Convolutional Neural Network and Stochastic Variational Gaussian Process for Heating Load Forecasting

Federico Bianchi*, Pietro Tarocco*, Alberto Castellini, Alessandro Farinelli

Verona University, Department of Computer Science,
Strada Le Grazie 15, 37134 Verona, Italy,
{federico.bianchi, pietro.tarocco, alberto.castellini,
alessandro.farinelli}@univr.it
* These authors contributed equally to this work.

Accepted version.
The final authenticated version is available online at https://www.springerprofessional.de/convolutional-neural-network-and-stochastic-variational-gaussian/18748442

Abstract. Heating load forecasting is a key task for operational planning in district heating networks. In this work we present two advanced models for this purpose, namely a convolutional neural network (CNN) and a Stochastic Variational Gaussian Process (SVGP). Both models are extensions of an autoregressive linear model available in the literature. The CNN outperforms the linear model in terms of 48-hours prediction accuracy and its parameters are interpretable. The SVGP has performance comparable to the linear model but it intrinsically deals with prediction uncertainty, hence it provides both load estimations and confidence intervals. Models and performance are analyzed and compared on a real dataset of heating load collected in an Italian network.

Keywords - Heating load forecasting, smart grids, convolutional neural networks, Stochastic Variational Gaussian Processes, model interpretability.

1 Introduction

Energy management for smart grids gained strong interest from the artificial intelligence community [17]. A branch of smart grids concerns District Heating Networks (DHNs), centralized heating plants that provide heating to residential and commercial buildings through a network of pipes. In particular these measurements consider the temperature and the water flow rate. Accurate prediction of heating load plays a key role in energy production, supplying planning and energy saving, with economical and environmental benefits.

Data-driven forecasting [4,3,5] involves learning models of a variable of interest (i.e., the heating load in our case) from historical data of the same and other

variables (e.g., meteorological or social factors) to predict future values of the variable of interest. Several methodologies are available in the literature for this purpose. Autoregressive linear models [15,1] predict the target variable considering a linear combination of environmental and social factors (day of the week, calendar events). These models are usually simple to interpret but they have a quite rigid function form that can limit their performance in case of complex variable relationships. In [1] a multiple equation autoregressive linear approach is proposed, where the heating load of each pair (hour of the day, day of the week) is modeled independently, resulting in 168 equations.

Recurrent Neural Networks (RNN), in particular Long Short-Term Memory (LSTM) [10] and convolutional-LSTM [19], are among the most used methods for time series forecasting. The disadvantage of these models is that they require large training datasets to be learned and they are hardly interpretable. CNNs have been used for energy load forecasting and other problems related to time series forecasting [14]. What differentiates these works from ours is that we focus on the specific problem of heating load forecasting and provide a simple CNN model having good interpretability and better forecasting performance than available linear regression models.

Gaussian processes (GPs) [16] are other approaches used for time series forecasting. Their advantage is that they explicitly consider uncertainty in analyzed data, hence their predictions are equipped with both expected values and confidence intervals. On the other hand, exact learning of these models is very time consuming and unfeasible for large datasets, hence approximated training methods are used. Usage of GPs for time series analysis has recently gained interest [18,8], and applications to the energy forecasting domain are present in the literature [2,7,20]. The main differences between these approaches and our model is that we use stochastic variational Gaussian processes (SVGPs), which enable model training on a two-years dataset in few minutes using GPUs, and that our model uses specific variables for heating load forecasting.

In the following of this paper we propose two models for heating load forecasting, a CNN and a SVGP. Both models take inspiration from the autoregressive model proposed in [1] (see model called ARM_4) and extend it with specific features of CNN and SVGP, respectively. An analysis of the models is proposed with the aim to explain [6] how they extend the autoregressive model. Models are trained, tested and their performance are compared on a real dataset collected in a DHN located in Verona (Italy), containing hourly heating load produced by the plants in years 2016, 2017 and 2018. Novelties of this work are i) CNN outperforms a state-of-the-art autoregressive linear regression model, ii) SVGP has slightly lower performance but it provides useful confidence intervals on the prediction, iii) first step towards model interpretability. Comparison with other methods (e.g., LSTM and T-CNN) has been considered in some of our experiments but in this paper we only presented the two models with best results in terms of both performance and interpretability. The main contributions to the state-of-the-art are summarized in the following:

- a CNN and a SVGP model are proposed for heating load forecasting in DHNs;
- model parameters are analyzed and explained highlighting connections with parameters of the autoregressive linear model in the literature;
- model performance are tested and compared on a real-world dataset.

The rest of the manuscript is organized as follows. Section 2 presents the framework for model comparison, the dataset, the used methodologies and the performance measures. Result and performance are analyzed in Section 3. Conclusions and future directions are described in Section 4.

2 Material and Methods

In this section we formalize the problem of heating load forecasting for DHNs, we describe the dataset composed by heating load and environmental variables. We finally introduce the modeling methodologies and performance measures we used to compare the models.

2.1 Problem definition and system overview

District heating networks are plants in which a power station, often through co-generation, produces heat and distributes it through a network of pipes connected to commercial and residential buildings. The heating load is collected by direct measurements performed in the plant. Forecasting methods are an important task for improving the process of planning, production and distribution of heating. In Figure 1 an overview of data analysis framework is displayed. In the first phase, models are trained using real-world data. In the second phase models are tested on a different test set by performing all possible 48-hours predictions of heating load. In the last phase model parameters and performance are analyzed and compared.

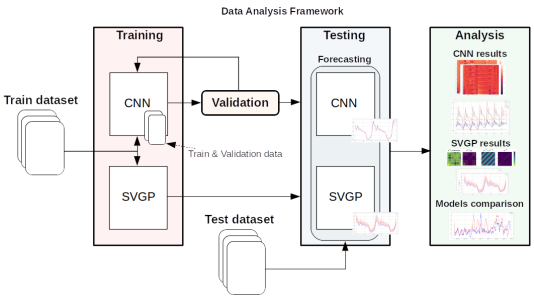


Fig. 1. Overview of the data analysis framework

2.2 Dataset

In the present work we use a real dataset provided by AGSM[1] an Italian utility company that manages a DHN in Verona. Data was collected from 01.01.2016 to 21.04.2018 with hourly sampling interval including historical load l and forecast of weather variables like temperature T, relative humidity (R_H), rainfall (R), wind speed (W_S), wind direction (W_D). We first selected only observations belonging to time intervals in which the heating is on (this is regulated by law in intervals from 01.01.2016 to 11.05.2016, from 11.10.2016 to 14.05.2017, and from 16.10.2017 to 21.04.2018). We engineered new variables according to similar applications in the literature [15]. The complete list of variables is display in Table 1. Models are trained using data related to years 2016 and 2017 (10140 observations) and tested on 2018 data (2497 observations).

Table 1. List of variables used in the models.

Variable	Description	Variable	Description
l	Heating load [MW] (target)	W_S	Wind speed [m/s]
l_i	$i \in [1, 7]$ Heating load i days ago	W_D	Wind direction [0,9], 9=no wind
T	Temperature [°C]	R	Rainfall (1 = rain, 0 = no rain)
T^2	Square of T	H	Holiday (0 = false, 1 = true)
$T_{ma(7)}$	Moving avg of T last 7 days	h	Hour of the day [0, 23]
T_M	Maximum T of the day	d	Day of the week [1, 7]
T_M^2	Square of T_M of the day	w	Weekend (0 = false, 1 = true)
T_{Mp}	T_M of the previous day	Sa	Saturday (0 = false, 1 = true)
T_{Mp}^2	T_M^2 of previous day	Su	Sunday (0 = false, 1 = true)
R_H	Relative humidity [%]		

2.3 Convolutional Neural Network model

CNNs are neural networks that use a linear mathematical operator called *convolution* in at least one of their layers [10]. Each convolutional neuron takes two functions $x(t)$ and $k(t)$ as inputs and generates a new function $f(t)$ which is defined, when t is discrete, as $f(t) = (x*k)(t) = \sum_{i=-\infty}^{+\infty} x(i)k(t-i)$ where function x is often referred to as *input*, function k as *kernel* (or *filter*) and function f as *feature map*. The kernel is learned by suitable algorithms to allow the network to approximate a function of interest, in our case study, the future values of heating load from past and present values of environmental and social factors. Working with time series data, we apply convolution over a single dimension, i.e., time t, hence our kernels are bidimensional matrices of parameters having one column for each input variable and one row for each time instant considered in the convolution.

[1] `https://www.agsm.it/`

The CNN presented in this work has a simple architecture which however outperforms the autoregressive linear model presented in [1], showing the strong capabilities of CNNs to forecast future values of heating load. The network architecture is displayed in Figure 2.a. It takes as an input a matrix having one column for each variable (22 columns in total) and one row for each time instant considered for predicting the heating load of the next hour (168 rows in total). The first layer is a CNN layer with five neurons. Each neuron performs a convolution of the input using a kernel of the same dimension of the input itself (i.e., 3697 weights are used in each kernel, bias included) and then applies a ReLU activation function (i.e., $ReLU(v) = max(v, 0)$) generating a single real value for each neuron. The five feature maps $f_0, \ldots, f_4$ thus obtained are then passed to a *dense* neuron, which computes their linear combination $y = w_0 f_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 + w_5$ where $w_i \in \mathbb{R}$ are the weights and the bias of the dense neuron. This operation can be seen as an extension of the linear model presented in [1] since that linear model has one autoregressive equation for each pair (weekday, hour), i.e., 168 equations with 20 variables for a total of 3360 parameters, while this CNN model is the composition of 5 convolutions made by kernels having a parameter for each pair *(variable, time instant)*, namely, 22 variables and 168 time instants for each of the 5 convolutions plus 6 parameters used to compose the convolutions, plus 6 biases, for a total of 18491 parameters. The CNN model is trained using a dataset containing weather variables T, R_H, W_S, W_D, R, T^2, $T_{ma(7)}$, T_M, T_M^2, T_{Mp}, T_{Mp}^2, historical heating load variables $l_i, 1 \leq i \leq 7$ and social factors H, h, d, w (see Table 1 for variable definitions).

Weight initialization for each layer is performed by *Xavier* normal initializer [9]. We trained the model using Keras[2], splitting further training dataset into train (6500 observations) and validation (3460 observations) to improve the model selection and avoid overfitting problem, typical of neural networks. The weights are learned by optimized gradient descent Adam [13], for 20 epochs using batch sizes of 32. Early stopping procedure monitors the training process, saving the best set of weights that minimize a loss function computed at the end of each epoch as mean squared error (MSE) over the validation dataset.

2.4 Stochastic Variational Gaussian Process model

Let X be a finite set of input points $x_1, \ldots, x_n$ (they can be scalars or vectors), Gaussian processes assume the probability distribution of function values $p(f(x_1), \ldots, f(x_n))$ at those points to be jointly Gaussian, namely $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, K)$ where matrix K is called the *covariance matrix* or *kernel* [16,18]. It has dimension $n \times n$, where n is the number of inputs of the training set. During model training the kernel matrix is filled in with covariance values between all possible pairs of inputs in the training sets. A key point of GP model design is the choice of kernels. We use *periodic kernels* $k_{\text{Periodic}}(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x-x'|/p)}{\ell^2}\right)$ for modeling cyclical behaviors due to social factors and Radial Basis Function

[2] https://keras.io/

(RBF) kernels $k_{\mathrm{RBF}}(x, x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$ for environmental factors, such as temperature. The parameters of periodic kernel are the output variance σ, the period length p and the length scale l. Those of RBF kernels are the output variance σ and the length scale l.

The posterior distribution of the function values on the testing locations f_* (i.e. load predictions) is jointly Gaussian distributed with the function values f on the training locations. GP predictions provide both expected values and *confidence intervals*, which are extremely important in forecasting applications. The downside of this approach is its computational cost, the time and space complexity for training.

A solution to the complexity problem is approximate training methods. We use, in particular, *Stochastic Variational Gaussian Processes* (SVGPs) that use stochastic optimization to scale GP training to large datasets. The main idea of this model is to select a set of datapoints called *inducing inputs* or pseudo inputs on which the covariance matrix is generated. The position of these points in the dataset is optimized together with the model parameters through gradient-based optimization with the aim to maximize the evidence lower bound (ELBO) [11], a lower bound of the log-marginal likelihood. Improving the ELBO improves the variational posterior approximation by minimizing the Kullback-Leibler divergence between the true posterior and the variational approximation. Since inducing inputs variational parameters and not model parameters, they can be optimized without risk of overfitting. We perform SVGP training by *Adamax* [13] with *predictive log likelihood* loss function [12]. *Batches* of 256 points are used and 500 *inducing inputs* are chosen from all input dimensions. Parameter optimization was iterated for 100 *epochs*. Finally, the *Cyclical Learning Rate* (CLR) [21] method is used to optimize the learning rate of the model during the training phase. Models are trained using *GPyTorch*[3] on GPUs provided by *Google Colab*[4].

The model proposed in this work uses 13 variables, namely, T, R_H, W_S, W_D, R, T^2, $T_{ma(7)}$, T_M, T_M^2, T_{Mp}, T_{Mp}^2, *Sa and Su* (see Table 1). For each of these variables we introduce an RBF kernel because future heating load values should be inferred from past load values having similar values for these variables. Then we introduce two periodic kernels for considering the daily and the weekly cycle of the heating load due to social factors. We finally compose the two periodic kernels by summing them (to consider both periodicities) and we multiply the result by the product of all the RBF kernels of environmental factors. We multiply kernels because the effect of multiplication is similar to the intersection (logical *and*) of data filters, hence we predict future heating loads considering more important past loads having similar values of all social and environmental factors. The final kernel is $k_{\mathrm{F}}(\mathbf{x_1}, \mathbf{x_2}) = (k_{\mathrm{P\_24h}}(\mathbf{x_1}, \mathbf{x_2}) + k_{\mathrm{P\_168h}}(\mathbf{x_1}, \mathbf{x_2})) * \prod_{v \in V} k_{\mathrm{RBF\_v}}(\mathbf{x_1}, \mathbf{x_2})$ where $k_{\mathrm{P\_24h}}(\mathbf{x_1}, \mathbf{x_2})$ is the daily periodic kernel, $k_{\mathrm{P\_168h}}(\mathbf{x_1}, \mathbf{x_2})$ is the weekly periodic kernel and $k_{\mathrm{RBF\_v}}(\mathbf{x_1}, \mathbf{x_2})$ is the ard-version of the RBF kernel. We notice that in the proposed SVGP model, variables related to previous loads

[3] https://gpytorch.ai/

[4] https://colab.research.google.com

$l_i, 1 \leq i \leq 7$ are not used because the model intrinsically computes loads as a weighted sum of past loads corresponding to similar environmental and social conditions.

2.5 Performance measure

Performance is evaluated by Root Mean Square Error (RMSE) on 48-hours forecasting horizon. Given an observed time-series with n observations $y_1, \ldots, y_n$ and predictions $\hat{y}_1, \ldots, \hat{y}_n$, the formula is $RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2}$. Performance is evaluated on the overall test set, therefore we iterate the computation of the RMSE on a sliding window of 48-hours, moving from the beginning to the end of the test set. For example, starting from the first point p_1 we forecast the next 48-hours and compute the RMSE on the interval [p_1,p_{48}], then we move to the next point p_2 repeating the previous step on the interval [p_2,p_{49}], and so on. The measure thus obtained is called ($\overline{RMSE}$) in the following and it is the average RMSE over all 48-hours predictions in the test set. The RMSE was computed on a 48-hours basis for because of a specific application requirement.

3 Results

In this section we evaluate the proposed models, first analyzing the CNN and the SVGP independently, then for CNN we show some kernel parameters and how single CNN neuron signals are composed to generate the heating load prediction. For SVGP we display the kernels of a few single variables and their composition. Some details are provided to investigate the interpretability and the evaluation of models performance on test set.

3.1 CNN model

The CNN model described in the previous section (Figure 2.a) computes the heating load as a weighted sum of five signals (i.e., $f_0, \ldots, f_4$) generated by convolution of the multivariate input signal. One of the five kernels used to perform the convolution, namely kernel 0, is displayed as a heatmap in Figure 2.c, where rows are time instants (i.e., index 168 corresponds to one hour before and 0 corresponds to 168 hours before the current time), columns are variables and colors values of parameters.

Interestingly enough, temperature T and previews day load l_1 have value of parameters with a quite direct interpretation, although CNN are known to be hardly interpretable. In Figure 2.d we show the values of kernel for only these two variables in a line chart having time in the x-axis and value of parameter in the y-axis. Temperature parameters (blue line) are negative in 168 (i.e., one hour before the prediction instant) and they increase (from right to left) to about 0.05 moving towards 0 (i.e., one week before the prediction). These values

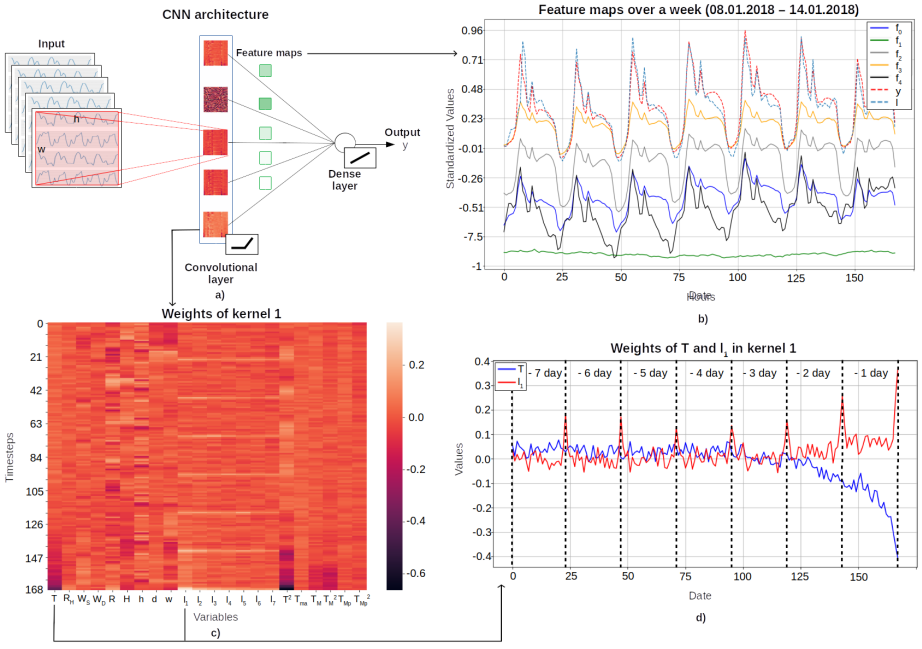
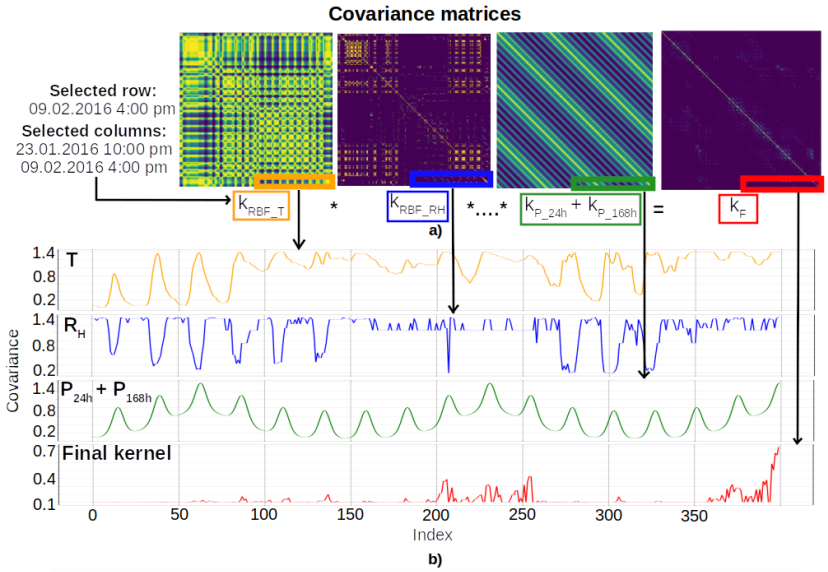


Fig. 2. a) CNN's architecture; b) Neuron outputs for one week of prediction; c) Kernel of neuron 1; d) Weights of T and l_1 variables in kernel 1.

show that the temperature in the previous two days have a negative impact on the heating load, and its informativeness becomes almost null after two days. This behaviour makes sense since we know that low temperatures imply high heating load to warm up buildings. Also the decrease of the absolute value of the parameter when moving back in time seems to make sense, since it means that recent temperatures are more informative than old one for the prediction. The parameters related to the load of the previous day (red line) have even more interesting behavior, with daily peaks that decrease from 168 to about 75 (from right to left) and then increase from 75 to 0. All peaks are positive because past load have a positive influence on future loads. The daily peaks show the social component of load, namely, to predict today's load at time $\bar{t}$ it is more informative the yesterday's load at $\bar{t}$ than loads at different hours of the day. This is because people usually warm up buildings differently in hours of the day. Moreover, the increase of the peak corresponding to indices 48 and 24 highlights the weekly pattern of the load, due to the fact that to predict heating load on day $\bar{d}$ (e.g., Sunday) it is more informative to use past loads observed in day $\bar{d}$ than in other days, because people use heating differently in days of the week.

Finally, the charts of Figure 2.b show the output of each of the 5 neurons of the convolutional layer (blue, green, gray, orange and black lines), the output of the network (red dashed line), and the true load (blue dashed line) for the

week from 08.01.2018 to 14.01.2018. We first observe that feature map f_1 has an almost constant negative value, while feature map f_4 shows the typical load peaks more than others. Considering the weights of the dense layer, i.e., $w_0 = 0.900, w_1 = -0.561, w_2 = 0.221, w_3 = 0.928, w_4 = 0.298, w_5 = 0.030$, we notice that all of them but w_1 are positive, and w_0 and w_3 have higher absolute values, hence feature maps f_0 and f_3 have stronger influence to the final prediction than others. Finally, the load prediction y which is the weighted sum of convolution signals is very similar to the true load signal l.



Kernels	Hyperparam.	Kernels	Hyperparam.
$k_{\text{P\_24h}}(\mathbf{x_1}, \mathbf{x_2})$	l = -2.1672 p = 23.9999	$k_{\text{RBF\_}W_S}(\mathbf{x_1}, \mathbf{x_2})$	l = 11.5921
$k_{\text{P\_168h}}(\mathbf{x_1}, \mathbf{x_2})$	l = 8.0042 p = 168.3604	$k_{\text{RBF\_}W_D}(\mathbf{x_1}, \mathbf{x_2})$	l = 14.7726
$k_{\text{RBF\_Sa}}(\mathbf{x_1}, \mathbf{x_2})$	l = 2.3707	$k_{\text{RBF\_}T_{Mp}}(\mathbf{x_1}, \mathbf{x_2})$	l = 3.8777
$k_{\text{RBF\_Su}}(\mathbf{x_1}, \mathbf{x_2})$	l = 1.1507	$k_{\text{RBF\_}T^2}(\mathbf{x_1}, \mathbf{x_2})$	l = 10.6826
$k_{\text{RBF\_T}}(\mathbf{x_1}, \mathbf{x_2})$	l = 2.4025	$k_{\text{RBF\_}T_M}(\mathbf{x_1}, \mathbf{x_2})$	l = 4.4703
$k_{\text{RBF\_}T_{ma(7)}}(\mathbf{x_1}, \mathbf{x_2})$	l = 0.7115	$k_{\text{RBF\_}T^2}(\mathbf{x_1}, \mathbf{x_2})$	l = 6.4616
$k_{\text{RBF\_}RH}(\mathbf{x_1}, \mathbf{x_2})$	l = 4.7928	$k_{\text{RBF\_}T^2_{Mp}}(\mathbf{x_1}, \mathbf{x_2})$	l = 3.9429
$k_{\text{RBF\_R}}(\mathbf{x_1}, \mathbf{x_2})$	l = 11.1029		

c)

Fig. 3. a) SVGP kernels; b) Highlight on kernel parameters (similarity) of observations taken on 09/02/2016 at 4:00pm against observations taken from 23/01/2016 at 10:00pm to 09/02/2016 at 4:00pm; c) Kernel hyperparameters after training.

3.2 SVGP model

The SVGP model is based on a product of $n \times n$ kernels. Figure 3.a shows the final kernel on the right and some of the factor kernels in the center. They

are all depicted as heatmaps in which yellow (and bright) to high parameter values and blue (and dark) corresponds to parameters close to zero. Notice that pictures show only a submatrix of each kernel and that each cell (i, j) of the final kernel contains the product of the corresponding (i, j) cells of all factor kernels. Patterns in the colors correspond to correlation patterns since each cell contains a similarity measure between two values. In Figure 3.b we explain them in the particular case of the row corresponding to 09.02.2016 at 4:00 pm. The x-axis value 400 corresponds to the same date and moving from 400 back to 0 the time decreases of one hour at each step. The day cycle is quite visible, since night temperatures are lower than the temperature of 09.02.2016 at 4:00 pm, hence the RBF kernel produces a smaller covariance value (see the periodical lower values in the chart). Similar pattern, although with differences, can be seen in the relative humidity kernel which is also RBF. Kernel ($k_{\text{P\_24h}} + k_{\text{P\_168h}}$) has a more stable trend with a clear daily period (with peaks at 4:00 pm) summed to a weekly period (with peaks on Tuesdays 09.02.2016). The corresponding values of the final kernel (red line) tend to be high only when all factor values are high. As the figure shows, recent values (close to 400) have high values since they are very important to predict the heating load of the next hour. The parameters (i.e., length scale and periodicity) of the various kernels are listed in Table 3.c.

3.3 Model comparison

The main properties of the two proposed models are listed in Figure 4.a, where also the autoregressive linear model presented in [1] is reported for comparison. The best performance is achieved by the CNN which however has a very high number of parameters. The SVGP model has a slightly worse performance than ARM but it has a small number of parameters and it provides confidence intervals on prediction, which is a key feature in some applications. The training time of the CNN is also low, because the network is very simple and the optimizer reaches good performance with a small number of training epochs. However, the training times of ARM and SVGP are also low for our application, since the model is required to be updated only every 24 hours.

In Figure 4.b we show the trend of RMSE for the CNN (blue line) and the SVGP (red line) on the test set (year 2018). Each point (x,y) represents a RMSE (y) of a 48-hours forecast starting at instant x. The blue and red points show the maximum and minimum RMSE of, respectively, the CNN and the SVGP model. CNN has the maximum RMSE on February 25th at 9:00 am with $3.295MWh$, while the SVGP model has it maximum RMSE on February 12th at 6:00 am with $3.184MWh$. Minimum RMSEs are achieved on February 5th at 10:00 am by CNN with $0.652MWh$, whereas for SVGP on April 14th at 9:00 am with $0.662MWh$. The 48-hours predictions that generate these (best and worst) performance are displayed in Figure 4.c-d-e-f). For SVGP confidence intervals are also provided. Blue lines represent the true load and red lines the predicted load. The heating load here displayed is standardized, to guarantee the privacy of the dataset, as requested by the utility company.

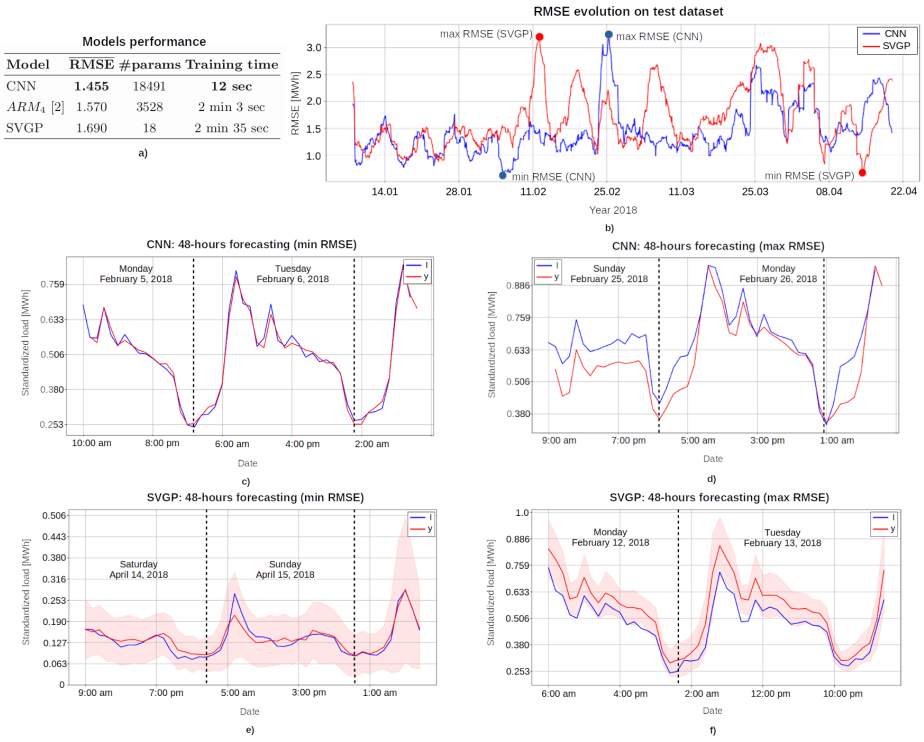


Fig. 4. a) Models performance; b) RMSE evolution of CNN and SVGP for each 48 hours forecast performed in 2018. Each point (x,y) represents the RMSE (y) of a 48 hours forecast starting at instant x; c-d-e-f) 48-hours forecasting with minimum and maximum RMSE error for CNN and SVGP.

4 Conclusion and ongoing work

CNN and SVGP models have been used to predict heating load in a real DHN. Results show that the CNN outperforms a state-of-the-art autoregressive linear regression model and the SVGP has slightly lower performance but it provides useful confidence intervals on the prediction. Both models have been analyzed and interpreted. More complex CNN architectures were also tested, obtaining slightly better performance in terms of $\overline{RMSE}$ (i.e., up to 1.397) but with a large increase of parameters (i.e., up to $121,255$). These architectures will be developed in future work and their interpretability will be further investigated. Future work concerns the improvement of model explainability and the integration of automatic feature engineering in neural network and Gaussian process based models.

Acknowledgments

The research has been partially supported by the projects "Dipartimenti di Eccellenza 2018-2022, funded by the Italian Ministry of Education, Universities and Research (MIUR), and "GHOTEM/CORE-WOOD, POR-FESR 2014-2020", funded by Regione del Veneto.

References

1. F. Bianchi, A. Castellini, P. Tarocco, and A. Farinelli. Load forecasting in district heating networks: Model comparison on a real-world case study. In *Machine Learning, Optimization, and Data Science (LOD)*, pages 553–565. Springer, 2019.
2. M. Blum and M. Riedmiller. Electricity demand forecasting using gaussian processes. In *Proc. 15th AAAIWS*, page 1013. AAAI Press, 2013.
3. A. Castellini, G. Beltrame, M. Bicego, D. Bloisi, J. Blum, M. Denitto, and A. Farinelli. Activity recognition for autonomous water drones based on unsupervised learning methods. In *Proc. 4th Italian Workshop on Artificial Intelligence and Robotics (AI*IA 2017)*, volume 2054, pages 16–21, 2018.
4. A. Castellini, M. Bicego, F. Masillo, M. Zuccotto, and A. Farinelli. Time series segmentation for state-model generation of autonomous aquatic drones: A systematic framework. *Engineering Applications of Artificial Intelligence*, 90, 2020.
5. A. Castellini and G. Franco. Bayesian clustering of multivariate immunological data. In *Machine Learning, Optimization, and Data Science (LOD)*, pages 506–519. Springer International Publishing, 2019.
6. A. Castellini, F. Masillo, R. Sartea, and A. Farinelli. eXplainable Modeling (XM): Data Analysis for Intelligent Agents. In *Proc. 18th Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2342–2344. IFAAMAS, 2019.
7. A. Dahl and E. Bonilla. Scalable gaussian process models for solar power forecasting. In *Data Analytics for Renewable En. Int.*, pages 94–106. Springer, 2017.
8. R. Frigola-Alcalde. *Bayesian Time Series Learning with Gaussian Processes*. PhD thesis, University of Cambridge, 2015.
9. X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.
10. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
11. J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational gaussian process classification. In *18th Int Conf. on Artif. Int. and Stat.*, 2015.
12. M. Jankowiak, G. Pleiss, and J.R. Gardner. Sparse gaussian process regression beyond variational inference. *CoRR*, 2019.
13. D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.
14. I. Koprinska, D. Wu, and Z. Wang. Convolutional neural networks for energy time series forecasting. In *2018 Int. Joint Conf. Neural Nets (IJCNN)*, pages 1–8, 2018.
15. R. Ramanathan, R. Engle, C.W.J. Granger, F. Vahid-Araghi, and C. Brace. Short-run forecast of electricity loads and peaks. *Int. J. Forecasting*, pages 161–174, 1997.
16. C.E. Rasmussen and C.K.I Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
17. M. Raza and A. Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Ren. Sust. En. Rev.*, 50:1352–72, 2015.

18. S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Phil. Trans. Royal Soc. (Part A)*, 371, 2013.
19. T. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *ICASSP*, pages 4580–4, 2015.
20. M. Shepero, D.V.D. Meer, J. Munkhammar, and J. Widn. Residential probabilistic load forecasting: A method using gaussian process designed for electric load data. *Applied Energy*, 218:159 – 172, 2018.
21. L.N. Smith. No more pesky learning rate guessing games. *CoRR*, 2015.