SURYA GROUP OF INSTITUTIONS

NAAN MUDHALVAN

IBM – ARTIFICIAL INTELLIGENCE

NAME:VINOTHINI B

REG NO:422221104048

TEAM NO:04

PROJECT:MEASURE ENERGY CONSUMPTION

MEASURE ENERGY CONSUMPTION

Abstract

The forecasting of building energy consumption remains a challenging task because of the intricate management of the relevant parameters that can influence the performance of models. Due to the powerful capability of artificial intelligence (AI) in forecasting problems, it is deemed to be highly effective in this domain. However, achieving accurate predictions requires the extraction of meaningful historical knowledge from various features. Given that the exogenous data may affect the energy consumption forecasting model's accuracy, we propose an approach to study the importance of data and selecting optimum time lags to obtain a high-performance machine learning-based model, while reducing its complexity.

Regarding energy consumption forecasting, multilayer perceptron-based nonlinear autoregressive with exogenous inputs (NARX), long short-term memory (LSTM), gated recurrent unit (GRU), decision tree, and XGboost models are utilized. The best model performance is achieved by LSTM and GRU with a root mean square error of 0.23. An analysis by the Diebold–Mariano method is also presented, to compare the prediction accuracy of the models. In order to measure the association of feature data on modeling, the "model reliance" method is implemented. The proposed approach shows promising results to obtain a well-performing model. The obtained results are qualitatively reported and discussed.

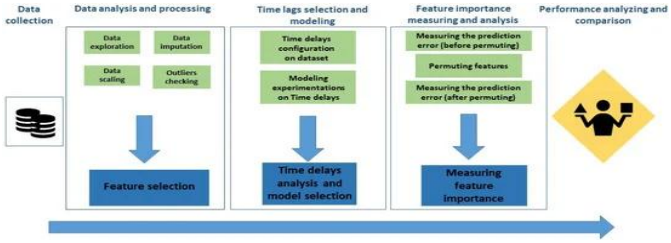
Introduction

Currently, climate change and natural resource shortages have become significant issues. According to the research of Gaya Herrington [1], resources will run out in a few decades if the consumption rate remains stable. France has been involved internationally in combating climate change with the multiannual energy plan (MAEP), which was published on 25 January 2019 [2]. Residential and industrial buildings are the highest consuming sectors in France, with a share of almost 44% of the total final energy consumption [2,3]. Hence, there are substantial investments to accelerate the transition from traditional to smart buildings. Smart buildings have remarkable resource management and control capabilities.

The ability of future smart buildings to forecast energy consumption not only can enhance the energy consumption optimization in buildings, but also, at a higher level, can play a vital role in planning the energy demand response in smart grids. Artificial intelligence techniques are widely used in this domain, and they show their powerful influence, though there remains a wide range of studies to be performed to advance in this investigation area. One of the study areas that needs more attention, and for which there is still a shortage of work, is the importance of feature data and the role they can play to obtain not only an accurate model, but also efficient model construction. The investigation of this area is always complicated due to the unclear participation of various data that can improve or deteriorate the model performance. Having a clear idea of this subject can enhance the model performance and decrease its computation cost. Due to the abovementioned factors, in this investigation, we would like to dig deeper into energy consumption forecasting and study the influence of different data features.

Methods and Protocols

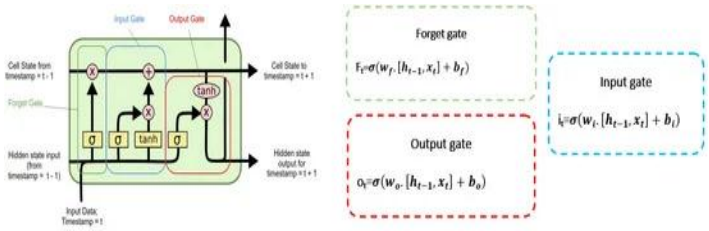
In this section, the approach and methods that are utilized in this research work are presented. The schema of the considered protocol for advancing the proposed research idea is illustrated in Figure 1. It is divided into 5 parts: The first part is the data collection to form the needed dataset. In the case of this research work, an open-access dataset is utilized, which is explained in Section 3. The second part is regarding the representation of the raw data, which is followed by data analysis and processing. In this part, the raw data will be prepared for the training process. As shown in the figure, several processes, such as data imputation and scaling of data, are included in this section. The third part is leading to the construction of machine learning models according to different algorithms and input features. The optimum time lags are realized in this part of the study, based on a protocol that is presented in Section 4 of this article. The fourth part illustrates the important inputs and their roles in the learning process of the model based on error measurement and permutation of data features in a defined framework, which is explained comprehensively. Finally, the last part presents the results of the work for analysis, comparison, and discussion. In the next Section 2.1 and Section 2.2, the utilized materials and techniques for conducting the research are presented.



Machine Learning Algorithms for Modeling Energy Consumption Forecaster

The methodological approach of this research is based on data-driven energy consumption forecasting. Due to the effectiveness of machine learning methods in mimicking complicated time series patterns, LSTM, NARX-MLP, GRU, decision tree, and XGboost are applied.

LSTM is a widely used recurrent neural network in time series forecasting. Its performance in solving time series problems is remarked upon in several works [14,16,17]. LSTM neural networks achieve temporal dependency using special units called memory blocks, which is the main difference between RNNs and ANNs. LSTM is an improved form of RNNs that is capable of overcoming the vanishing gradient problem [18]. The information is passed through a mechanism known as cell states, with three gates to update the previous hidden state. Figure 2 shows the gates and architecture of LSTM [19], where, W_f, W_o, and W_i are the weight matrices, and b_f, b_o, and b_i are the bias vectors. X_t is the current input. h_t and h_{t-1} are the output at the current time t and the previous time t-1, respectively. Finally, σ represents the sigmoid function.



Feature Importance Analysis by "Model Reliance" Method

In this research work, the model reliance method (MR) permits us to select the most efficient model that has the highest performance with less computational cost, regarding the number of participant features and time delays. The model reliance method [23] is based on an analysis of prediction errors. A machine learning method relies on the learning features to perform the prediction. However, depending on relations between the input and output of the model, the reliance on features can differ. If one of the features of the model is permuted, it implies that the association of the permuted feature with other features is broken. As a consequence, in the prediction phase of the model, it is expected that the error based on a permuted feature varies from the original feature. While there are several methods that aid in finding feature importance (e.g., XGboost [24]), MR permits the study of these aspects, by the learning algorithm, for the particular constructed model. In fact, MR is more interpretable for explaining the operation of utilized machine learning algorithms for modeling.

The variation of error based on permuted features is dependent on the importance of the permuted feature in the learning phase. Due to this fact, if the error is jumped more, it indicates that the permuted feature is more important, and the model relies more on that feature. Once the original error is computed, the permuted error for each feature can be calculated by dividing the collected samples of the considered feature into two groups, and swapping the first half with the second half. By doing that, the association between the permuted feature with other features will be broken, and model reliance can be calculated and evaluated. The following equations present the calculation of model reliance.

$$\text{coriginal} = L(y,f(x)) \text{ coriginal} = Ly,fx$$

(2)

$$\text{epermuted} = L(y,f(x\text{permuted})) \text{epermuted} = Ly,fx\text{permuted}$$

(3)

$$\text{epermuted} = 12\lceil n2\rceil \sum \lceil n2/\rceil i\lceil L\{f,(yi,x1\lceil i+n2\rceil,x2,xm\}\rceil + \lceil L\{f,(y\lceil i+n2\rceil,x1,x2\lceil i+n2\rceil,xm\lceil i+n2\rceil\}\rceil \text{epermuted} = 12\lceil n2\rceil \sum i\lceil n2\rceil Lf,(yi,x1i+n2,x2,xm + Lf,(yi+n2,x1,x2i+n2,xmi+n2$$

(4)

where, $e_{original}$ is the original error of the machine learning model, $e_{permuted}$ is the permuted error on the machine learning model, L is the function that calculates the error, f is the machine learning model, n is the number of incidences (samples) in the dataset, y is the true output of the machine learning model, and $\{x_1, x_2 \ldots x_m\}$ are the features.

Then, for calculating MR, the following equation is applied:

$$MR = e_{original} e_{permuted} \quad MR = e_{original} e_{permuted}$$

(5)

The more that MR is larger than one (1 < MR), the more influence it has on the modeling. In the case that MR is strictly less than one (1 > MR), there would be another model that performs better.

Now, considering the proposed materials in the above sections, and regarding Figure 1, in the next step, the dataset will be presented, analyzed, and processed. It provides the needed data to study the selection of tapped delayed line parameters for learning. In time series forecasting problems it is one of the parameters for which there is never a clear approach. Following that, the modeling by different machine learning algorithms is proposed, and several statistical analyses of the results are presented. Finally, a study on features by MR is presented to discuss the performance of models based on exogenous data.

Dataset Presentation, Analysis, and Processing

Dataset Presentation

The dataset in this study was previously collected and is publicly available. The dataset was collected in an office of the University of Calabria, which is a public building located in the south of Italy (39°21′58.6″ N 16°13′30.9″ E) with Mediterranean weather conditions. The area of the concerned office is 19 m^2 and its height is 2.50 m. The room has two wing windows that face the west. The windows dimensions are 68 × 76 cm. The room is equipped with desktop computers and printers, and its heating and cooling systems are autonomous [25,26]. The data are numerical data. Occupancy data were collected only taking into account the working days and the hours between 8 a.m. and 9 p.m. The occupancy count is performed manually by the person in the monitored office. The considered dataset is sampled every 1 min, from 13 May 2016 to 12 May 2017, using different types of sensors: two CO_2 sensors and air quality thermometers. The state of the door and the window were

monitored using magnetic switches. Figure 7 presents the list of measured features of the dataset.

Dataset Analysis and Processing

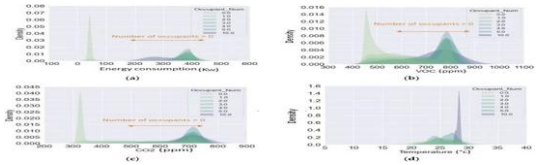
The missing values of the dataset are imputed by interpolation techniques. The autocorrelation between each two features is computed based on the following equation:

$$Autocorr=Cov(x^1(t), x^2(t))\sigma x^1(t)\sigma x^2(t)Autocorr=Covx1t, x2t\diamond x1t\diamond x2t$$

(6)

where, $x_1(t)$,x_2(tx1t,x2(t) are the variables at time, Cov is the covariance, and σ is the standard deviation. The three features highly correlated with energy consumption are the number of occupants, CO_2, and volatile organic compounds (VOC). Their correlations are 0.76, 0.64, and 0.45, respectively. CO_2 and VOC are two variables that are directly related to occupants of spaces in closed environments. The autocorrelations between occupants, CO_2, and VOC are 0.63 and 0.45 respectively.

The density of each continuous variable for each number of occupants is presented in Figure 8. In fact, by density, it shows how many times a measurement is repeated for each feature based on different occupancy numbers, and illustrates the distribution.



Considering Figure 8, it is understood that CO_2, VOC, and occupancy play undeniable roles as exogenous data for modeling. In fact, according to Equation (1), while considering them as sole inputs to the model, it could be argued that energy consumption in time t is a function of CO_2, VOC, and occupancy. Relatively, CO_2 and VOC can be considered as a function of occupancy in covered environments:

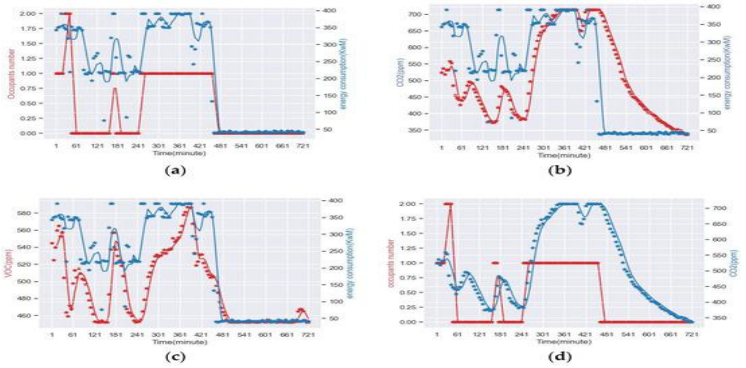
$$E(t)=f(CO^2(Occ(t)),VOC(occ(t)),OCC(t))Et=fCO2Occt,VOCoccct,OCCt$$

(7)

where, occ(t) illustrates the occupation level in time t, f is the function that shows the relation of occupancy level to CO_2, g represents the relation between CO_2 and energy consumption, and E(t) depicts the energy consumption in time t. The following equation reveals the effect of changes in occupancy level and CO_2 concentration on energy consumption by a derivative of energy consumption in Equation (8), with respect to occupancy (occ(t)):

$$dE(t)dOCC=(dfdCO2\times dCO2dOCC)+(dfdVOC*dVOCdOCC)+dfdOCC\ dEtdOCC=dfdCO2\times dCO2dOCC+dfdVOC*dVOCdOCC+dfdOCC \quad (8)$$

It shows that the changes in occupancy lead to changes in CO_2, and energy consumption changes following the change in CO_2. Figure 9 shows the variations between the abovementioned features during 12 h, with a granularity of 5 min.



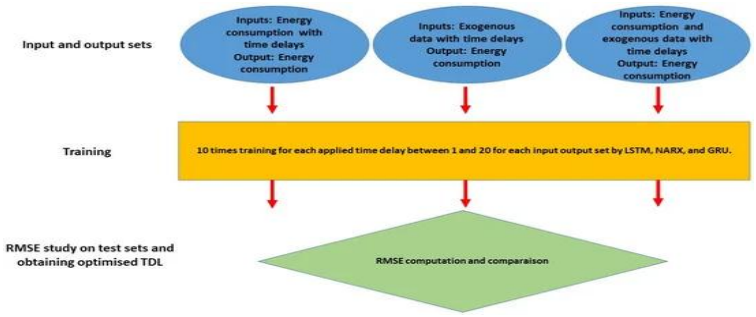
Experimentation and Results

The experimentation of this investigation is implemented based on several protocols and criteria. Three different models, based on three types of matched input features, are constructed for each proposed algorithm:

1. The first set of input features is the energy consumption with applied time lags (ECTL) to predict the energy consumption in their next steps.
2. In the second model, only the exogenous data with applied time lags (ETL), which lacks the energy consumption data, are utilized as inputs to study the model performance and the association of exogenous data.
3. In the third model, energy consumption and exogenous data with applied time lags (ECETL) are used as the inputs to study their roles in the final model's performance.

After finding the appropriate model for each abovementioned case, the model reliance can be implemented to analyze the important features associated with the constructed models. However, before that, it is imperative to find a model with good performance. Applying time lags to features is the most crucial parameter that plays an undeniable role in time series forecasting problems. They affect the computation cost, complexity, and accuracy of the models. Due to this fact, a protocol is presented in the next section to obtain the optimized number of time lags for each case of modeling, based on the different abovementioned input sets.

To study the plausibility of the TDL selection method, several models are constructed by LSTM, NARX based on MLP, GRU, decision tree, and XGboost. For each delay between 1 and 20, the training and testing are performed 10 times to evaluate the models; in each training phase, the weights and biases are initialized randomly (200 models for LSTM, NARX, and GRU for each predefined feature set as inputs). Regarding the decision tree and XGboost, 20 models are constructed for each set of inputs. Considering three sets of inputs, the total number of trained models is 1920. The process of implementation is performed by Python and by several machine learning and deep learning packages. The hyperparameters for NARX, LSTM, and GRU are configured by Bayesian grid search techniques and by trial and error. Of the data, 80 percent is used for training and 20 percent is used for testing. Regarding NARX-MLP, after several trials and errors, two hidden layers with sizes of 25 and 6, with a tangent hyperbolic activation function, are utilized. The maximum iteration is 30. Referring to LSTM and GRU, two layers with sizes of 32 and 16 are used. The activation functions for the two are rectified linear unit (ReLU) and scaled exponential linear unit (SeLU). The batch size is 128 with 10 epochs. Finally, the maximum, minimum, and average root mean square errors of the models for each TDL are computed. Regarding decision tree and XGboost hyperparameters, a grid search is implemented. Figure 12 presents the graph of the implementation of the protocol. The same protocol is implemented for decision tree and XGboost. However, instead of training 10 times for each time delay, a search grid is implemented (as decision tree training is not based on the initialization of weights).



Finally, the models are assessed by the calculation of three metrics, RMSE (root mean squared error), MAE (mean absolute error), and R2R2 (R-squared), defined, respectively, as follows:

$$RMSE=1N\sum ni=n(yi-yi)2----------\sqrt{}RMSE=1N\sum i=nnyi-yi\hat{}2$$

(10)

$$MAE=\sum ni=1|yi-yi|NMAE=\sum i=1nyi-yi\hat{}N$$

(11)

$$R2=1-\sum ni=1(yi-yi)2\sum ni=1(yi-y\square\square\square\square\square)2\ R2=1-\sum i=1nyi-yi\hat{}2\sum i=1n(yi-y^{-})2$$

(12)

where, y_i is the actual energy consumption, $\hat{y}$ is the predicted energy consumption, and n is the number of samples. Table 1 presents the results regarding the models with the optimized number of time delays for each case.

Table 1. Results summary of model performances based on different input features based on LSTM, NARX, GRU, decision tree, and XGboost.

In addition to assessing the models individually, for comparing the accuracy of the models, each against the other, the Diebold–Mariano (DM) test is implemented [27,28,29]. It is a statistical approach that permits us to make a comparison of the prediction accuracy. It assumes:

$$\text{Assumption DM: } \begin{cases} \square\square\square\square\square\square E(d_{12t}) = \mu, \forall t \text{Cov}(d_{12t}, d_{12(t-\tau)}) = \gamma(\tau), 0 < \text{var}(d_{12t}) = \sigma^2 < \infty \forall t \text{Assumption} \\ \text{DM: } Ed_{12t} = \blacksquare, \forall t \text{Cov} d_{12t}, d_{12t-\blacksquare} = \blacksquare\blacksquare, 0 < \text{var} d_{12t} = \blacksquare^2 < \infty \forall t \end{cases}$$

(13)

where, d_{12} is the loss differential between predictions one and two. $E(d_{12})$ represents the hypothesis of equal predictive accuracy, which is $E(d_{12}) = 0$, under the retained assumption DM:

$$DM_{12} = d\square\square\square\square\square_{12} \hat{\sigma} d\square\square\square\square\square\square_{12} \rightarrow_d N(0,1) DM12 = d\bar{}12 \blacksquare \wedge d\bar{}12 \rightarrow_d N0,1$$

(14)

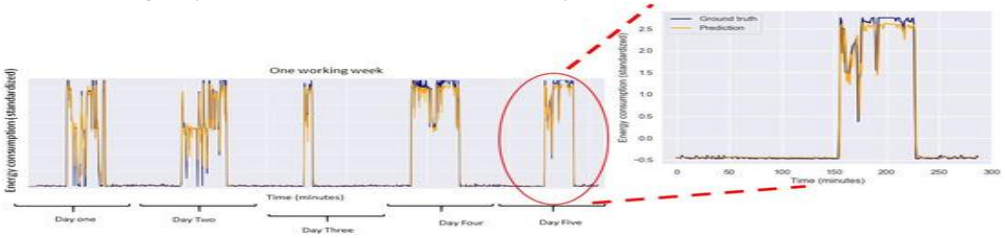
where, $d\square\square\square\square\square_{12} d\bar{}12$ is the average of the sample of loss differential and $\hat{\sigma} d\square\square\square\square\square\square_{12} \blacksquare \wedge d\bar{}12$ is a consistent estimate of the standard deviation of $d\square\square\square\square\square_{12} d\bar{}12$. If the assumption of DM is maintained, consequently, the $N(0,1)$N0,1, which is the limiting distribution of the static test, should be preserved.

To begin, it has to be noted that in the case of decision tree and XGboost there is just one value for each metric, which is due to the different configurations of decision tree and XGboost. NARX, GRU, and LSTM are neural networks and, in the training phase, each of the time weight biases are initialized randomly. However, decision tree and XGboost fit the data based only on tuning of the hyperparameters. The best performance, by considering RMSE as the critical condition, is achieved by the LSTM and GRU models, with the energy consumption time delay (ECTL) selected as input. The RMSE for the concerned models is 0.23. However, GRU achieved this RMSE by 3 time lags, and LSTM by 10 time lags. The time delay in the case of ECTL for LSTM is higher than all of the other algorithms. However, it should be noted that, for instance, according to Figure 13a,d, LSTM still has a better performance than NARX in time lag four. In addition, the selected time lag of 10 does not mean that LSTM could not perform better with a less computationally demanding and complex model.

In the case where only ambient data is used for modeling (ETL model), concerning RMSE values and the number of delays, GRU and LSTM perform the opposite of other algorithms. LSTM and GRU have lower error in lower time lags, but on the other hand, other algorithms need more historical data in order to perform as well as LSTM and GRU. This indicates their ability to achieve better results with lower historical data, in this case. Although, NARX and decision tree, in two other cases (ECTL and ECETL) and with lower time lags (two and three), performed more comparably. In the case of ECETL, it should be mentioned that decision tree with a lower number of time lags has a better performance than NARX, and a little higher than GRU and LSTM.

In general, while energy consumption is included in the input data, the performance of the models is much higher than in the case where only the ambient data is utilized as input. In both MAE and RMSE criteria, the performances of the models for ECTL and ECETL are

comparable, with a slightly better performance for ECTL. However, it should be noted that in conditions where multistep prediction (where the predicted energy consumption is output with a closed-loop feedback to the input) is considered, the exogenous data will influence the performance, and ECETL can show its advantage over the other two [21]. Figure 14 presents the prediction of energy consumption during one working week, and the zoomed in portion shows one working day (for better visualization) by LSTM-ECTL.



Conclusions

In this article, an approach for modeling a well-performing energy consumption forecasting model is proposed and analyzed. This article has three stages for modeling and analyzing. In the first stage, the inputs that are most correlated to the outputs are selected, and, according to defined protocols, three different types of models based on different inputs are constructed by two well-known algorithms in this domain, which are LSTM, NARX by MLP, GRU, decision tree, and XGboost. In the second stage of this article, an efficient approach is utilized to obtain the best time delays for each proposed model. The goal is to optimize the number of time delays parameter to achieve a less complex model that has the highest performance. The highest performance is achieved with ECTL and ECETL, where they perform with almost 0.07 as the minimum MAE. The lowest performance is exhibited by the models where the input features are solely ambient data (ETL). The MAE of these models for LSTM, NARX, GRU, decision tree, and XGboost are 0.22, 0.21, 0.2, 0.22, and 0.22, respectively. The DM test is also clarified, statistically, the accuracy of the models' predictions. In most cases, it confirms that there is not much difference between the models, as illustrated by the resulting metrics. In the last stage, the model reliance method is applied in order to quantify the contribution of features and time delays in the constructed models. The results show that, in the case of modeling just with ambient data, occupancy participates the most; in the other two cases, it is the energy consumption with time lags. Regarding the time delay MR score, the highest model reliance score is achieved by the first time lag. As the time step increases, the score falls. The results of the model reliance analysis also confirm the proposed method regarding obtaining an optimized number of delays as, in all cases, the scores are higher than 1. In the end, among utilized algorithms for modeling of ECETL and ECTL, NARX, with less complex architecture and computation, appears to be a better choice for this case study. Where the input and output features are not highly correlated, and the model's MR score is low, which is the ETL case, LSTM appears to be the better choice.

In the next step of the analysis, as promised, MR is applied to time delay slices for the most important features that have the higher scores. presents MR scores versus time delays of the most important features, according to (features with the highest MR scores). They are energy consumption for the first and third case, and occupancy for the second case. In all situations, despite a slight fluctuation, a declining orientation is observed from dI (which is one time step before the predicted output) to further time steps. Indeed, delayed dI is the most correlated to the output of the model (as is also illustrated in). It is interesting to note that all MR scores of time delays are higher than I, which shows the effectiveness of the approach to choose the optimum time delays in the last section of the investigation.