A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing

Innocent Mpawenimana[1], Alain Pegatoquet[1], Valérie Roy[2], Laurent Rodriguez[1], Cécile Belleudy[1]

1.Univerisité Côte d'Azur : LEAT/CNRS, Sophia Antipolis France
2.Ecole Mines ParisTech : Centre des Mathématiques Appliqués, Sophia Antipolis France
Innocent.Mpawenimana@univ-cotedazur.fr, Alain.PEGATOQUET@univ-cotedazur.fr, laurent.rodriguez@univ-cotedazur.fr, Cecile.Belleudy@univ-cotedazur.fr, valerie.roy@mines-paristech.fr

Abstract— **Energy load prediction plays a central role in the decision-making process of energy production and consumption for smart homes with systems based on energy harvesting. However, forecasting energy load turned out to be a difficult problem since time series data used for the prediction involve both linear and non-linear properties. In this paper, we proposed a system which can predict a daily future energy load in a smart home based on LSTM and ARIMA models. To improve the energy load forecasting accuracy, we propose a new data preprocessing algorithm called STDAN (Same Time a Day Ago or Next) to fill the missing values. This technique is compared with well-known techniques using previous or mean values. A comparison between LSTM and ARIMA is provided for short and medium-term load forecasting. Results show that LSTM outperforms ARIMA in all cases. Finally, we also evaluated our training model based on LSTM with a new data set and the model provides an around 80% accuracy.**

Keywords-Machine Learning, Deep Learning, Electric load, Forecasting, LSTM, ARIMA, Data Preprocessing;

I. Introduction

Over the last few years, an increase of the global average temperature on earth has been observed mainly due to a continuous raise of greenhouse gas emissions. To stem this increase, many solutions have been proposed to reduce environment impacts by organizations and states. A sector where considerable progress could be made in the energy management concerns the residential sector. According to [1], residential sector is indeed one of the main consumers of energy and accounts for instance in 35% of the total French energy consumption in 2016. In order to improve household energy efficiency, it is therefore essential to have a system able to predict with accuracy its future energy consumption. Predictions are indeed useful to determine the adequate sizing of solar panels and battery for improving the self-consumption, thus reducing power flow in the grid. Energy load forecast are more and more based on algorithms requiring large scale energy consumption data.

Every day, new smart plugs appear on the market. These smart plugs have made energy consumption data available [2], making data statistical modeling possible [4]. To collect data in a house or more generally within a building , two types of approach can be distinguished [3]: Non-intrusive and Intrusive. A Non-intrusive approach is based on a single sensor (one smart meter) placed at the household electricity's point. An Intrusive approach rather uses multiple sensors deployed in the home. Sensors are indeed plugged into the different appliances of the household. According to the prediction period, three different classes of energy load forecasting can be considered: (1) Short-term-load when prediction varies from 1 hour to 1-week, (2) Medium-term-load, forecasting ranging from 1 week to 1 year, and finally (3) Long-term load when the model makes predictions longer than one year [5]. Whatever the category, energy load prediction has been shown to be a complex matter. Moreover, load prediction at individual household level has shown to be even more difficult than global load forecasting[6]. In this work, the aim is to compare the performance of linear (AutoRegressive Integrated Moving Average (ARIMA)) and non-linear (Long Short-Term Memory (LSTM)) predictors for energy load forecast on short and medium term with actual dataset IHEPCD [1] [25]. In [7]–[10], the authors predicted the energy load in short interval without preprocessing the original dataset. However, a dataset may contain some missing values due to system or network connection failures. The lack of this information may reduce the accuracy of the prediction model. In this paper, we investigate the impact of different data preprocessing techniques to fill missing values. Our prediction was first only based on active power data collected within a household during a period of 47 months and a time resolution of one minute. Prediction is done either using a deep learning algorithm (LSTM) or a stochastic process (ARIMA). Both approaches are explored for performing energy load forecasting and tested on a dataset containing electricity consumption data for a single household with one-minute time resolution. To validate our training model, another data set collected in a household with a 5-second resolution is used.

The rest of this paper is organized as follows. Section II gives an overview of related works. Section III describes our methodology for energy load forecasting in terms of data preprocessing, ARIMA and LSTM models, as well as performance metrics. Section IV describes our experiments and results obtained using ARIMA and LSTM predictors. Section V presents concluding remarks and future works.

[1] Individual Household Electric Power Consumption Dataset

II. RELATED WORKS

Over the last few years, energy load forecasts have received an increasing attention from researchers. A lot of studies have used forecasting models with time series dataset such as electrical power consumption. In [11], the authors studied a load forecasting model based on a one step ahead forecast for monthly electric energy consumption in Lebanon. Two approaches, ARIMA and AR (1), are used with a high pass filter. The best accuracy is obtained using AR (1) high pass filter. In [12], the authors studied the issue of the sustained growth of household energy consumption in China from 1980 to 2009. ARIMA and BVAR (Bayesian vector autoregression) were used as forecasting models and showed that both methodologies are appropriate to predict the sustained growth of household energy consumption (HEC) trends.

Recently, new algorithms based on deep learning have been proposed to cope with the challenges related to the load forecasting models. Despite deep learning is quite a new approach to address energy load prediction problems, this kind of methods has gained popularity among private companies as well as academics over the last few years [24]. In [4], Artificial neural network (ANN) ensembles were used to perform energy load forecasting. In [13]–[17], the authors have explored in details ANNs for short, medium and long term periods of load forecasting. In [18], support vector machines (SVM) coupled with empirical mode decomposition were used to perform long term energy load forecasting. In [19], the authors proposed kernel based multi-task learning approaches to predict electricity demand. In [20], Deep Belief Networks were used to perform short term electricity load forecasting on a Macedonian hourly electricity consumption dataset. In [7]–[10], the authors predicted electrical consumption using a forecasting model based on LSTM. However, their predictions were performed without any preprocessing of the raw data, thus including some missing values in the LSTM forecasting model [21]. In [27]- [28], authors provided a comparison between ARIMA and LSTM for sales forecasting in retail, as well as for financial market price prediction. Their experiments showed that LSTM provides a better accuracy than ARIMA. In [23], authors studied the most suitable forecasting period for electrical load. Two stochastic approaches, ARMA and ARIMA, using a preprocessing technique based on previous value to fill missing samples were compared. As energy load exhibits both linear and nonlinear patterns, we propose a system which can predict a day ahead power consumption based on LSTM and ARIMA models. The choice of these models was motivated by the following reasons. LSTM is able to identify structures and patterns of data such as non-linearity and complexity in time series. On the other hand, ARIMA is known to perform well on linear time series data and stationary data. Finally, we investigate three different techniques to fill missing values from the original dataset: previous, mean and same time a day ago or next (STDAN) values.

III. ENERGY LOAD PREDICTION METHODOLOGY

In this work, our main objective is to compare the performance of linear and non-linear models (ARIMA and LSTM respectively) for energy load forecasting. The main issues addressed in this work are two-fold: Which algorithm has the best prediction accuracy for time series data? Which data preprocessing technique to fill missing values is better for energy load forecasting?

A. Data set and Data preprocessing

For collecting information in household, an architecture based on smart sensors (or smart plugs) is required. Through a wireless communication, sensing information are sent to a gateway and can be then visualized on a dashboard. As it is illustrated on Figure 3, we developed an architecture to collect data in a household using different smart plugs such as Fibaro Plug [30], Coco plug [31], Energy meter gen5 [32] and Multisensor6 [33]. This architecture is built around a Raspberry Pi3 computer [34] and a home automation system called Domoticz [35]. This web frontend is interesting since various sensor devices can be monitored and configured very easily. To visualize sensed data on a dashboard, we used an open source analytics and monitoring system called Grafana [36]. This sensing prototype has been deployed in a house located in the south of France (Nice), to get power measurements for different appliances as well as for the overall consumption. Unfortunately, measures performed so far are still not large enough to construct a relevant dataset and apply our forecasting models. We therefore decided to use a freely available dataset of electric power consumption [25].

This dataset is the result of acquisitions performed within a French household between December 2006 and November 2010 (47 months), using a one minute sampling rate. The dataset contains 2,075,259 measures of a household's electrical energy consumption. This data set contains the following 9 household attributes: date, time, global active power, global reactive power, voltage, global current, energy sub-metering 1, energy sub-metering 2, energy sub-metering 3. Power, voltage and current measures are averaged over one minute. Energies sub-metering's have been used in this house to detect groups of appliances that consume more energy than others.

As mentioned in [25], all appliances in this household are considered to be supplied by the electrical network. In this dataset, appliances include a dishwasher, an oven, a microwave (hot plates are not electric but gas power), a washing machine, a tumble dryer, a refrigerator, a light, a water heater and an air conditioner. Some forecasting studies have been already performed in the past using this dataset [29]. In [29], the authors compared CNN, ANN, SVM and LSTM models using power consumptions samples with one-minute sampling rate. Their experiments showed that LSTM provides a better accuracy than others. It is worth noticing that in this work, we first only consider active power as feature set. Unfortunately, this dataset contains around 1.25% missing values (a "?" character in the csv file) and is therefore not usable as this for a prediction model. The lack of these information (by replacing missing values by zeros) may indeed reduce the predictive efficiency of the forecasting model as shown in Figure 4. To cope with this issue, we investigated three techniques to replace missing values using 1) the previous sample, 2) the mean calculated over all the samples, or 3) the sample at the same time a day ago or next. Both previous and mean values are already implemented in panda's python library. The proposed STDAN (Same Time a Days Ago or Next) algorithm in python is depicted on Figure 1. The STDAN approach is dedicated for time series data. As shown in Figure 1, STDAN consists in detecting a NaN (not a number) and then replacing it by the value at the same time a day ago if the operation (row – one_day) returns a positive value, or the next day at the same time if the operation returns a negative value. To assess the benefit of this algorithm, we randomly generated 5% to 90% missing values from the original dataset. As we are interested in predicting a daily total power consumption, we aggregate the minute-by-minute dataset into daily observations by using Eq. (1).

$$P_{daily} = \sum_{i=1}^{N=24*60} P_i \qquad (1)$$

Where P_{daily} represents the daily total power consumed, P_i the instantaneous power consumed every minute and N the number of samples per day. We got therefore a new dataset with 1442 samples. In the next section, we describe the different models used in this work.

```
def fill_missing_nouveau_apreschangement (values,n_samples_in_day):
    one_day = n_samples_of_day  = 24*60
    for row in range (values.shape[0]):
        for col in range(values.shape[1]):
            if np.isnan (values[row,col]):
                if (row-one_day) < 0:
                    i=0
                    if isfinite (values[row + one_day + i, col]):
                        values[row, col] = values[row + one_day + i,col]
                    else:
                        while isnan (values[row + one_day + i,col]):
                            i=i + one_day
                            values[row,col] = values[row + one_day + i,col]
                            if isfinite (values[row,col]):
                                break
                else:
                    values[row,col] = values[row-one_day,col]
```

Figure 1: Algorithm proposal STDAN

B. ARIMA

ARIMA is a time series prediction model initially proposed by Box and Jenkins [22] to address the limitations related to ARMA. ARMA is only used for so-called stationary time series. A time series is stationary if the mean and variance are almost constant over the time. ARIMA is a generalization of the auto-regressive moving average model to which a differencing process to make time series stationary is added. This differencing process is based on the difference between consecutive observations. An ARIMA model captures the following key elements of the model:

- *AR*: Autoregression. It is a regression model that exploits the dependencies between an observation and a number of lagged observations (p).
- *I*: Integrated. This parameter is used to make the time series stationary. To do so, the differences of observations are measured at different times (d).
- *MA*: Moving Average. It is an approach able to consider the dependency between observed samples and the residual error terms when a moving average model is used with a number of lagged observations (q).

An AR model can be written as a linear regression given by [27]:

$$x_t = b + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t \qquad (2)$$

Where x_t is the stationary variable at time t, b is a constant, and $\emptyset_i$ are the autocorrelation coefficients to be estimated from lags 1 to p. Finally, ε_t are the residuals.
An MA model of order q, i.e MA(q), can be written in the form:

$$x_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \qquad (3)$$

Where μ is the expected x_t , and θ_i the coefficients to be estimated (with $\theta_0 = 1$) . We assume that ε_t is a Gaussian white noise series with mean zero and variance σ_ε^2 . The ARIMA model of order (p,0,q) can be obtained by combining AR et MA models.

$$x_t = b + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \qquad (4)$$

In order to explore different combinations of parameters p (from 0 to 10), d (from 0 to 2) and q (from 0 to 5), we have used a grid search (an exhaustive searching algorithm with a manual specified subset for hyperparameters optimization). For each combination of parameters, we fit and evaluate the ARIMA model with mean squared error. From this grid search method, the optimized p, d and q values were identified as follows: p=1, d=0 and q=2.

C. LSTM

LSTM models have been observed as a solution to overcome the problem of time series prediction due to their capacity of remembering patterns for short and long-term periods of time [24]. A recurrent LSTM network is composed of different memory blocks called cells. A LSTM model can modify, remove or add information as it crosses different layers as shown in Figure 2. Information flows through a mechanism known as cell states which allows memorizing or forgetting things in a selective way. At a particular cell state, the information has three different dependencies, called respectively the previous cell state c_{t-1}, the previous hidden state h_{t-1}, and the input at the current time step x_t. Three gates called forget, input and output, are responsible of saving things and manipulations of a cell. To optimize the performance of the LSTM network, a forget gate (f_t) decides which information needs to be thrown away using a sigmoid layer by looking at h_{t-1} and x_t, and outputs a number between zero and one for each number in the cell state c_{t-1}. A value of 1 means "completely keep this", while a 0 indicates "completely get rid of this".

$$f_t = \sigma\left(W_f.\left[h_{t-1}, x_t\right] + b_f\right) \quad (5)$$

The memory gate $\tilde{c}_t$ chooses which new information needs to be stored in the cell state. First, a sigmoid layer called the "input gate layer" chooses which values will be updated. Then, a tanh layer creates a vector of new candidate values, $\tilde{c}_t$, that could be added to the state. Then, it combines these two to create an update of the state.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (6)$$

$$\tilde{c}_t = tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (7)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (8)$$

The output gate (o_t) decides what will be yield out of each cell. The obtained values are based on the cell state along with the filtered and newly added data. Then, the cell state is filtered through tanh (to get the values between −1 and 1) and multiply it by the output of the sigmoid gate, so that only the chosen parts are sent to the output layer.

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (9)$$

$$h_t = o_t * \tanh(c_t) \quad (10)$$

Where W_f, W_c, W_i, and W_o represent rectangular weight matrices, b_f, b_c, b_i *and* b_o are bias vectors and σ is the logistic sigmoid.

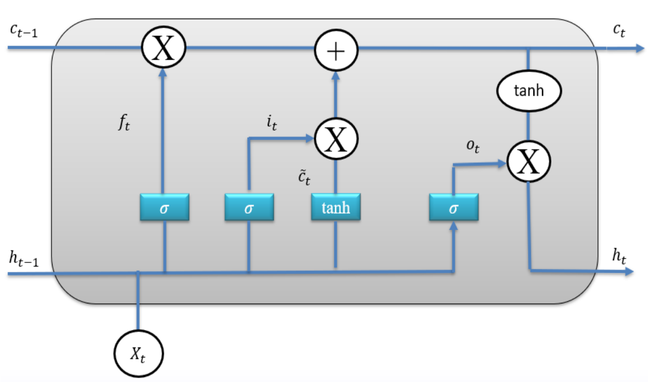

Figure 2: The internal structure of an LSTM

A standard LSTM has been designed for our propose. The network structure consists of 2 hidden layers LSTM with 50 memory units, one fully (dense) connected layer of 25 neurons with a linear activation function and an output layer that makes a single value prediction. The choice of the network structure has been decided following a trial and error approach. The network also uses the Mean Squared Error as a loss function and the ADAM [26] algorithm as an optimizer. Networks were fitted with 70 epochs and a batch size of 16.

D. Performance metrics

To evaluate the performance of LSTM and ARIMA models, two different evaluation metrics were used: RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Errors). RMSE and MAPE are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\boldsymbol{y_j} - \boldsymbol{\hat{y}_j}\right)^2} \quad (\mathbf{11})$$

$$\text{MAPE} = \frac{\mathbf{100\%}}{\boldsymbol{N}}\sum_{j=1}^{N}\left|\frac{y_j - \hat{y}_j}{y_j}\right| \quad (\mathbf{12})$$

Where N is the total number of observations, $\boldsymbol{y_j}$ is the actual value; and $\boldsymbol{\hat{y}_j}$ is the predicted value. The RMSE metric is used to measure, over a forecasting period, the global error between the actual energy consumed and the corresponding energy estimated by forecasting models. It is a quadratic scoring rule that also measures the average magnitude of the error. MAPE is a metric used to measure prediction accuracy of a forecasting.

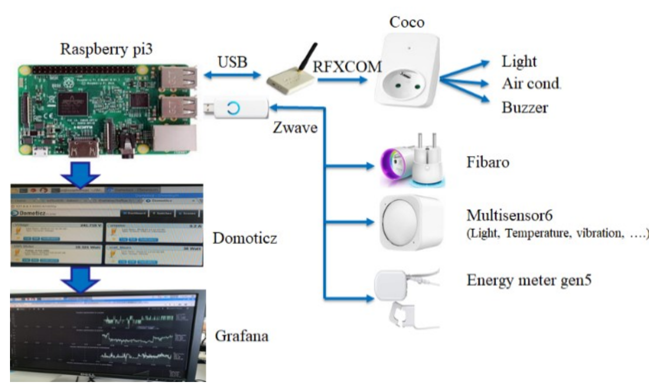

Figure 3: Prototype for collecting data and making decision in smart home.

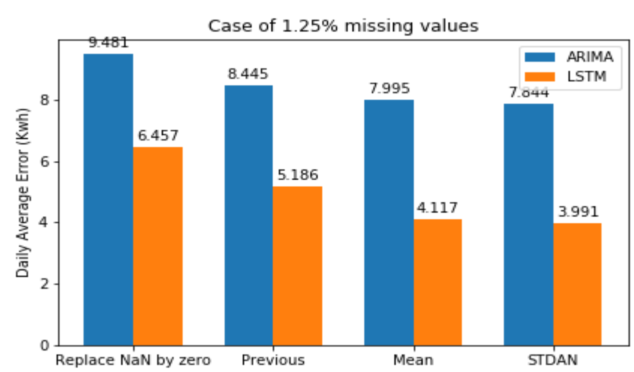

Figure 4: Comparison of LSTM and ARIMA for different preprocessing

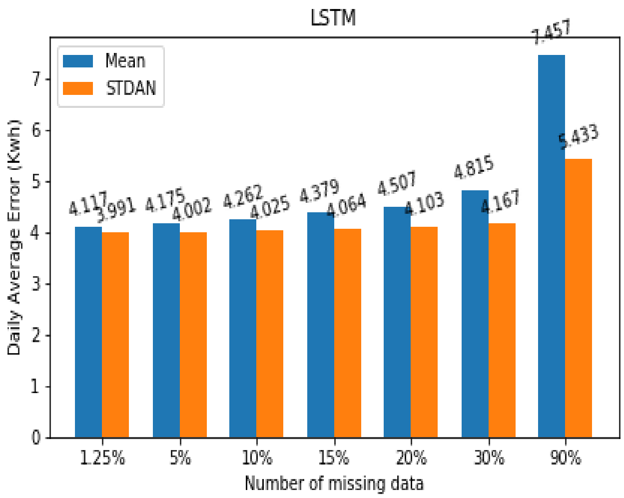

Figure 5: Comparison for different missing values percentage

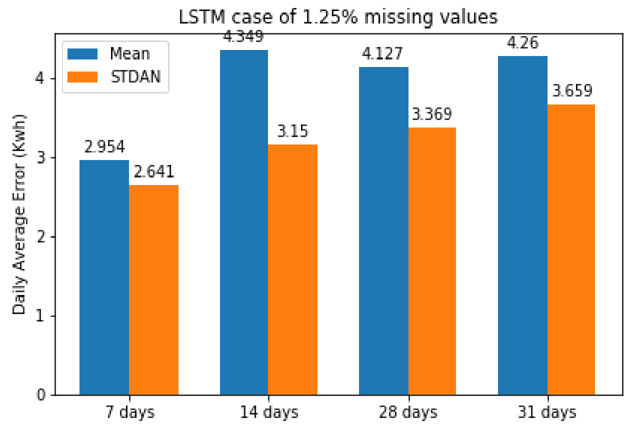

Figure 6: LSTM for short- and medium-term load forecasting

IV. Experiment and Results

A. LSTM and ARIMA prediction models comparison

To analyze the performance of ARIMA and LSTM models, experiments have been conducted using the IHEPCD dataset [25]. The samples were grouped to obtain daily total power consumption and then split into two groups. The first group was considered as the training set: it contains data from 2006-12-16 to 2009-12-31 and represents 78% of the dataset. The second group was used for test and contains data from 2010-01-01 to 2010-11-26, representing 22% of the dataset. ARIMA and LSTM models were implemented using the python ecosystem. Statsmodel library was used to fit the ARIMA model by calling the *ARIMA* function along with the p, d and q parameters. Then *fit* and *predict* functions were called to train the model and make predictions respectively. SciPy environment with Keras deep learning library using the TensorFlow backend was used for the LSTM model. Our objective is to perform a one-day ahead forecast using ARIMA and LSTM models. The RMSE and MAPE metrics are used to evaluate both models using the three preprocessing techniques presented in section III.A. As shown on Figure 4, LSTM provides a significant improvement in accuracy (RMSE) compared to ARIMA model whatever the preprocessing technique. The RMSE decreases from 7.844 to 3.991 kWh using STDAN as a preprocessing technique. For LSTM, the best performance is obtained when using our STDAN preprocessing approach. To further evaluate STDAN approach, we randomly generated 5% to 90% missing values within the test set. As it can be seen in Figure 5, and as expected, the daily average error increases with the number of missing values for both preprocessing techniques. However, STDAN outperforms the mean preprocessing technique.

We then tested the short (7 days) and medium (14 to 31 days) term load forecasting with LSTM. Obtained results are shown in Figure 6. As it can be seen, LSTM performs better with short load term forecasting than with medium term when using STDAN. This can be explained by the k-step ahead out samples forecasts which accumulate the error terms, thus resulting in lower accuracy in medium-term forecasting performance. Overall, we can conclude that LSTM performs better than ARIMA for energy load forecasting. Results also demonstrate that STDAN provides the best prediction accuracy.

Table I : Validation of new data set

Algorithm	Data set	Prediction period	RMSE (KWh)	MAPE (%)
LSTM Univariate	IHEPCD (Paris)	Test set = 322 days	3.99	13.29
	New dataset (Nice)		5.99	20.94

B. LSTM prediction model validation on new dataset

In order to validate our LSTM prediction model based on the IHEPCD dataset, where samples were collected in areas close to Paris, we also performed experiments using another dataset from a household close to Nice in the south of France. This new dataset contains only active power measurements with a 5-second sampling rate. To get daily observations, samples are aggregated using the Eq. (1) with N=17280 (i.e. 12*60*24). As this new dataset contained missing values as well, all missing values were replaced using our STDAN approach. As it can be seen on Table I, our model exhibits around 80% accuracy with this new household.

V. Conclusion

In this paper, our objective was to perform a one-day ahead forecast of household energy load using ARIMA and LSTM models. To do so, we first presented different techniques to replace missing values from the original dataset. We also evaluated the accuracy of ARIMA and LSTM, as representative models when forecasting time series data. Both models were trained using a set of residential power consumption data. Obtained results showed that LSTM outperforms ARIMA whatever the preprocessing technique. To validate the effectiveness of the LSTM model, predictions were conducted using a new data test from a household located far away from the one used during the learning phase. Using this new dataset, the prediction error is around 20%, which can be considered as acceptable. In future works, we plan to study other deep learning algorithms as well as other regularization approaches to improve the generalization of our models.

REFERENCES

[1] Florian, "Consommation Électrique 2018 en France: Statistiques et Analyses," *Prix-elec by Selectra*, 27-Aug-2018. [Online]. Available: https://prix-elec.com/energie/comprendre/statistiques-consommation-france.

[2] M. Manic, K. Amarasinghe, J. J. Rodriguez-Andina, and C. Rieger, «Intelligent Buildings of the Future: Cyberaware, Deep Learning Powered, and Human Interacting," *IEEE Industrial Electronics Magazine*, vol. 10, no. 4, pp. 32–49, Dec. 2016.

[3] A. Ridi, C. Gisler, and J. Hennebert, "A Survey on Intrusive Load Monitoring for Appliance Recognition," 2014, pp. 3702–3707.

[4] J. G. Jetcheva, M. Majidpour, and W.-P. Chen, "Neural network model ensembles for building-level electricity load forecasts," *Energy and Buildings*, vol. 84, pp. 214–223, Dec. 2014

[5] Vahid Mansouri and Mohammad E. Akbari, Efficient Short-Term Electricity Load Forecasting Using Recurrent Neural Networks, Journal of Artificial Intelligence in Electrical Engineering, Vol. 3, No. 9, June 2014

[6] Elena Mocanu, Phuong H. Nguyen, Madeleine Gibescu, Wil L. Kling, Deep learning for estimating building energy consumption, Sustainable Energy, Grids and Networks, Volume 6, June 2016, Pages 91-99

[7] N. Thokala, A. Bapna, M. Chandra, "A Deployable Electrical Load Forecasting Solution for Commercial Buildings," IEEE International Conference on Industrical Technology (ICIT), Lyon, France, Feb. 2018, pp. 1101-1106

[8] H. Zheng, J. Yuan and L. Chen, "Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation", Energies, vol. 10, no. 8, p. 1168, 2017

[9] B. Stephen, X. Tang, P. Harvey, S. Galloway and K. Jennett, "Incorporating Practice Theory in Sub-Profile Models for Short Term Aggregated Residential Load Forecasting", IEEE Transactions on Smart Grid, vol. 8, no. 4, pp. 1591-1598, 2017

[10] A. Marinescu, C. Harris, I. Dusparic, S. Clarke, and V. Cahill, "Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods," Proc. 2nd Int. Workshop Softw. Eng. Challenges Smart Grid (SE4SG), San Francisco, CA, USA, May 2013, pp. 25–32

[11] Samer Saab, Elie Badr and George Nasr, "Univariate modeling and forecasting of energy consumption: the case of electricity in Lebanon, «Energy, vol.26, 2001, pp. 1- 14

[12] Qing Zhu, Yujing Guo, Genfu Feng, "Household energy consumption in China forecasting with BVAR model up to 2015,"2012 Fifth International Joint Conference on Computational Sciences and Optimization, 2012

[13] C. Roldán-Blay, G. Escrivá-Escrivá, C. Álvarez-Bel, C. Roldán-Porta, and J. Rodríguez-García, "Upgrade of an artificial neural network prediction method for electrical consumption forecasting using an hourly temperature curve model," *Energy and Buildings*, vol. 60, pp. 38–46, May 2013

[14] M. Q. Raza and Z. Baharudin, "A review on short term load forecasting using hybrid neural network techniques," in *2012 IEEE International Conference on Power and Energy (PECon)*, 2012, pp. 846–851

[15] M. D. Felice and X. Yao, "Short-Term Load Forecasting with Neural Network Ensembles: A Comparative Study [Application Notes]," *IEEE Computational Intelligence Magazine*, vol. 6, no. 3, pp. 47–56, Aug. 2011

[16] R. Kumar, R. K. Aggarwal, and J. D. Sharma, "Energy analysis of a building using artificial neural network: A review," *Energy and Buildings*, vol. 65, pp. 352–358, Oct. 2013.

[17] S. M. Sulaiman, P. A. Jeyanthy, and D. Devaraj, "Artificial neural network-based day ahead load forecasting using Smart Meter data," in *2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE)*, 2016, pp. 1–6.

[18] L. Ghelardoni, A. Ghio, and D. Anguita, "Energy Load Forecasting Using Empirical Mode Decomposition and Support Vector Regression," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 549–556, Mar. 2013

[19] J. B. Fiot and F. Dinuzzo, "Electricity Demand Forecasting by MultiTask Learning," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp.1–1, 2016.

[20] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network-based electricity load forecasting: An analysis of Macedonian case," *Energy*, vol. 115, Part 3, pp. 1688–1700, Nov. 2016

[21] N. Kim, M. Kim, and J. Choi, "LSTM Based Short-term Electricity Consumption Forecast with Daily Load Profile Sequences," 2018, pp. 136–137

[22] G. Box, G. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day, 1970

[23] P. Chujai, N. Kerdprasop, and K. Kerdprasop, "Time Series Analysis of Household Electric Consumption with ARIMA and ARMA Models," *Hong Kong*, p. 6, 2013.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997

[25] Hebrail G. UCI machine learning repository: individual household electric power consumption data set. Tech rep, University of California, Irvine, School of Information and Computer Sciences, 00002; 2012. <Https://archive.ics.uci.edu/>

[26] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, Dec. 2014.

[27] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "Forecasting Economic and Financial Time Series: ARIMA vs LSTMA Comparison of ARIMA and LSTM," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, 2018, pp. 1394–1401.

[28] A. Elmasdotter and C. Nyströmer, 'A comparative study between LSTM and ARIMA for sales forecasting in retail', Dissertation, 2018.

[29] K. Amarasinghe, D. L. Marino and M. Manic, "Deep neural networks for energy load forecasting - *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, Edinburgh, 2017, pp. 1483-1488.

[30] F. D. Department, "FIBARO | The smart outlet - Wall Plug," *fibaro.com*. [Online]. Available: https://www.fibaro.com/en/products/wall-plug/.

[31] "Prises télécommandées." [Online]. Available : https://chacon.com/fr/40-prises-telecommandees.

[32] "Z-Wave electricity meter • Aeotec." [Online]. Available: https://aeotec.com/z-wave-home-energy-measure/

[33] "MultiSensor 6 Z-Wave motion, light, temperature sensor." [Online]. Available: https://aeotec.com/z-wave-sensor/

[34] "Buy a Raspberry Pi 3 Model B – Raspberry Pi." [Online]. Available: https://www.raspberrypi.org.

[35] "Domoticz." [Online]. Available: https://www.domoticz.com/.

[36] "Grafana: The open observability platform," *Grafana Labs*. [Online]. Available: https://grafana.com/