2nd German-West African Conference on Sustainable, Renewable Energy Systems SusRES – Kara 2021

Machine Learning

https://doi.org/10.52825/thwildauensp.v1i.25

Published: 15 June 2021

Short-Term Electricity Generation Forecasting Using Machine Learning Algorithms: A Case Study of the Benin Electricity Community (C.E.B)

Agbassou Guenoupkati[1], Adekunlé Akim Salami[1], Mawugno Koffi Kodjo[1], and Kossi Napo[2]

[1]Department of Electrical Engineering, Ecole Nationale Supérieure d’Ingénieurs (ENSI), University of Lome, P.O. Box 1515 Lome Togo.

[2]Solar Energy Laboratory; L.E.S; Faculty of Sciences, University of Lomé, P.O. Box: 1515, Lomé, Togo

Abstract. Time series forecasting in the energy sector is important to power utilities for decision making to ensure the sustainability and quality of electricity supply, and the stability of the power grid. Unfortunately, the presence of certain exogenous factors such as weather conditions, electricity price complicate the task using linear regression models that are becoming unsuitable. The search for a robust predictor would be an invaluable asset for electricity companies. To overcome this difficulty, Artificial Intelligence differs from these prediction methods through the Machine Learning algorithms which have been performing over the last decades in predicting time series on several levels. This work proposes the deployment of three univariate Machine Learning models: Support Vector Regression, Multi-Layer Perceptron, and the Long Short-Term Memory Recurrent Neural Network to predict the electricity production of Benin Electricity Community. In order to validate the performance of these different methods, against the Autoregressive Integrated Mobile Average and Multiple Regression model, performance metrics were used. Overall, the results show that the Machine Learning models outperform the linear regression methods. Consequently, Machine Learning methods offer a perspective for short-term electric power generation forecasting of Benin Electricity Community sources.

Keywords: Linear regression models, Short-term forecasting, Electric power generation, Machine Learning Algorithms

Introduction

Nowadays, with the liberalization and technological advances in the energy sector, several electric companies are in perpetual competition in the energy market to satisfy customer demand. In addition to this, the evolution of demand has huge uncertainties and follows stochastic processes due to several complex factors such as the time, weather, seasonality, economic activity, days, preferential tariffs, occasional events, etc. which are all non-linear. At any given time, the energy supply must equal demand. unfortunately, the amount of electricity generated and the consumption of electricity should be balanced because there is no developed system that can store the electricity that should be generated in case of sudden demand.

Unbalanced demands for electricity generation lead to economic losses and user dissatisfaction. It is therefore important for electricity providers to maintain this balance. Overestimating future load can lead to unnecessary waste of resources, which in turn can lead to additional cost in capital expenditures. However, underestimation of future demand can also result in certain malfunctions or failures that may influence the long-term stability of the power system [1]. In this context, a robust forecasting tool remains essential for decision-making on planning generation sources and improving the national economy. The electricity produced at each point in time is equal to the sum of demand and line losses. However, forecasting errors can, in fact, cause significant operational costs [2]. According to Hobbs et al. [3], a reduction in the average forecast error of 1% can save thousands or even millions of dollars in a power generation unit. Depending on the planning horizon, the different types of forecasts can be classified into four categories: ultra-short-term (less than 1 hour), very short-term (1 to 1 day, or 1 week), medium-term (1 week to 1 year) and long-term (1 year to 10 years).

Forecasting techniques can be classified into two groups, namely statistical models and artificial intelligence (AI) models. Traditional statistical models include regression analysis, moving average, exponential smoothing, stochastic time series models, etc. Machine learning, data mining, artificial neural networks, genetic algorithms, fuzzy time series and expert systems are based on AI techniques. Neural network algorithms are the most popular models for nonlinear time series problems compared to methods that have limitations when the aforementioned exogenous variables that influence power generation are considered. Several works have also focused on the development of ensemble methods in machine learning and hybrid models to improve the accuracy of electricity forecasting. Moreover, many recent studies have been conducted on load prediction using different deep learning techniques [4]. Deep Learning uses artificial neural networks that are inspired by the functioning of the human brain. These networks are composed of a multitude of hidden layers of neurons, each receiving and interpreting information from the previous layer.

The objective of this work is to develop Machine Learning models Support Vector Regression, Multi-Layer Perceptron, and the Long Short-Term Memory Recurrent Neural Network that have a strong generalization capability to predict the electric power production of the Electricity Community of Benin in order to minimize the Mean Absolute Percentage Error (MAPE) and improve the Coefficient of Determination (R^2) and other metrics are used as performance indicators. The CEB is an international organization co-owned by the governments of Benin and Togo. The contributions of this work are declined into five (05) points presented as follows:

- to develop an efficient one-step-ahead forecasting system for electricity generation companies and industries (CEB) for reducing the generating and operating cost;
- to investigate the application of appropriate techniques and tools of forecasting on electricity for Benin and Togo with minimum forecasting error;
- to investigate the application of the generated results as a guideline for the better performance of different Machine Learning models of Communauté Électrique du Bénin;
- to show the power and prospects of Machine Learning algorithms;
- identify the key parameters that influence the electricity generated by the electricity generation companies and industries (CEB).

Modeling

Suppose we have a training data set D containing T pairs of vector x and scalar y given by Eq. (1). Where y_t is a time series and x_t is a vector of dimension d $x_t = [x_1, ..., x_d]^T$. All input vectors are often combined into a matrix X, and the output values into the output vector Y.

$$D = \{(x_t, y_t) \mid t = 1, ..., T\} \tag{1}$$

The general model of a time series is given by Eq. (2):

$$y_t = f(x_t; \theta) + \varepsilon_t \tag{2}$$

Where f is a function that corresponds to the input, x_t the observation at time t, θ the parameter vector, ε_t is a random error term of zero mean that is assumed to have a Gaussian distribution unless otherwise specified by Eq. (3).

$$\varepsilon_t \sim \mathcal{N}(0, \sigma_N^2) \tag{3}$$

The forecast one at horizon h is done by evaluating the function f at the test point x_{T+h}.

$$y_{T+h} = f\left(x_{T+h}, \theta\right) \tag{4}$$

Where θ is the vector of parameters from the training on the training data set D [5]. With this general model, the following section offers a global view of all the prediction models developed in this study.

Multiple linear regression model

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses multiple explanatory variables to predict the outcome of a response variable. The objective of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and the response (dependent) variable. In case of multidimensional analysis. the MLR model is expressed by Eq. (5).

$$y = \beta_0 + \beta_1 \cdot x_1 + \ldots + \beta_n \cdot x_n + \varepsilon \tag{5}$$

Where y is the dependent variable, the x_i are the independent variable, the β_i are the parameters, and the ε are the error.

ARIMA model

There are three distinct integers (p, d, q) used to parameterize the ARIMA models. Hence the contracted notation ARIMA (p, d, q). Together, these three parameters account for seasonality, trend and noise in the data sets. ARIMA models are applied in some cases where the data show evidence of non-stationarity, where an initial differentiation step can be applied one or more times to eliminate non-stationarity [6]-[8]. The autoregressive part of the model (p) allows the effect of past values to be incorporated into the model. The integrated part of the model (d) includes the model terms that incorporate the amount of differentiation to be applied to the time series. The moving average part of the model (q). This allows us to define the error of our model as a linear combination of the error values observed at previous times in the past. An ARIMA (p, d, q) model using the lag polynomial L is expressed by Eq. (6).

$$\left(1-\sum_{i=1}^{p} \varphi_i L^i\right)(1-L)^d = \left(1+\sum_{j=1}^{q} \theta_j L^j\right)\varepsilon_t \tag{6}$$

Where L^i is the lag operator, the φ_i are the parameters of the autoregressive part of the model, the θ_j are the parameters of the moving average part and the ε_t are error terms. Model selection can be performed based on the values of specific criteria such as the standard Akaike information criteria (AIC) [9]. The Akaike information criterion is written by Eq. (7).

$$\mathrm{AIC} = 2k - 2\ln\left(L\right) \tag{7}$$

Where k is the number of parameters to be estimated for the model and L is the maximum of the likelihood function (a function of the parameters of a statistical model calculated from observed data) of the model. If we consider a set of candidate models, the model chosen is the one with the lowest AIC value. This criterion is therefore based on a trade-off between the goodness of fit and the complexity of the model. We solved this problem by programmatically selecting the optimal parameter values for our ARIMA (p, d, q) time series model. We will use a "grid search" to iteratively explore different parameter combinations. Once we have explored the entire parameter landscape, our optimal set of parameters will be the one that gives the best performance for our criteria of interest. This process is called grid search (or hyper parameter optimization) [10] for model selection. Let us start by generating the different parameter combinations we want to evaluate. When evaluating and comparing statistical models with different parameters, each can be compared based on its fit to the data or its ability to accurately predict future data points. We will use the AIC (Akaike Information Criterion) value, which is returned with the ARIMA models. AIC is used to determine how well a model fits the data while taking into account the overall complexity of the model. A model that fits the data very well while using many features will be assigned a higher AIC score than a model that uses fewer features to achieve the same fit. Therefore, we are interested in finding the model that gives the lowest AIC value.

Multi-Layer Perceptron model (MLP)

Artificial neural networks are one of the approaches to artificial intelligence that are being developed through the methods by which humans are still trying to imitate nature and reproduce their own modes of reasoning and behavior. A neuron is essentially an integrator that performs a weighted sum of its inputs. The results of this sum are then transformed by a transfer function f which produces the output y of the neuron. Following the notations presented in the previous section, the n inputs of the neuron correspond to the vector X , while represents W the vector of weights of the neuron. The output s of the integrator is given by Eq. (8).

$$O(t) = W^T \cdot X(t) \pm b \tag{8}$$

This output corresponds to a weighted sum of the weights and inputs plus what is called the bias b of the neuron. The result s of the weighted sum is called the activation level of the neuron. The bias b is also called the activation threshold of the neuron. When the activation level reaches or exceeds the threshold b, then the argument of becomes positive (or zero). Otherwise, it is negative. We will assume that all neurons are synchronous, i.e. that at each time t, they will simultaneously compute their weighted sum and produce an output given by Eq. (9).

$$y(t) = f\left(O(t)\right) = f\left(W^T \cdot X(t) \pm b\right) \tag{9}$$

The activation function f plays a very important role in the behavior of the neuron. It returns a value representative of the activation of the neuron, this function has as parameter the weighted sum of the inputs and the activation threshold. The nature of this function differs according to the network. There are various types of activation function presented in the literature. The ReLU (Linear Rectification Unit) function is a generalization function that solves the saturation problem of the Sigmoid and Hyperbolic Tangent functions. It is the most widely used function in deep network learning.

$$\mathrm{ReLu} = \begin{cases} y(t) = 0 \quad \mathrm{si} \quad O(t) < 0 \\ \\ y(t) = 1 \quad \mathrm{si} \quad O(t) \geq 0 \end{cases} \tag{10}$$

One of the most common algorithms used in this study is backpropagation. This algorithm changes the weights of a network whose architecture is fixed by the operator, each time an example $y_i = f(x_i)$ is presented. This change is done in such a way as to minimize the error between the desired output and the response of the network to an input x_i . At each iteration the input signal propagates in the network in the input-output direction, an output is thus obtained, the error between this output and the desired output is calculated and then by backpropagation intermediate errors, corresponding to the hidden layer are thus calculated and allow the adjustment of the weights $w_{ij}(t)$ of the hidden layer. The gradient backpropagation algorithm thus has 2 phases:

- propagation: at each step, an example is presented to the network as input. This input is propagated to the output layer.
- correction: For sure, the network will not provide exactly what was expected. We therefore calculate an error (usually the mean square sum of the errors for all the output neurons) which we back-propagate in the network. This process is interrupted as soon as the global error is estimated to be sufficient

Support Vector Regression (SVR)

Support Vector Regression (SVR) is an adaptation of Support Vector Machines (SVM) to the regression problem. SVMs and SVRs are a class of supervised learning algorithms, based on the same principles as neural networks. They are based on the search for the optimal hyperplane, which, when possible, correctly classifies or separates the data while being as far away as possible from all observations. The use of SVMs as much in classification optimization as in regression algorithms consists in determining the optimal Lagrange multipliers. The principle is therefore to find a classifier, or a discrimination function, whose generalization capacity is as large as possible. The model of the classifier is built from a training set de $\mathbb{N}$of examples labeled (x_i, y_i) with $x_i \in \mathbb{N}^p$ and $y_i \in \{-1; 1\}$ according to the class represents the dimension of the input vectors or the number of features in the input examples). The training allows, in the case where the examples are linearly separable, to build the decision function f also called separator hyperplane of the form defined by Eq. (11).

$$f(X) = \text{sign}(\langle W, X \rangle + b) \tag{11}$$

With $W \in \mathbb{N}^p$ and b the parameters to determine in which part of the hyperplane. The figure shows the linearly separable case, where the margin Δ is defined by the minimum distance between the two points of the different classes. The principle of SVMs is taken up and adapted by SVRs to model a regression problem. The goal is to approximate a set of data (x_i, y_i) by a function f in the form given by Eq. (12).

$$f(X) = \langle W, X \rangle + b \tag{12}$$

such that the error is expressed by Eq. (11).

$$|f(x_i)\text{-}y_i| \leq \gamma \tag{13}$$

With $i \in \{1,..., N\}$. The idea is to minimize the term w while being under the constraint of not exceeding an error rate γ. If we consider the minimization of $\|w\|^2$ we obtain the quadratic optimization problem. This description of the problem therefore assumes that a linear function f exists that approaches all examples with precision γ. In practice, this is not always the case. In the presence of outliers, it is also more important to allow some errors. In this case, the concept of flexible margin is used. It consists in introducing slack variables ξ_i and ξ_i^* to make the constraints of the optimization problem feasible in Eq. (14).

$$\begin{cases} \min : \dfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \\ \text{subject to} \begin{cases} y_i - w^T x_i - b \le \varepsilon + \xi_i \\ w^T x_i + b - y_i \le \varepsilon + \xi_i^* \end{cases} \end{cases} \tag{14}$$

ξ_i and ξ_i^* representing respectively the positive and negative errors. The constant $C > 0$ is a hyper parameter to adjust the tradeoff between the allowed error and the flatness of the function f . Using the dual formulation and the Lagrange equation, the resulting function can be written by :

$$f(x) = \sum_{i=1}^{n}\left(\alpha_i + \alpha_i^*\right)\cdot K(x, x_i) + b \tag{15}$$

with α_i and α_i^* the Lagrange multipliers from the dual formulation. $K(\ )$ is a kernel function that induces a nonlinear transformation of the data to an intermediate space of higher dimension. Some commonly used kernel functions in the literature are Linear function, polynomial function, Radial basis function (RBF) and Sigmoid. This paper used the Gaussian radial basis function (RBF) as the kernel function, because RBF is the most effective for the nonlinear regression problems. The RBF can be expressed by Eq. (16).

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \tag{16}$$

Where σ is the standard deviation. The selection of optimal hyperparameters was done by cross-validation.

Long Short-Term Memory Recurrent Neural Network (LSTM)

Long Short-Term Memory (LSTM) is an artificial recurrent neural network (RNN) architecture [1] used in the field of deep learning. Unlike neural networks. However, the hidden units are replaced by memory blocks. An LSTM unit consists of a cell (c), an input gate (i) , an output gate (o) and a forget gate (f). The output vector (h) represents the state of the LSTM hidden layers. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The module has three gate activation functions σ_g (sigmoid function), σ_c (hyperbolic tangent), and σ_h (hyperbolic tangent) which is sometimes $\sigma_h(x) = x$. The operation performed by the LSTM layers is given by Eqs. (17).

$$f_t = \sigma_g\left(W_f \cdot x_t + U_f \cdot c_{t-1} + b_f\right) \tag{17}$$

$$i_t = \sigma_g\left(W_i \cdot x_t + U_i \cdot c_{t-1} + b_i\right) \tag{18}$$

$$o_t = \sigma_g\left(W_o \cdot x_t + U_o \cdot c_{t-1} + b_o\right) \tag{19}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c \cdot x_t + b_c) \tag{20}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{21}$$

During model fitting, the mean square error is used as the loss function to optimize the LSTM model parameters and Adam is used as the optimization algorithm for the loss function.

Methodology

Historical power generation data is provided by CEB recorded every 1 hour collected from 1st January 2015 to 31st December 2019. This time series requires pre-processing, especially because there are many missing values and outliers in the original raw data. The presence of these outliers alters the accuracy of the predictions which could be lower. In this project, a filtering technique is used. Then the system is loaded with a dataset composed of 24 input variables using slide windows technique. The data is normalized using the min-max scaling method and then divided into training (1st January 2015 to 30th November 2019) and test sets (1st to 31st December 2019). Then, the SVR, MLP, LSTM, ARIMA and MLR models are built and trained. After that, the models are used to forecasting.

Result and Discussion

The objective of this project is to minimize the mean absolute percentage error (MAPE) which is used as a measure of accuracy. The results are presented in Table 1. According to the table, errors of ARIMA is the highest. Overall, the Machine Learning model performs better than the linear models. The ANN model can predict the electric power generation the minimum MAPE and maximum R^2. After examining the results, we realize that the best model is the Artificial Neutral Network (MLPRegressor) and it has the besSeveral statistical indicators including Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), Normalized RMSE (NRMSE) and Correlation Coefficient (R^2) as well as other statistical tools can be used to provide a proper comparative evaluation of the forecasting models. Table 1 presents the different metrics used to test diffrent model permormance for the 1-hour-ahead forecast.

Table 1. Model performance evaluating using RMSE, nRMSE, MAPE and R^2

Model	RMSE	nRMSE	MAPE	R^2 value	Rank
ARIMA	17.9477	0.064058	14.3430	0.7629	5th
MLR	17.14236	0.061237	4.063912	0.7835406	4th
SVR**	16.6050	0.05931727	3.83646	0.796898	2nd
ANN*	16.2753	0.05825749	3.83245	0.80427	1st
LSTM	18.4812	0.06595	4.26894	0.749155	3rd

Conclusion

The objective of this project is to develop a system for forecasting the electricity balance between supply and demand using the Machine Learning technique and to evaluate its performance by comparing it to other linear regression techniques. Time series forecasting in the energy sector is important for utilities for decision making to ensure the sustainability and quality of electricity supply, and the stability of the power system. Unfortunately, the presence of some exogenous factors such as weather conditions, electricity prices, etc.… complicates the task with the use of linear regression models that become inadequate. Finding a robust predictor would be a valuable asset for utilities. To overcome this difficulty, Artificial Intelligence is distinguished from these prediction methods by Machine Learning algorithms that have been successful in the last decades in predicting multilevel time series. This work proposes the deployment of three univariate machine learning models: Support Vector Regression, Multilayer Perceptron, and Long Term Memory Recurrent Neural Network to predict the electricity production of the Benin Electricity Community. In order to validate the performance of these different methods, compared to the autoregressive integrated moving average model and the multiple linear regression model. Performance metrics were used. Overall, the results show that the machine learning models except LSTM perform better than the linear regression methods. Therefore, machine learning methods offer a perspective for short-term forecasting of electric power generation.

Future Work

- Try on the new input structure more.
- Train the model using more datasets
- Apply another deep learning technique
- Adopted the K-Fold Cross-validation methodology when selecting the best parameter for a single model
- Extend the results of this paper
- Elaborate the work methodology
- Use Google Colab or Amazone Web Service for the study.

References

[1] Zjavka L, Snášel V. Short-term power load forecasting with ordinary differential equation substitutions of polynomial networks. Electric Power Systems Research. 2016 08;137:113-123. https://doi.org/10.1016/j.epsr.2016.04.003

[2] Haida T, Muto S. Regression based peak load forecasting using a transformation technique. IEEE Transactions on Power Systems. 1994;9(4):1788-1794. https://doi.org/10.1109/59.331433

[3] Hobbs B, Jitprapaikulsarn S, Konda S, Chankong V, Loparo K, Maratukulam D. Analysis of the value for unit commitment of improved load forecasts. IEEE Transactions on Power Systems. 1999;14(4):1342-1348. https://doi.org/10.1109/59.801894

[4] Hippert H, Pedreira C, Souza R. Neural networks for short-term load forecasting: a review and evaluation. IEEE Transactions on Power Systems. 2001;16(1):44-55. https://doi.org/10.1109/59.910780

[5] Swastanto B. Gaussian Process Regression for Long-Term Time Series Forecasting. Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology; 2016.

[6] Singh S, Parmar KS, Kumar J, Makkhan SJS. Development of new hybrid model of discrete wavelet decomposition and autoregressive integrated moving average (ARIMA) models in application to one month forecast the casualties cases of COVID-19. Chaos, Solitons & Fractals. 2020 06;135:109866. https://doi.org/10.1016/j.chaos.2020.109866

[7] Zhang L, Lin J, Qiu R, Hu X, Zhang H, Chen Q, Tan H, Lin D, Wang J. Trend analysis and forecast of PM2.5 in Fuzhou, China using the ARIMA model. Ecological Indicators. 2018 Dec;95:702-710. https://doi.org/10.1016/j.ecolind.2018.08.032

[8] Khan FM, Gupta R. ARIMA and NAR based prediction model for time series analysis of COVID-19 cases in India. Journal of Safety Science and Resilience. 2020 09;1(1):12-18. https://doi.org/10.1016/j.jnlssr.2020.06.007

[9] Ma T, Antoniou C, Toledo T. Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. Transportation Research Part C: Emerging Technologies. 2020 02;111:352-372. https://doi.org/10.1016/j.trc.2019.12.022

[10] Mohamed H, Negm A, Mohamed Z, Oliver C. S. Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: case study el burullus lake. International Water Technology Journal. 2015 December;5:352-372.