Predictive model generation for load forecasting in district heating networks

Alberto Castellini*, Federico Bianchi, and Alessandro Farinelli

District heating networks (DHNs) are promising technologies to increase the efficiency and reduce emissions of heat distribution to residential and commercial buildings. The advent of the smart grid paradigm has introduced the usage of heating load forecasting tools in DHNs. They provide estimates of future heating load, improving the planning of heat production and power station maintenance. In this work, we propose a methodology based on the integrated use of regularized regression and clustering for generating predictive models of future heating load in DHNs. The methodology is tested on a real case study based on a dataset provided by AGSM, an Italian utility company that manages a DHN in the city of Verona, Italy. We generate a set of multiple-equation models having different degrees of complexity and show that models generated by the proposed approach outperform those trained by standard methods. Moreover, we provide an interpretation of patterns encoded by these models, and show that they identify real operational states of the network. The approach is completely data-driven.

Index Terms—Heating load forecasting, predictive model generation, regularized regression, model clustering, model interpretability, time series analysis, dynamical systems.

1 Introduction

Predictive models are key tools in modern intelligent systems. They allow to forecast in advance future system states and to choose optimal actions, accordingly. Several application domains require increasingly accurate predictive models, from autonomous driving to churn prediction [1] , robotics and planning , to provide a few examples. In the context of smart grids [2], predictive models have recently gained strong interest because they enable to improve planning of energy production and power station maintenance. A specific branch of smart grids concerns *district heating networks* (DHNs), in which centralized heating plants generate the heat and distribute it through a pipe system via exchangers to residential and commercial buildings (see Figure 1.a). DHNs are acknowledged as a promising technology for heat distribution, since they ensure better efficiency and pollution control than standard heating systems.

The heating production strongly depends on the environment temperature, represented by variable T in Figure 1.a, but also other weather and social factors may influence it. In Figure 1.a we represent, for instance, the relative humidity (RH), the rainfalls (R), and holidays (H), an important social factor in this context. Load forecasting models are analytical tools designed to predict future heating load from these factors. In particular, past and present values of heating load, and past, present and future (predicted) values of social/weather factors are used to predict future values of the load. Different model variants can make predictions at different times in the future (e.g., one hour or one day) and use different modeling frameworks to generate the prediction. Another key problem concerns the identification of the most informative factors and lags to be used for prediction. In this work, we focus on model selection and assume the set of predictors to be fixed and provided by prior knowledge. Our problem here concerns instead

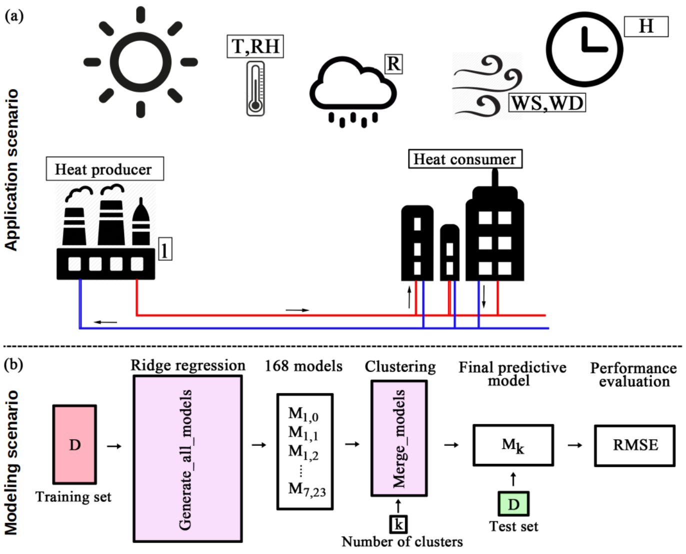


Fig. 1. System overview and problem definition. (a) District heating network and factors influencing the heating load. (b) Overview of the model generation process. Functions *Generate_all_models* and *Merge_models* are defined in Algorithm 1.

- *All authors are with the Department of Computer Science, University of Verona, Italy.*
E-mail: alberto.castellini@univr.it

Manuscript received Xxxxxx XX, XXXX; revised Xxxxxxx XX, XXXX.

the generation of parsimonious (i.e, compact, in terms of number of parameters) and interpretable predictive models able to outperform a model presented in [3].

We present a methodology based on the integrated usage of regularized regression and k-means clustering for generating compact and interpretable multiple-equation autoregressive models able to predict future heating load in DHNs. The approach is based on two steps: *i)* the partition of the observations in the dataset into N sets having same weekday and hour-of-the-day (e.g., all observations from Monday at midnight are grouped together, the same for observations of Monday at 1.00, and so on, until Sunday at 23.00), and use ridge regression to generate one autoregressive model for each weekday-hour pair; *ii)* the merging, by k-means clustering, of autoregressive models having similar parameters, obtaining models having $k \leq N$ autoregressive equations. Using this original strategy we reach two important goals: namely, we get compact, interpretable and accurate models for heating load forecasting, and we automatically discover different operational states of the DHN from these models. Importantly, these states are detected in a completely data-driven way. Moreover, each state is related to a specific set of weekday-hour pairs and it is represented by the parameters of an autoregressive model that predicts future loads only for these pairs. Parameters of autoregressive models correspond to cluster centroids computed by a clustering strategy that favours the grouping of autoregressive models having homogeneous parameters.

The proposed approach is applied to a real case study based on a DHN located in Verona, Italy. The dataset used contains hourly heating loads produced by 3 heating plants in 2016, 2017 and 2018. We first analyze and compare the performance of models having different degrees of compression (i.e., number of clusters). Then we select two compact models having good forecasting performance, analyze their parameters and the states of the DHN they represent.

The main contributions of this work to the state-of-the-art are summarized in the following points:

- we propose an original methodology based on the combination of ridge regression and k-means clustering for generating compact multiple-equation autoregressive linear models for load forecasting in DHNs;
- we apply the approach to a real case study and show that the generated models outperform models provided by standard methods;
- we analyze the dependence between model complexity and model performance;
- we analyze the parameters of two compact models and provide an interpretation of some clusters as states of the DHN.

The original contribution that mainly differentiates this work from other approaches, concerns the combination of regression and clustering methods for, respectively, building linear predictive models on subsets of samples having simple and homogeneous relationships between weather/social factors and heating load, and merging models having similar parameters. This allows to build compact models having different abstraction levels and possibly showing different mechanisms of the process under investigation. In the next sections the main differences between our method and the literature are also highlighted.

The rest of the manuscript is organized as follows. Section 2 presents the state-of-the-art on this topic. Section 3 describes the dataset, proposed methodology and performance measures used to evaluate the models. Results are analyzed in Section 4 and conclusions are drawn in Section 5.

2 Related work

We found two main connections between our work and the literature. One is methodological and the other concerns the application domain. From the methodological point of view, clusterwise regression [4] and mixture regression are the techniques more similar to ours. They aim to partition the dataset and, simultaneously, make regression models for each partition. They differ from our approach because they consider all possible sample partitions and do not use any prior knowledge to make the partition, as we do with the weekday-hour factor. Our methodology therefore solves a simpler problem but it is also more efficient, since it takes advantage of the prior knowledge about sample partitioning.

Regarding the specific application domain of load forecasting for smart grids, there exist an extended literature dating back to the eighties [2]. Some recent literature concerns also DHNs [5], [6], [7]. Multiple linear regression models and SARIMA models are used, considering both weather conditions and social components. Other relevant social components are calendar events [6], such as holidays which have different behaviours than working days, leading to systematic variations in the heating load. In [7] an optimization is developed on combined heat and power to minimize the production, distribution and net operating cost. In [6] linear regression, multilayer perceptron, and support vector regression are compared to forecast the heat load of a district heating system in Aarhus, Denmark. Models are trained on six years of hourly data including load, weather (outdoor temperature, wind speed, and solar irradiation) and social factors (hour-of-the-day, day-of-the-week, weekend, month-of-the-year, and holidays). Several other papers propose the usage of standard and extended methods to improve prediction performance, such as the recent [8], where linear regression, neural networks and support vector regression are used, together with feature selection, and compared to each other. Multiple-equation linear models with one equation for each hours of the day are built in [8] and [9] but those equations are not merged as we propose in this paper to improve model parsimony.

3 Material and Methods

In this section we formalize the problem, describe the dataset, introduce the model generation procedure and define the performance measures used to evaluate the model.

3.1 System overview and problem definition

An overview of the system under investigation is depicted in Figure 1.a. The heat is produced by a power plant (on the left) and distributed through a water pipe system to commercial and residential buildings (on the right). The red line represents hot water moving from power plant to buildings, and the blue line represents cold water moving back to the power plant. The main factors possibly affecting the heating load (l) are the temperature (T), relative humidity (RH), rainfalls (R), wind speed (WS), wind direction (WD) and holidays (H). The model generation process is summarized in Figure 1.b. We want to develop interpretable models for predicting heating load in the next 48 hours from informative variables contained in the training set. The 48-hours time horizon is useful for operational planning and control.

3.2 Dataset

We merged data from three sources, namely, a dataset of hourly heating loads for a DHN managed by an Italian utility company called AGSM (https://www.agsm.it/); weather data from a nearby meteorological station (https://rp5.ru/); and a dataset of Italian holidays (https://pypi.org/project/workalendar/). Data was collected from 01/01/2016 to 21/04/2018 (i.e., 842 days, 20208 observations, date format in the overall manuscript is *DD/MM/YYYY*). The first source (which is proprietary) provides data for three power stations that we summed together to obtain the total hourly load provided to the network. From the meteorological data archive (which is freely available) we took the five weather variables listed in the previous subsection (namely, *T, RH, R, WS, WD*) and processed them to obtain an hourly sampling interval. Holidays were mapped to a binary variable (0=working day, 1=holiday).

These variables were further manipulated to obtain the final training and test sets. 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). Then we engineered new variables expected to have predictive power in our context, according to similar applications in the literature [9]. The final list of twenty independent variables and one dependent variable (i.e, the heating load) is reported in Table 1 with short descriptions. Finally, we used standardized data from 2016 and 2017 (10128 observations) as a training set and from 2018 (2496 observations) as a test set.

3.3 Model generation procedure

The methodology proposed in this work aims to generate a model of heating load l at time t_j given the 20 independent variables of Table 1. For instance, to predict the load of next Monday at 8.00 we need the temperature (or some forecasts) in the previous week, the relative humidity and other weather variables of next Monday at 8.00, then we need to know if next Monday will be a holiday, and the heating load of the previous Sunday, Saturday, . . ., Monday, at 8.00 (i.e., variables $l_i, 1 \leq i \leq 7$).

A possible mathematical form for this model is a single equation linear regression model computed by the Ordinary Least Squares method [10] (we will call this model M_1^{OLS} in the following). However, to generate a good model we need to identify some invariants (i.e., rules or patterns) in the data. The main assumption we use in this work is that, in the context of district heating networks, data taken in the same *weekday-hour* (e.g., on Monday at 8.00), are characterized by specific statistical properties. We therefore use a *multi-equation* strategy, based on autoregressive models, wherein each equation deals with a specific weekday-hour. First, we learn an autoregressive model for each weekday-hour, obtaining 168 models (24 models for each weekday). Then we use k-means clustering [10] on vectors of autoregressive model parameters to merge models having similar parameters, achieving a more compact and interpretable final prediction model based on cluster centroids. In this model, clusters correspond to different *states* of the heating load and cluster centroids correspond to the parameters of "average" autoregressive models representing the invariants of these states. Notice that the weekday-hour factor here considered to partition the samples can be substituted with more complex factors, such as month-weekday-hour, that could provide information about seasonality. But seasonality can also be dealt with by introducing specific variables in the set of independent variables.

TABLE 1
List of variables used in the models.

Variable	Description
l	Heating load [MW] (target variable)
$l_i, 1 \leq i \leq 7$	Heating load i days before [MW]
l_p	Load in previous day peak (6:00 AM) [MW]
T	Temperature [°C]
T^2	Square of temperature [°C]
$T_{ma(7)}$	Moving avg of temperature last 7 days [°C]
T_M	Maximum temperature of the day [°C]
T_M^2	Square of maximum temperature of the day [°C]
T_{Mp}	Maximum temperature of the previous day [°C]
T_{Mp}^2	Square of max temperature of previous day [°C]
RH	Relative humidity [%]
WS	Wind speed [m/s]
WD	Wind direction [0, 9], 9=no wind
R	Rainfall (1 = rain, 0 = no rain)
H	Holiday (0 = false, 1 = true)

Algorithm 1, in the following, provides the pseudo-code and the mathematical notation of two procedures (also depicted in Figure 1.b) representing the core of the proposed approach. The first procedure is called *Generate_all_models*. It is used to generate the complete set of 168 autoregressive models, called $M_{d,h}, d = 1, \ldots, 7, h = 0, \ldots, 23$, where d represents the weekday and h the hour of the day. The second procedure is called *Merge_Models* and is used to generate more compact models, having k autoregressive equations (with $1 \leq k \leq 168$) represented by cluster centroids. We use the notation M_k to represent a generic prediction model composed of k autoregressive equations.

Ridge regression. Single autoregressive models are trained by *ridge regression* [10], instead of standard OLS, because it is more robust to variable correlations. In fact, some of the variables described in Table 1 are significantly correlated to each other, hence in some cases they could be interchangeably used to generate models having different parameters but similar behaviors. Ridge regression avoids this issue by imposing a constraint on parameter size that

Algorithm 1: Model generation procedure

```
Generate_all_models(D) // D: dataset
  for d in 1..7  // d: weekday
    for h in 0..23  // h: hour
      $M_{d,h}$=Ridge_regression($D_{d,h}$)
    end
  end
  return $\{M_{d,h}, d = 1, \ldots, 7, h = 0, \ldots, 23\}$
end

Merge_Models($\{M_{d,h}, d = 1, ..., 7, h = 0, ..., 23\}, k$)
  $M_k$=K_means($\{M_{d,h}, d = 1, \ldots, 7, h = 0, \ldots, 23\}, k$)
  return $M_k$
end
```

forces the solution with minimal L_2 norm to be selected among all solutions having the similar goodness of fit. The Lagrangian form of the optimization problem solved by ridge regression is shown in the following [10]:

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \quad (1)$$

where β^{ridge} is the vector of model parameters computed by ridge regression, n is the number of observations, y_i is the i-th observation of the dependent variable in the training set (i.e., the heat load in our case), x_{ij} is the value of the j-th independent variable (see Table 1) for the i-th observation of the training set, β_0 is the intercept, β_j is the j-th parameter of the regression model, and $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage. The larger the value of λ, the greater the amount of shrinkage. The optimal λ parameter is selected by 10-folds cross-validation. Due to the shrinkage constraint, model coefficients generated by ridge regression are more uniform than those generated by OLS hence they can be used in the subsequent clustering phase without having any problem of multiple clusters with same predictive behavior.

K-means. The k-means algorithm is a well known iterative descent clustering method [10] which aims at minimizing the objective function $J = \sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic} \parallel x_i - \mu_c \parallel^2$, where $r_{ic} \in \{0, 1\}$ is a binary indicator of point-cluster membership, x_i is a data point (i.e., a vector of model parameters in our case), μ_c is the centroid of cluster c, n is the number of data points and k the number of clusters. A clustering is a set of centroids that minimizes J.

We implemented our approach in Python using the Scikit-learn (https://scikit-learn.org) library. For ridge regression we used function *RidgeCV* with parameters $alphas \in [0.005, 5e^9]$, $intercept = True$, $normalize = False$, and $cv = 10$. For k-means we used function *KMeans* with Euclidean distance $\| \cdot \|^2$, and re-initialized the algorithm 100 times.

3.4 Performance measures

Models are evaluated by four indices, namely, the coefficient of determination (R^2) [10] on the training set, the root mean square error (RMSE) on the test set, the Akaike information criterion (AIC), and the number of parameters in the model.

Root mean square error (RMSE). Given a time-series with n observations $y_1, \ldots, y_n$ and a prediction $\hat{y}_1, \ldots, \hat{y}_n$, the RMSE is computed as [10],

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \quad (2)$$

and represents the average prediction error over all time instants in the time series. Since we are interested in evaluating our models on predictive horizons of 48 hours, we computed the average RMSE on all 48-hours predictions that can be made on the test set ($\overline{RMSE}$ in the following).

Akaike information criterion (AIC). Given a model with k parameters and a likelihood L, its AIC is [10],

$$AIC = 2 \cdot k - 2 \cdot log(L). \quad (3)$$

Since we used this formula to evaluate multi-equation autoregressive models, we substituted the log-likelihood $log(L)$ with a measure of goodness for this specific type of model, namely $\frac{n}{2} \cdot \log(\overline{RMSE}^2)$, where n is the number of samples in the test set.

4 Results

In this section we first present model M_{168}, namely, the model generated by procedure *Generate_all_models* and using one autoregressive model for each weekday-hour. Then we analyze the performance of models M_k with $k < 168$ (generated from model M_{168} by procedure *Merge_models*) and investigate the dependence of this performance on k. We finally focus our attention on a small subset of models, namely M_1, M_2, M_6 and M_{168}, having specific properties (i.e., smallest number of parameters, best AIC and best RMSE) and analyze their parameters, discovering a set of states of the heating network.

4.1 Model $\mathrm{M_{168}}$

Model M_{168} is composed of 168 autoregressive models. Each autoregressive model has 21 parameters (i.e., one parameter for each variable and one intercept) hence the total number of parameters is 3528. Figure 2 shows the distribution of parameters for each variable. For instance, the first box plot, on the left, shows the distribution of the intercepts across the 168 autoregressive models, which has a median of 14.091 and all values larger than 10. The second boxplot displays the distribution of parameters related to the temperature (i.e., variable T). The median is -2.456 and the interquartile range is negative with only few positive outliers, which is reasonable since temperature is usually negatively related to heating load (namely, when the temperature decreases the heating load increases). The opposite behavior is observed for variable l_1 (i.e., the heating load one day before) which has all positive parameters since it is directly related to the predicted heating load.

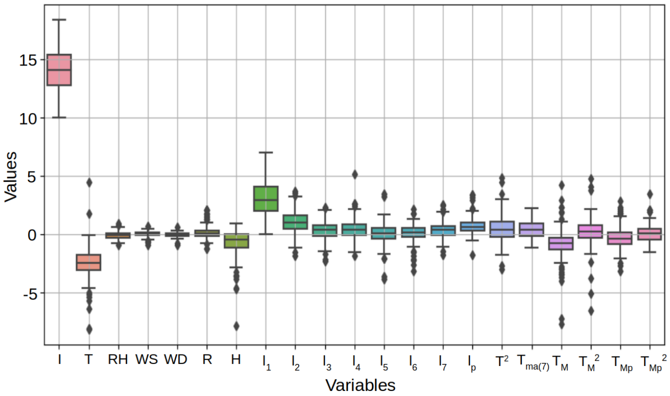


Fig. 2. Distribution of model parameters across the 168 autoregressive models of model M_{168}.

TABLE 2
Model performance.

Model	$\overline{\text{RMSE}}$	$\bar{\text{R}}^2$	AIC	# params
M_{168}^{OLS} [3]	1.570	0.984	14193	3528
M_{168}	**1.377**	0.980	12623	3528
M_6	1.657	0.953	**3968**	126
M_2	1.736	0.941	**3954**	42
M_1	1.855	0.930	4107	**21**
M_1^{OLS}	1.982	0.948	3822	21
M_1^{Ridge}	2.136	0.942	4287	21

4.2 Dependence of model performance on *k*

The question we answer in this section concerns the relationships between the number of autoregression models k (with $1 \leq k \leq 168$) and the performance of related model M_k. The analysis proposed in the following allows us to understand this relationship and to identify some models having good balance between performance and parsimony, which results also in the discovery of some states of the heating network. Models with $k < 168$, generated by procedure *Merge_Models*, are more parsimonious than model M_{168} in terms of number of parameters, but they have also lower performance due to the reduction of the degrees of freedom of the model. For each model M_k we compute the average coefficient of determination $\bar{R}^2$ (over all k autoregressive models in M_k), the $\overline{RMSE}$ (over all 48-hours predictions on the test set), and the AIC. Figure 3.a shows the distribution of $\bar{R}^2$ of models M_1, M_2, M_3, M_6, M_{100} and M_{168} (only 6 models are displayed to improve the readability of the figure). Model M_{168} has $\bar{R}^2$ equal to 0.980 (see the second row of Table 2 and the red point on the right hand side of Figure 3.a), which is the highest since the model employs the largest number of autoregressive models. As k decreases, the $\bar{R}^2$ decreases as well, but the performance are high for all k, with a minimum $\bar{R}^2$ of 0.930 for model M_1 (see Table 2). The chart also shows the $\bar{R}^2$ of models M_1^{OLS} and M_1^{Ridge}, that are generated using ordinary least squares (OLS) [10] and ridge regression, respectively, on the entire training set (i.e., without splitting it by weekday-hour). Moreover, we display the mean performance of model M_{168}^{OLS} which has 168 autoregressive models and is generated by OLS instead of ridge regression. The coefficient of determination of M_1^{OLS} and M_1^{Ridge}, respectively 0.948 and 0.942 (see Table 2), are slightly better than that of M_1, and the $\bar{R}^2$ of M_{168}^{OLS}, namely 0.984, is slightly better than that of M_{168}. We see in the next paragraph that this order of performance changes when performance is computed on the test set instead of on the training set, showing that models computed by Algorithm 1 have improved generalization capability than standard models.

Figure 3.b provides information about model $\overline{RMSE}$ on the test set. The values of $\overline{RMSE}$ of models computed by Algorithm 1 (see red points connected by a red line in Figure 3.b, and related values in the second column of Table 2) range from 1.377 of model M_{168} to 1.855 of model M_1. Boxplots show the distribution of $\overline{RMSE}$ for each M_k. Interesting enough, the $\overline{RMSE}$ of model M_{168}, which is computed by ridge regression, is lower than that of M_{168}^{OLS} (namely, 1.570), that is the best model presented in [3] (see the two red points on the right of Figure 3.b). This improvement is achieved by the regularization term introduced by ridge regression, which enhances model generalization. Another very interesting result concerns the lower $\overline{RMSE}$ of M_1 with respect to M_1^{OLS} and M_1^{Ridge} (respectively 1.982 and 2.136 on the left of Figure 3.b). It shows that the proposed methodology outperforms standard model generation methods, such as OLS and also ridge regression, if they are applied to the complete dataset.

Since both $\bar{R}^2$ and $\overline{RMSE}$ improve when the number of autoregressive models increases, we use the AIC to identify models having a good balance between parsimony and prediction performance. We are in fact interested in compact models because it is realistic to believe that there is a high level of redundancy in model M_{168}. Figure 3.c shows that the AIC has two local minima in $k = 2$ and $k = 6$, with similar AIC values of 3954 and 3968, respectively (see also Table 2). The AIC then increases in model M_1, because of the poor goodness of fit, and in models M_k with $k \geq 7$, because they use many parameters.

Figure 3.d compares the RMSE of models M_1 (red), M_6 (green) and M_{168} (blue) on all predictions made on the test set. The x-axis contains dates from 07/01/2018 to 22/04/2018, the y-axis shows RMSE values. Each point represents the RMSE of the 48-hours prediction starting from the date in the x-axis. Model M_1 is the most compact but it has the worst performance (i.e., the highest RMSE), model M_{168} is the most complex and has the best performance (i.e., lower RMSE), and models M_2 and M_6 have intermediate performance. We notice that in some points, such as point P_1 in Figure 3.d, all the models have a peak of RMSE, while in other points, such as point P_2 and P_3, only some models (namely, M_1 and M_6) have an error increase. Figure 3.e, 3.f and 3.g show the 48-hours prediction of models M_6 (green line) and M_{168} (blue line), with the true heating load of the network (red line) in points P_1, P_2 and P_3, respectively.

4.3 Parameters and interpretation of selected models

Models M_2 and M_6 have, respectively, 42 and 126 parameters (see Table 2), that correspond to a reduction of 99.10% and 96.43% of parameters with respect to model M_{168}. This strong shrinkage yields, however, a quite limited decrease

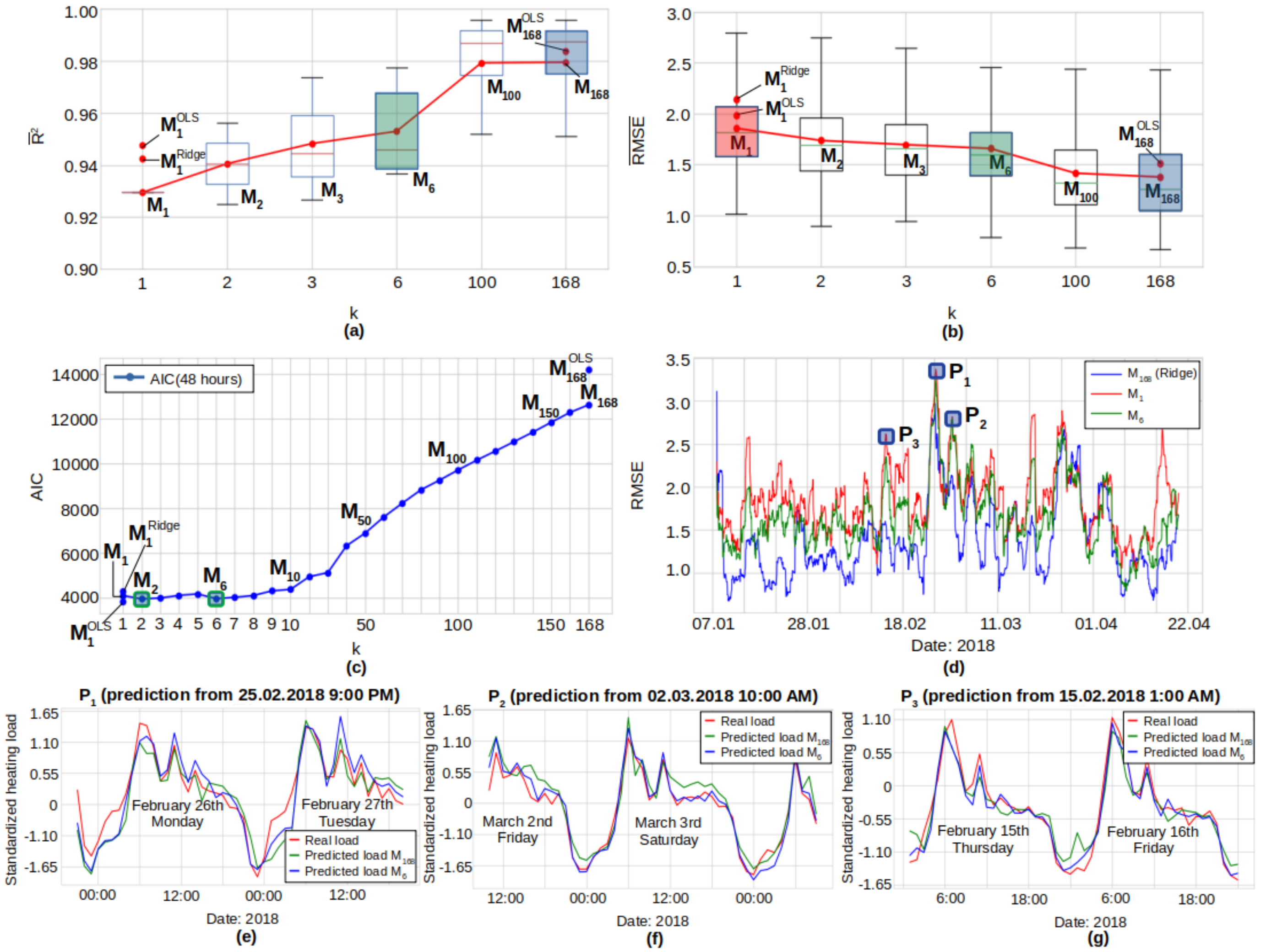
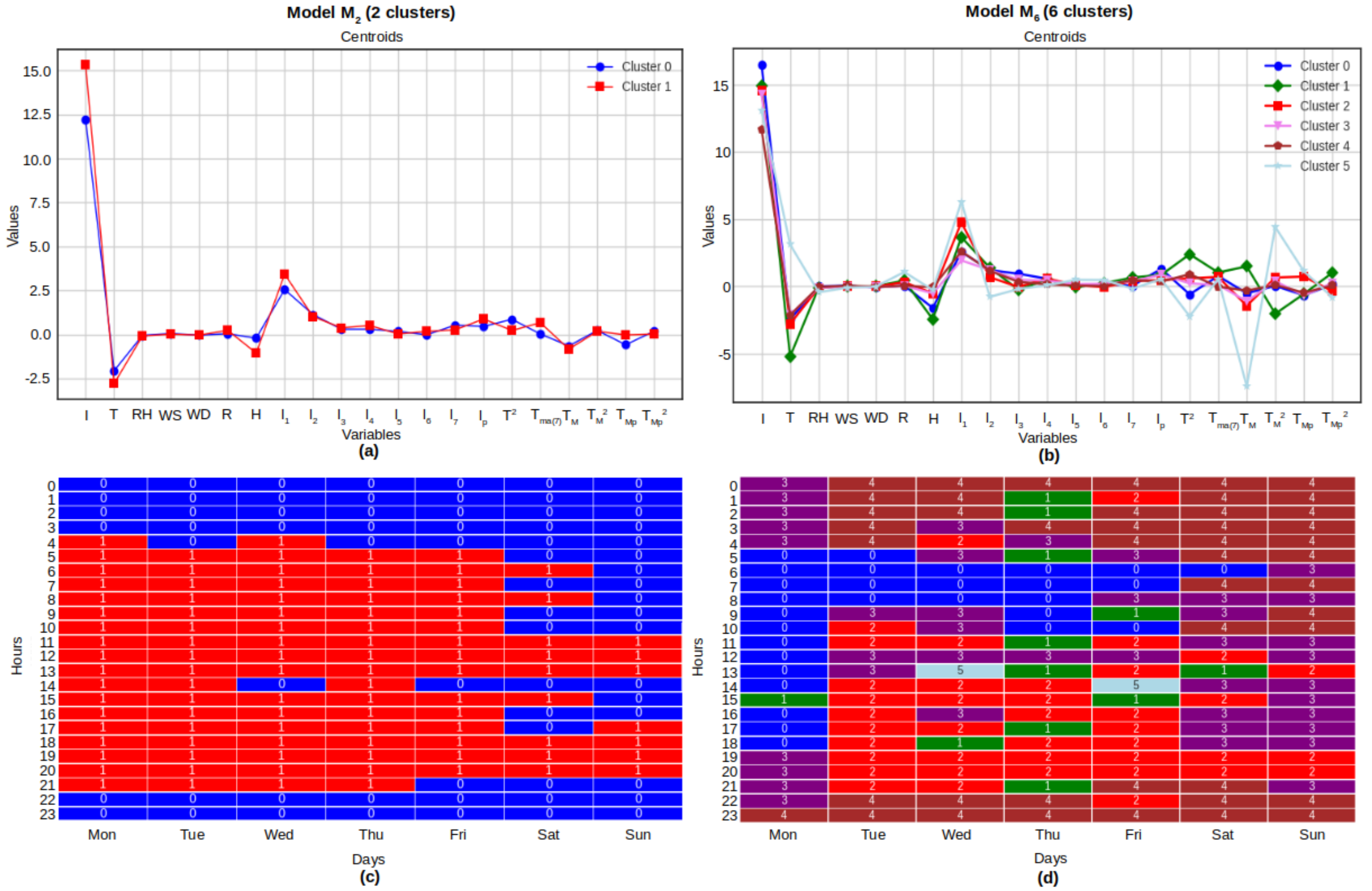


Fig. 3. Model performance. (a) Dependence of $\bar{R}^2$ on k; (b) Dependence of $\overline{RMSE}$ on k; (c) Dependence of AIC on k; (d) RMSE of models M_1, M_6 and M_{168} over the test set; (e)(f)(g) comparison between real evolution of load and predictions of models M_6 and M_{168} for three peaks of error P_1, P_2 and P_3.

of performance, since the $\overline{RMSE}$ of M_2 grows by 26.1% (from 1.377 to 1.736, see Table 2) and the $\overline{RMSE}$ of M_6 grows by 20.3% (from 1.377 to 1.657). This behavior points out that the proposed merge strategy, based on k-means, preserves the prediction capabilities of the final model while clustering similar autoregressive models. The analysis of cluster centroids (i.e., parameters of models M_2 and M_6) and related weekday-hour pairs therefore provides model interpretation and allows the identification of states of the heating network, as explained in the following.

Figures 4.a and 4.b show the coordinates of cluster centroids for models M_2 and M_6, respectively. The x-axis contains the list of variables, namely, the intercept (I) and all independent variables of Table 1. The y-axis displays the value of variable parameters. Different centroids (corresponding to different autoregressive models in a model M_k) are depicted by different colors. Figures 4.c and 4.d show, by two heatmaps, the distribution of clusters across weekdays (columns) and hours of the day (rows) for models M_2 and M_6, respectively.

Focusing on model M_2 (see Figures 4.a and 4.c), we observe that cluster C_0 (in blue) is characterized by an intercept of 12.227 which is smaller than the intercept of cluster C_1, namely 15.322. Other significant differences are present, for instance, between the parameters of temperature (see variable T in the x-axis), which have a value of -2.065 for cluster C_0 and of -2.279 for cluster C_1. Another parameter of interest is *holiday* (variable H) which has value -0.212 for cluster C_0 and -1.059 for cluster C_1. These differences can be directly interpreted in terms of heating network states. The properties of cluster C_1 say that this cluster represents a state in which larger heating load is provided (under the same conditions of other variables) than in the state represented by C_0 (according to the higher intercept). Moreover, the heating load is more influenced by the *temperature* in state C_1 than in state C_0. The same holds for variable *holiday*, namely, the heating load is more influenced by holidays in state C_1 than in state C_0. All these factors identify two specific states of the network, that are also projected to different intervals of hour of the days and weekdays, as clearly displayed in Figure 4.c. This figure shows that cluster C_0 (blue) mainly corresponds to time intervals when the network is little used, i.e., night and weekends, while cluster C_1 (red) corresponds to time



Hours	Mon	Tue	Wed	Thu	Fri	Sat	Sun
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	1	0	1	0	0	0	0
5	1	1	1	1	1	0	0
6	1	1	1	1	1	1	0
7	1	1	1	1	1	0	0
8	1	1	1	1	1	1	0
9	1	1	1	1	1	0	0
10	1	1	1	1	1	0	0
11	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1
14	1	1	0	1	0	0	0
15	1	1	1	1	1	1	0
16	1	1	1	1	1	0	0
17	1	1	1	1	1	0	1
18	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1
21	1	1	1	1	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0

Hours	Mon	Tue	Wed	Thu	Fri	Sat	Sun
0	3	4	4	4	4	4	4
1	3	4	4	1	2	4	4
2	3	4	4	1	4	4	4
3	3	4	3	4	4	4	4
4	3	4	2	3	4	4	4
5	0	0	3	1	3	4	4
6	0	0	0	0	0	0	3
7	0	0	0	0	0	4	4
8	0	0	0	0	3	3	3
9	0	3	3	0	1	3	4
10	0	2	3	0	0	4	4
11	0	2	2	1	2	3	3
12	0	3	3	3	3	2	3
13	0	3	5	1	2	1	2
14	0	2	2	2	5	3	3
15	1	2	2	2	1	2	3
16	0	2	3	2	2	3	3
17	0	2	2	1	2	3	3
18	0	2	1	2	2	3	3
19	3	2	2	2	2	2	2
20	3	2	2	2	2	2	2
21	3	2	2	1	4	4	3
22	3	4	4	4	2	4	4
23	4	4	4	4	4	4	4

Fig. 4. Detail on models M_2 and M_6. (a) Centroids (i.e., parameters of autoregressibe models) of model M_2; (b) Centroids of model M_6; (c) Distribution of clusters in model M_2 across weekdays and hours; (d) Distribution of clusters in model M_6 across weekdays and hours.

instants when the heating network is more strongly used, i.e., daytime and working days.

The proposed methodology allows to enhance the granularity of the discovered states by increasing the number of clusters k. In this way more specific states can be identified. The highest granularity is provided by model M_{168} but in our investigation we noticed that the more interesting states, able to generalize over similar behaviors of the network, are found with $k < 10$. Here we show the clusters identified by model M_6, which has the second best AIC after model M_2. The distribution of clusters over weekdays and hours of the day, in Figure 4.d, shows four big clusters and two small ones. Interesting enough, two of the big clusters split the "strong usage" state (i.e., C_1) of M_2, and the other two big clusters split the "little usage" state (i.e., C_0) detected by M_2. In particular, cluster C_0 of model M_6 (in blue) is specialized on very strong usage of the network, which occurs on peak hours (around 6.00 during working days and Saturday) and on Monday, namely, when people turn on the heaters after the night or after the weekend. Cluster C_2 of model M_6 (in red) represents a state of strong but slightly lower usage of the network, which mainly occurs in the afternoon and evening. On the other hand, clusters C_3 and C_4 of model M_6 split in two parts cluster C_0 of model M_2 (i.e., low usage of the network).

The remaining clusters C_1 and C_5 of model M_6 group together only 12 and 2 weekday-hour pairs, respectively. These clusters have different parameters than other clusters and they mainly concern time slots on Thursday (cluster C_1), and Wednesday/Friday at 13.00/14.00 (cluster C_5). As such, it is difficult to associate these two clusters to clear operational states (e.g., holidays or week-end), however we cannot exclude the presence of specific behaviors on those time slots because the heating load signal is very complex and is generated by behaviours of both residential and commercial consumers. A key point to highlight is that these clusters are statistically significant because *i)* their average coefficient of determination is very high (i.e., $\bar{R}^2 = 0.98$ for C_1 and $\bar{R}^2 = 0.99$ for C_5) hence they correctly fit the training data, *ii)* they are compact and well separated clusters (i.e., silhouettes are 0.71 and 0.88, respectively, and silhouettes of clusters C_0, C_2, C_3 and C_4 are 0.58, 0.72, 0.58 and 0.60, respectively), *iii)* they are stable over different number of clusters (i.e., cluster C_1 appears in model M_4 and cluster C_5 appears in model M_5, and they both keep their configuration of weekday-hours fixed until M_6 and beyond), *iv)* they provide performance improvements.

A possible interpretation for cluster C_5 is obtained by observing that its centroid has a positive parameter (i.e., 3.117) for temperature (T), and a strongly negative parameter (i.e., -7.428) for the maximum temperature of the previous day (T_M). Since the maximum temperature in winter (when the heating is on) is at about 13.00 or 14.00, and this cluster concerns the same hours we can observe that the linear combination of the two variables $3.511T_M - 7.111T$ actually represents a weighted difference between the maximum temperature of the previous day (T_M) and the maximum temperature at the current day. Hence this parameter configuration could suggest that in the specific time slots of cluster C_5 the composed variable related to temperature difference is more informative than variables selected in other clusters for predicting the heating load. Feature engineering is out of the scope of this work but this result is interesting because it suggests novel informative variables to insert in the variable set. Our analysis also confirmed that temperature variations between consecutive days can strongly affect the prediction performance (peak P_1 in Figure 3 corresponds to a decrease of the daily average temperature from $7°C$ to $-1°C$ in 24 hours). Cluster C_1 follows similar principles.

To conclude this section we briefly provide a performance comparison with a popular modeling framework, namely Convolutional Neural Networks (CNNs), to show that the proposed method has comparable performance on the same dataset. The best model we generated with a relatively small CNN model has two layers, six neurons, 18491 parameters and $\overline{RMSE}$ of 1.455. Larger networks reach only slightly better performance. This performance is comparable with that of more parsimonious linear models presented in this paper (in particular, it is slightly worse than M_{168} and slightly better than M_6) but the number of parameters used by CNNs is much higher. Similar results have been achieved with Long Short-Term Memory (LSTM) networks. The main motivation for these results seems to be that CNN and LSTM models were trained on the overall dataset, which contains complex dependencies between weather/social factors and heating load, while our multiple-equation linear models were trained on subsets of samples in which simpler relationships are present. Neural networks cannot be trained on the same subsets of samples because they need large number of samples to avoid overfitting. These results further support our strategy based on sample partitioning, generation of simple models on those partitions, and fusion of similar models using clustering.

5 Conclusion

In this paper we propose a methodology for generating interpretable predictive models in the context of load forecasting in district heating networks. Our analysis shows that in datasets having a predefined partitioning structure, such as, weekday-hour, a predictive model can be generated by first creating an autoregressive model for each subset of observations and then merging, by clustering, similar autoregressive models having homogeneous parameters. We used ridge regression for estimating parameters of single autoregressive models, since it is robust to variable correlation, and k-means for model clustering. Results show that *i)* predictive models generated by the proposed technique maintain a good accuracy also when a small number of clusters k is used, *ii)* the clustering of autoregressive models yields interpretable “average” models representing significant states of the heating network. Future work concerns the integration of the current approach with (interpretable) feature engineering and feature selection methods able to automatically generate and select variables that can improve prediction performance while preserving model interpretability.

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] P. Berger and M. Kompan, “User modeling for churn prediction in e-commerce,” *IEEE Intelligent Systems*, vol. 34, no. 2, pp. 44–52, March 2019.

[2] P. Mirowski, S. Chen, T. Ho, and C. Yu, “Demand forecasting in smart grids,” *Bell Labs Technical Journal*, pp. 135–158, 2014.

[3] 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 2019*. Springer International, 2019, pp. 553–565.

[4] Y. W. Park, Y. Jiang, D. Klabjan, and L. Williams, “Algorithms for generalized clusterwise linear regression,” *INFORMS Journal on Computing*, vol. 29, no. 2, pp. 301–317, 2017.

[5] M. Gong, H. Zhou, Q. Wang, S. Wang, and P. Yang, “District heating systems load forecasting: a deep neural networks model based on similar day approach,” *Advances in Building Energy Research*, pp. 1–17, 2019.

[6] M. Dahl, A. Brun, O. Kirsebom, and G. Andresen, “Improving short-term heat load forecasts with calendar and holiday data,” *Energies*, 2018.

[7] T. Fang and R. Lahdelma, “Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system,” *Applied Energy*, vol. 179, pp. 544–552, 2016.

[8] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk, and C. Johansson, “Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods,” *Energy*, vol. 157, pp. 141 – 149, 2018.

[9] R. Ramanathan, R. Engle, C. Granger, F. Vahid-Araghi, and C. Brace, “Short-run forecast of electricity loads and peaks,” *International Journal of Forecasting*, pp. 161–174, 1997.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2009.

Alberto Castellini is assistant professor at the University of Verona, computer science department. His current research mainly focuses on artificial intelligence, statistical learning and data analysis, in particular on time series analysis and autonomous planning under uncertainty. He received his Ph.D. in computer science from Verona University and had research and teaching experiences at both Verona University and Potsdam University/Max Planck Institute in Potsdam, Germany. He co-authored about 12 peer-reviewed scientific contributions in international journals (among them EAAI, RAS) and about 25 peer-reviewed contributions in international conferences (among them IJCAI, AAMAS and ACM SAC). Contact him at alberto.castellini@univr.it.

Federico Bianchi is a research scholarship holder at the University of Verona, computer science department. He received his Master’s degree in Computer Science and Engineering, Computer systems security curricula, from the University of Verona. His research interests are related to statistical learning, machine learning, predictive modeling and their interdisciplinary applications to security domain. Contact him at federico.bianchi@univr.it.

Alessandro Farinelli is associate professor at University of Verona, since December 2014. His research interests comprise theoretical and practical issues related to the development of Artificial Intelligent Systems applied to robotics. In particular, he focuses on coordination, decentralised optimisation and information integration for Multi-Agent and Multi-Robot systems, control and evaluation of autonomous mobile robots. He was principal investigator for several national and international research projects in the broad area of Artificial Intelligence for robotic systems. He co-authored more than 100 peer-reviewed scientific contributions in top international journals and conferences. Contact him at alessandro.farinelli@univr.it.