Short-term forecasting of Italian residential gas demand

Andrea Marziali$^{1,2}$, Emanuele Fabbiani$^2$, and Giuseppe De Nicolao$^2$

$^1$Modelling and pricing, A2A
$^2$Department of Electrical, Computer and Biomedical Engineering, University of Pavia

19th February 2019

Abstract

Natural gas is the most important energy source in Italy: it fuels thermoelectric power plants, industrial facilities and domestic heating. Gas demand forecasting is a critical task for any energy provider as it impacts on pipe reservation and stock planning. In this paper, the one-day-ahead forecasting of Italian daily residential gas demand is studied. Five predictors are developed and compared: Ridge Regression, Gaussian Process, k-Nearest Neighbour, Artificial Neural Network, and Torus Model. Pre-processing and feature selection are also discussed in detail. Concerning the prediction error, a theoretical bound on the best achievable root mean square error is worked out assuming ideal conditions, except for the inaccuracy of meteorological temperature forecasts, whose effects are properly propagated. The best predictors, namely the Artificial Neural Network and the Gaussian Process, achieve an RMSE which is twice the performance limit, suggesting that precise predictions of residential gas demand can be achieved at country level.

Keywords — natural gas; time series forecasting; statistical learning; Gaussian Process; neural networks.
1 Introduction

In Italy, natural gas is the most common fuel for both power plants and domestic heating. Moreover, several industrial facilities burn gas for either heating and powering productive processes. According to SNAM Rete Gas [1], the Italian Transmission System Operator (TSO), in 2017 about 70.59 billions of cubic meters of natural gas were consumed, with an increase of 5.6% over the previous year. Overall, the increase in demand between 2015 and 2017 was 11%. Out of the total gas demand in 2017, 35.9% was due to thermoelectric power plants, 22.4% to industrial facilities and 41.7% to residential users.

Forecasting natural gas demand is a key task for energy companies for several reasons. It provides relevant information to effectively reserve pipe capacity and plan stocks. Furthermore, energy regulations impose the balance of the network by charging providers with a fee proportional to their unbalanced quantity. Finally, demand is a critical input to forecast gas price, which is in turn an important driver for several business decisions.

Comprehensive reviews of the literature on gas demand forecasting are [2] and [3]. According to Sebalj et al, papers can be classified along four dimensions. The prediction horizon can range from hourly to yearly, the reference area from single nodes of the network to a whole country; adopted models include time series, mathematical and statistical approaches, neural networks, and others; input features can be demand history, temperature, calendar, and other minor ones. Our study focuses on day-ahead forecasting of Italian Residential Gas Demand (RGD). We develop and compare five models with an input dataset made of 22 features derived from the calendar, past gas demand and forecasted temperature.

Several previous studies have focused on country- or regional-level daily forecasting. Mathematical and statistical models based on parametric non-linear functions have been used in [4] to explain the factors which affect the demand. A different multi-factor approach has been developed in [5] and a model based on the physical relation between gas demand and temperature has been presented in [6]. An adaptive network-based fuzzy inference system (ANFIS) has been described in [7], where the authors showed the better performances of their model with respect to ANN and conventional time series methods. A statistical learning model, based on support vector machine (SVM), has been developed in [8] for UK demand, and compared to ANN and an autoregressive moving average (ARMA) predictor. A hybrid model, exploiting many different techniques such as wavelet transform, genetic algorithm, ANFIS and ANN, has been used in [9].

Neural networks are popular and effective models to predict gas demand: they have been applied in [10, 11, 12, 13, 14] to perform hourly and daily forecasts on cities and regions. [15] has shown how ANNs, combined with Principal Components Correlation Analysis (PCCA), provide robust and precise forecasts on regional demand.

Several studies about long-term forecasting have also been presented: [16] has discussed gas demand in Bangladesh, showing how population growth and Gross Domestic Product (GDP) are important drivers of the demand. Similar conclusions have been achieved in [17], where a breeder model has been used to outperform other approaches in forecasting Turkish demand.

To the best of our knowledge, no study about Italian RGD has been published yet. This paper aims at filling the gap by providing: (i) an analysis of the features of the series, (ii) a discussion of preprocessing and feature selection steps, (iii) a comparison among five predictors. In absence of a benchmark, a theoretical bound on the best achievable root mean square error is worked out assuming ideal conditions, except
for the inaccuracy of meteorological temperature forecasts, whose effects are properly propagated.

The paper is organized as follows. In Section 2, we formulate the problem and present the available data. In Section 3, we provide a statistical characterization of target and input variables, discussing both preprocessing and feature selection. Section 4 describes models, including training process and hyperparameter tuning. In Section 5 we derive the performance limit, which is used as the ultimate benchmark in Section 6, where the results are presented and discussed. Finally, Section 7 is devoted to some concluding remarks.

2 Problem Statement

The task addressed in this paper is the one-day-ahead forecasting of daily Italian Residential Gas Demand (RGD). RGD represents the main part of the overall Italian gas consumption, accounting for household usage for cooking, water heating and, most importantly, environment heating.

The available dataset covers 11 years, from 2007 to 2017, and is made of 3 fields: date (t), forecasted average temperature (\( \hat{T} \)) and residential gas demand (RGD). Forecasted temperature is provided by Meteologica, a well-known service, and is relative to the Northern regions of Italy. In the preliminary analysis, we also took into consideration a weighted average of the temperatures in different zones of Italy, but we noticed a weaker correlation with RGD. This is explained by the role of domestic heating in Northern Italy, where winters are colder than in other regions.

The profile of RGD from 2007 to 2017 is displayed in Fig. 1.

![Figure 1: Italian Residential Gas Demand (RGD): years 2007-2017.](image-url)
3 Exploratory analysis and feature selection

3.1 Residential Gas Demand

RGD magnitude greatly oscillates with the season: during the cold months, from October to March, it represents about 56% of the overall Italian demand, while it drops to about 28% during the warm months, from April to September. In fact, when the temperature climbs above 17-18 Celsius degrees, domestic heating is typically switched off. Thus, during the cold period lower temperatures cause a larger RGD, while, during summer, weather influence is negligible, while a seasonal pattern becomes evident, with lower RGD during weekends compared with working days. Due to the lack of dependence on weather conditions, the profile of summer RGD is remarkably repeatable from year to year. All these features are visible in Fig. 2, which displays eleven years of Italian RGD, overlapped with a proper shift to align weekdays.

![Figure 2: Italian Residential Gas Demand (RGD): years 2007-2017. The time series are shifted to align weekdays: weekly periodicity is particularly visible in summer. The yearly seasonal variation is mostly explained by heating requirements. In the inset, two weeks of July’s demand are zoomed.](image)

As expected, the autocorrelation function, estimated on the whole dataset, exhibits a clear yearly seasonality and a much smaller weekly periodicity, see Fig. 3.

Most of the spectral density, see Fig. 4, is concentrated at period 365.25 days. A smaller yet relevant spike can be found at a period of 7 days, accounting for the weekly periodicity. In both cases, smaller peaks at lower periods are ascribable to harmonics.
Figure 3: RGD autocorrelation function estimated on 2007-2017 data. The 365-day yearly periodicity is evident. In the inset, weekly waves witness the presence of a 7-day periodicity of smaller amplitude.

Figure 4: RGD periodogram. Left panel: periods from 0 to 8 days; right panel: periods from 0 to 500 days. The yearly periodicity is highlighted by peaks at 365.25 days, while the weekly one by the smaller spike at a period of 7 days. Other notable values are caused by harmonics.

The autocorrelation of lag 1 can be assessed through the scatter plot in Fig. 5a, where RGD at time $t$ is plotted against RGD at time $t - 1$. The correlation coefficient computed on the entire dataset is 0.988, and it increases to 0.995 if Saturdays and Mondays are discarded. This is an evidence of a different behavior between working days and weekends, visually confirmed in the plot, where Monday’s RGD (orange dots) stays in the upper part of the cloud whereas Saturday’s RGD (green dots) lies in the lower part.

As for the lag-7 autocorrelation, in Fig. 5b the scatter plot of RGD at times $t$ and $t - 7$ is displayed. The scatter plot in Fig. 5b is narrower when the demand is low, that is during warm months, while it gets wider in winter when the demand is high. This is due to the variability of weather from one week to the next one.
In order to characterize the yearly seasonality, it is convenient to introduce the notion of similar day. The following definitions hold:

- \( \text{year}(t) \) is the year to which day \( t \) belongs;
- \( \text{weekday}(t) \) is the weekday of day \( t \), e.g. Monday, Tuesday, etc;
- \( \text{yearday}(t) \) is the day number within \( \text{year}(t) \) starting from January 1, whose \( \text{yearday} \) is equal to 1.

**Definition 1** (Similar Day). If \( t \) is not a holiday, its similar day \( \tau^* = \text{sim}(t) \) is

\[
\tau^* = \arg \min_{\tau} |\text{yearday}(\tau) - \text{yearday}(t)|
\]

subject to

- \( \text{year}(\tau) = \text{year}(t) - 1; \)
- \( \text{weekday}(\tau) = \text{weekday}(t); \)
- \( \tau \) is not a holiday.

If \( t \) is a holiday, its similar day \( \tau^* = \text{sim}(t) \) is the same holiday in the previous year.

According to the Italian calendar, holidays are: 1 January, 6 January, 25 April, 1 May, 2 June, 15 August, 1 November, 8, 25 and 26 December, Easter and Easter Monday.

The relationship between RGD and RGD in the similar day is shown in Fig. 5c: again, the correlation is higher when the demand is lower, due to the smaller influence of temperature.

It can also be of some interest to take into account the similar day of \( t - 1 \). The scatter plot in Fig. 5d shows that the difference \( \text{RGD}(t - 1) - \text{RGD}(\text{sim}(t - 1)) \) is a good proxy to the difference \( \text{RGD}(t) - \text{RGD}(\text{sim}(t)) \).

Due to these considerations, we use \( \text{RGD}(t - 1) \), \( \text{RGD}(t - 7) \), \( \text{RGD}(\text{sim}(t)) \), and \( \text{RGD}(\text{sim}(t - 1)) \) as inputs to forecast RGD at time \( t \).
Figure 5: Scatter plots between RGD and potential features to be used for its prediction.

3.2 Temperature

The RGD time series shows a strong relation with temperature, especially when, during the winter season, temperature falls below $18^\circ C$ and household heating becomes relevant. As shown in the left panel of Fig. 6, the relationship is piecewise linear: a line with negative slope below $18^\circ C$, followed by an approximately constant line above $18^\circ C$. In order to transform the piecewise linear dependence into a linear one, it is useful to make reference to the so-called Heating Day Degrees (HDD):

**Definition 2** (Heating Day Degrees (HDD)),

$$\text{HDD}(T) = \max(18^\circ - T, 0)$$

In the right panel of Fig. 6, the scatter plot of RGD vs HDD highlights an approximately linear relationship, with a positive correlation of 0.97. The correlation of HDD
with RGD is even more evident when we look at the time series of RGD and HDD during 2017, see Fig. 7.

Figure 6: Left panel: scatter plot of daily RGD vs average daily temperature. Right panel: scatter plot of daily RGD vs HDD. Inset: HDD as a function of the temperature.

Figure 7: Time series of RGD and HDD in 2017. The instantaneous correlation between the two series is apparent.

As shown in Fig. 6, HDD are more correlated to gas demand than plain temperatures. Thus, we consider $\text{HDD}(\hat{T}(t))$ as a feature, where $\hat{T}(t)$ denotes the one-day-ahead forecast of $T(t)$. As additional features also $\text{HDD}(\hat{T}(t-1))$, $\text{HDD}(\hat{T}(t-2))$...
### Feature Reference time Type

| Feature               | Reference time | Type    |
|-----------------------|----------------|---------|
| RGD                   | t-1            | continuous |
| RGD                   | t-7            | continuous |
| RGD                   | sim(t)         | continuous |
| RGD                   | sim(t - 1)     | continuous |
| Forecasted temperature| t              | continuous |
| Forecasted temperature| t-1            | continuous |
| Forecasted temperature| t-7            | continuous |
| Forecasted temperature| sim(t)         | continuous |
| Forecasted HDD        | t              | continuous |
| Forecasted HDD        | t-1            | continuous |
| Forecasted HDD        | t-7            | continuous |
| Forecasted HDD        | sim(t)         | continuous |
| Weekday               | t              | categorical |
| Holiday               | t              | binary   |
| Day after holiday     | t              | binary   |
| Bridge holiday        | t              | binary   |

Table 1: Features

7). HDD(\(\hat{T}(\text{sim}(t))\)) are included. For completeness, also \(\hat{T}(t), \hat{T}(t - 1), \hat{T}(t - 7)\) and \(\hat{T}(\text{sim}(t))\) are considered.

### 3.3 Calendar features

As shown in the previous paragraphs, weekdays and holidays have a great influence on RGD. To capture this phenomena, the following categorical calendar features are taken into account.

**Weekday.** In view of the weekly periodicity, the seven days of the week are taken as explanatory features. By resorting to the one-hot encoding method they are transformed in 7 dichotomic time series.

**Holiday.** A binary feature which takes value 1 in correspondence of holidays.

**Day after holiday.** A binary feature which takes value 1 the first working day after a holiday. A working day is a day different from Saturday and Sunday that is not a holiday.

**Bridge holiday.** A binary feature which takes value 1 on isolated working days, that is working days where both the day before and the day after are either Saturday, Sunday or a holiday.

All the features are summarized in Table 1.

### 4 Prediction models

The classical methods used for time series forecasting are linear Box-Jenkins models such as SARIMA, where the forecast is based only on past values of the time series,
and SARIMAX, that accounts also for exogenous variables. A major drawback of classical linear models is given by discontinuities due to holidays and the possible presence of other nonlinear phenomena. In order to overcome these difficulties, herein RGD forecasting is formulated as a statistical learning problem.

Based on the availability of \( n \) data pairs \((x_i, y_i), i = 1, \ldots, n\), known as the training data, a prediction rule \( f(\cdot) \) is designed with the objective of using \( f(x_\ast) \) as prediction of \( y_\ast \), where \((x_\ast, y_\ast)\) is any novel input-output pair. In this context, \( x_i \in \mathbb{R}^p, p < n \), is a vector whose entries are given by the \( p \) features associated with the target \( y_i \).

Herein, the \( p \) features are the 22 covariates discussed in the previous section and shown in Table 1. In the following, with reference to the training data, \( y = y_i \in \mathbb{R}^n \) will denote the vector of the targets and \( X = \{x_{ij}\} \in \mathbb{R}^{n \times p} \) will denote the matrix of the training input data, where \( x_{ij} \) is the \( j \)-th feature of the \( i \)-th training pair \((x_i, y_i)\).

Below, five approaches are presented:

- ridge regression;
- torus model [18];
- Gaussian Process (GP);
- k-nearest neighbour (KNN);
- artificial neural network (ANN).

### 4.1 Ridge regression

Ridge regression [19] is a technique to identify a linear model in the form:

\[
\hat{f}(x) = \sum_{j=1}^{p} x_{ij} \beta_j = X \beta, \quad \beta \in \mathbb{R}^p
\]

To prevent overfitting, besides the standard squared sum of the residuals, the loss function includes the squared norm of the parameter vector \( \beta \):

\[
\beta_{\text{ridge}} := \arg\min_\beta \|y - X \beta\|^2 + \lambda \|\beta\|^2
\]  

where \( \lambda \) is the so-called regularization parameter, an hyperparameter which controls the flexibility of the learning algorithm. Assuming that \( X \) is full rank, the solution of (2) is

\[
\beta_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y
\]  

that highlights the shrinking effect with respect to the standard least squares estimator \( \beta_{LS} = (X^T X)^{-1} X^T y \).

Since the parameters are obtained in closed form (3), the ridge regression model is completely specified by the choice of \( \lambda \), that can be calibrated following different approaches [19].

A normalized assessment of the amount of regularization associated with a given \( \lambda \) is provided by the so-called effective degrees of freedom

\[
\text{df}(\lambda) = \text{tr} \left( X (X^T X + \lambda I)^{-1} X^T \right)
\]

In fact, \( \text{df}(\lambda) \) ranges from \( p \) to 0 as \( \lambda \) goes from 0 to infinity [19].
4.2 Gaussian processes

Let $\tilde{y} = [y, y^T]^T$, $\tilde{x} = [x^T, x_1^T, \ldots, x_n^T]^T$ and assume that, conditional on $\tilde{x}$, the vector $\tilde{y}$ is normally distributed as follows

$$\tilde{y}|\tilde{x} \sim N(0, \Sigma(\tilde{x}) + \sigma^2 I_n)$$

where the kernel $\kappa(\cdot, \cdot)$ is a suitable function whose choice reflects the available prior knowledge on the characteristics of the prediction rule. It is worth noting that the previous hypothesis is equivalent to assuming that

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n$$

where $\epsilon_i \sim N(0, \sigma^2)$ are independent errors and $f(\cdot)$ is the realization a zero-mean continuous-time Gaussian Process (GP) with autocovariance $\kappa(\tilde{x}_i, \tilde{x}_j)$ [20, 21]. The estimation of a new target value $y_*$ relies on the following property of normally distributed random vectors.

**Lemma 1** (Distribution of jointly Gaussian variables). Let $z_*$ and $z$ be jointly Gaussian random variables:

$$\left[\begin{array}{c} z_* \\ z \end{array}\right] \sim N\left(\left[\begin{array}{c} 0 \\ 0 \end{array}\right], \left[\begin{array}{cc} \Sigma_{z_*z_*} + \sigma^2 \Sigma_{z_*z} & \Sigma_{z_*z} \\ \Sigma_{z_*z} & \Sigma_{zz} + \sigma^2 I_n \end{array}\right]\right)$$

Then, the posterior distribution of $z_*$ conditional on $z$ is:

$$z_*|z \sim N\left(\Sigma_{z_*z} (\Sigma_{zz} + \sigma^2 I_n)^{-1} z, \Sigma_{z_*z_*} + \sigma^2 - \Sigma_{z_*z} (\Sigma_{zz} + \sigma^2 I_n)^{-1} \Sigma_{z_*z}\right)$$

In view of the previous lemma, it is possible to use the posterior expectation as prediction rule.

$$f(x_*) = \mathbb{E}[y_*|x_*, y, x] = \sum_{i=1}^n c_i \kappa(x_*, x_i)$$

$$c = (\Sigma(x) + \sigma^2 I_n)^{-1} y$$

The main distinctive feature of GP models is the learning process, which aims directly at obtaining the predictive function rather than inferring its parameters.

A zero-mean GP is completely defined by its covariance function $\kappa(x_i, x_j)$, also called kernel. When it is a function of the distance $r = ||x_i - x_j||$ between $x_i$ and $x_j$, i.e. $\kappa(x_i, x_j) = \kappa(r)$, the kernel is said to be stationary and isotropic. Within this class, a popular and flexible choice is the family of Matérn kernels, defined by:

$$\kappa_{\text{Matérn}}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l}\right)^\nu K_{\nu}\left(\frac{\sqrt{2\nu} r}{l}\right)$$

where $\nu$ and $l$ are hyperparameters to be tuned and $K_{\nu}$ is a modified Bessel function [22]. The parameter $l$ defines the characteristic length-scale of the process, whereas $\nu$ defines the specific covariance function in the Matérn class. If $\nu$ tends towards infinity, the Matérn formula reduces to the widely used squared exponential function

$$\kappa_{se}(r) = \exp\left(-\frac{r^2}{2l^2}\right)$$
while if \( \nu = 1/2 \) it becomes an exponential function

\[
\kappa_{\text{exp}}(r) = \exp\left(-\frac{r}{l}\right)
\]

Different approaches are possible in order to tune the hyperparameters \( \nu, \lambda, \) and \( \sigma^2 \). According to an empirical Bayes, the hyperparameter vector \( \eta \) is chosen as the maximizer of the marginal likelihood \( p(y|x, \eta) \).

### 4.3 K-Nearest neighbours

K-Nearest neighbours (KNN) relies on the distance between samples in the feature space: given a test sample \( x_* \), the prediction of \( y_* \) is computed by averaging \( K \) training samples \( y_i, i \in \mathcal{C} \), where \( \mathcal{C} \) denotes the set identified by the \( K \) feature vectors \( x_i \) that are closest to \( x_* \), according to some distance measure, e.g. the Euclidean norm that was adopted herein.

In order to specify a KNN estimator, one has to choose the distance metric, e.g. Euclidean, Minkowsky, Manhattan, etc, and the type of weighted average, e.g. uniform or inverse distance, and to calibrate one hyperparameter, viz the number \( K \) of neighbours. Too small values of \( K \) lead to overfitting to the training data, while including too many neighbours reduces the variance at the cost of jeopardizing model flexibility.

### 4.4 Neural Networks

Artificial Neural Networks (ANN) are complex non-linear models, capable of capturing non-linear patterns and relations. A comprehensive explanation of their structure and the most common training algorithms can be found in [23].

In this study, we focused on the Multi-Layer Perceptron (MLP) or fully connected ANN.
The Rectified Linear Unit (ReLu) activation function and the Mean Squared Error (MSE) loss function were adopted. Training was performed by means of gradient descent as implemented in the Adaptive Moment Estimation (ADAM) algorithm [24]. The hyperparameters to be tuned include the number of neurons in each layer, the parameters entering the definition of the activation functions, and optimization para-

Figure 8: The ANN with 22 input features, three dense layers of 24, 12 and 4 neurons and an output neuron.
meters such as number of epochs, batch size, and learning rate.

4.5 Torus model

The torus model [18] is a linear model based on sinusoidal functions, originally developed to predict power load. Herein, its short-term version is adapted to forecast RGD.

Following [18], a logarithmic transformation of the RGD is performed in order to mitigate the effect of its skewness. The long-term model is

$$\ln \hat{\text{RGD}}_{\text{long}}(t) = L(t) + F(t) + \sum_{i} H_i(t)$$

where the forecast is given by the sum of three elements: the trend or level $L$, the potential $F$, which accounts for seasonality, and the effect of holidays $\sum_i H_i$.

The potential $F$ is modelled by a linear combination of sinusoidal functions:

$$F(t) = \sum_{i=1}^{(1+2N_d)(1+2N_w)} \theta_i h_i(t) \quad \{h_i(t)\} = \mathcal{D} \otimes \mathcal{W}$$

where the functions $h_i$ are given by the product of the $j$-th element in $\mathcal{D}$ with the $k$-th element in $\mathcal{W}$, for suitable $j$ and $k$, and

$$\mathcal{D} = \{\cos(j\Psi t), j \in [0, N_d]\} \cup \{\sin(j\Psi t), j =1, \ldots, N_d\}$$
$$\mathcal{W} = \{\cos(k\Omega t), k \in [0, N_w]\} \cup \{\sin(k\Omega t), k =1, \ldots, N_w\}$$

The frequencies of the sinusoidal functions are $\Psi = \frac{2\pi}{365.25}$ and $\Omega = \frac{2\pi}{7}$. The number of harmonics, respectively $N_w$ for 7-day and $N_d$ for 365.25-day periodicity, are hyperparameters of the model.

We add to the original model the dependency on temperature, expressed in HDD(t), and its daily difference HDD(t) – HDD(t − 1), by including these two features in the set of regressors. The terms related to trend and holidays are kept as presented in [18].

Finally, to get a short-term predictor, we correct the long-term model with the consumption of the previous day:

$$\hat{\text{RGD}}(t) = \hat{\text{RGD}}_{\text{long}}(t) \frac{\text{RGD}(t-1)}{\hat{\text{RGD}}_{\text{long}}(t-1)}.$$ 

The number of harmonics $N_w$ and $N_d$ were tuned by minimizing the AIC index.

4.6 Technical notes on model implementation

All the models, except the torus one, were implemented in Python, using scikit-learn and keras; automated hyperparameters tuning exploited the GridSearchCV function of scikit-learn. The torus model was implemented in MATLAB, as well as its hyperparameter tuning routine.
5 Performance limit

As shown in Section 3, temperature is the most important exogenous variable. Unfortunately, the actual temperature is not available when forecasting future RGD: only a forecast is available, affected by a small yet non-negligible error, which inevitably impacts the performance of the model. The scope of this section is to assess the impact of the temperature error on the precision of RGD forecast.

Let us assume that RGD is a deterministic function $g$ of the temperature $T$ and some other factors $x = (x_1, x_2, ...)$: $\text{RGD} = g(T, x)$. In view of the analysis and the charts presented in Section 3, a reasonable first-level approximation of the relationship between RGD and $T$ is a linear function of HDD, while the dependence on the other factors can be represented as an additive term $\bar{g}(x)$:

$$\text{RGD} = g(T, x) = \bar{g}(x) + \alpha \text{HDD}$$

where $\alpha$ is the sensitivity of the gas demand to the temperature expressed as HDD. The formula is of general validity and applies to both regional and national gas markets. Note that $\alpha$ depends on the size of the considered market and can be estimated from historical data. Consider now the ideal case when $\alpha$ and also the function $\bar{g}$ are perfectly known, yet, only a forecast $\hat{T}$ of the correct temperature $T$ is available

$$\hat{T} = T + \epsilon$$

where $\epsilon$ is a zero-mean error with variance $\sigma^2$. The optimal forecast $\hat{\text{RGD}}$, given $\hat{T}$ is:

$$\hat{\text{RGD}} = \bar{g}(x) + \alpha \text{HDD}(\hat{T})$$

In order to obtain the mean squared error of $\hat{\text{RGD}}$, we first compute the conditional variance of $\hat{\text{RGD}}$:

$$\text{Var} \left[ \hat{\text{RGD}} \mid T \geq 18^\circ \right] = \text{Var} \left[ \bar{g}(x) + \alpha \cdot 0 \right] = 0$$

$$\text{Var} \left[ \hat{\text{RGD}} \mid T < 18^\circ \right] = \text{Var} \left[ \bar{g}(x) + \alpha (\hat{T} - 18^\circ) \right] = \alpha^2 \text{Var} [\epsilon] = \alpha^2 \sigma^2$$

Since $E[\epsilon] = 0$, it follows that $E[\hat{\text{RGD}}] = \text{RGD}$. Thus:

$$E \left[ (\hat{\text{RGD}} - \text{RGD})^2 \right] = P(T < 18^\circ) \text{Var} [\bar{g} \mid T < T_c]$$

$$= P(T < 18^\circ) \alpha^2 \sigma^2$$

This last equation provides an estimate of the mean squared error due to the forecasting error on the temperatures. Since it has been derived under an ideal setting ($\alpha$ and $\bar{g}(\cdot)$ perfectly known) it provides a lower limit to the precision that can be achieved by the best possible forecaster.

To find a numeric value for the bound, we first estimate $P(T < T_c)$ by computing the ratio between the number of samples such that $T < T_c$ and the total number of available data, then we compute $\alpha$ through a least square fit and finally we estimate $\sigma^2$ as the variance of the difference between the temperature forecast and the actual values. In the 3-year period 2015-2017, $P(T < T_c)$ ranges from 54% to 67%, while $\sigma^2$ ranges from 0.05 to 0.09, and $\alpha$ from 9.85 to 10.96. Considering altogether the years 2015-2017, we have $P(T < T_c) = 63\%$, $\sigma^2 = 0.07$, $\alpha = 10.56$, corresponding to a best achievable Root Mean Squared Error (RMSE) of 2.20 Millions of Standard Cubic Meters (MSCM).
6 Results

As mentioned in Section 2, available data range from 2007 to 2017. Three test sets were defined, each one year long, namely 2015, 2016, and 2017. To each of them, a training set was associated, spanning from 2007 to the day before the start of the test set. Thus, training sets cover the years 2007-2014, 2007-2015, 2007-2016. In the following, each training set is identified by the year of the corresponding test set, e.g. we will write "training set 2016" to indicate the second training set, spanning from 2007 to 2015.

On each test set, the performance of each of the five models was measured using the Mean Absolute Error (MAE).

\[
MAE = \frac{1}{N} \sum_{j=1}^{N} |R_{GDj} - \hat{R}_{GDj}|
\]

MAE is preferred over MAPE due to the highly non-stationary behaviour of RGD series. Moreover, MAE is proportional to the monetary loss sustained by energy companies because of errors in nomination due to inaccurate forecasts.

Nevertheless, in order to allow a comparison with forecasting performances achieved in the UK market, we will also refer to the MAPE

\[
MAPE = \frac{100}{N} \sum_{j=1}^{N} \left| \frac{R_{GDj} - \hat{R}_{GDj}}{R_{GDj}} \right|
\]

In fact, the comparison between two different markets calls for the use of a relative metric. To avoid the confounding effect of small absolute errors that are amplified by MAPE during the Italian Summer, the comparison between Italy and UK was limited to the cold months, when gas demand is relatively high, see Section 6.2.

Finally, as the performance limit derived in Section 5 poses a lower bound to the mean squared error, the Root Mean Square Error (RMSE) was also used as a comparison metric.

6.1 Hyperparameters

All the five models include hyperparameters that were tuned by cross-validation.

For ridge regression, the regularization parameter \( \lambda \) was tuned on an interval ranging from \( 10^{-4} \) to 50 in logarithmic steps. Line search selected 0.236, \( 10^{-4} \) and \( 10^{-4} \) as optimal values for the three test sets. In all cases, the effective degrees of freedom \( df(\lambda) \) were close to 21. Since the number of parameters is \( p = 22 \), this suggests that regularization affected the estimate only marginally.

For KNN we optimized the number of neighbours, in the interval [1, 30], and also the weighting strategy, choosing between uniform and inverse of distance. We obtained 7 neighbours for training set 2015 and 6 for the two remaining ones. In all the three cases, the "inverse of distance" weights were selected.

As for the Gaussian Process, the maximization of the marginal likelihood yielded \( \nu = 1.5 \), \( l = 10 \), and \( \sigma^2 = 10 \), with minimal variations among all training sets.

For the ANN models, a trial and error procedure led to an architecture with an input layer of 24 neurons, two hidden layers of 12 and 4 neurons, and an output layer of a single neuron, as shown in Fig. 8. By 5-fold cross-validation, we obtained a learning
rate of 0.001, a number of epochs of 1000, and a batch size of 32, being the sets of possible values \{0.1, 0.01, 0.001\}, \{300, 500, 1000\} and \{16, 32, 64\}, respectively.

For what concerns the Torus model, the minimization of AIC led to the choice of \(N_w = 3\) and \(N_d = 1\) for all the training sets.

### 6.2 Prediction results

We first compare the performances of the adopted methods in terms of RMSE, for which a lower bound has been derived in Section 5. Considering the three years, ANN achieves the best overall performance, closely followed by GP; both maintain the RMSE close to or even below 4 MSCM. The Torus model is also a good performer, except for 2015. Ridge regression yields a slightly higher but stable RMSE across the three years. KNN achieves the poorest performance, with an RMSE around 8 MSCM. The best achievable performance, computed according to Section 5, is always close to 2 MSCM. Hence, all the forecasters but KNN achieve an RMSE that is about twice the theoretical limit. Recalling that the limit of performance assumes ideal conditions, in which the only error is the one associated with temperature forecasts, the results suggest that highly precise RGD predictions were achieved in four cases out of five.

Next, we compare models in terms of their MAE. Results on the test sets are shown in Table 3. Now, GP is the best overall performer, achieving an average MAE of 2.60 MSCM over the three test years. ANN, Torus and Ridge Regression follow in the order. KNN is again the worst model, with an average MAE of 5.43 MSCM.

The differences between the rankings based on MAE and RMSE are explained by the non-Gaussianity of the prediction errors. In case of perfectly Gaussian and stationary prediction errors, it should be \(\text{MAE}/\text{RMSE} = \sqrt{2/\pi} \sim 0.798\), yielding identical rankings, irrespective of the adopted metrics. As a matter of fact, it occurs that \(\text{MAE}/\text{RMSE} < 0.798\) for all the models: about 0.61 for GP and Torus, 0.65 for KNN and ANN and 0.71 for Ridge. This is explained by the different degree of non-Gaussianity of the prediction errors, possibly associated with the presence of "fat tails" in their distributions. In particular, from Fig. 9, it is apparent that different error variances are observed in the cold and warm seasons, so that the overall error distribution is akin to a mixture, which can produce fat tails when the variances in the two seasons are much different. The shapes of error distributions in 2017 are displayed in Fig. 10.

Due to the seasonal behaviour of RGD, it is of interest to study the monthly errors. In Table 4 we show the average of the MAE and MAPE achieved by each model in each month, throughout the 2015-2017 test years. It appears that GP is the best performer during the summer period, especially from June to October, whereas in the winter months, from December to February, ANN is more accurate. A possible explanation is that the GP model is better at capturing the effects of the weekly seasonality, that explains most of the Summer variability, while ANN better allows for the non-linear effect of temperature, mostly relevant during the cold months.

As already mentioned, to the best of our knowledge, there are no direct benchmarks in the literature for the forecasting task addressed in this paper. A somehow similar problem was studied by Zhu et al. [8] relative to UK gas demand in 2012. Still, their results are not completely comparable to ours, for two main reasons: first, the authors considered the whole UK demand and not just the residential one; second, UK climate is colder than the Italian one. Nonetheless, we can use a relative error metrics, such as the MAPE in order to obtain a first level comparison, limited to 6 cold months (from October to March). Our best model in terms of average MAPE over 2015-2017, i.e.
the GP, achieves 3.11%, while Zhu’s false neighbours filtered-support vector regression local predictor (FNF-SVRLP) achieves 3.88% on the same six cold months of 2012. Although no definite conclusion can be drawn, these numbers suggest some degree of consistence between performances achievable in forecasting gas demand at country level.

| Year | 2015 | 2016 | 2017 | 2015-2017 |
|------|------|------|------|-----------|
| GP   | 4.33 | 4.28 | 4.14 | 4.25      |
| KNN  | 7.82 | 8.05 | 8.72 | 8.21      |
| Torus| 5.31 | 4.32 | 3.93 | 4.56      |
| ANN  | 4.34 | 4.11 | 3.64 | 4.04      |
| Performance limit | 2.15 | 2.02 | 1.98 | 2.20 |

Table 2: Performance on test sets: yearly RMSE (MSCM) of the five forecasters and test sets and performance limit.

| Year | 2015 | 2016 | 2017 | 2015-2017 |
|------|------|------|------|-----------|
| Ridge | 3.34 | 3.15 | 3.00 | 3.16      |
| GP    | 2.65 | 2.58 | 2.56 | 2.60      |
| KNN   | 5.39 | 5.42 | 5.48 | 5.43      |
| Torus | 3.18 | 2.66 | 2.54 | 2.79      |
| ANN   | 2.76 | 2.68 | 2.43 | 2.62      |

Table 3: Yearly MAE (MSCM) on test sets.

| Month  | Ridge | GP   | KNN | Torus | ANN | MAPE(%) | MAE (MSCM) |
|--------|-------|------|-----|-------|-----|---------|------------|
| January| 3.10  | 3.01 | 6.45| 3.21  | 2.93| 5.79    | 5.67       |
| February| 2.75 | 2.84 | 4.77| 3.33  | 2.62| 4.52    | 4.59       |
| March  | 3.94  | 4.20 | 7.13| 3.83  | 4.27| 4.59    | 4.89       |
| April  | 6.17  | 4.80 | 14.70| 4.89  | 5.09| 3.56    | 2.89       |
| May    | 5.76  | 2.67 | 6.08| 3.21  | 2.50| 2.37    | 2.22       |
| June   | 4.57  | 1.32 | 6.02| 3.37  | 1.92| 1.46    | 0.43       |
| July   | 3.78  | 1.16 | 3.65| 1.50  | 1.54| 1.11    | 0.35       |
| August | 9.39  | 3.00 | 19.44| 3.86  | 4.50| 2.24    | 0.71       |
| September| 5.36 | 1.06 | 3.18| 1.33  | 1.81| 1.92    | 0.38       |
| October| 4.10  | 2.81 | 6.22| 3.30  | 3.42| 2.23    | 1.78       |
| November| 2.70 | 3.14 | 5.50| 3.03  | 2.9 | 3.39    | 3.76       |
| December| 2.78 | 2.68 | 4.27| 2.70  | 2.52| 4.83    | 4.58       |

Table 4: Monthly MAPE and MAE (MSCM) on test sets 2015-2017: best performers in terms of MAE are highlighted in boldface.
7 Conclusions

In this paper, one-day-ahead forecasting of the daily Italian residential gas demand was addressed. We discussed the main features of the time series and its covariates, such as temperature, as well as the most relevant aspects of the preprocessing and
feature extraction steps. Five different models were developed and compared: Ridge regression, Gaussian Process, K-nearest neighbour, Artificial Neural Network and the Torus model.

To our best knowledge, no specific benchmark was available in the literature. It is therefore of some interest the theoretical performance limit that we derived assuming ideal conditions except for the inaccuracy of meteorological temperature forecasts, whose effect was properly propagated to the gas demand forecasting errors, see Section 5.

Our best model in terms of RMSE, is the Artificial Neural network, closely followed by the Gaussian Process. They achieve forecasting errors whose RMSEs are about twice the performance limit. If the MAE is taken as error measure, the GP becomes the best model, although by a narrow margin. From the analysis of monthly performance, we realize that GP model is more accurate in tracking the weekly periodicity, which is predominant in summer period, while the ANN may better allow for the non-linear influence of temperature, whose contribution is greater during the winter period.

References

[1] Italian natural gas demand report. http://pianodecennale.snamretegas.it/it/domanda-offerta-di-gas-in-italia/domanda-di-gas-naturale.html. Accessed: 2019-01-31.
[2] Dario Šebalj, Josip Mesarić, and Davor Dujak. Predicting natural gas consumption—a literature review. In 28th International Conference* Central European Conference on Information and Intelligent Systems*, 2017.
[3] Božidar Soldo. Forecasting natural gas consumption. *Applied Energy*, 92:26–37, 2012.
[4] Marek Brabec, Ondřej Konár, Emil Pelikán, and Marek Malý. A nonlinear mixed effects model for the prediction of natural gas consumption by individual customers. *International Journal of Forecasting*, 24(4):659–678, 2008.
[5] Primož Potočnik, Marko Thaler, Edvard Govekar, Igor Grabec, and Alojz Poredos. Forecasting risks of natural gas consumption in Slovenia. *Energy policy*, 35(8):4271–4282, 2007.
[6] Salvador Gil and J Deferrari. Generalized model of prediction of natural gas consumption. *Journal of energy resources technology*, 126(2):90–98, 2004.
[7] A Azadeh, SM Asadzadeh, and A Ghanbari. An adaptive network-based fuzzy inference system for short-term natural gas demand estimation: uncertain and complex environments. *Energy Policy*, 38(3):1529–1536, 2010.
[8] Lixing Zhu, MS Li, QH Wu, and L Jiang. Short-term natural gas demand prediction based on support vector regression with false neighbours filtered. *Energy*, 80:428–436, 2015.
[9] Ioannis P Panapakidis and Athanasios S Dagoumas. Day-ahead natural gas demand forecasting based on the combination of wavelet transform and anfis/genetic algorithm/neural network model. *Energy*, 118:231–245, 2017.
[10] Fatih Taşpınar, Numan Celebi, and Nedim Tutkun. Forecasting of daily natural gas consumption on regional basis in turkey using various computational methods. *Energy and Buildings*, 56:23–31, 2013.
[11] Ömer Fahrettin Demirel, Selim Zaim, Ahmet Çalışkan, and Pınar Özuyar. Forecasting natural gas consumption in istanbul using neural networks and multivariate time series methods. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(5):695–711, 2012.
[12] Jolanta Szoplik. Forecasting of natural gas consumption with artificial neural networks. *Energy*, 85:208–220, 2015.

[13] Božidar Soldo, Primož Potočnik, Goran Šimunović, Tomislav Šarić, and Edvard Govekar. Improving the residential natural gas consumption forecasting models by using solar radiation. *Energy and buildings*, 69:498–506, 2014.

[14] Zlatko Tonković, Marijana Zekić-Sušac, and Marija Somolanji. Predicting natural gas consumption by neural networks. * Tehnički vjesnik*, 16(3):51–61, 2009.

[15] Nan Wei, Changjun Li, Jiehao Duan, Jinyuan Liu, and Fanhua Zeng. Daily natural gas load forecasting based on a hybrid deep learning model. *Energies*, 12(2):218, 2019.

[16] Zia Wadud, Himadri S. Dey, Md. Ashfanoor Kabir, and Shahidul I. Khan. Modeling and forecasting natural gas demand in bangladesh. *Energy Policy*, 39(11):7372–7380, 2011. Asian Energy Security.

[17] Yusuf Karadede, Gultekin Ozdemir, and Erdal Aydemir. Breeder hybrid algorithm approach for natural gas demand forecasting model. *Energy*, 141:1269–1284, 2017.

[18] Alice Guerini. Long and short term forecasting of daily and quarter-hourly electrical load and price data: a torus-based approach. 2016.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York, 2009.

[20] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning series. University Press Group Limited, 2006.

[21] K.P. Murphy and F. Bach. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine. MIT Press, 2012.

[22] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1965.

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.