Journal of Energy

VOLUME 73 Number 2 | 2024

journal homepage: http://journalofenergy.com/

Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada)

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman

***Summary* — Load forecasting is crucial for power systems optimal operation and allows power utilities to overcome technical and economic issues. Some forecasting techniques are currently being deployed on a large scale to meet the requirements of increased energy demand while balancing it with the production to achieve socio-economic benefits for sustainable development. In this paper, we are diving into the forecasting using the regression method. We are focusing on short-term load forecasting and how it can give businesses valuable insights into future sales, labor needs, and more. Power utilities use short-term load forecasting technology to make reasonable power systems. A forecasting model with low prediction errors helps reduce operating costs and risks for the operators leading to models' optimization. To make things real, we are using actual load and weather data from the Hydro-Quebec database. We will be exploring the capabilities, advantages, and limitations of this method, all while keeping an eye on the changing landscape of electricity supply and demand. Our study is centered around the Mascouche region in Quebec, Canada, where the load fluctuates between 60 to 140 megawatts.**

***Keywords* — Forecast, regression, short-term load forecasting, data, optimization, power systems**

I. Introduction

In many decision-making processes, prediction plays a key role and should take into account the stochasticity in its outcome. Load forecasting is essential to maintain the balance of power supply and demand in power grids, and serves as the foundation of power market operation. Power systems planning and operation rely on accurate load forecasting on various time horizons [1]. Electricity is an essential guarantee for industrial production and social life. To meet the consumers' satisfaction and generate profit, electric power companies should balance the supply and need by scheduling a series of generators in the most efficient manner [2]. Power systems' operators need to make power generation plans in advance to achieve economical and reliable operation. The electricity demand is increasing with the economic growth. The emergence of a large number of electric vehicles and various household appliances has brought more uncertainties to the management of power grids. In addition, the increase in intermittent renewable energy sources (RES) connected to the power grids, especially photovoltaic and wind turbine, makes it more challenging to manage power systems. The increased RES penetration has raised the need for spinning reserves to offset the fluctuation of RES generation [3]. These two new challenges pushed electric power companies to improve their operational capabilities. As a result, decision-makers have higher requirements for the accuracy of load forecasting.

Due to measurement errors, lack of knowledge of input data, and model approximation errors (e.g. due to imperfections in the model formulation, the estimation process, etc.), prediction uncertainty can arise. In load forecasting, the multiple linear regression method is used to seek a statistical insight into the relationship between dependent and independent variables. Regression analysis does so by using ordinary least square estimation to draw a linear relationship between load and its determinants.

In energy systems, uncertainty in the prediction of the key factors, due both to the stochasticity in the data and the approximation of the prediction models, can cause high costs to the market participants (generators, customers, etc.) when not properly accounted for. Particularly, in intermittent renewable energy integrated power systems (wind and solar PV), the impact of such a highly variable energy sources on system reliability is an important aspect that must be assessed when wind and solar power penetration is significant. Therefore, considering the high penetration of intermittent power sources in the new competitive power systems, the necessity of having access to reliable methods of power prediction has become more evident for the sustainability and efficient management of the energy market: combining accurate short-term load forecasts enables operators to commit the balance of the generation fleet to economically and securely serve future load.

This paper presents a real time case study of Mascouche region through the use of regression learner application as short-term load forecasting method. Different methodologies in load forecasting are discussed. Then performance evaluation indices are introduced and thoroughly presented. Finally, the regression method is applied to predict load using load and weather data followed by discussions that show the different errors out of Gaussian Process Regression (GPR) models found as the best tool for training the prediction model, and insights into future directions.

(Corresponding author: Tahar Tafticht)

Mouctar Tchakala, Tahar Tafticht and Md Jahidur Rahman are with the University of Québec in Abitibi-Témiscamingue (UQAT), Rouyn-Noranda, Canada (e-mails: mouctar.tchakala@uqat.ca, tahar.tafticht@uqat.ca, mdjahidur.rahman@uqat.ca)

Iqbal Messaïf is with the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria (e-mail: imessaif@usthb.dz)

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman, Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada), Journal of Energy, vol. 73 Number 2 (2024), 13–20
https://doi.org/10.37798/2024732502

II. Materials and Methods

To better understand the framework of the proposed model, this section introduces the relevant methods, including two- stage forecasting model, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Attention Mechanism (AM), and Multi-layer Perceptron (MLP).

A. Two Stage Forecasting Model

The two-stage forecasting model is shown in figure 1. The proposed model consists of an LSTM-based module with the AM for multi-step forecasting and an MLP-based module for residual modification. The inputs of the sequence to sequence (seq2seq) module are over the past *n* hours where *n* is the length of the input window, including MoY_t (month of year, 1 to12 represent January to December), DoW_t (day of week, 1 to 7 represent Monday to Sunday), HoD_t (hour of day, 0 to 23 represent 0:00~1:00 to 23:00~24:00), L_t (load), and T_t (temperature). At time *t*, the seq2seq module processes the input, $X = (X_{t-n+1}, X_{t-n+2}, \dots, X_t\ )^T$, where $X_i = (MoY_i, DoW_i, HoD_i, L_i, T_i)$ and outputs the predictions of the next 24 h load demand via the fully connected (FC) layer, $y' = (y'_{t+1}, y'_{t+2}, \dots, y'_{t+24})$ [2].

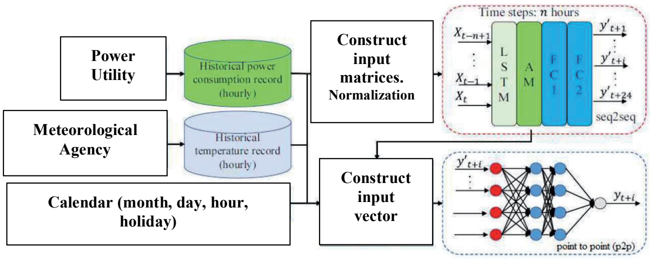

Fig. 1. The framework of the two-stage short-term load forecasting (STLF) model

B. Recurrent Neural Network

The RNN is a generalization of feedforward neural network that has an internal state (i.e., memory), making it applicable to process sequences of inputs, such as speech recognition [4], natural language processing [5], and time series prediction [6], [7]. When processing a sequence of input, the RNN performs the same function for each input of data. After producing the output of current input of data, it is duplicated and sent back into the RNN as a component of the next input. Figure 2 shows the structure of a simple RNN. The mapping from the input X_t to output y_t can be described using following equations [4]:

$$h_t = f(W_{xh}.X_t + W_{hh}.h_{t-1}) \tag{1}$$

$$y_t = g(W_{hy}.h_t) \tag{2}$$

Where h_t is the hidden state at time *t; W*$_{xh}$ are shared weight matrix at current input state, previous hidden state, and output state, respectively; *f(.)* and *g(.)* are the activation functions.

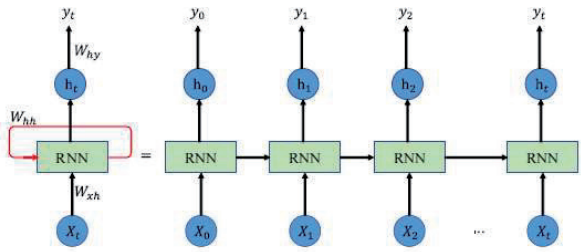

Fig. 2. The structure of a simple reurrent neural network (RNN)

Initially, the RNN takes X_0 from the sequence of input and generates hidden state h_0 and output y_0. In the next step, h_0 and X_1 are the input. The RNN repeats this process till the end of the sequence. In this way, the RNN keeps remembering the previous information. Thus, it is good at processing the sequence whose contexts are intrinsically related. However, the RNN is trained using backpropagation algorithm, and, therefore, gradient vanishing problem may occur when the sequence became very long.

C. Long Short-Term Memory (LSTM)

The LSTM network is a variant of RNN, which has several gates to control the input, memory (i.e., cell state), and output, making it remembers past information more efficiently [8]. So that the gradient vanishing problem is resolved. The structure of an LSTM network is shown in Figure 3.

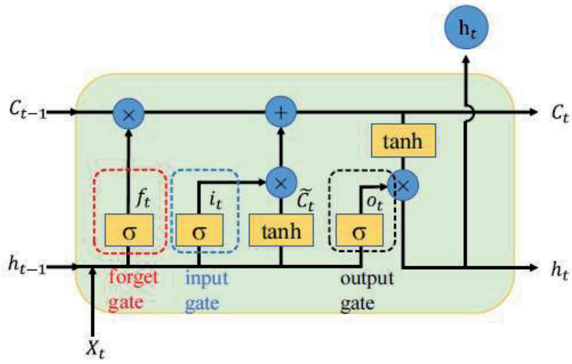

Fig. 3. The structure of long short-term memory (LSTM)

The first step in the LSTM sequence process, is to decide which information in the memory will be thrown. The decision is made by the forget gate via the sigmoid function, whose output is a value between 0 and 1. The larger the output value is, the more past information is kept in the memory. The calculation of the forget gate can be expressed as [8]:

$$f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{3}$$

Where W_f and b_f are the weight matrix and bias of the forget layer, h_{t-1} is the output (i.e, hidden state) at time t-1, X_t is the input of current state.

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman, Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada), Journal of Energy, vol. 73 Number 2 (2024), 13–20
https://doi.org/10.37798/2024732502

The next step is to determine which new information will be stored in the memory by the input gate. The tanh layer creates a candidate of the input. Meanwhile, the input gate will generate a value it between 0 and 1 through the sigmoid layer. The candidate, $\check{C}_t$, will be scaled by i_t and added to the memory to culations of $\check{C}_t$, i_t, and C_t are as follows [9]:

$$\check{C}_t = tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + bi) \tag{5}$$

$$C_t = f_f * C_{t-1} + i_t * \check{C}_t) \tag{6}$$

Where W_C and b_C are the weight matrix and bias of the tanh layer, W_i and b_i are the weight and bias of the sigmoid layer of the input gate, respectively.

Finally, the LSTM gives the output controlled by the output gate. We put the current cell state Ct through a tanh layer and multiply it by the scalar generated by the output gate.

The scalar and output of the LSTM can be computed by [9]:

$$o_t = \sigma(W_0 \cdot [h_{t-1}, X_t] + b_0) \tag{7}$$

$$h_t = o_t * tanhC_t \tag{8}$$

Where WO and bO are the weight and bias of the sigmoid layer of the output gate respectively.

Where W_C and b_C are the weight matrix and bias of the tanh layer, W_i and b_i are the weight and bias of the sigmoid layer of the input gate, respectively.

D. Attention Mechanism

The AM in Deep Learning is based on the concept of directing the focus, making the networks pay greater attention to certain factors when processing the input of data. It is used to manage and quantify the interdependence within the elements of the input sequence. Therefore, when generating the output over any time step, this layer has viewed the whole input sequence and captured the relationships between any two timeslots. [2]

A self-attention mechanism can be described as mapping a query and a set of key-value pairs to an output. The self- attention layer operates an input sequence, $a = (a_1, a_2, \ldots, a_n)$ where $a_i \in \mathbb{R}^{da}$, and generates a new sequence $b = (b_1, b_2, \ldots, b_n)$ where $b_i \in \mathbb{R}^{db}$. The query, keys, and values are computed by [33]:

$$q^i = W^q . a_i \tag{9}$$

$$k^i = W^k . a_i \tag{10}$$

$$v^i = W^v . a_i \tag{11}$$

Where W^q, W^k, $W^v \in \mathbb{R}^{da \times db}$ are the training parameters.

The output b_i is computed as weighted sum of a linear transformed input elements [10]:

$$b_i = \sum_{j=1}^{n} \propto_{ij} (x_j W^v) \tag{12}$$

The $\propto_{ij}$ represents the weight coefficient between elements x_i and x_j which is computed using a softmax function [10]:

$$\propto_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} exp\ e_{ik}} \tag{13}$$

The attention score e_{ij} can be calculated by [10]:

$$e_{ij} = \frac{(x_i W^q)(x_j W^k)^T}{\sqrt{d_b}} \tag{14}$$

E. Multilayer Perceptron

The MLP is a class of feedforward Artificial Neural Networks, consisting of a series of interconnected neurons (i.e., nodes). The structure of a simple MLP network [11] with one hidden layer is shown in Figure 4. The network consists of an input layer with d units, one hidden layer with m units, and one output layer with one The MLP is a class of feedforward Artificial Neural Networks, consisting of a series of interconnected neurons (i.e., nodes). The structure of a simple MLP network [11] with one hidden layer is shown in Figure 4. The network consists of an input layer with d units, one hidden layer with m units, and one output layer with one node. Since MLP networks are fully connected, each node in one layer connects with a certain weight to every node in the following layer. The outcomes of the MLP can be described by Equation (15) [11].

$$y = \sum_{j=1}^{m} \lambda_j . \psi \left(\sum_{i=1}^{d} w_{ij} . x_i \right) \tag{15}$$

Where w_{ij} denotes the weight from ith neuron of input layer to jth neuron of hidden layer, λ_j denotes the weight from jth neuron of hidden layer to output layer, and ψ is the activation function. One of commonly used activation functions is the rectified linear unit (ReLU) function, which can be written as:

$$\psi(x) = \max(0, x) \tag{16}$$

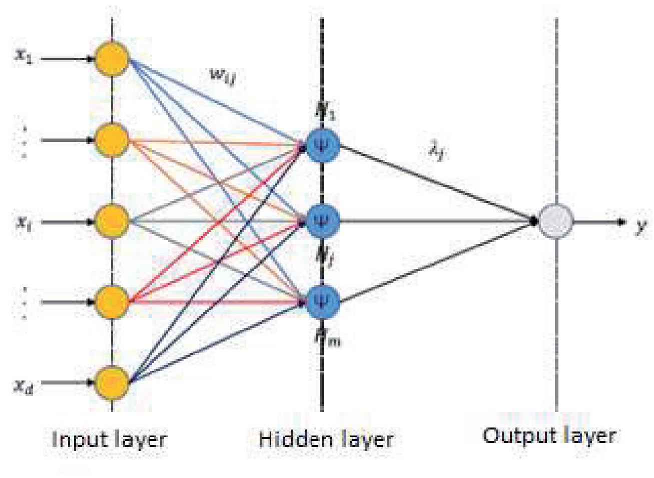

Fig. 4. A multilayer perceptron (MLP) network with one hidden layer

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman, Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada), Journal of Energy, vol. 73 Number 2 (2024), 13–20
https://doi.org/10.37798/2024732502

The MLP can approximate highly non-linear functions between the input and output without any complex mathematical formula. It has been approved that the performance of MLP in forecasting applications outperforms regression-based methods [12]. The MLP may have more than one hidden layer and multiple units in the output layer depending on different cases.

F. Advantages And Disadvantages Of Different Machine Learning Models

The advantages of linear regression include simplicity, interpretability, and fast training processes. However, this method may yield misleading results if the real relationship in the dataset is not linear. Additionally, it may be limited in expressing very complex relationships and may not model nonlinear relationships correctly. Advantages of clustering algorithms include the ability to discover structures in the dataset, better understand datasets, identify power sources with similar properties, and reveal similarities of different power sources. However, it should be noted here that these algorithms are sensitive to the data and metrics used. Additionally, in some cases it is possible to have difficulty in making a clear distinction when data points need to be separated into different groups. Advantages of artificial neural networks (ANNs) include the ability to identify and learn complex relationships, flexibility, performing well on large datasets, and scalability. Moreover, ANNs may reveal hidden patterns within data and analyze complex structures, owing to their learning ability. However, ANNs might sometimes encounter overfitting problems. This means that the network overfits the dataset and reduces generalizability. Additionally, training and configuring ANNs may require time and computational power. The advantages and disadvantages of the machine learning models discussed in this study are listed in Table 1.

Table I.
Advantages And Disadvantages Of Ml AndAnn Models

Model	Advantage	Disadvantage
Linear Regression	Simple to apply, easy to interpret	It is limited to linearity, not suitable for use if there is no relationship between variables, and is sensitive to noise
Decision Tree	It is robust to independent variables, adaptable to nonlinear data, has high prediction success, can deal with categorized values, and does not need too much data	Variable to data change, unstable with noise, overfitting, not smooth or continuous, unsuitable for unstable data classes, computationally costly for large data
Support Vector Machine	Works well when separation is clear, becomes more effective as size increases, and is memory efficient	It is not suitable for large datasets; noise negatively affects the results
Efficient Linear	It is easy to apply and understand	As the number of variables increases, adaptability decreases
Gaussian Process Regression	Provides reliable intervals; suitable for uncertainty estimation	It is only for data with positive values; not suitable for data with noise
Kernel	It can capture complex and nonlinear patterns, save memory, and provide flexibility	Interpretation and explanation of the model are difficult. As feature size increases, extension becomes difficult and overfits new data
Ensemble of Trees	Good for complex and noisy problems in decision trees, adjusts variance adjustment	For multiple models, computational costs are high, and interpretation and explanation are difficult
Neural Network	It provides effective visual capability, the ability to process data even if it is not edited, and it is adaptable and offers a user interface	It has high hardware requirements, data-based operation, poor control capability, and may produce incomplete results
Long Short-Term Memory	It can capture long dependencies and handles sequential data quite well	Hardware costs are high; the dataset must be large

G. Merit Based Selection Of The Gaussian Process Regression (GPR)

In this paper, we are designing a load forecasting method using data which are: (i) positive values; (ii) noiseless; and (iii) suitable for regression techniques application.

Futhermore, GPR provides reliable intervals and is suitable for uncertainty estimation which is the main concern of this paper: efficient load forecasting to reduce uncertainty in power forecasting which seriousely hinders power supply through power grids and smart/mini-grids.

Based on the above, we chose to use GPR as it is clearly suitbale to the technique being designed and the nature of the real-time data used.

H. Performance Evaluation Indices

The forecasted load is compared with the actual measured load for each regression model. By calculating three different statistical evaluations, the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE), the load forecasting capacity of each method and model accuracy can be assessed [13], [14].

H.1 Mean Absolute Error (MAE)

The MAE measures the average magnitude of the errors. It is calculated by:

$$MAE = \frac{\sum_{t=1}^{n} |Y_t - \hat{Y}_t|}{n} \tag{17}$$

Where $\hat{Y}_I$ is the prediction, Y_I is the true value from field recording, and *n* is the number of measurement points.

H.2 Mean Absolute Percentage Error (MAPE)

This error percentage is a measure of the prediction accuracy of a forecasting method in statistics. It produces a measure of the relative overall fit, which can be calculated by:

$$MAPE = \frac{\sum_{t=1}^{n} \frac{|Y_t - \hat{Y}_t|}{Y_t}}{n} \times 100 \tag{18}$$

Where $\hat{Y}_t$ is the prediction, Y_t is the true value from field recording, and *n* is the number of measurement points.

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman, Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada), Journal of Energy, vol. 73 Number 2 (2024), 13–20
https://doi.org/10.37798/2024732502

H.3 Root Mean Square Error (RMSE)

The RMSE is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are. Hence, it is a measure of how spread out these residuals are. It is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\mathrm{Y_t} - \hat{\mathrm{Y}}_\mathrm{t})^2}{n}} \tag{19}$$

Where $\hat{\mathrm{Y}}_\mathrm{t}$ is the prediction, $\mathrm{Y_t}$ is the true value from field recording, and *n* is the number of measurement points.

H.4 R-SQUARED

R-Squared is a statistical measure of how close the fitted regression line is to the results. R-squared lies between 0 and 1. Generally, the higher the R-squared value, the better the model.

The following criteria is used to evaluate load forecasting performance using the error indices:

- The RMSE is always positive and a smaller RMSE value indicates a good model.
- The R-squared lies between 0 and 1. R-Squared near 1 indicates a good model.
- The MSE is the square of the RMSE and a smaller MSE value indicates a successful model.
- The MAE is positive, similar to RMSE, and a smaller MAE value implies a successful model.
- An error percentage very close to zero means the predicted values are very relative to actual values.

III. The Proposed Load Forecasting Approach

The procedure to implement the proposed approach is demonstrated in the following six steps:

- *Step 1: Data collection*. Two datasets need to be collected: one is load demand data; another is meteorological data.
- *Step 2: Models selection*. Some models are used in the proposed method, and suitable ones are selected.
- *Step 3: Input parameters selection*. Important input parameters such as weather parameters are evaluated and selected.
- *Step 4: Models creation and load forecasting conducted using them*. The selected ones in Step 2 will be trained and tested and then will be used to proceed.
- *Step 5: Comparison of the performance*. To compare the performance of the regression models, the forecasted load is compared with the actual measured load, and statistical error matrices are used to evaluate their accuracy.
- Step 6: *Recommendation of the ones with the highest accuracy*. Based on previous steps, the models with the highest accuracy will be selected.

The procedure is shown in the Figure 5.

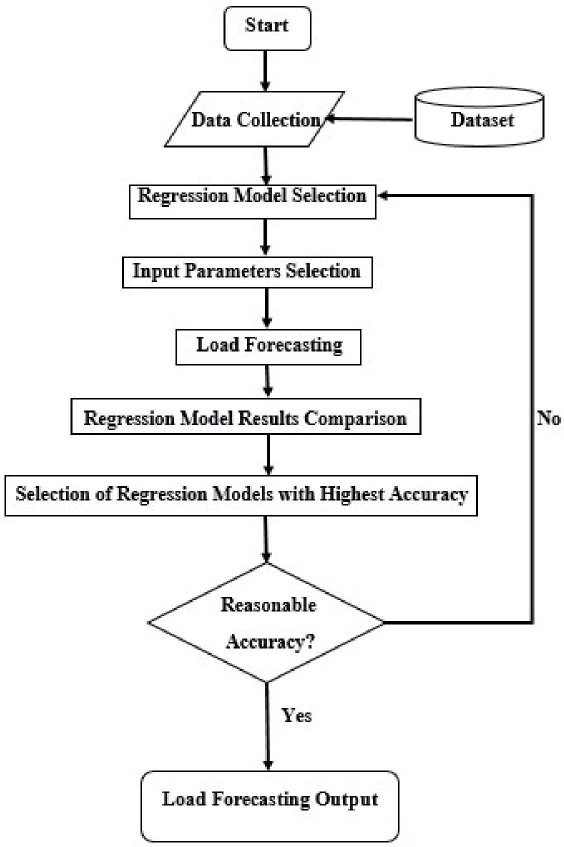

Fig. 5. The flowchart of the proposed load forecasting method

Five families of regression model algorithms provided in the Matlab regression toolbox are suitable for the short-term load prediction. They are Linear Regression, Regression Trees, Support Vector Machines (SVM), Gaussian Process Regression (GPR), and Ensemble of Trees. Table II shows these regression models. The GPR is concerned in this paper.

Table II.
Regression Toolbox

Family of regression models	Types of regression models
Gaussian Process Regression (GPR)	Rational Quadratic GPR
	Squared Exponential GPR
	Matern 5/2 GPR
	Exponential GPR
Ensemble of Trees (ET)	Boosted tress
	Bagged Trees
Support Vector Machines (SVM)	Linear SVM
	Quadratic SVM
	Cubic SVM
	Fine Gaussian SVM
	Medium Gaussian SVM
	Coarse Gaussian SVM
Linear Regression (LR)	Linear Regression Model
	Interactions Regression Model
	Robust regression model
	Stepwise Linear Regression Model
Regression Trees (RT)	Medium Tree
	Coarse Tree
	Fine Tree

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman, Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada), Journal of Energy, vol. 73 Number 2 (2024), 13–20
https://doi.org/10.37798/2024732502

IV. The Proposed Methodology And Data Exploration

Two files (mascouche_1 and mascouche_2) are provided having hourly real load in MW for two months, one month in each file having only working days, i.e., weekends are not included, and here the two months are in wintertime. Mascouche_2, having the file (data 2), is used for training the neural network, and Mascouche_1, having the data 1 file, is used for validating. Each of these two data files has 5 columns: Load, Days, Hours, Temperature, and the average temperature over the last five hours.

Inputs of the network are temperature (T) and the average of the temperature (T_{avg}), columns 4 and 5 respectively (training). The desired output (target) of the network will be the load, column 1, in MW.

The mascouche_1 file is used for validating (testing) the neural network. It can be divided into (three days) windows, i.e., 7 windows having 3 days each:

- Windows 1: days (1, 2, and 3), Windows 2: days (4, 5 and 6).
- Windows 3: days (7, 8, and 9), Windows 4: days (10,11 and 12).
- Windows 5: days (13, 14, and 15), Windows 6: days(16, 17, and 18).
- Windows7: days (19, 20, and 21).

The purpose is that, with the temperature of the next three days, can we predict the hourly load for these three days. Comparison is using the Root Mean Squared Error (RMSE). The RMSE is the standard deviation of the residuals [15-18]. Residuals are a measure of how far from the regression line data points are, so RMSE is a measure of how spread out these residuals are. It can be calculated as follows:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \tag{20}$$

Where $\hat{y}_i$ is the prediction, y_i is the true value from field recording, and n is the number of measurement points.

Below are dataset details:

Data 1 is composed of five (05) columns (Load, Days, Hours, Temperature, and the average temperature over the last five hours) with 528 values each;

Data 2 is composed of five (05) columns (Load, Days, Hours, Temperature, and the average temperature over the last five hours) with 552 values each.

Dataset extent and accuracy are respectively suitable for validation and training the prediction model.

V. The Regression Learner Application

The regression learner application trains models to predict data using supervised machine learning (ML). Using this application, we can explore data, select features, specify validation schemes, train, and assess results. We can perform automated training to search for the best model type, including linear, trees, Gaussian Process (GPR), support vector machines, kernel approximation, ensembles of trees, and neural network models.

We perform supervised ML by supplying a known set of observations of inputs (predictors) and known responses. Use the observations to train a model that generates predicted responses for new input. To use the model with new data, or to learn about programmatic regression, you can export the model to the workspace or generate MATLAB code to recreate the trained model [19-21]. The below flow chart (Figure 6) shows a common workflow for training models in the Regression Learner Application.

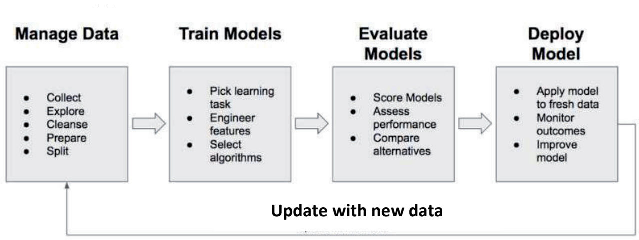

Fig. 6. Regression models training workflow

VI. Results and Discussion

Using ML in load forecasting may significantly improve the efficiency, performance, and quality of power supply.

Below, in Table III, is shown the global simulation results:

Table III.

Model Performances Based On The Mae, Mse, R2, And Rmse

Regression model	Performance Evaluation on Training Dataset				Prediction speed	Training time
	RMSE	R^2	MSE	MAE		
GPR	2.947	0.960	8.685	2.216	~ 40 000 obs/sec	1.683 sec

The above performances table confirm the choice of the RMSE as performance indice considering the low value of R- squared (the higher the R-squared, the better); the high value of MSE; and the drawbacks of the MAE such as: (i) it doesn't penalise large errors as much as small errors, meaning that it might not reflect the true accuracy of the model; (ii) it is not differentiable at zero, meaning optimising using gradient-based methods is harder.

The use of the Regression Learner Application shows its advantages which are (i) better prediction speed, (ii) good performance, (iii) better errors estimation; and its limits such as

(i) less efficient than artificial neural networks, (ii) problem of the right choice of predictors and response for some specific data. The well-known Gaussian Process (GPR) is found as the best tool for training using the method studied with less error (e_{GPR} = 2.947) corresponding to the Matern 5/2 type. The Table IV shows errors' values and types of GPRs. Figures. 7, 8, and 9 show predicted and actual load as well as errors of prediction between both data.

Mouctar Tchakala, Tahar Tafticht, Iqbal Messaïf, Md Jahidur Rahman, Regression Learner Application Model-Based Short-Term Load Forecasting for Mascouche (Quebec, Canada), Journal of Energy, vol. 73 Number 2 (2024), 13–20
https://doi.org/10.37798/2024732502

Table Iv.

Gpr Models And Equivalent Simulation Errors' Values

Model N°	GPR Type	Error Value
1.16	Squared Exponential	3.1420
1.17	Matern 5/2	2.9470
1.18	Exponential	3.6167
1.19	Rational Quadratic	2.9601

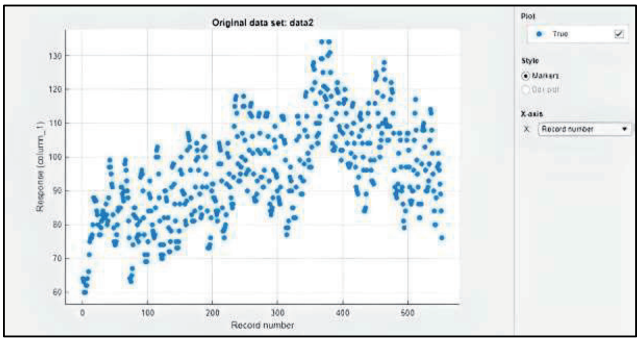

Fig. 7. Original data set with real load

Figure 7 shows the load profile with data 2 values. As indicated, the minimum load is 60 MW and the maximum load is 140 MW.

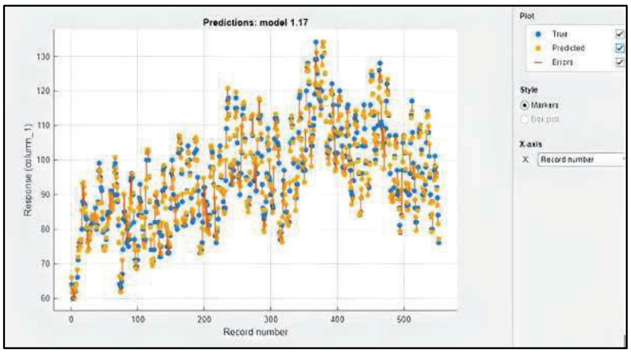

Fig. 8. Response plot

The actual load, predicted and prediction errors are plotted in the figure 8 highlighting the performance of the prediction as the similarity between actual (true) and predicted load is evidenced. In the below figure 9, only real and predicted loads are plotted without errors.

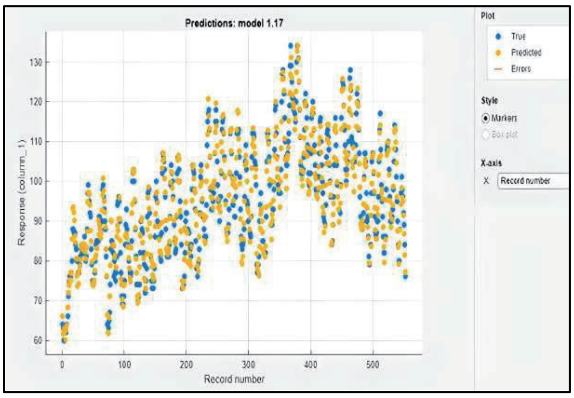

Fig. 9. Response plot without errors

The fineness graph between the actual and the predicted data by the used method is shown in Figure 10. The quality of the prediction following the distribution of the cloud of points indicates how the predicted fits with the actual data. Deviation areas along the perfect prediction line are noticeable.

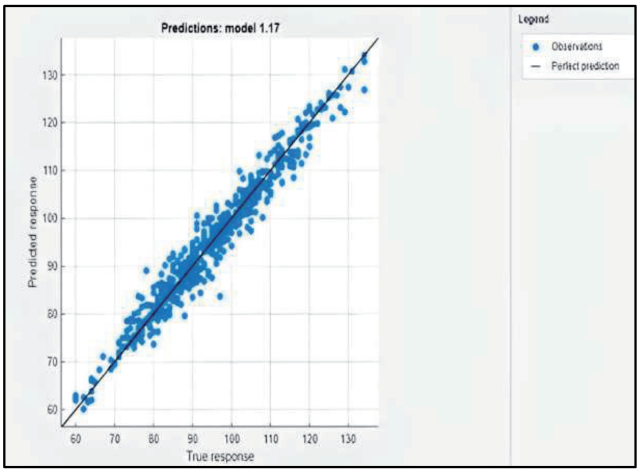

Fig. 10. Validation predicted vs. actual plot

As we strive to improve our model, we plotted the residuals. The distance from the line at zero is how bad the prediction is for that value. We remark that the majority of the points are close to the zero line which implies that the model delivers a good performance (Figure 11).

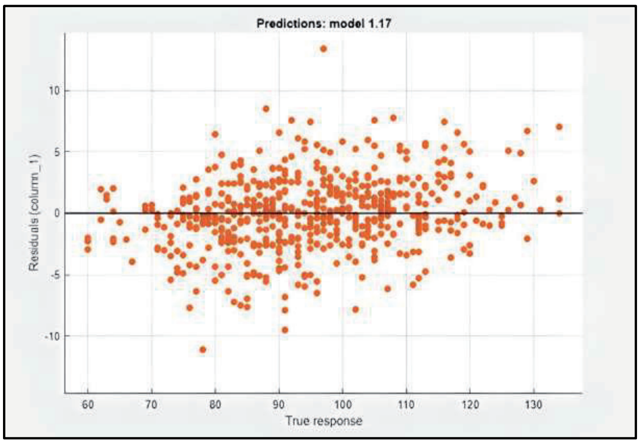

Fig. 11. Validation residuals plot for the regression model

VII. Conclusion And Future Trends

In this study, we've applied the Regression Learner Application to forecast load using weather data. We have highlighted the advantages and limitations of this tool and showcased its capabilities without relying on load history. The recommended models are GPR, which are nonparametric kernel-based probabilistic models. Through our research, we have found that GPR is a viable methodology. It is nonparametric, meaning it is not limited to a specific function, and can calculate the probability distribution over all possible functions that fit the data. GPR is computationally efficient and provides a predictive distribution with mean values and variances. In our case, the Matern 5/2 model is recommended due to its lower error. However, the study has limitations. Hence, we're planning future works to compare and classify the best prediction models and methods. Future trends that can be improved are: (i) detailed simulations with all the regression learner techniques along with comparison; (ii) optimizing data with multi-sources systems management tool (EnergyPLAN) before simulation with ML techniques; and (iii) prediction with the same data using other ML architectures such as SVM, ET, LR, and RT.

References

[1] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, Short-term load forecasting with deep residual networks. *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943_3952, Jul. 2019.

[2] Xie, Y.; Ueda, Y.; Sugiyama, M. A Two-Stage Short-Term Load Forecasting Method Using Long Short-Term Memory and Multilayer Perceptron. Energies 2021, 14, 5873. https://doi.org/10.3390/en14185873

[3] Javadi, M.S.; Lotfi, M.; Gough, M.; Nezhad, A.E.; Santos, S.F.; Catalão, J.P.S. Optimal Spinning Reserve Allocation in Presence of Electrical Storage and Renewable Energy Sources. In Proceedings of the 2019 EEEIC/I&CPS Europe, Genova, Italy, 11–14 June 2019; pp. 1–6.

[4] Mikolov, T.; Kombrink, S.; Burget, L.; C ernocký, J.; Khudanpur, S. Extensions of recurrent neural network language model. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 5528–5531.

[5] Yin, W.P.; Kann, K.; Yu, M.; Schütze, H. Comparative Study of CNN and RNN for Natural Language Processing. arXiv 2017, arXiv: 1702.01923.

[6] Lee, K.Y.; Cha, Y.T.; Ku, C.C. A study on neural networks for short-term load forecasting. In Proceedings of the First International Forum on Applications of Neural Networks to Power Systems, Seattle, WA, USA, 23–26 July 1991; pp. 26–30.

[7] Lee, K.Y.; Choi, T.I.; Ku, C.C.; Park, J.H. Short-term load forecasting using diagonal recurrent neural network. In Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, Yokohama, Japan, 19–22 April 1992; pp. 227–232.

[8] Tian, C.; Ma, J.; Zhang, C.; Zhan, P. A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. Energies 2018, 11, 3493. [CrossRef]

[9] David, K.; Michael, M.; Stephan, S. Short-term temperature forecasts using a convolutional neural network - An application to different weather stations in Germany. Mach. Learn. Appl. 2020, 2, 100007.

[10] Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-Attention with Relative Position Representations. 2018. Available online: https://arxiv.org/pdf/1803.02155.pdf (accessed on 5 September 2021).

[11] Attali, J.G.; Pagès, G. Approximations of Functions by a Multilayer Perceptron: A New Approach. Neural Netw. 1997, 10, 1069–1081. [CrossRef]

[12] Gardner, M.W.; Dorling, S.R. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. Atmos. Environ. 1998, 32, 2627–2636. [CrossRef]

[13] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, vol. 72. Cham, Switzerland: Springer, 2015. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-10247-4#about

[14] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[15] Mamun, A.A.; Sohel, M.; Mohammad, N.; Haque Sunny, M.S.; Dipta, D.R.; Hossain, E. A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models. IEEE Access 2020, 8, 134911–134939

[16] Nanae, K.; Yu, F.; Satoshi, K.; Motonari, H.; Yasuhiro, H. Sparse modeling approach for identifying the dominant factors affecting situation-dependent hourly electricity demand. Appl. Energy 2020, 265, 114752

[17] David, K.; Michael, M.; Stephan, S. Short-term temperature forecasts using a convolutional neural network—An application to different weather stations in Germany. Mach. Learn. Appl. 2020, 2, 100007

[18] Wang, S.X.; Wang, X.; Wang, S.M.; Wang, D. Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. Int. J. Electr. Power Energy Syst. 2019, 109, 470–479

[19] Tian, C.; Ma, J.; Zhang, C.; Zhan, P. A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. Energies 2018, 11, 3493

[20] Moon, J.; Kim, Y.; Son, M.; Hwang, E. Hybrid Short-Term Load Forecasting Scheme Using Random Forest and Multilayer Perceptron. Energies 2018, 11, 3283.

[21] Kuo, P.H.; Huang, C.J. A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting. Energies 2018, 11, 213.

https://doi.org/10.37798/2024732502