE_t = \left[ \frac{1}{H} \sum_{sa=1}^H \left( \frac{y_{t,sa} - \hat{y}_{t,sa}}{\bar{y}_{sa}} \right)^2 \right]^{\frac{1}{2}} \quad (25)$$

$$(26)$$

The continuous lines refer to the whole dataset, while the dotted lines are the ECDFs referring to the clear sky dataset, which was obtained using only observations for which  $s_t = 1$ . It is clear that when the NWP for  $GHI$  for the next 24 hours are close to the extraterrestrial irradiance, the forecasting error for the whole horizon is significantly lower. Fig. 15 shows the same results when NWP forecasts for  $GHI$  and  $T$  are used. Also in the case of clear days, modeling the PV does not significantly increase the accuracy of the forecasts, since when no clouds are present, splitting  $GHI$  in its direct and diffuse components is easier, and the map which links PV production to  $GHI$  is much easier to learn.

### 1.5.6 Distributed hierarchical forecasting

It is easy to see that if consistency is not respected (single household power forecasts do not sum up to the prediction of the aggregated power profile), distributed control algorithms will just disregard forecasts of upper levels of the hierarchy. This is detrimental in all the cases in which the time series at the bottom level of the hierarchy are much harder to forecast with respect to the forecasts of the root node. This is true when the bottom time series present a low signal to noise ratio (with respect to the root node) and are mutually uncorrelated. Consistency can be enforced encoding the hierarchical structure in a learning algorithm. One way to do this is to firstly obtain forecasters for all the levels of the hierarchy, and then reconcile them based on the hierarchical structure. Following this approach, in [24] the authors used ordinary least squares (OLS) regression to reconcile the forecasts in the hierarchy. Elaborating on this approach, [25] proposed a trace minimization method in which the covariance matrix of the forecasters error is estimated to perform a weighted least squares regression. In [26], an elastic net penalization was proposed in order to induce sparseness in the forecasters adjustments, and benefit was shown on the reconciliation of the forecasts for the power consumption of residential consumers. We included this variant in our analysis. In the following we present a distributed algorithm to obtain hierarchical reconciliation of the different time series. Being able to reconcile time series through a distributed algorithm allows to not fully disclose informations about individual time series, respecting the privacy of individual prosumers. In fact, this information would be available only in an aggregated form to the upper level of the structure. Apart from the basic case in which hierarchical reconciliation techniques are useful due to geographical smoothing of the power consumption, the distributed mechanism can be also beneficial in the case in which prosumers do possess additional information about their future consumption, for example the internal scheduling of their heat pump or electric vehicles. Moreover this technique can be applied to temporal hierarchy [27], but in this case decomposing the problem is of few interest, since typically a single entity would be in possess of the information needed to apply the technique in this case.

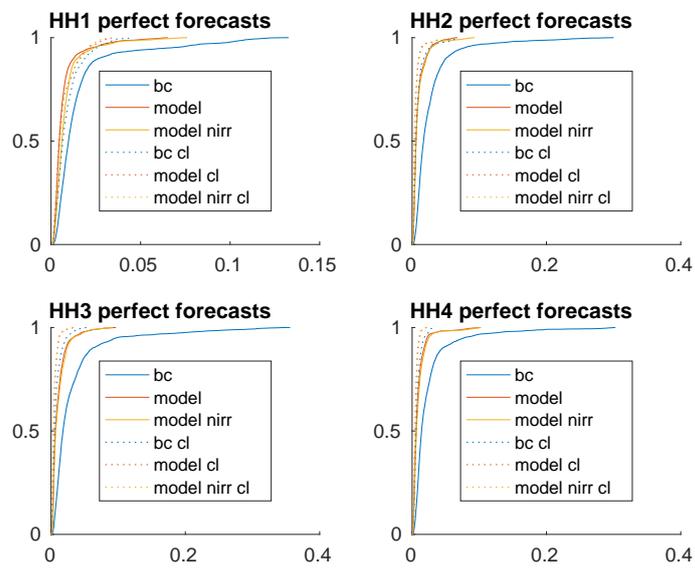


Figure 14: ECDF of the horizon nRMSE for the base forecast and the two PV model forecasts, for each household. The dotted lines are referred to the clear sky day dataset.

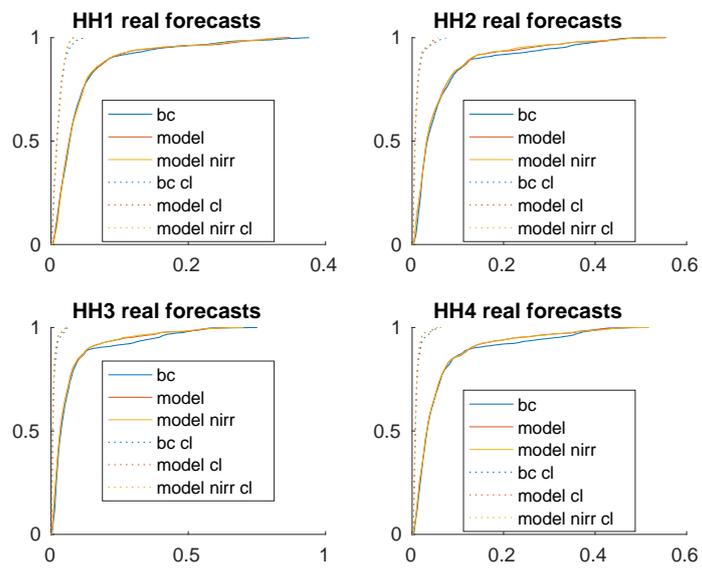


Figure 15: ECDF of the horizon nRMSE for the base forecast and the two PV model forecasts, for each household. The dotted lines are referred to the clear sky day dataset.

### 1.5.7 Problem formulation

We consider a hierarchical structure which can be described by a rooted tree, which is a unidirected acyclic graph, with every node having exactly one parent, except for the root node. Each node is identified by a tuple  $(d_1 \dots d_i \dots d_l)$  where  $l$  is the level to which the node belongs, and each entry represent the enumeration of its  $i_{th}$  level ancestor. Formally, we indicate with  $\tau$  the set of all the nodes in the tree. Given the forecasts for the next  $t$  timesteps of all the  $n$  nodes of the rooted tree, called the base forecasters, we can collect them in the matrix  $T \in \mathbb{R}^{t \times n}$ . Reconciling the forecasts is then equal to the task of finding the set of bottom level forecasts  $X \in \mathbb{R}^{t \times n_b}$  which minimizes the residual  $\epsilon$

$$T = [T_u, T_b] = XS^T + \epsilon \quad (27)$$

where  $T_u \in \mathbb{R}^{t \times n - n_b}$  and  $T_b \in \mathbb{R}^{t \times n_b}$  are the matrices of the upper level and bottom level base forecasters. The intuition behind this is that we are seeking for a set of latent variables,  $X$ , which generate the forecasts at all the levels, through the summation matrix  $S$ . For example, for a 3 levels hierarchy with two nodes in the second level, with 4 bottom forecasters, the matrix  $S$  would be the following:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

If the covariance matrix of  $\epsilon$ ,  $W$ , is known, the optimal solution of the reconciliation is then given by solving the generalized least squares problem [24]

$$X^* = \underset{X}{\operatorname{argmin}} \frac{1}{2} \|T - XS^T\|_W \quad (29)$$

where  $X^*$  is the set of reconciled bottom forecasts, with the analytical solution

$$X^* = (S^T W^\dagger S)^{-1} S^T W^\dagger T^T \quad (30)$$

where  $W^\dagger$  is the pseudoinverse of  $W$ . In the following we limit ourself to the case in which  $W$  is diagonal, the simplest case being the one proposed in [24], in which  $W$  is the identity matrix. We start splitting problem (29) among the nodes of the rooted tree:

$$\underset{X}{\operatorname{argmin}} \sum_{i=1}^n \frac{w_i}{2} \|\hat{t}_i - X s_i^T\|_2^2 \quad (31)$$

where  $s_i \in \mathbb{R}^{n_b}$  are the  $i_{th}$  rows of the summation matrix  $S$  and  $\hat{t}_i$  are the base forecasts. We can than decompose the problem introducing additional variables  $y_i$ :

$$\begin{aligned} \underset{X, Y}{\operatorname{argmin}} \sum_{i=1}^n \frac{w_i}{2} \|\hat{t}_i - y_i\|_2^2 \\ \text{s.t. } y_i = X s_i^T \quad \forall i \in \{n\} \end{aligned} \quad (32)$$

we proceed with an augmented Lagrangian relaxation to turn the problem in a set of unconstrained optimizations, and solve it with an ADMM [28] strategy. The overall problem becomes:

$$\underset{X, Y}{\operatorname{argmin}} \sum_{i \in \mathcal{P}} \frac{w_i}{2} \|\hat{t}_i - y_i\|_2^2 + \frac{1}{2\rho} \|y_i - X s_i^T + \lambda_i\|_2^2 + \sum_{i \in \mathcal{B}} \frac{w_i}{2} \|\hat{t}_i - x_i\|_2^2 \quad (33)$$

where  $\lambda_i \in \mathbb{R}^T$  are the Lagrangian multipliers associated to the constraints in (32),  $\mathcal{P}$  is the set of nodes not belonging to the set of terminal nodes  $\mathcal{B}$ , formally  $\mathcal{P} = \tau \setminus \mathcal{B}(\tau)$ . Problem 33 can be interpreted as a sharing problem, in which each node of the hierarchy tries to minimize the distance of its decision variable (the latent variables, that is the reconciled forecasts) from its target (the original base forecasts  $\hat{t}_i$ ), while being subject to the structural constrained encoded by the summation matrix  $S$ . We then use a parallelized formulation of the sharing problem, which makes use of ADMM [28]. The resulting formulation is the following:

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}} \frac{w_i}{2} \|\hat{t}_i - x_i\|_2^2 + \frac{1}{2\rho} \|x_i - r_i\|_2^2 \quad \forall i \in \mathcal{B} \\ y_i^{k+1} &= \underset{y_i}{\operatorname{argmin}} \frac{w_i}{2} \|\hat{t}_i - y_i\|_2^2 + \frac{1}{2\rho} \|X^{k+1} s_i^T - y_i + \lambda_i\|_2^2 \quad \forall i \in \mathcal{P} \\ \lambda_i^{k+1} &= \lambda_i^k + X^{k+1} s_i^T - y_i^{k+1} \quad \forall i \in \mathcal{P} \end{aligned} \quad (34)$$

where  $r_i$  is a reference signal coming from the parent node of node  $i$ :

$$r_i = \sum_{a \in \mathcal{A}_i} \left( y_i - X^k s_a^T \right) / n_a + x_i^k - \lambda_a \quad (35)$$

where  $\mathcal{A}_i$  is the set of ancestors of node  $i$  and  $n_a$  is the number of the children of the ancestor  $a$ . Intuitively, (34) and (35) divides equally (division by  $n_a$  in (35)) the quadratic loss needed to respect the consistency constraints when moving from the target (the original forecasts). Note that the minimizations in (34) have analytical solutions, so that the final algorithm can be rewritten as:

$$\begin{aligned} x_i^{k+1} &= \frac{r_i + \hat{t}_i \rho w_i}{n_i + \rho w_i} \quad \forall i \in \mathcal{B} \\ y_i^{k+1} &= \frac{\rho w_i \hat{t}_i + X^{k+1} s_i^T + \lambda_i}{1 + \rho w_i} \quad \forall i \in \mathcal{P} \\ \lambda_i^{k+1} &= \lambda_i^k + X^{k+1} s_i^T - y_i \quad \forall i \in \mathcal{P} \end{aligned} \quad (36)$$

where  $n_l$  is the number of levels in the hierarchy. The algorithm can thus be computed only using summation and multiplication; furthermore, it can be solved following a forward-backward strategy. The forward passage consist in each parent

node sending the updated Lagrangian multipliers  $\lambda_i$  downward through the hierarchy. When the Lagrangian is received by a non-terminal node, this will send it, together with its own Lagrangian, to its children. This allows terminal nodes to compute  $x_i$ , since these depend on the the sum of the  $\lambda_i$  coming from all of their ancestors, as described in (35). In the backward passage, the terminal nodes compute their update for  $x_i$  as in (36), and send it upward to their ancestors. Note that each ancestor only needs information from its own children to compute its minimization, since  $X^{k+1} s_i^T$  filters out all the other optimization variables in  $X^{k+1}$ . As soon as the ancestors computes their optimization, they send the solution, and so on, up to the root node. Lagrangian multipliers are updated in a similar fashion.

### 1.5.8 Inducing regularization

It is easy to see that when  $W$  is the identity matrix, the solution of (29) just retrieves a set of bottom level forecasts which minimize the distance of aggregate consistent forecasts with the original base forecasts  $T$ . This would totally ignore the historical accuracy of the base forecasters. In fact, in the case in which some base forecasters present a higher accuracy with respect to the others, we should include it in the reconciliation. This is possible through matrix  $W$ . Anyway, estimating  $W$  is difficult and was avoided in [24], where it was replaced with the identity matrix. In [25]  $W$  is directly estimated for historical error covariance matrix. We follow the approach reported there, also used in [26]:

$$\begin{aligned}
W &= \theta W_d + (1 - \theta) W_1 \\
W_d &= \text{diag}(W_1) \\
W_1 &= \mathbb{E}(ee') \\
\theta &= \frac{\sum_{i \neq j} \text{Var}(\hat{r}_{i,j})}{\sum_{i \neq j} \hat{r}_{i,j}^2}
\end{aligned} \tag{37}$$

where  $\hat{r}_{i,j}$  are the elements of the one step ahead sampled covariance matrix. More details on the computation of  $\theta$  can be found in [29]. In practice, though, this method alone can induce too large adjustments in the bottom level forecasts, since it does not allow the base forecasters to be unchanged, and could lead to reconciled forecasts with poor prediction accuracy. Regularization technique can be applied to the reconciliation problem with favorable results, as shown in [26], where large displacement of  $X^*$  from the bottom level base forecasters are punished, inducing a sparsity structure in the forecasts corrections. This method can be readily included in our distributed algorithm, adding to the second line of (34) a punishment for the deviation of the upper level forecasts from the one generated by the original bottom level forecasts  $T_b$ :

$$k \left( \frac{1 - \alpha}{2} \|y_i - T_b s_i^T\|_2^2 + \alpha \|y_i - T_b s_i^T\|_1 \right) \tag{38}$$

Similarly, the same punishment can be added for the bottom level forecasts (first line of (34)). In this case, the minimization in (34) has still a closed form, which is equal to the proximal operator of the L1 norm, also known as the soft threshold operator [30]. For completeness, the final equations are reported below:

$$x_i^{k+1} = \begin{cases} \frac{a_{x,i}-\rho\alpha}{b_{x,i}} & \mathbf{if} \quad \frac{a_{x,i}-\rho\alpha}{b_{x,i}} > 0 \\ \frac{a_{x,i}+\rho\alpha}{b_{x,i}} & \mathbf{if} \quad \frac{a_{x,i}+\rho\alpha}{b_{x,i}} < 0 \\ t_i & \textit{otherwise} \end{cases} \quad y_i^{k+1} = \begin{cases} \frac{a_{y,i}-\rho\alpha}{b_{y,i}} & \mathbf{if} \quad \frac{a_{y,i}-\rho\alpha}{b_{y,i}} > 0 \\ \frac{a_{y,i}+\rho\alpha}{b_{y,i}} & \mathbf{if} \quad \frac{a_{y,i}+\rho\alpha}{b_{y,i}} < 0 \\ t_i & \textit{otherwise} \end{cases} \quad (39)$$

where  $t_i$  are the original base forecasters of the  $i_{th}$  node, and

$$\begin{aligned} a_{x,i} &= t_i(\rho w_i + \rho(1-\alpha)) + r_i \\ b_{x,i} &= \rho w_i + \rho(1-\alpha) + n_l \\ a_{y,i} &= \rho w_i t_i + X s_i^T + \lambda_i + \rho(1-\alpha) T_b s_i^T \\ b_{y,i} &= \rho w_i + \rho(1-\alpha) + 1 \end{aligned} \quad (40)$$

and the  $\lambda$  update is the same as in (34).

### 1.5.9 Results

In order to estimate the effect of forecast reconciliation, we used 2200 power profiles coming from a smart meter trial measurement campaign in Great Britain provided by AECOM, which is available at <https://www.ukdataservice.ac.uk/>. Fig. 16 shows the performance of the reconciliation technique by means of RMSE, as a function of the step ahead. The first part shows the forecasters performance referred to all the hierarchy, the second part refers to the upper level, while the last one refers to the bottom level. The *consistent* labels show the sum-consistent forecasts obtained summing the original bottom level base forecasters, using the summation matrix  $S$ , that is, following a bottom-up strategy. We can achieve better performance using the ordinary least squares approach, which results in an improvement in the bottom level forecasters. The same solution can be retrieved through the distributed algorithm (36). The dashed lines refers to the minT solution, that is, the one achieved using matrix (37) as weight matrix of the generalized least squares problem. In this case, both the top level and the bottom level accuracy increase.

## 2 Achievement of deliverable

### 2.1 Date

See next section.

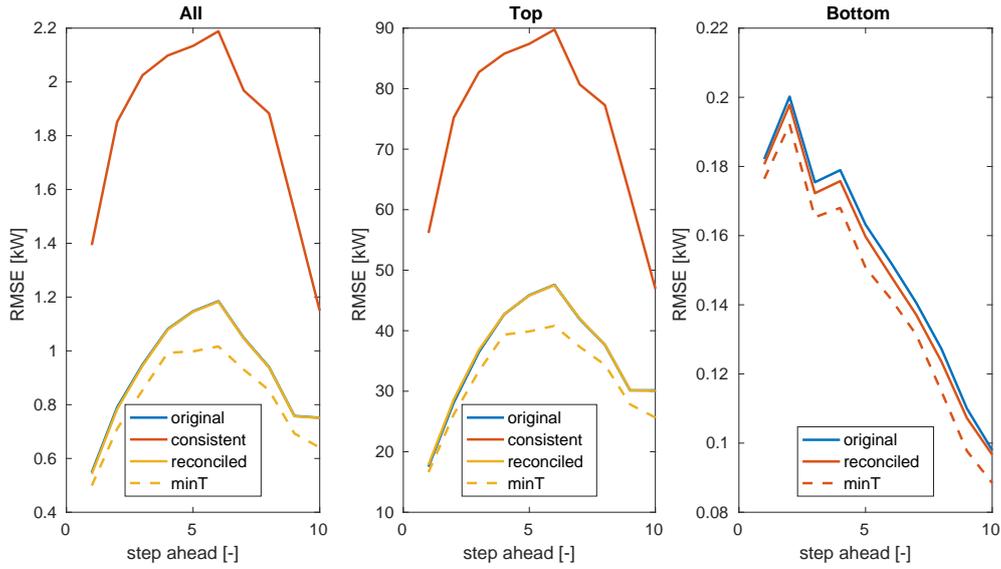


Figure 16: RMSE as a function of step ahead, divided by level of aggregation. Original: base forecaster. Consistent: forecasted profile starting from the original bottom level base forecasters. Reconciled: reconciled forecasts using ordinary least squares. minT: using (37) as weighting matrix.

## 2.2 Demonstration of the deliverable

This deliverable consists of data analysis work. The comparison of the performance of the various forecasting techniques is presented in the previous sections. Preliminary analysis shows that the performance of the forecasting algorithms is satisfactory. An in depth analysis of their adequacy for the distributed control design will be the subject of the rest of the work performed by SUPSI in WP1.

## 3 Impact

This work compares the performances of different multi-step-ahead forecasters for single residential power measurements and their aggregate, up to 24 hours. The output of this deliverable is essential for the work that will be performed in the WP1 tasks involving the design and test of distributed demand side management control algorithms. Indeed, those algorithms will make use of the different forecast techniques presented above.

## References

- [1] G. Frison and J. B. Jørgensen, “A fast condensing method for solution of linear-quadratic control problems,” *Proceedings of the IEEE Conference on Decision and Control*, pp. 7715–7720, 2013.
- [2] G. Frison and J. B. Jorgensen, “Efficient implementation of the Riccati recursion for solving linear-quadratic control problems,” *2013 IEEE International Conference on Control Applications (CCA)*, pp. 1117–1122, 2013.
- [3] J. B. Jørgensen, J. B. Rawlings, and S. B. Jørgensen, “Numerical Methods for Large Scale Moving Horizon Estimation and Control,” *7th International Symposium on Dynamics and Control of Process Systems (DYCOPS)*, 1996.
- [4] C. V. Rao, S. J. Wright, and J. B. Rawlings, “<IP methods and MPC Rao 1998.pdf>,” vol. 99, no. 3, pp. 723–758, 1998.
- [5] Alessandro Alessio and A. Bemporad, *A Survey on Explicit Model Predictive Control*. 2009.
- [6] Y. Wang and S. Boyd, “Fast Model Predictive Control Using Online Optimization,” vol. 18, no. 2, pp. 267–278, 2010.
- [7] L. Ljung, *System Identification Theory for User*. 1987.
- [8] N. Gröwe-Kuska, H. Heitsch, and W. Römisch, “Scenario reduction and scenario tree construction for power management problems,” in *2003 IEEE Bologna PowerTech - Conference Proceedings*, 2003.
- [9] Z. B. Bouallegue, T. Heppelmann, S. E. Theis, and P. Pinson, “Generation of scenarios from calibrated ensemble forecasts with a dual ensemble copula coupling approach,” pp. 4737–4750, 2016.
- [10] J. D. Wichard, “Forecasting the NN5 time series with hybrid models,” *International Journal of Forecasting*, 2011.
- [11] S. Ben Taieb, G. Bontempi, A. F. Atiya, and 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*, vol. 39, no. 8, pp. 7067–7083, 2012.
- [12] C. M. Bishop, *Patterns Recognition and Machine Learning*. 2006.
- [13] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to Statistical Learning*, vol. 7. 2000.
- [14] N. Meinshausen, “Quantile Regression Forests Nicolai,” *Journal of Machine Learning Research*, vol. 131, pp. 65–78, 2014.

- [15] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, “Extreme learning machine: algorithm, theory and applications,” *Artificial Intelligence Review*, 2015.
- [16] C. C. Holt, “Forecasting seasonals and trends by exponentially weighted moving averages,” *International Journal of Forecasting*, 2004.
- [17] E. S. Gardner, “Exponential smoothing: The state of the art,” *Journal of Forecasting*, 1985.
- [18] F. Golestaneh, P. Pinson, and H. B. Gooi, “Very Short-Term Nonparametric Probabilistic Forecasting of Renewable Energy Generation; With Application to Solar Energy,” *Power Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–14, 2016.
- [19] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de Pison, and F. Antonanzas-Torres, “Review of photovoltaic power forecasting,” *Solar Energy*, vol. 136, pp. 78–111, 2016.
- [20] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, “Solar forecasting methods for renewable energy integration,” *Progress in Energy and Combustion Science*, vol. 39, no. 6, pp. 535–576, 2013.
- [21] C. Voyant, G. Notton, S. Kalogirou, M. L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, “Machine learning methods for solar radiation forecasting: A review,” *Renewable Energy*, vol. 105, pp. 569–582, 2017.
- [22] L. Nespoli and V. Medici, “An unsupervised method for estimating the global horizontal irradiance from photovoltaic power measurements,” *Solar Energy*, 2017.
- [23] F. Sossan, L. Nespoli, V. Medici, and M. Paolone, “Unsupervised Disaggregation of PhotoVoltaic Production from Composite Power Flow Measurements of Heterogeneous Prosumers,” 2018.
- [24] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang, “Optimal combination forecasts for hierarchical time series,” *Computational Statistics and Data Analysis*, 2011.
- [25] S. L. Wickramasuriya and G. Athanasopoulos, “Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization,” *Working Paper*, no. June, 2017.
- [26] S. B. Taieb, R. Rajagopal, S. Ben Taieb, J. Yu, M. Neves Barreto, and R. Rajagopal, “Regularization in Hierarchical Time Series Forecasting With Application to Electricity Smart Meter Data,” *Aaai*, no. 2011, pp. 4474–4480, 2017.
- [27] D. Yang, H. Quan, V. R. Disfani, and C. D. Rodríguez-Gallegos, “Reconciling solar forecasts: Temporal hierarchy,” *Solar Energy*, 2017.

- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [29] J. Schäfer and K. Strimmer, “A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics,” *Statistical Applications in Genetics and Molecular Biology*, 2005.
- [30] A. Beck, *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.
- [31] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, “Smart\*: An Open Data Set and Tools for Enabling Research in Sustainable Homes,” *SustKDD*, no. August, p. 6, 2012.
- [32] T. V. Jensen and P. Pinson, “Data Descriptor: RE-Europe, a large-scale dataset for modeling a highly renewable European electricity system,” *Scientific Data*, vol. 4, pp. 1–18, 2017.
- [33] S. Hosseini, S. Kelouwani, K. Agbossou, A. Cardenas, and N. Henao, “A semi-synthetic dataset development tool for household energy consumption analysis,” *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 564–569, 2017.
- [34] T. Afjei, U. Schonhardt, C. Wemhöner, M. Erb, H. R. Gabathuler, H. Mayer, G. Zweifel, M. Achermann, R. von Euw, and U. Stöckli, “Standardschaltungen für Kleinwärmepumpenanlagen Teil 2: Grundlagen und Computersimulationen. Schlussbericht,” tech. rep., 2002.
- [35] G. Reynders, J. Diriken, and D. Saelens, “Quality of grey-box models and identified parameters as function of the accuracy of input and observation signals,” *Energy and Buildings*, vol. 82, pp. 263–274, 2014.
- [36] P. Bacher and H. Madsen, “Identifying suitable models for the heat dynamics of buildings,” *Energy and Buildings*, vol. 43, pp. 1511–1522, jul 2011.
- [37] F. S. Office, “Bâtiments: aperçu général selon les cantons 2016,” 2016.
- [38] L. Pampuri, N. Cereghetti, P. G. Bianchi, and P. Caputo, “Evaluation of the space heating need in residential buildings at territorial scale: The case of Canton Ticino (CH),” *Energy and Buildings*, vol. 148, pp. 218–227, 2017.
- [39] European Commission, “European Building Database.”
- [40] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*. 2007.

- [41] T. Cholewa, M. Rosiński, Z. Spik, M. R. Dudzińska, and A. Siuta-Olcha, “On the heat transfer coefficients between heated/cooled radiant floor and room,” *Energy and Buildings*, vol. 66, pp. 599–606, 2013.
- [42] P. Wallenten, “Heat Transfer Coefficients in a Full Scale Room With and Without Furniture,” *Lund Institute of Thechnology*, pp. 1–8, 1999.
- [43] T. Schütz, H. Harb, R. Streblov, and D. Müller, “Comparison of models for thermal energy storage units and heat pumps in mixed integer linear programming,” *PROCEEDINGS OF ECOS 2015 - THE 28TH INTERNATIONAL CONFERENCE ON EFFICIENCY, COST, OPTIMIZATION, SIMULATION AND ENVIRONMENTAL IMPACT OF ENERGY SYSTEMS JUNE 30-JULY 3, 2015, PAU, FRANCE*, 2015.
- [44] R. D. Coninck, R. Baetens, D. Saelens, A. Woyte, L. Helsen, A. Mechanics, and B. P. Section, “Rule-based demand side management of domestic hot water production with heat pumps in zero energy neighbourhoods,” *Journal of Building Performance Simulation*, 2013.
- [45] L. Nespoli, A. Giusti, N. Vermes, M. Derboni, E. Rizzoli, and L. M. Gambardella, “Distributed demand side management using electric boilers,” *Computer Science - Research and Development*, 2016.

## 4 Appendices

### 4.1 Synthetic power profile generation

To our knowledge, one of the most fungible datasets for residential energy consumption forecasting is the Apartment dataset of the UMass Smart\* Dataset [31]. Unfortunately, these data come from American consumers, which are not very representative of European energy consumptions. Moreover they do not possess PV installations which we would like to include in the analysis. Due to the scarcity of energy consumption datasets, researchers in this field have recently tried to produce synthetic datasets for forecasting and analysis, based on statistical analysis or on simulation [32][33].

We decided to follow the latter strategy, producing our dataset through a dynamic simulation. For this task we built the code from scratch relying on the standard scipy ODE integrator.

In the following we give a detailed description of the reference system we have considered for the simulation. Since it is not of general interest to describe all the possible configurations of the simulated systems, we just describe a typical configuration, on which all the simulated systems are based. In order to obtain a representative dataset for Switzerland, we used the STASCH6 standard [34] and its variants as a reference for the heating system and the control logic.

#### 4.1.1 Heating system and control logic

The STASCH6 standard comprehends 3 main components: an heatpump (HP), a water tank used as an energy buffer, and a heating element delivering heat to the building. The HP control logic is based on two temperature sensors placed at different heights of the water tank, while the circulation pump connecting the tank with the building's heating element is controlled by an hysteresis on the temperature measure by a sensor placed inside the house.

We describe the control logic in a sequential way, following the heating components of the system. The first decision is taken by the building central controller, which decides its working mode, that is, if the building needs to be cooled or heated, based on a moving average of the historical data of the external temperature:

$$\begin{cases} wm_t = -1 & \mathbf{if} \ T_{ma,t} > T_{max,ma} \\ wm_t = 1 & \mathbf{if} \ T_{ma,t} < T_{min,ma} \\ wm_t = 0 & otherwise \end{cases} \quad (41)$$

where the working mode  $wm_t$  is negative when the building requires to be cooled, positive when heating is required, and 0 when no actions are needed.  $T_{max,ma}$  and  $T_{min,ma}$  represent the maximum and minimum values of the external temperature's moving average, which is based on the past 7 days. The actual activation of the

heating element is controlled by the hysteresis on the internal temperature of the building,  $T_z$ . If the working mode is positive, this is given by:

$$\begin{cases} s_{hy,t} = 1 & \mathbf{if} \ (T_z < T_{min,hy} - \Delta T/2) \\ & \mathbf{or} \ (T_z < T_{min,hy} + \Delta T/2 \ \mathbf{and} \ s_{hy,t-1}) \\ s_{hy} = 0 & \mathit{otherwise} \end{cases} \quad (42)$$

where  $s_{hy,t}$  is the state of the hysteresis at time  $t$ , 1 meaning that the circulation pump of the heating element must be activated, and  $DT$  was chosen to be equal to  $1^\circ C$ . For completeness, we report also the control logic when the building is in cooling mode:

$$\begin{cases} s_{hy,t} = 1 & \mathbf{if} \ (T_z > T_{max,hy} + \Delta T/2) \\ & \mathbf{or} \ (T_z > T_{max,hy} - \Delta T/2 \ \mathbf{and} \ s_{hy,t-1}) \\ s_{hy} = 0 & \mathit{otherwise} \end{cases} \quad (43)$$

The incoming water temperature in the heating element is then modulated linearly through a 3-way valve between a maximum and minimum value, based on the external temperature, both in the heating and cooling modes. When operative, the heating element requests hot or cold water to the water tank, which control logic is based on two temperature sensors located in two different layers. When the building is in heating mode, the control logic is a simple hysteresis based on the temperature of the sensor in the uppermost layer, which is identical to the one in (42). When in cooling mode, the control logic is the following:

$$\begin{cases} s_{hy,t} = -1 & \mathbf{if} \ (T_{up} > T_{max}^c + \Delta T/2) \\ & \mathbf{or} \ T_{low} > T_{max}^c + \Delta T/2 \\ s_{hy,t} = 0 & \mathbf{if} \ (T_{low} < T_{min}^c) \ \mathbf{or} \ (T_{up} < T_{max}^c - \Delta T/2) \\ s_{hy,t} = s_{hy,t-1} & \mathit{otherwise} \end{cases} \quad (44)$$

where  $T_{up}$  and  $T_{low}$  are the temperature measured by the upper and lower sensors, respectively, and  $T_{min}^c$  and  $T_{max}^c$  are the minimum and maximum desired temperatures of the water in the tank while in cooling mode.

The value of  $s_{hy,t}$  is then communicated to the HP. In the case in which the HP is also used for the domestic hot water (DHW), the DHW tank is always served with priority by the HP.

#### 4.1.2 Building model

We modeled the building thermal dynamics with a simple one state RC equivalent model. The main reason for this choice is that it is hard to generalize RC models with higher number of states, since no value can be found in the literature for the needed parameters. Estimating an RC model from data requires different mea-

measurements of temperatures, internal and solar gains, at a resolution of at least 10 minutes. This kind of datasets are extremely hard to find, and limited to only a few, often undwelled, cases. These equivalent RC circuit parameters could, in theory, be estimated starting from first principles, but several studies show that this can give worse results than estimating a model from data. The second reason is that, while a higher order model leads in general to smaller one step ahead residuals compared to a lower order model, the loss of accuracy passing from a one state model to an higher order one when considering a longer period of simulation is much lower [35]. Last, when considering RC models for buildings with a number of states higher than 3, the chances of overfitting are high, and additional measurements such as the heat fluxes between thermal zones are required to guarantee observability. Alternatively, pseudo-random binary sequences must be applied to the heating systems in order to excite the system in a wide range of frequencies [36], while being uncorrelated with other exogenous inputs, which technique induce high changes in internal temperature of the building and cannot clearly be applied to occupied building.

We adopted the following methodology to retrieve representative R values for the single state RC equivalent model. We retrieved the distribution of year of construction for residential buildings in Switzerland from the swiss Federal Statistical Office [37]. We then combined them with the estimated mean heating needs per squared meter, based on construction year [38]. In this way we get a distribution of energy demand for heating,  $E_d$  [ $kWh/m^2/year$ ]. We then retrieved the U values per squared meter (the inverse of the R parameter per squared meter), dividing  $E_d$  for 1500 equivalent hour of the building's heating system per year. This gave us the estimated U value distribution in [ $kW/m^2$ ]. In Fig. 17, the final distribution for Switzerland is shown. For comparison, we also plotted the distribution of the declared U values from the EU28 members, which is available for the year 2014 at [39]. For additional comparison, we identified a one state RC equivalent circuit from a monitored building located in Biel-Banken. The model is the following;

$$C \frac{\partial T_z}{\partial t} = \frac{T_{ext} - T_z}{R} + kQ_h + A_{eq}I_s \quad (45)$$

where  $T_{ext}$  is the the external temperature,  $R$  is the equivalent thermal resistance for the building,  $k$  is a parameter weighting the estimated power coming from the heating system  $Q_h$ ,  $I_s$  is the incoming solar radiation and  $A_{eq}$  is the estimated equivalent window area.

### 4.1.3 Floor heating

Modeling floor heating requires to simulate an N-states system, since the temperature of the water in a given point of the serpentines depends on in a non-trivial way on all the temperatures of the previous portion of the serpentines and of the surrounding floor. Furthermore, simulating the temperature of the water in the serpentine in a dynamic way could lead to prohibitive computational time (considering

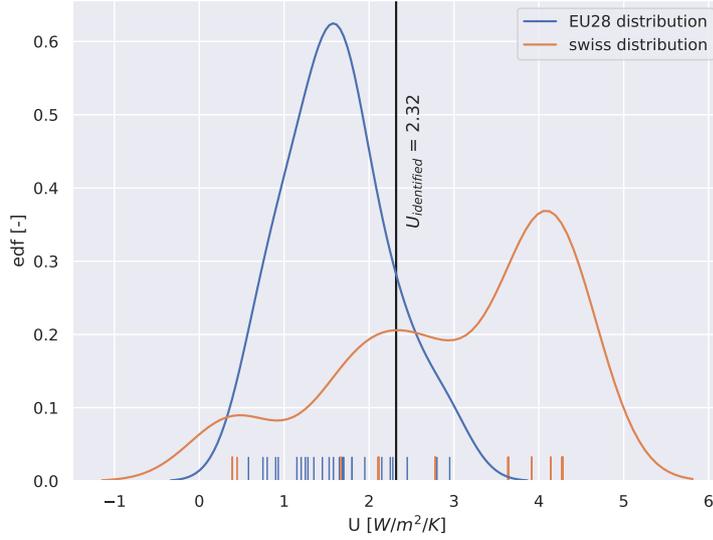


Figure 17: Comparison of U value distributions for Switzerland and for the member of EU28. The vertical line shows the identified U value for a monitored building located in Biel-Benken

we want to simulate hundreds of buildings), due to the CourantFriedrichsLewy condition. Considering a typical mass flow in the serpentine of  $0.1 [kg/s]$ , a radius of the tubes of  $2 [cm]$ , and a discretization of 1 meter along the serpentine, the maximum allowable time-step is in the range of 2 seconds (considering implicit solution of the transport equation inside the tube). Since we do not simulate thermal activated building structures (TABS), in which the water of the heating system flows inside the building's concrete structure, but only underfloor heating pipes, which effects due to thermal inertia are less significant, we chose to neglect the thermal transient of the screed layer. Considering a fixed and uniform temperature for the ground and the building internal temperature at each time-step and stationary conditions, we can retrieve the analytical expression of the temperature profile along the pipe, through the energy balance on an infinitesimal element of the pipe. This can be expressed as:

$$\frac{\partial T_x}{\partial t} = \Phi_x - \Phi_{x+\partial x} + \dot{q}_{up} + \dot{q}_{down} \quad (46)$$

where  $x$  is the distance from the pipe entrance,  $T_x$  is the temperature of the water inside the pipe at  $x$ ,  $\Phi$  are enthalpy flows at the entrance and exit of the considered infinitesimal volume,  $\dot{q}_{up}$  and  $\dot{q}_{down}$  are the heating powers from the building and from the ground. Expressing the latter through equivalent resistance taking into account convective and conductive effects, the balance in steady state can be rewritten as:

$$\frac{\dot{m}c_p}{\rho^*} \frac{\partial T_x}{\partial x} = \frac{R_{down}T_z + R_{up}T_g}{R_{down} + R_{up}} - T_x = T^a - T_x \quad (47)$$

where  $T^a$  is the asymptotic temperature and where:

$$R_{down} = \frac{1}{h_{in}w} + \frac{1}{h_{u,eq}w} + R_u \quad (48)$$

$$R_{up} = \frac{1}{h_{in}w} + R_g \quad (49)$$

$$\rho^* = \frac{R_{up} + R_{down}}{R_{up}R_{down}} \quad (50)$$

where  $w$  is the diameter of the tube,  $h_{in}$  is the internal coefficient of heat transfer, which can be retrieved using available empirical relations for fully developed flow with fixed temperature at the boundary conditions [40],  $h_{u,eq}$  is the heat transfer coefficient between the floor and the building air including both the effect for natural convection and radiation. The values of  $h_{u,eq}$  can be found in the literature [41],[42]. The value of the thermal resistances  $R_u$  and  $R_g$ , towards the floor and the ground, can be found in the literature as well. We can reformulate (47), making it adimensional through a change of variable:

$$\frac{\partial \Theta}{\partial \mathcal{X}} = -\Theta \quad (51)$$

from which solution we can retrieve the temperature profile of the water inside the pipe:

$$T_x = T^a + (T_0 - T^a)e^{\frac{-x\rho^*}{\dot{m}c_p}} \quad (52)$$

where  $T_0$  is the temperature of the water at the pipe inlet. We can use (52) to retrieve the heating power flowing into the building, integrating  $\dot{q}_{up}(x)$  along the pipe.

$$\dot{Q}_{up} = \int_0^L \dot{q}_{up}(x)dx = \int_0^L \frac{T(x) - T_z}{R_{up}} dx \quad (53)$$

where  $L$  is the length of the serpentine. Integrating, we obtain

$$\dot{Q}_{up} = \frac{(T^a - T_z)L - (T_L - T_0)\frac{\dot{m}c_p}{\rho^*}}{R_{up}} \quad (54)$$

where  $T_L$  is the temperature of the water at the outlet of the serpentine. Note that the equation (54) tends to  $(T_L - T_0)\dot{m}c_p$  when  $R_{down}$  increases and  $R_{up}$  is kept fixed. The nominal mass flow of the heating system and the length of the serpentine are found as the solution of the following optimization problem:

$$\operatorname{argmin}_{L, \dot{m}} \left( \dot{Q}_{up}(L) - \dot{Q}_{nom} \right)^2 + 10^{-3} (\dot{m} - \dot{m}_{nom})^2 \quad (55)$$

where  $\dot{m}_{nom}$  is a reference mass flow, equal to  $0.1 [kg/s]$  and  $\dot{Q}_{nom}$  is the power required to keep the building internal temperature constant under reference conditions (we used an external temperature of  $-4^\circ C$  and a desired internal temperature

of 20 °C):

$$\dot{Q}_{nom} = \frac{\Delta T_{ref}}{R} \quad (56)$$

where  $R$  is the resistance of an equivalent RC circuit describing the heating dynamics of the building.

#### 4.1.4 Water tanks and boilers

The water tank connected with the floor heating, which is used as a buffer by the heat pump, and the boiler for the DHW, are modeled as a N-states fully-mixed stratified tanks. Despite not being able to model buoyancy driven effects such as heat plumes and transient de-stratification, this kind of models are suitable for 1D simulations and control [43].

The dynamic equation describing the evolution of the temperature of the tank's layers is the following:

$$C \frac{\partial T_i}{\partial t} = \dot{Q}_{buo,i}^u + \dot{Q}_{buo,i}^d + \dot{Q}_{h,i} + \dot{Q}_{loss,i} + \dot{Q}_{cond,i}^u + \dot{Q}_{cond,i}^d + c_p \dot{m}(T_{i-1} - T_i) \quad (57)$$

where  $T_i$  is the temperature of the  $i_{th}$  layer,  $Q_{buo}^u, Q_{buo}^d, Q_{cond}^u, Q_{cond}^d$  are the thermal powers due to buoyancy and conduction, from the lower and upper layer, respectively. The last term represents the enthalpy flow due to mass exchange, while  $C$  is the thermal capacity of the layer, in  $[J/K]$  and  $Q_{h,i}$  is the thermal power due to an electric resistance (for the boiler) or an heat exchange (for the heating system buffer). The expression for the above thermal power are the following:

$$\dot{Q}_{buo,i}^u = k \max(T_{i+1} - T_i, 0)N, \quad 0 \quad for \quad i = N \quad (58)$$

$$\dot{Q}_{buo,i}^d = k \max(T_{i-1} - T_i, 0)N, \quad 0 \quad for \quad i = 1 \quad (59)$$

$$\dot{Q}_{cond,i}^u = u_{amb}(T_{i+1} - T_i), \quad 0 \quad for \quad i = N \quad (60)$$

$$\dot{Q}_{cond,i}^d = u_{amb}(T_{i-1} - T_i), \quad 0 \quad for \quad i = 1 \quad (61)$$

$$\dot{Q}_{loss,i} = u_{amb}(T_{ext} - T_i) \quad (62)$$

$$\dot{Q}_{h,i} = \dot{Q}_{tot}/n_h \quad if \quad i \in \mathcal{I} \quad (63)$$

$$(64)$$

where  $N$  is the number of layers,  $u_{amb}$  is the equivalent thermal loss coefficient with the ambient and  $\mathcal{I}$  is the set of the  $n_h$  layers heated by the heat exchange (or electric resistance). The buoyancy model is the one proposed in the IDEAS library [44]. Detailed description of the parameters for the boiler model can be found in [45].

#### 4.1.5 Heat pump model

The heat pump is modeled by means of interpolated tables, in which heating and electrical power are available as a function of the evaporator and the condenser temperatures. The tables were taken from the energy simulation software Polysun (Vela Solaris AG, Winterthur, Switzerland). When the heat pump produces heat

for both the heating system and the domestic hot water, its control logic prioritizes the latter, meaning that the buffer is heated as long as the DHW tank temperature sensor reaches the upper bound of its hysteresis control.