

## DEEP LEARNING MODELS FOR NATURAL GAS DEMAND FORECASTING: A COMPARATIVE STUDY OF MLP, CNN AND LSTM

**Artemis Aidoni**<sup>1</sup>

**Konstantinos Kofidis**<sup>2</sup>

**Catalina Lucia Cocianu**<sup>2</sup>

**Lazar Avram**<sup>1</sup>

<sup>1</sup> Petroleum-Gas University of Ploiesti, Romania

<sup>2</sup> Bucharest University of Economic Studies, Romania

email: artemisaidoni60@gmail.com

**DOI: 10.51865/JPGT.2023.01.12**

### ABSTRACT

This study aims to investigate the use of various deep learning techniques to predict future residential natural gas consumption in Italy, with a particular emphasis on the correlation between gas consumption and temperature. Four models were evaluated, including Multi-layer Perceptron (MLP), Convolutional Neural Network (CNN), Simple Long-Short Term Memory (LSTM), and Stack-LSTM, with the latter chosen due to its two-layer LSTM and potential to improve forecasting accuracy. Feature scaling was conducted with the MinMaxScaler method to ensure uniform values among variables. Statistical analysis was performed using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared accuracy metrics, with ANOVA tests and boxplots, used to visualize the distribution of accuracy metrics across test and full datasets. Results implied that the CNN and Stack-LSTM models were more effective in accurately predicting the target variable compared to the other models, as indicated by MSE and R-squared scores, as well as graphical comparisons of actual and predicted values. Finally, the research recommends the utilization of supplementary features in future research to increase the precision of forecasts.

**Keywords:** gas consumption, forecasting, temperature, deep learning algorithms

### INTRODUCTION

Natural gas is a key component for energy consumption around the world, making up 25% of the global energy supply as reported by the Statistical Review of World Energy by BP p.l.c. (Fig.1). Additionally, it plays a role in reducing pollution and creating a clean, healthy environment. Its many uses-residential, commercial, and industrial-make it important to accurately forecast its demand for efficient management of energy resources.

Europe is heavily reliant on gas for energy generation, transport, and heating. In 2021, fossil fuels produced two-thirds (76%) of Europe's energy - oil (31%), coal (31%), and gas combustion (34%) providing one-third of Europe's energy. Nuclear energy provided for 10% of total energy, with renewable sources like hydropower, solar, wind, and biofuels making up the remaining 14%.

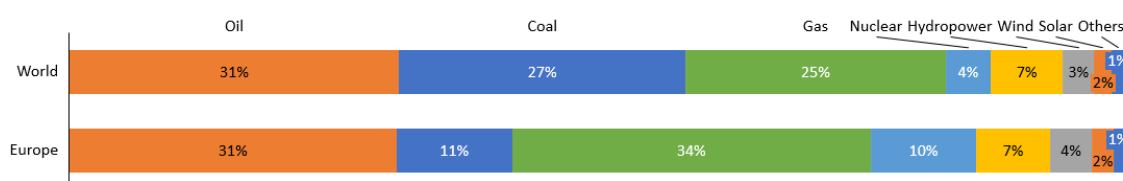


Fig. 1. 2021 World Energy Consumption by source [1]

Estimation and forecasting of natural gas consumption have drawn significant attention from the literature. Xin Zhang et al. [2] developed methods to improve the prediction of energy consumption by understanding the variation of gas consumption in relation to temperature changes. The authors have used two techniques: Empirical Mode Decomposition (EMD) and linear regression analysis, along with outlier detection. EMD divides original data into several Intrinsic Mode Functions (IMFs). Outliers from real-time gas consumption and temperature data points were removed using the Mahalanobis distance-based technique. The correlation coefficient between gas load and temperature is computed to assess the weather-sensitive parts of demand for gases when compared on both real-time dataset and processed data via the EMD method. The results showed a higher correlation between temperatures and loads after applying the EMD than when using raw/real-time values.

The research reported in [3] focuses on the connection between winter temperatures and residential gas consumption. Nine subregions in the U.S. east of the Rocky Mountains from 1989-2000 are taken into consideration. There are two temperature indices, days below percentile (DBP) and heating degree-days (HDD), that are developed from daily maximum and minimum temperatures from 1949-2000. Results show the highest correlations between DBP/HDD and gas consumption in the Great Lakes-Ohio Valley region both monthly and seasonally, with values ranging from 0.89 to 0.97. On the other hand, a significantly lower correlation value is established in New England and across the South, ranging from 0.62 to 0.80. Additionally, the percentiles with the highest correlation with gas consumption are slightly higher in northern regions than in the south. As far as HDD correlations go, they are lower in colder northern regions than farther south. However, all HDD reference temperatures greater than 10°C (15°C) in northern (southern) regions yield similar correlations. These findings indicate that accurate seasonal temperature forecasts could lead to strong predictability of gas consumption.

Ahmet Goncu et al. proposed a new methodology for forecasting residential and commercial natural gas consumption which combines demand estimation with stochastic temperature modeling [4]. They used daily data on natural gas consumption and local temperatures from Istanbul to estimate both processes separately before deriving the distribution of expected natural gas usage based on temperature conditions. Then they forecasted future levels by using Monte Carlo simulations or an analytical solution depending upon their model specifications to evaluate how well these models perform against actual values through back-testing methods as well as establishing relationships between traded weather derivatives (HDD/CDD futures) and expected natural gas consumptions so suppliers can partially hedge their risk via those instruments if needed. The last sentence is too long, split it into 2-3 parts.

In recent years, due to the development of technology, research has been conducted to create new machine learning models that can accurately predict energy demand. Jie Wang et al. [5] reviewed the evolution of natural gas consumption forecasting, outlining changes in forecasting horizons, influencing factors, and performance. The evolution is analyzed taking into account four stages: the initial stage (1950-1980), the conventional stage (1981-2000), the AI stage (2001-2015), and the all-round stage (>2016). Each period is outlined along with the typical models utilized during that timeframe. Long-term forecasts were mainly affected by production, population, and economic variables, while medium-term were influenced by temperature & economics; short terms mostly depend on weather conditions & date type. Time series models produced the best results for long-term predictions, while the ANN-based models had an average mean absolute percentage error rate of 2.21% / 4.98% for mid/short-range forecasting. A framework has been proposed for model selection as well as suggestions for future research directions such as considering data sets spatiotemporally or using more Deep Learning-based approaches.

A similar approach can be seen in the study conducted by Panek, W. et al [6] that analyses the impacts of various factors on natural gas consumption as well as how they can be used to estimate future demand. The authors collected data from a medium-sized city in Poland, then used neural networks, MLR (Multiple Linear Regression), and RF (Random Forest) methods to model temporary and future natural gas consumption for municipal consumers. Results from the various forecasting techniques recommended RF as the best predictor of demand based on the input data. This study also uncovered differences in the impact of different factors on natural gas demands as well as the prediction accuracies across each algorithm per time horizon

Machine learning models are used by Brian de Keijzer et al. [7] to solve the problem of energy forecasting in dwellings. Data collected was nine months' worth of mean gas consumption from 52 dwellings, with six months used for training and three months for evaluating different models. It was found that Deep Neural Network (DNN) had the best results predicting one-hour resolution forecasts, while Multivariate Linear Regression (MVLRL) had better accuracy at daily and weekly resolutions. Further studies are suggested to obtain more accurate results, such as using additional features like electricity consumption or increasing training data samples over a longer period; however, this could require more computational power and could be limited by hardware availability

A novel deep learning model, multi-channel DNN (MC-DNN), is proposed in [8] to forecast the gas consumption in the forging factory in Busan for a specific elapsed time. The proposed algorithm uses three different channels to extract hidden patterns from the data: the Time Variable Channel records information about how long it takes for the heating process; the Environmental Channel captures the temperature inside the furnace which affects gas usage; the Raw Consumption channel provides historical trends of gas consumption. The extracted features are combined as fusion features used in prediction tasks with two loss functions that also increase accuracy. To assess the efficiency of the model the author first compares the MC-DNN model with existing ones such as SVR, LR, LSTM, CNN, DNN, and CNN-LSTM. Additionally, to further validate the effect of the number of nodes, the author conducted some experiments on real datasets using multiple evaluation metrics. MC-DNN model shows accurate results both for the consumed number of gases and elapsed times at the same time.

The research from Dr. Nil Aras [9] looks into the utilization of genetic algorithms to forecast short-term demand for natural gas in residential areas. In this study, residential demand was assumed to be affected by time, heating degree-day value (a measure used in energy management), and consumer price index (CPI). The outcomes revealed that utilizing this approach provides more accurate predictions than conventional methods without requiring any assumptions concerning the underlying functions or models. The approach was tested using monthly data from Turkey's residential sector which uses 23% imported natural gas - demonstrating its potential and superiority when it comes to forecasting demands for natural gas use in residential areas.

This paper provides strong evidence to prove temperature weight significantly in gas consumption and it is focused on forecasting demand for residential consumption of natural gas based on temperature. The study starts with a brief introduction on natural gas importance and a literature review as presented above, in section 2 the research methodology of this work is described. Section 3 shows the results of the forecasting models and their evaluation. Major conclusions and future research directions and suggestions for natural gas consumption forecasting are presented in Section 4.

## **METHODOLOGY**

This section describes how the data was gathered and processed. Details on the deep learning techniques utilized and the metrics chosen for the model evaluation are provided. Also, a series of indicators are used to prove the strong correlation between gas consumption and temperature.

Natural gas consumption data is from the national system and transport operator website (SNAM) with the unit of consumption converted in GWh/day. The dataset includes 1460 daily observations for the period from January 1, 2018, to December 31, 2022, covering household, industrial, and gas consumption for generating electricity. The dataset of daily average temperatures for Italy is obtained for the same period from Refinitiv Eikon.

The methods to quantify the correlation between gas demand and temperature include a scatterplot, cross-correlation coefficient, and SARIMAX. Scatterplots give the correlation a visual representation. Additionally, makes it simple to see any patterns or trends in the data. It also helped identify outliers and strange data.

The cross-correlation coefficient is a widely used statistical measure in many fields and is widely recognized as a quantitative measure of correlation. The relationship between temperature and gas consumption was quantitatively measured using this method. The coefficients range from -1 to 1, with -1 indicating a completely negative correlation, 0 indicating no correlation, and 1 indicating a completely positive correlation. When two variables are positively correlated, it indicates that as one variable rises, the other rises as well, and when they are negatively correlated, the other variable falls.

A popular method for studying and predicting trends in time series data is the ARIMA model. It is especially useful in the context of correlation analysis as it can be used to model relationships between two or more variables. The degree and direction of correlation between variables, and any patterns or trends that may exist, can all be determined by fitting an ARIMA model to the data. In addition, ARIMA models can be used to predict future values of variables based on their correlation with other variables.

ARIMA models are useful for correlation analysis to determine causal relationships between variables and make data-driven decisions.

After the correlation is identified the focus moves to the evaluation of four different machine learning models. Specifically, the Long Short-Term Memory (LSTM), Multilayer perceptron (MPL), Convolutional Neural Network (CNN), and Stacked Long Short-Term Memory.

The basic architecture of an artificial neural network consists of three types of neuron layers: an input layer, a series of hidden layers, and an output layer. Artificial neurons in one layer are connected, fully or partially, to the artificial neurons in the next layer. Feedback connections to previous layers are also possible [10].

The first deep learning model used in the presented work is the Multilayer perceptron. Multilayer Perceptron (MLP) has been used in a variety of application domains, and most of its benefits come in classification or regression problems by modeling the input data. Despite the advantages of this model, it has been observed that inserting more inputs increases the size parameter of the network, causes memory problems, and reduces computational power. A Multilayer perceptron (MLP) works by combining three layers: an input layer, a hidden layer, and an output layer. [11]

Artificial neurons in one layer are fully or partially connected to artificial neurons in the next layer. These three layers are connected by bonds, the strength of which is called weight. These weights make the network very flexible to adapt to the data; these are free parameters and their number corresponds to the degrees of freedom of the network [12]

The output of the MLP model is calculated based on forward propagation and the weights are trained with backward propagation. The formula that calculates the output in each node  $n \in N$

$$o_n(X) = \sigma\left(\sum_{j \in H} w_{jn} \cdot \sigma\left(\sum_{i \in L} w_{ij} d_i\right)\right) \quad (1)$$

where  $L$  is the inputs layer,  $d_i$  is the input of node  $i \in L$ ,  $H$  is the hidden layer, and  $N$  is the output layer.

The backward propagation updates the weights by adjusting them based on the loss rate that obtains from the previous iteration. The loss function is also modified since the model can have multiple output units  $y$ :

$$loss_d(W) = \frac{1}{2} \sum_{k \in y} (t_{dk} - o_k(d))^2 \quad (2)$$

$$\frac{\partial loss}{\partial w_{ij}} = \frac{\partial loss}{\partial in_j} \frac{\partial in_j}{\partial w_{ij}} = \delta_j \frac{\partial}{\partial w_{ij}} \sum_{k \in y} w_{kj} \cdot out_k = \delta_j \cdot out_i \quad (3)$$

The calculation  $\delta_j = \frac{\partial loss}{\partial in_j}$  depends on what node  $j$  belongs to. If  $j$  belongs to an output unit, then

$$\begin{aligned}\delta_j &= \frac{\partial loss}{\partial in_j} = \frac{\partial}{\partial in_j} \frac{1}{2} \sum_{k \in N} (t_k - o_k(in_k))^2 = \frac{\partial}{\partial in_j} \frac{1}{2} (t_j - o_j(in_j))^2 \\ &= (o_j(in_j) - t_j) \cdot \acute{o}_j(in_j)\end{aligned}\quad (4)$$

In case  $j$  belongs in a hidden unit node then

$$\begin{aligned}\delta_j &= \frac{\partial loss}{\partial in_j} = \sum_{k \in up(j)} \frac{\partial loss}{\partial in_k} \frac{\partial in_k}{\partial in_j} = \sum_{k \in up(j)} \delta_k \cdot \frac{\partial}{\partial in_j} \sum_{l \in down(k)} w_{lk} \cdot o_l(in_l) \\ &= \sum_{k \in up(j)} \delta_k \cdot w_{jk} \cdot \acute{o}_j(in_j)\end{aligned}\quad (5)$$

where  $in_i$  is the input of node  $i$ ,  $o_i$  is the output of the function of node  $i$ ,  $out_i = o_i(in_i)$  is the output of the node  $i$ ,  $up(i)$  the nodes that closer to input in the above layer  $i$ ,  $down(i)$  the nodes that closer to input in the below layer  $i$  and  $w_{ij}$  the applied to the output nodes  $i$  weights of node  $j$ . Finally, the weights can be updated based on the following rule

$$\Delta w_{ij} = -a \cdot \delta_j \cdot out_i \quad (6)$$

$$\hat{w}_{ij} = w_{ij} + \Delta w_{ij} \quad (7)$$

where  $a$  is the learning rate. [13], [14]

The proposed MLP model starts with a fully connected layer, typically referred to as a dense layer, denoted by:

$$y = activation(W * x + b) \quad (8)$$

where  $y$  is the output of the dense layer,  $x$  is the input to the layer,  $W$  is the weight matrix,  $b$  is the bias vector, and  $activation$  is the activation function, which is set to the rectified linear unit (ReLU) activation function. The input shape of the layer is defined as a two-dimensional tensor with dimensions (batch size, number of features), where the number of features is based on the input data. The dense layer has 32 units and uses the He normal initializer for the kernel weights.

The model then includes a Dropout layer with a rate of 0.05, denoted by:

$$y = dropout(x, rate) \quad (9)$$

where  $y$  is the output of the Dropout layer and  $x$  is the input to the layer. The Dropout layer helps prevent overfitting by randomly dropping out a portion of the neurons during training. The model also includes two additional dense layers, each with 15 units, which produce the final prediction for the dependent variable. The activation function for the final layer is not specified, implying that a linear activation is used.

Another algorithm used to predict gas demand is the Long Short-Term Memory. LSTM models are the most used and most successful for forecasting time series data. LSTM is the preferred choice for complex problems because the network can remember both short-term and long-term values. A LSTM network is a combination of layers built from LSTM units. The unit consists of three gates, the entrance gate, the exit gate, and the oblivion gate, which govern the flow of information. [15]. For each gate, the calculations are happening in every time step and the formulas that are used are given seen below:



$$f_t = \sigma(W_f \circ [y_{t-1}, x_t] + b_f) \quad (10)$$

$$i_t = \sigma(W_i \circ [y_{t-1}, x_t] + b_i) \quad (11)$$

$$o_t = \sigma(W_o \circ [y_{t-1}, x_t] + b_o) \quad (12)$$

where  $i_t$  is the input gate,  $f_t$  is the forget gate,  $o_t$  is the output gate,  $\sigma$  is the sigmoid function,  $W_{i,f,o}$  is the weight matrix,  $y_{t-1}$  is the input of the previous time step,  $x_t$  is the input of the current time step,  $b_{i,f,o}$  is the bias of the current vector.

The formula shows that at time  $t$ , the first iteration stores new information, the second discards information, and the third gives the output. [16]

For our research, we will employ two approaches regarding Long Short-Term Memory (LSTM) models. The first one is a basic LSTM model with just one LSTM layer, while the second one's a stacked LSTM model with two layers of LSTM. Both of the models are sequential neural networks that we're setting up with the Keras API.

The initial layer of the model consists of an LSTM layer with 50 units, and uses the ReLU activation function. This layer is intended to capture long-term correlations within the time series data and transmit them to the succeeding layers. The form of the input is specified as a three-dimensional tensor, with measurements (batch size, time steps, the number of features) determined by the input data.

A Dropout layer is then added, with a rate of 0.05, to help avoid overfitting by randomly dropping out neurons during the training process. An additional LSTM layer with 50 units and the `return_sequences` parameter set to `False` follows, which further processes the output from the prior layer and consolidates information from the sequence of prior time steps.

Finally, this model consists of two Dense layers; the first has 15 units while the second has just 1 unit. These layers merge the outputs from the prior layers and transform them into one value, thus producing a prediction of the dependent variable.

The last algorithm tested is Convolutional Neural Networks, the most common model for vision and image processing. CNN is a mathematical construct that typically consists of three types of layers: convolution, pooling, and fully connected layers. Convolutional layers play an important role in CNNs. It consists of a stack of mathematical operations such as convolution, a specialized type of linear operation [17]. The advantage of this model over other models is that it can recognize important features without human supervision. [18].

In this study, we present a deep-learning architecture utilizing Conv1D layers and MaxPooling1D layers. The initial layer is a Conv1D with 256 filters, a ReLU activation function, and a kernel size of 2. This layer implements the convolution operation on the data, allowing the model to recognize localized patterns in the temporal dataset. The input shape is structured as a three-dimensional tensor, with dimensions (batch size, time steps, number of features), where the time steps and the number of features is determined based on the input data.

The subsequent layer is a MaxPooling1D layer with a pool size of 1, which is employed to reduce the size of the input data and consequently decrease the computational complexity of the model. This is then followed by another Conv1D layer incorporating

128 filters, ReLU activation function, and a kernel size of 1. The second Conv1D layer puts into effect another convolution operation on the output of the preceding layer, allowing the model to recognize higher level patterns in the temporal series data.

The next step is to transform the output of the second MaxPooling1D layer into a one-dimensional tensor, which is then followed by a Dropout layer with a rate of 0.075. This helps to prevent overfitting by randomly dropping out several neurons during training. Subsequently, a Dense layer with 50 units and ReLU activation are included to further process the data from the preceding layers and consolidate information from all the features. Finally, a Dense layer with 1 unit is included, which produces the conclusion for the dependent variable by combining the outputs from the previous layers and transforming them into a single value.

The Convolutional Neural Network is comprised by:

$$1D \text{ convolution } (f * g)(i) = \sum_{j=1}^m g(j)f \left( \left[ i - j + \frac{m}{2} \right] \right) \quad (13)$$

with kernel defined as  $g$  that has length  $m$  and vector  $f$ . This convolutional layer contains gates with  $k \times k$  grid inputs and the weights of each gate are tied together so they can recognize the same features. Different features can be learned by using multiple layers consisting of collections of these gates.

The pooling layer is used as a down sampling layer that selects the maximum or average value of all  $k \times k$  input grids and reduces them to a single gate used to scan the layer, so that the non-overlapping coverage of the layer is provided. The dimension of the pooling layer will be  $(H_1 \times W_1 \times D_1 \text{ input})$ :

$$H_2 \times W_2 \times D_2 \text{ where } (H_2, W_2, D_2) = \left( H_1 - k/Z_s + 1, W_1 - k/Z_s + 1, D_n \right) \quad (14)$$

where  $k$  is the size of the kernel,  $D_n$  the number of kernel window and  $Z_s$  the step to develop the pooling layer. [19]

For description and comparing the results of prediction, three indicators were calculated, Mean Square Error (MSE), Mean Absolute Error (MAE), and R-squared ( $R^2$ ). The mathematical equations are shown below:

$$MAE = \frac{1}{T} \sum_{i=1}^T |\hat{Y}(i) - Y(i)| \quad (15)$$

$$MSE = \frac{1}{T} \sum_{i=1}^T (\hat{Y}(i) - Y(i))^2 \quad (16)$$

$$R^2 = 1 - \frac{\sum_{i=1}^T (\hat{Y}(i) - Y(i))^2}{\sum_{i=1}^T (\hat{Y} - Y(i))^2} \quad (17)$$

where  $Y(i)$  is the actual value,  $\hat{Y}(i)$  is the forecasted value;  $T$  represents the dataset size and  $\hat{Y}$  is the average value of  $Y(i)$ .

MAE calculates the difference between the actual value and the predicted value. MSE demonstrates a measure of the accuracy of predictive data.  $R^2$  is used in the regression



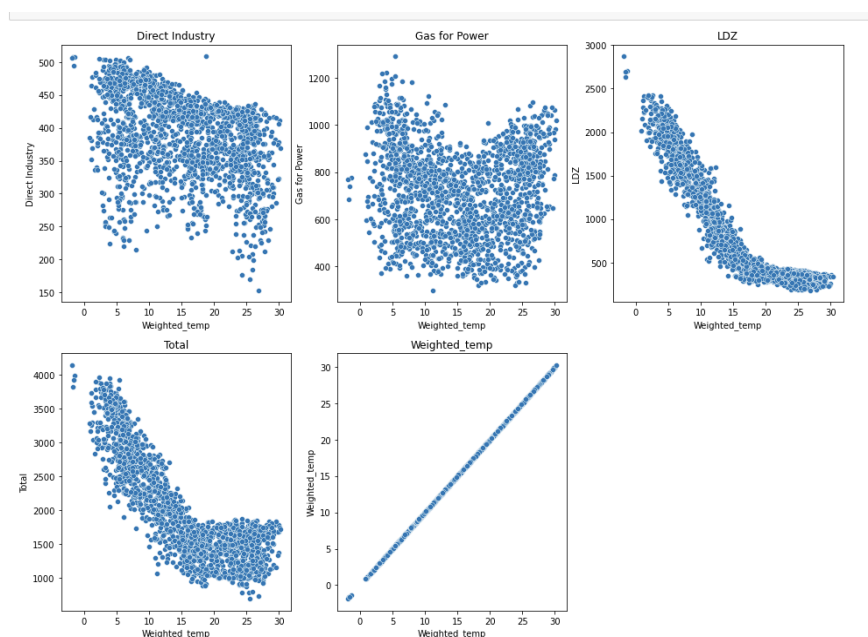
model and determines the accuracy of the predicted value. The  $R^2$  value can quickly determine the accuracy of the predicted values, but exceptions can be made for variant variables. [20]

## EXPERIMENTAL RESULTS

The presented research provided valuable results and conclusions. In Figure 2, can be visualized the correlation, through scatter plots, between natural gas consumption and temperature.

The graphs in question present a differentiated analysis of natural gas consumption in Italy, taking into account three distinct consumption categories: Direct Industry, Gas for Power, and LDZ. Direct Industry represents the consumption of natural gas by large industrial consumers, Gas for Power represents the natural gas used for electricity generation, and LDZ stands for Local Distribution Zone and refers to household consumption. Furthermore, the graphs also depict the Total daily natural gas consumption, which is the aggregate of the three aforementioned categories.

The results obtained from the scatter plot analysis show a linear relationship between natural gas consumption and temperature. This linearity suggests that the temperature change has a direct and proportional impact on the amount of natural gas consumed. In other words, an increase in temperature leads to an increase in natural gas consumption and vice versa.



*Fig. 2 Scatterplot for gas consumption and temperature*

It is also noteworthy to differentiate between household and non-household natural gas consumption based on temperature, the first relationship being stronger. This disparity is due to the greater temperature sensitivity of household natural gas consumption. Large consumers usually have a fixed energy use time schedule and the gas consumption for

power generation is based on several factors, such as the availability and price of alternative sources (coal), government policies, economic conditions, etc. As a result, temperature changes are more likely to have a substantial impact on household natural gas consumption than on non-household consumption.

This is emphasized in Figure 3 where there are presented the cross-correlation coefficient results. The values for direct industry and gas for power suggest a weak negative correlation, whereas for LDZ a strong one.

The differentiation between household and non-household natural gas consumption highlights the importance of considering specific consumption patterns when analyzing energy consumption and its relationship with temperature. Therefore, the next two methods used focused only on household gas consumption and the models that were run predicted household gas consumption.

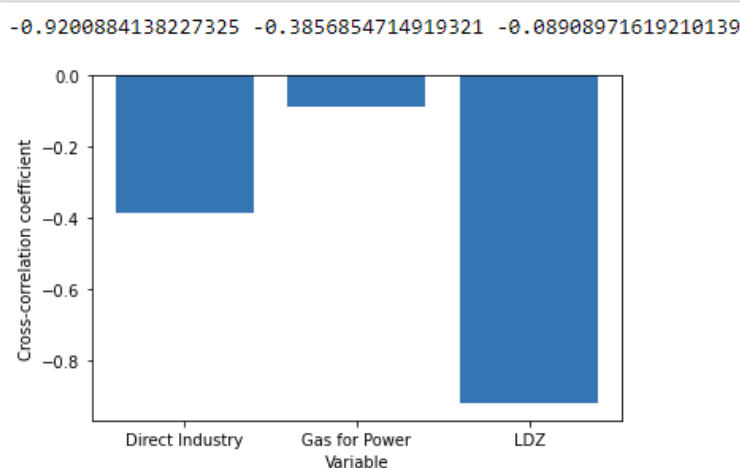


Fig. 3 Cross-correlation coefficient

The SARIMAX model analysis shows that the model fits well with the LDZ data. The coefficients of the model have low p-values and tight confidence intervals, which means they are significant. The log-likelihood of the model is -11001.594 and the AIC is 22009.187, which suggests that the model fits the data well.

The residuals of the model passed the Ljung-Box Q-statistic test, indicating that they are not related. Additionally, the heteroskedasticity test results show a low chance of heteroskedasticity in the residuals, further supporting that the SARIMAX model is a good choice for modeling the LDZ time series data and that there is a dependency between the 2 variables.

All the above suggests that temperature has a strong and meaningful impact on gas consumption. The results of this analysis provide valuable information for decision-makers, as they indicate that temperature should be considered a key factor in any analysis of gas consumption. This statement is highlighted by Jurado López [21] which states that weather conditions have the highest correlation to gas use.



SARIMAX Results						
Dep. Variable:	LDZ	No. Observations:	1817			
Model:	ARIMA(1, 1, 1)	Log Likelihood	-11001.594			
Date:	Fri, 06 Jan 2023	AIC	22009.187			
Time:	16:25:26	BIC	22025.701			
Sample:	0	HQIC	22015.280			
	- 1817					
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-0.5242	0.042	-12.524	0.000	-0.606	-0.442
ma.L1	0.7751	0.028	27.618	0.000	0.720	0.830
sigma2	1.07e+04	246.654	43.392	0.000	1.02e+04	1.12e+04
Ljung-Box (L1) (Q):			0.49	Jarque-Bera (JB):		448.19
Prob(Q):			0.48	Prob(JB):		0.00
Heteroskedasticity (H):			0.75	Skew:		0.43
Prob(H) (two-sided):			0.00	Kurtosis:		5.28

Fig. 4 SARIMAX table results

Before utilizing the data in machine learning models, it is necessary to prepare the data. The first step in this process involves transforming the date column into datetime objects. This is achieved through the use of the `pd.to_datetime()` method. This transformation is necessary to guarantee that the date values are correctly recognized and processed by subsequent functions.

The second operation normalizes the datetime values in the date column by subtracting the minimum value in the column from each value and dividing the result by the number of days. This facilitates a conversion of the date values into a numerical representation, which is an essential element for utilization in machine learning models. The outcome of this operation is a numerical representation of the number of days gone since the earliest date in the dataset.

The `MinMaxScaler` method from the `scikit-learn` library is then utilized for feature scaling on the input variables. Feature scaling is an essential step in machine learning as it assists in ensuring the variables possess the same range of values and stops one variable from having a higher influence than the other. The `MinMaxScaler` method scales the values of the variables between 0 and 1. The feature scaling is performed on two columns of the input data, date, and temperature, which are first extracted from the input data.

The target variable,  $y$ , is also transformed using the method of the `MinMaxScaler`, with the input to the method being a reshaped numpy array of the target variable. It is critical to apply feature scaling to both the input and target variables to guarantee that the range of values for each variable is consistent and to prevent one variable from overpowering the other. This is necessary to ensure that the predictions obtained from the models are precise and trustworthy.

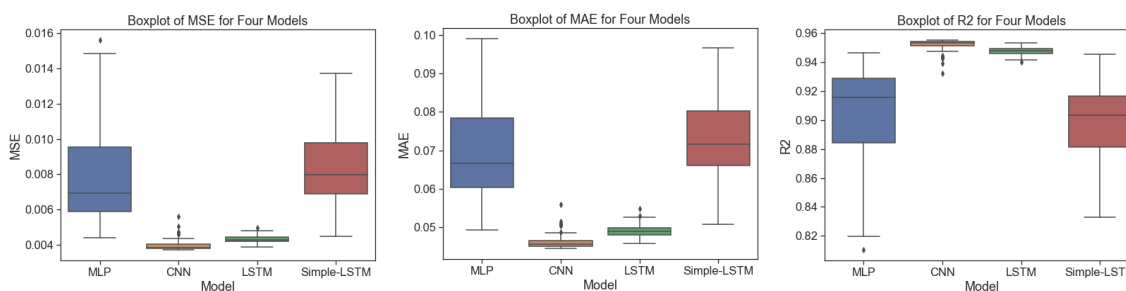
A total of 100 runs of each model were completed to obtain the outcomes. Following each run, the accuracy metrics were measured for both the full and test datasets. The evaluation was conducted for four distinct models: CNN, Stack-LSTM, MLP, and Simple-LSTM. A summary of the average performance of each model across both datasets, in terms of the MSE, MAE, and R2 accuracy metrics, is provided in Table 1.

**Table 1:** Mean performance metrics for each model on the test\_data and full\_data

Model	Test_data MSE	Test_data MAE	Test_data R2	Full_data MSE	Full_data MAE	Full_data R2
CNN	0.003944	0.046138	0.952043	0.003925	0.046030	0.951834
Stack-LSTM	0.004303	0.048978	0.947672	0.004364	0.049548	0.946455
MLP	0.007726	0.069059	0.906049	0.007775	0.069443	0.904601
Simple-LSTM	0.008402	0.073273	0.897826	0.008487	0.073847	0.895864

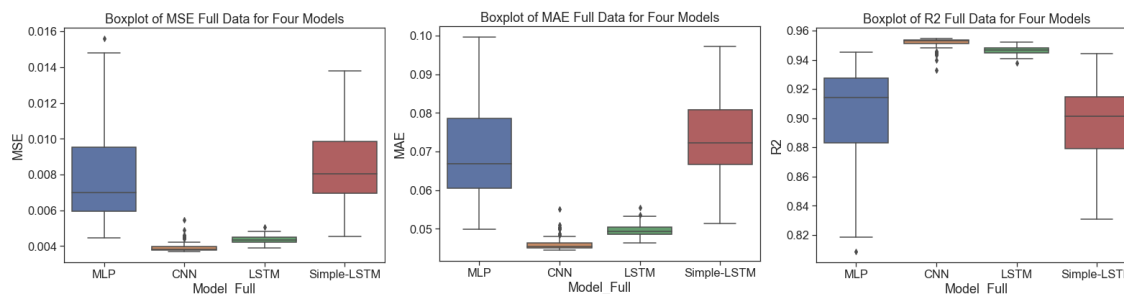
To determine whether there were any major distinctions in the performance between the models, separate ANOVA tests were conducted for each accuracy measure and dataset. The analysis of the test\_data revealed that the models exhibited noteworthy differences in both MSE ( $F(3, 396) = 202.86, p < 0.001$ ) and R2 ( $F(3, 396) = 202.86, p < 0.001$ ), however, there were no considerable variances when MAE was considered ( $F(3, 396) = 319.59, p < 0.001$ ).

The boxplots illustrated in Figure 5 provide an insight into the distribution of the accuracy metrics for each model on the test\_data. It can be observed that the CNN and LSTM models generally achieved better results than the MLP and Simple-LSTM for both MSE and R2, yet there were no remarkable discrepancies between the models concerning MAE.



**Fig 5** Boxplot of accuracy metrics test data for four models

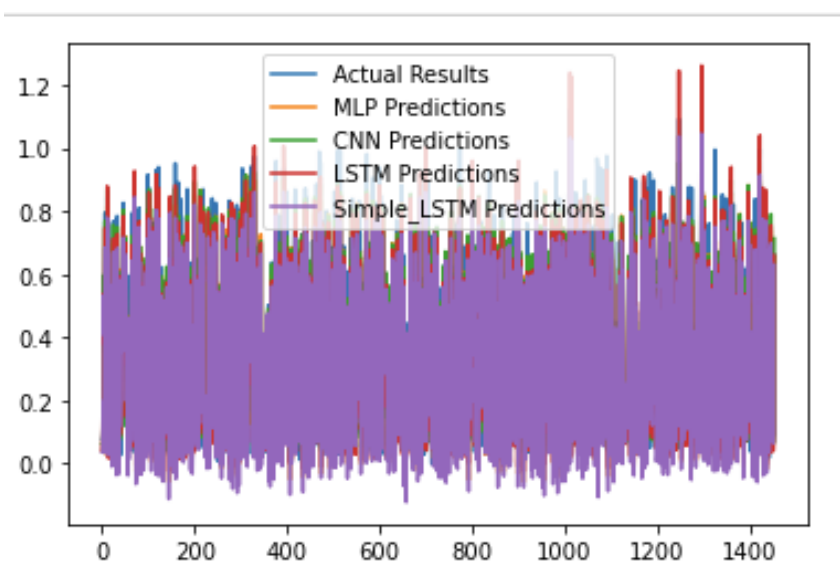
The results for the full\_dataset showed that there were significant differences between the models for MSE ( $F(3, 396) = 208.35, p < 0.001$ ) and R2 ( $F(3, 396) = 208.35, p < 0.001$ ), but not for MAE ( $F(3, 396) = 327.31, p < 0.001$ ). The graphical representation of the results in Figure 6 revealed that the performance of the models followed a similar trend as observed for the test\_data set, with the CNN and LSTM models yielding better results for MSE and R2 than the MLP and Simple-LSTM models; however, there were no significant differences between the models for MAE.



**Fig. 6** Boxplot of accuracy metrics full data for four models

The CNN and Stack-LSTM models are demonstrated to be the most effective in predicting the target variable, as they achieved superior results about the Mean Squared Error (MSE) and the R2 score on both the test\_data and the full\_data.

Finally, a comparison of the graphical representation of real and predicted values (Figure 7 and Figure 8) was conducted to assess the proficiency of the MLP, CNN, LSTM, and Simple LSTM models for the entirety of the dataset and the last thirty days. The results revealed that the CNN and Stack LSTM models were more successful in conforming to the predicted and actual values than the other models, whereas the MLP and Simple LSTM models demonstrated more considerable differences. As such, it can be concluded that the CNN and Stack LSTM models are more suitable for precise predictions.



**Fig. 7** Actual vs predicted values (full dataset)

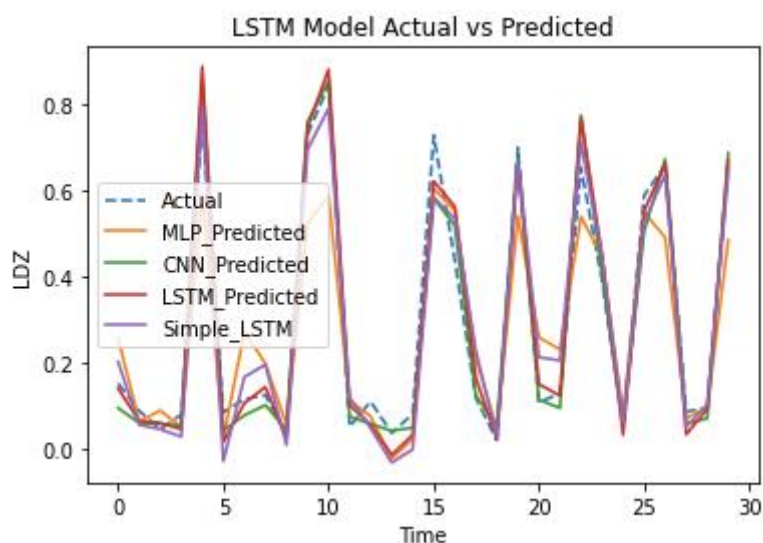


Fig. 8 Actual vs predicted values (full dataset last 30 days)

## CONCLUSION

To forecast future residential natural gas consumption, this study investigates a variety of deep learning techniques that combine natural gas demand prediction with a stochastic temperature model. Forecasts were made using daily data on natural gas consumption and temperature from Italy. It is clear from the correlation between gas consumption and temperature that gas load is negatively related to temperature.

The performance of four models is then evaluated, those being MLP, CNN, Simple LSTM, and Stack-LSTM. Stack-LSTM was chosen as it is a model with two layers of LSTM and it was expected that it would improve the forecasting ability of the Simple LSTM. To ensure that each variable has a consistent range of values and to stop one variable from having a greater influence than the others, the MinMaxScaler method from the scikit-learn library was implemented for feature scaling of the input and target variables.

A performance analysis was conducted by using the Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 accuracy metrics on both test data and full data. To illustrate the distribution of the accuracy metrics for each model across the two datasets, ANOVA tests were conducted and boxplots were utilized.

The results of the study indicate that the CNN and Stack-LSTM models are more successful in accurately predicting the target variable than the other models. This conclusion is supported by the MSE and R2 scores on the two datasets, as well as the graphs that compare the actual and predicted values, which demonstrate that the CNN and Stack LSTM models provide more accurate results than the MLP and Simple LSTM models.

Finally, the feature set was chosen to utilize the fewest number of features while requiring the least amount of computing resources. More features, such as wind speed, rain intensity, humidity, season, hour of the day, and day of week, should be used in future investigations to increase the chances of generating more accurate results.

## REFERENCES

- [1] BP p.l.c., BP Statistical Review of World Energy, UK, 2021.
- [2] Wang H., Gu C., Zhang X., Li F. and Gu L., Identifying the Correlation between Temperature and Gas Consumption in a Local Energy System, IEEE Transactions on Smart Grid, 2020.
- [3] Timmer R.P. and Lamb P.J., Relations between Temperature and Residential Natural Gas Consumption in the Central and Eastern United States, TIMMER AND LAMB, 2007.
- [4] Göncü A., Karahan M.O. and Kuzubas T.U., Forecasting Daily Residential Natural Gas Consumption: A Dynamic Temperature Modelling Approach, Bogazici Journal, 2019.
- [5] Liu J., Wang S., Nan W., Chen X., Xie H. and Wang J., Natural gas consumption forecasting: A discussion on forecasting history and future challenges, Journal of Natural Gas Science and Engineering, 90, 103930, 2021.
- [6] Panek W. and Włodek T., Natural Gas Consumption Forecasting Based on the Variability of External Meteorological Factors Using Machine Learning Algorithms, Energies, 15(1), 348, 2022.
- [7] De Keijzer B., De Visser P., Romillo V.G., Muñoz V.G., Boesten D., Meezen M. and Rahola T.B.S., Forecasting residential gas consumption with machine learning algorithms on weather data, E2S Web of Conferences, 2019.
- [8] Choi Y., Shao X. and Hwang H.S., Highly Accurate Short-Term Gas Consumption and Elapsed Time Forecasting Using Multi-Channel Deep Neural Network, IEEE Access, 2021.
- [9] Aras D.N., Forecasting Residential Consumption of Natural Gas Using Genetic Algorithms, Energy Exploration & Exploitation, Volume 26, pp. 241-266, 2008.
- [10] Fiandra C., Alparone A., Gallio E., Vecchi C., Balestra G., Bartoncini S., Rosati S., Ragona R. and Ricardi U., Automated Heuristic Optimization of Prostate VMAT Treatment Planning. International Journal of Medical Physics, Clinical Engineering and Radiation Oncology, vol. 7, 414-425, 2018.
- [11] Radhakrishnan P., What are Hyperparameters? and How to tune the Hyperparameters in a Deep Neural Network?, 2017. [Online]. Available: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>.
- [12] Sandhya S., Neural Networks for Applied Sciences and Engineering From Fundamentals to Complex Pattern Recognition, Auerbach Publications, 2007.
- [13] Mitchell T.M., Machine Learning, McGraw-Hill Science/Engineering/Math, 1997.
- [14] Wetzell M., Audio Segmentation for Robust Real-Time Speech Recognition Based on Neural Networks, Karlsruhe Institut für Technologie, 2016.



- 
- [15] Saxena S., Introduction to Long Short Term Memory (LSTM), 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>.
- [16] Surakhi O., Zaidan M., Fung P., Hossein Motlagh N., Serhan S., AlKhanafseh M., Ghoniem R. and Hussein T., Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm, *Electronics*, pp. 9-10, 2021.
- [17] Rikiya Y., Kaori T., Mizuho N. and Richard K.G.D., Convolutional neural networks: an overview and application in radiology, *Insights Imaging*, vol. 9, pp. 611-629, 2018.
- [18] Arden D., *Applied Deep Learning - Part 4: Convolutional Neural Networks, Towards Data Science*, 2017.
- [19] Bajaj C., *Statistics & Discrete Methods of Data Sciences, Lecture Notes: Geometry of Deep Learning: Multi-Layered Neural Networks*.
- [20] Choi, Y., Shao, X., Hwang, H.S., Highly Accurate Short-Term Gas Consumption and Elapsed Time Forecasting Using Multi-Channel Deep Neural Network, *IEEE Access*, vol. 9, pp. 157447-157457, 2021.
- [21] Jurado L.C., *Data-driven Predictive Control for Heating Demand in Buildings*, TU Delft, 2017.

---

Received: April 2023; Accepted: April 2023; Published: April 2023