An Efficient Data Model for Energy Prediction using Wireless Sensors

Michel Chammas[a], Abdallah Makhoul[b], Jacques Demerjian[c]

[a]*Digital Humanities Center, University of Balamand, El-Koura, Lebanon*
[b]*Femto-ST Institute, UMR CNRS 6174, Université de Bourgogne Franche-Comté, Besançon, France*
[c]*LARIFA-EDST, Faculty of Sciences, Lebanese University, Fanar, Lebanon*

Abstract

Energy prediction is in high importance for smart homes and smart cities, since it helps reduce power consumption and provides better energy and cost savings. Many algorithms have been used for predicting energy consumption using data collected from Internet of Things (IoT) devices and wireless sensors. In this paper, we propose a system based on Multilayer Perceptron (MLP) to predict energy consumption of a building using collected information (e.g., light energy, day of the week, humidity, temperature, etc.) from a Wireless Sensor Network (WSN). We compare our system against four other classification algorithms, namely: Linear Regression (LR), Support Vector Machine (SVM), Gradient Boosting Machine (GBM) and Random Forest (RF). We achieve state-of-the-art results with 64% of the coefficient of Determination R^2, 59.84% Root Mean Square Error (RMSE), 27.28% Mean Absolute Error (MAE) and 27.09% Mean Absolute Percentage Error (MAPE) in the testing set when using weather and temporal data.

Keywords: Energy Prediction, Multilayer Percetron (MLP), Data Mining,

Email addresses: michel.chammas@balamand.edu.lb (Michel Chammas), abdallah.makhoul@univ-fcomte.fr (Abdallah Makhoul), jacques.demerjian@ul.edu.lb (Jacques Demerjian)

Machine learning, classification algorithms

1. Introduction

Smart Home technologies are among the most emerging trends in Internet of Things (IoT) today. Latest studies show the intelligence of smart devices and sensor networks in improving the quality of people's life [1]. IoT devices are being used to collect data in order to analyze the behaviour and proper uses of energy. Whereas energy consumption has always been an important issue as the number of devices using electrical power are increasing. Since the mid 80's, researchers mentioned the importance of energy forecasting in the future of smart buildings as it plays an essential role in energy and cost saving. Whereas buildings represent around 40% of the world energy usage, they are considered as the most cost-effective areas to reduce energy consumption. [2]. Therefore, this issue created an important challenge for the researchers to analyze the energy consumption in order to reduce it. Many studies have concluded that weather is one of several conditions that could affect the energy usage in a building [3], and weather data performs an important role for energy prediction and performance assessment of smart buildings and urban environments [4].

Different models have been used to predict power consumption like Autoregressive Integrated Moving Average model (ARIMA) [5], simple and multiple Linear Regression, Neuro-Fuzzy model [6], Support Vector Regression, Support Vector Machines (SVM) [7], Artficial Neural Network (ANN) [8], Time-Series [9], or a combination of regression, Nearest Neighbor and ANN, whereas ANN has been considered to achieve the best results for energy prediction. Based on the statistics mentioned in [4], an overall of 47% of the energy consumption prediction models utilized ANN as machine learning algorithms, while 25% used SVM, 4% decision trees and 24% other statistical models. ANNs are the most used algorithms for energy consumption in buildings. In [10], it was used to predict energy consumption of a building simulation using EnergyPlus and presented a state-of-the-art result. They concluded that an ANN model with multiple outputs (larger number of neurons) has a better performance than an ANN with single output. Also a study in [11] mentioned the use of Long Short Term Memory (LSTM) with Auto-Encoders (AE) in predicting solar energy consumption using weather data and achieved best results over several Deep Learning (DL) models. Furthermore in [12], different approaches were suggested based on the energy forecasting time range, where they suggested Neural Network based Genetic Algorithm (NNGA) for short and middle term prediction (for a day or a month) and Neural Network based Particle Swarm Optimization (NNPSO) for long term prediction (for a year). Both approaches were compared with the Conventional Neural Network (CNN) and performed better results.

Many studies have compared the uses of classification and regression algorithms to Artificial Neural Network models in predicting energy consumption in buildings with varying results. In [13], ANN with two hidden layers and Back Propagation (BP) was compared to a decision tree algorithm (Random Forest (RF)). ANN performed better results than RF. The data used was a combination of weather (dew point temperature, outdoor air temperature, wind speed and humidity) and temporal information (month of the year, day of the week and hour of the day). In [14], ANN was tested along with SVM and times series algorithms. ANN and SVM showed adjacent results, but a better performance when combined together. While in [15], they reviewed different studies of hybrid approaches, where SVM, ANN and BPNN were combined with Swarm Intelligence (SI) method. SVM showed better performance than ANN.

On the other hand, some studies showed that deep learning methods have outperform the traditional machine learning techniques. In [16], Multilayer

Perceptron (MLP) was compared with Gaussian Processes, Support Vector Machines, Ensemble Boosting, Linear Regression and Regression Trees for predicting aggregated energy demand and showed a state of art results. While in [17], MLP showed almost the worst performance in comparing with 7 different algorithms (ARIMA, MLR, SVR, BT, RF, KNN and MARS) to predict next day energy consumption, where SVR and RF showed the best results. In [18] different deep learning methods (DBN, MLP, ANN-NAR and ANN-NARX) were compared against traditional statistical learning models (SVM, HMM and FHMM) for building energy prediction, where DBM showed a robust performance in different scenarios followed by MLP.

In this paper, we have used the same dataset of [3], which contains data for around 4.5 months. In their study [3], they compared four different models: Linear Regression (LR), decision tree models: Gradient Boosting Method (GBM), RF and SVM, where GBM presented the best results. Our objective was to create an efficient data model based on MLP and compare it with the results of the four classification models, to achieve better performance, and to reach the optimum configuration for our model.

The rest of this paper is organized as follow: Section 2 describes our developed system, how we configured it and its functionality, Section 3 presents the dataset and types of information used in our experiment, Section 4 discusses the experiment and results, and Section 5 concludes the paper.

2. Dataset

The used dataset contains 35 different variables of weather information (temperature, humidity, pressure, wind speed, visibility, dew point), appliances and light energy consumption, and temporal data (Table 1). It was collected from an indoor and outdoor sensors network of a two storey building and a nearby airport. The building contains 10 temperature sensors (9 indoor and 1 outdoor) and 8 humidity sensors (7 indoor and 1 outdoor) (Figures 1 and 2). The weather station provides: temperature, humidity, visibility, pressure, wind speed and dew point temperature. The data was recorded every 10 minutes for 137 days, including the light energy, everyday consumption and the number of seconds from midnight.

We implemented the same setup as in [3], the data was split using CARET algorithm, 75% for training and 25% for testing. CARET creates a balanced split using a maximum dissimilarity approach [19]. The features were scaled between 0 and 1 with min-max normalization $x' = \frac{x - \min x}{\max x - \min x}$, to ease the network training. Also in [3] they used BORUTA package [20] to identify the most important features, which defined the Number of Seconds from Midnight (NSM) and light energy as the most important variables in predicting the consumption. The dataset was tested on several scenarios by using all the features or by omitting some of them.

Table 1: Dataset variables

T1 Kitchen Temperature	RH1 Kitchen Humidity
T2 Living room Temperature	RH2 Living room Humidity
T3 Laundry room Temperature	RH3 Laundry room Temperature
T4 Office room Temperature	RH4 Office room Humidity
T5 Bathroom Temperature	RH5 Bathroom Humidity
T6 Temperature Outside	RH6 Humidity Outside
T7 Ironing room Temperature	RH7 Ironing room Humidity
T8 Teenager room Temperature	RH8 Teenager room Humidity
T9 Parents room Temperature	RH9 Parents room Humidity
To Weather Station Temperature	Weather Station Pressure
RHo Weather Station Humidity	Weather Station Windspeed
Weather Station Visibility	Weather Station Tdewpoint
Week status Weekday	Week status Weekend
Day of week Monday	Day of week Tuesday
Day of week Wednesday	Day of week Thursday
Day of week Friday	Day of week Saturday
Day of week Sunday	light energy
NSM Number of seconds from midnight	

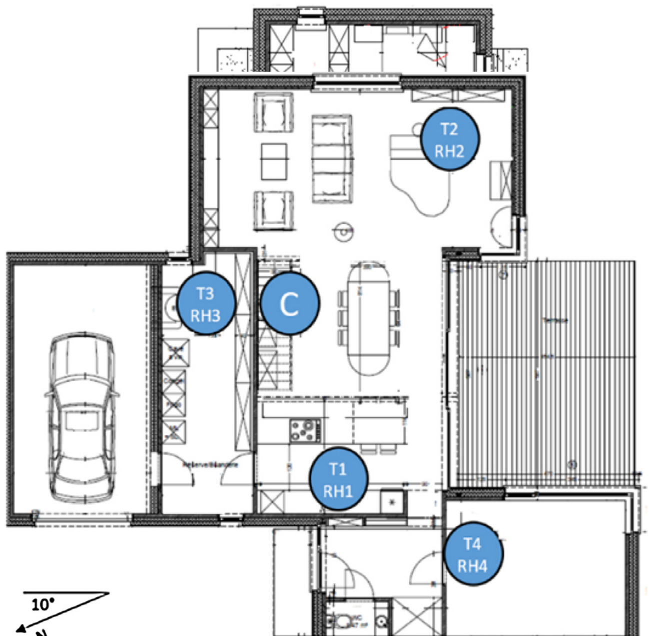

Figure 1: Building first floor [3]

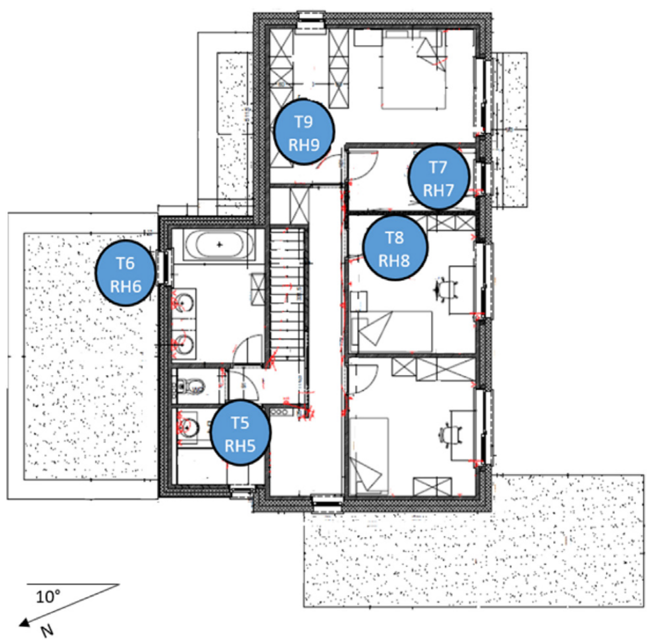

Figure 2: Building second floor [3]

3. System description

We model energy consumption with a multi-layer feed-forward neural network, where the feature vector at the input layer (Table 1) is mapped through a set of hidden layers to a single output representing the power consumption of the house (Figure 3). Each hidden layer in the network provides a specific representation of the input by building upon the output of the previous layer. With such level of abstraction, the network is able to build a robust representation of the input at high dimensions [21]. The connection between two adjacent layers in the network can be defined as:

$$y_j^n = \sum_i h_i^{n-1} w_{ij} + b_j^n$$
$$h_j^n = f(y_j^n)$$

where y_j is the output of neuron j at layer n, h_i^{n-1} the activation of neuron i at layer $n-1$, w_{ij} and b_j the weight and bias of the connection to neuron j from the previous layer, and f a nonlinear activation function.

3.1. Non-Linear activation function

We used the Rectified Linear Unit (ReLU) activation to achieve non-linearity; the function is used on the output of each neuron. It works by rectifying the input to 0 when $y <= 0$ or by preserving the signal when it is positive.

$$h_j^n = ReLU(y_j^n)$$
$$ReLU = f(y) = max(0, y)$$

The ReLU activation function is simple in terms of computation. It solves the vanishing gradient problem, and works better than sigmoid and other activation functions. ReLU is considered the most recommended activation function for feed forward neural networks, as it helps to generalize a variety of non-linear data [22].

3.2. Layer normalization

The distribution of the output activation at each layer is subject to high change during training; which is known as the internal covariate shift. This instability in the input may allow the network to get stuck in a saturated mode, which would lead to slow convergence. Layer normalization is used to reduce the covariate shift at each layer in the network by fixing the mean and variance of the input. More specifically, the input is normalized across all features, regardless of the batch size. The mean (μ) and variance (σ) across all hidden units at a given layer l are computed as follow:

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l$$
$$\sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^l - \mu^l)^2}$$

where H denotes the number of hidden units in a layer, and a_i^l is the activation unit i at a hidden layer l.

3.3. Weight initialization

The network weights allow for the preservation of a stable variance throughout the network layers. We initialized them as per [23] method, which improved the results and helped the network to converge faster. This method holds the signal from vanishing to zero or exploding to a high value. The weight matrices W_{ij} were initialized with a uniform distribution given as $W_{ij} \sim U(-\frac{\sqrt{6}}{n}, \frac{\sqrt{6}}{n})$, where n is the total number of input and output neurons at the layer (assuming all layers are of the same size).

3.4. Network training

We train the network via gradient descent with mean square error (MSE) as loss function, which is the squared difference between the predicted power consumption and the expected output. Adam optimizer is used with initial learning rate of 0.005 and batch size of 500. This algorithm is considered as an upgrade for RMSProp [24], which provides a bias correction and a momentum. Similar to Adadelta and RMSprop, it stores an exponentially decaying average of the past squared gradients and the past gradients (similar to momentum). It also offers a flexible learning rates for the stochastic gradient descent update, computed from the first and second moments of the gradients.

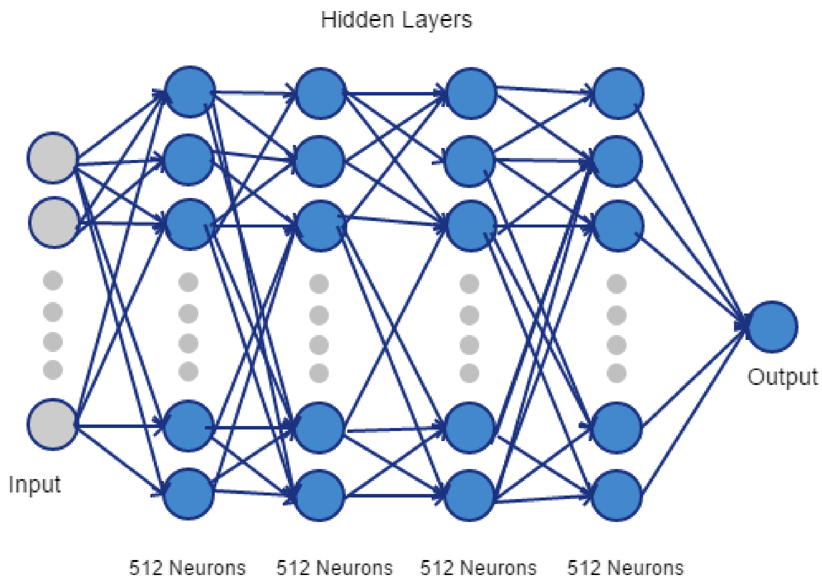

Figure 3: MLP with 4 hidden layers and back propagation

4. Experiments and Results

4.1. Evaluation metrics

We used four standard metrics to evaluate the performance of our system: Coefficient of determination R^2, Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Their formulas are shown in Table 2. The R^2 is the proportion of variance between the predicted and testing variable, whereas RMSE is the percentage of difference between them. MAE is the percentage of difference between the predicted variables, while MAPE express the accuracy.

Table 2: Model evaluation criteria

Coefficient of determination	R^2	$=$	$1 - \frac{\sum_{t=1}^{n}(Y_i - Y_i')^2}{\sum_{t=1}^{n}(Y_i - Y_i')^2}$
Root mean squared error	RMSE	$=$	$\sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_i - Y_i')^2}$
Mean absolute error	MAE	$=$	$\frac{1}{n}\sum_{t=1}^{n}(Y_i - Y_i')$
Mean absolute percentage error	MAPE	$=$	$\frac{1}{n}\sum_{t=1}^{n}\left\lvert\frac{(Y_i - Y_i')}{Y_i}\right\rvert$

4.2. Hyperparameters optimization

The selection of the number of layers and neurons in each layer can affect the model performance. To find the optimal parameters, we conducted a set of experiments with different configurations for the number of layers and neurons. In these experiments, all the features (weather, lights and temporal information) were used. To allow more generalization, we applied dropout [25] to the last hidden layer by randomly removing 50% of the neurons in that layer. However, we tested some models without dropout to access the improvements. Table 3 shows all the tested combinations, where L is the number of layers used in each model, N is the number of neurons per layer, DropOut shows the use of drop out at the last layer, Epochs is the number of complete passes through the data, and the four performance metrics represent the results. It is worth noting that layer normalization greatly improved network convergence. We noticed, for instance, a six fold increase in speed when training using layer normalization.

Our best system configuration is a MLP, with four hidden layers as illustrated in Figure 3. We used 512 neurons for each layer and we applied a dropout on the last layer. We also tried it without dropout and showed good results (a very slight difference), but we noticed a faster convergence when using dropout. The system needed a small number of Epochs to perform the best results, whereas, the second best configuration (3 hidden layers MLP without a dropout) used five times more epochs.

Table 3: Validating the best system configuration using all the data features

L	N	DropOut	Epochs	RMSE	R2	MAE	MAPE
1	128	Yes	19149	77.234	0.408	43.354	48.365
1	256	Yes	23425	74.285	0.450	41.870	46.829
1	512	Yes	11178	73.549	0.462	39.793	43.019
2	512	Yes	1940	69.415	0.526	34.570	35.714
3	256	Yes	24563	67.298	0.541	31.892	31.640
3	256	No	42610	68.889	0.534	32.293	31.999
3	512	Yes	4778	67.182	0.550	31.281	30.860
3	512	No	26748	66.245	0.567	30.599	29.160
4	**512**	**Yes**	**5401**	**66.295**	**0.567**	**29.556**	**27.961**
4	512	No	14549	66.743	0.562	30.094	28.550
5	512	Yes	3280	66.881	0.556	29.250	27.269
5	512	No	12560	66.974	0.558	29.688	28.392

4.3. Comparative results

After choosing the best configuration of layers and neurons, we tested our most performing MLP model with four different scenarios: 1) weather-only data which includes all the information from sensors inside the house and nearby airport station, 2) weather and lights information, 3) weather and temporal information (days of the week and Number of Seconds from Midnight (NSM)), and 4) all features data which includes weather, lights and temporal information. MLP yielded the best results with weather data and temporal information.

In [3], four algorithms were used to predict the power consumption: LR, RF, SVM, and GBM. The best model GBM was able to predict 57% of the variance

R^2 with 66.65% RMSE, 35.22% MAE and 38.29% MAPE when using all the features. GBM performed its best with no lights and achieved slightly better results 58% R^2, 66.21% RMSE, 35.24% MAE and 38.65% MAPE, where the number of seconds till mid-night was considered the best feature to predict the consumption for all the algorithms.

In our experiments, MLP showed considerably better error rate 61.75% RMSE, an accuracy R^2 61%, 28.52% MAE and 28.34% MAPE using only weather data. The importance of this result is that we omitted the most important feature (NSM) used by GBM in [3]. Our MLP achieved the best results at around 7k epochs. Table 4 shows the training (Trn) and Validation (Val) scores for the best scenario (without light energy variable). Figures 4, 5, 6 and 7 show the performance with respect to the four standard metrics.

We also tested the second best performing MLP (three hidden layers 512 neurons per layer without dropout) and showed high performance in the best scenario (using weather and temporal data): 63% R^2, 60.03% RMSE, 27.45% MAE and 26.85% MAPE. Furthermore, we tried our best MLP configuration (four layers 512 neurons per layer) with a dropout on each layer to test its performance on same scenario (using weather and temporal data) and we got the following results: 57% R^2, 65.63% RMSE, 28.29% MAE and 25.51% MAPE.

Table 4: Training and validation scores for the best scenario

	Epoch	**RMSE**	**R2**	**MAE**	**MAPE**
Trn	7557	20.059	0.962	10.810	13.254
Val	7557	59.840	0.643	27.283	27.096

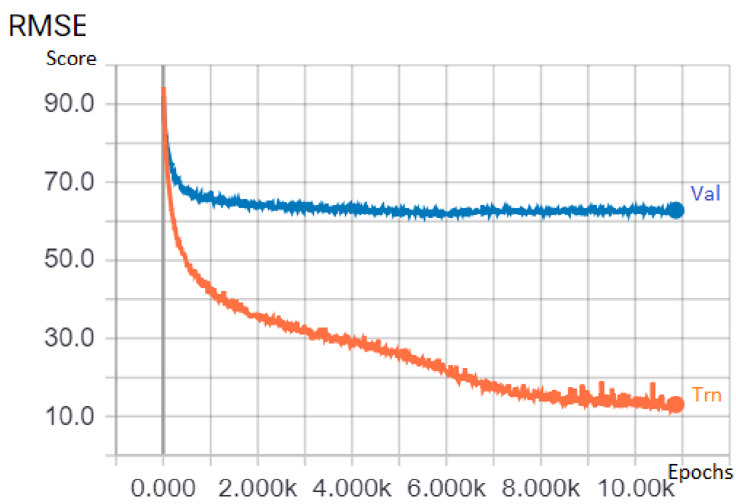

Figure 4: RMSE

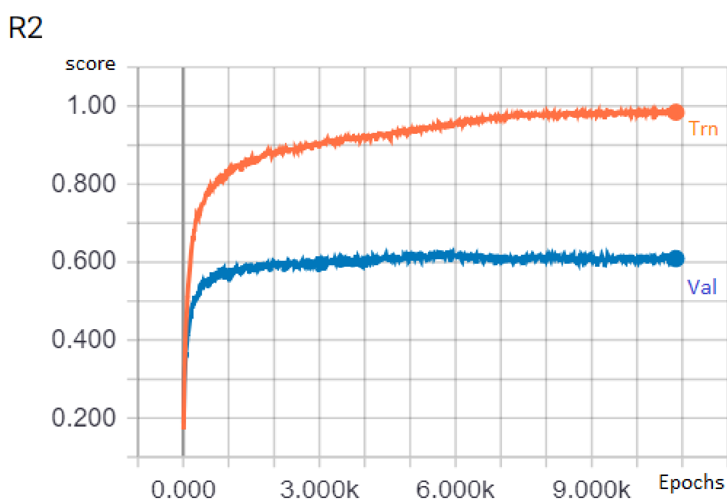

Figure 5: R^2

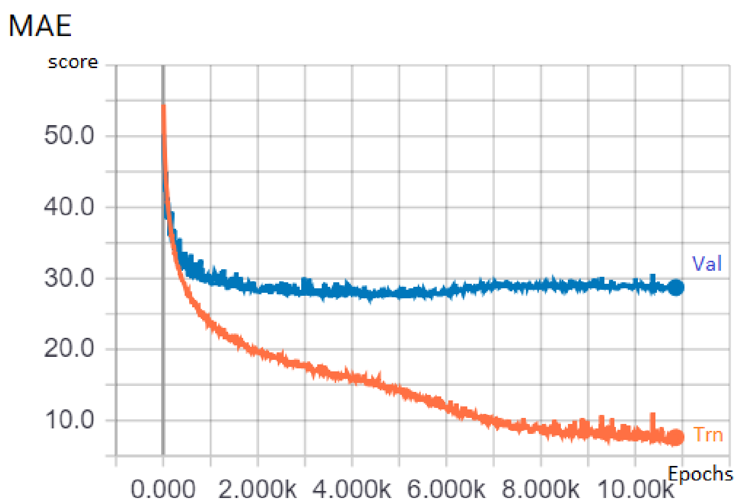

Figure 6: MAE

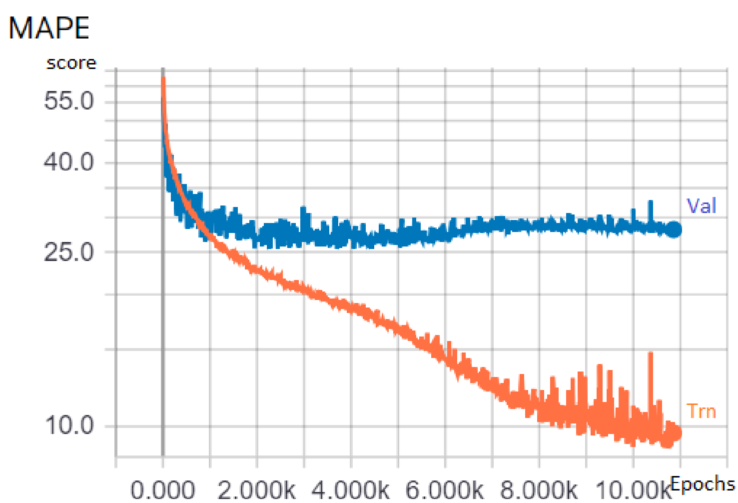

Figure 7: MAPE

The same experiment was repeated three times for MLP, first by adding the lights variable (55% R^2, 67.34% RMSE, 29.60% MAE and 28.03% MAPE), second by using all the features (56% R^2, 66.29% RMSE, 29.55% MAE and 27.96% MAPE), and third by using weather and temporal information only, without lights (64% R^2, 59.84% RMSE, 27.28% MAE and 27.09% MAPE). The last experiment showed the best results with a slight improvement of around 2% and yielded the best performance. Table 5 shows the performance of MLP, in the four different scenarios, in comparison with other models on the same dataset. We notice that adding the temporal information has improved the results by 1% in all the experiments, whereas MLP looks more sensitive than GBM to lights feature which degraded the performance by 6%.

Table 5: Models performance in the testing set

System	**RMSE**	$\boldsymbol{R^2}$	**MAE**	**MAPE %**
MLP weather only	61.75	0.61	28.52	28.34
MLP no date	67.34	0.55	29.60	28.03
MLP no lights	**59.84**	**0.64**	**27.28**	**27.09**
GBM no lights	66.21	0.58	35.24	38.65
MLP all features	66.29	0.56	29.55	27.96
GBM all features	66.65	0.57	35.22	38.29
RF all features	68.48	0.54	31.85	31.39
SVM all features	70.74	0.52	31.36	29.76
LR all features	93.18	0.16	51.97	59.93

5. Conclusions

One of the key factors for integrating systems and making homes and cities more intelligent is the energy savings that can be achieved. Therefore, energy prediction plays an important role in reducing power consumption. In this work, we propose to model a energy prediction system based on a Multilayer Perceptron neural network. Several data categories have been used, i.e.; light energy, temperature, humidity, day of the week, etc., collected from a Wireless Sensor Network installed in a two storey building. We compared the performance of our system against four classification algorithms, namely: Linear Regression, Support Vector Machine, Gradient Boosting Machine and Random Forest. We assessed the importance of temporal information and light energy as additional features for the prediction model. The temporal features slightly improved the performance for all systems, while the lights energy decreased the performance. For our method, we note a decrease by 10% in relative performance while the difference was negligible for the other classifiers (less than 1% for Gradient Boosting Machine). Our system outperformed the other four classifiers in all scenarios. Based on the conducted experiments, we conclude that weather information would be enough to predict energy consumption. This enables low-cost solutions for energy predictions. Further experiments should be performed on different datasets to validate our results.

References

[1] P. V. Paul, R. Saraswathi, The internet of things—a comprehensive survey, in: Computation of Power, Energy Information and Commuincation (ICCPEIC), 2017 International Conference on, IEEE, 2017, pp. 421–426.

[2] J. Lee, J. Kim, D. Song, J. Kim, C. Jang, Impact of external insulation and internal thermal density upon energy consumption of buildings in a temperate climate with four distinct seasons, Renewable and Sustainable Energy Reviews 75 (2017) 1081–1088.

[3] L. M. Candanedo, V. Feldheim, D. Deramaix, Data driven prediction models of energy use of appliances in a low-energy house, Energy and buildings 140 (2017) 81–97.

[4] K. Amasyali, N. M. El-Gohary, A review of data-driven building energy consumption prediction studies, Renewable and Sustainable Energy Reviews 81 (2018) 1192–1205.

[5] H. Kaur, S. Ahuja, Time series analysis and prediction of electricity consumption of health care institution using arima model, in: Proceedings of

Sixth International Conference on Soft Computing for Problem Solving, Springer, 2017, pp. 347–358.

[6] H. Pombeiro, R. Santos, P. Carreira, C. Silva, J. M. Sousa, Comparative assessment of low-complexity models to predict electricity consumption in an institutional building: Linear regression vs. fuzzy modeling vs. neural networks, Energy and Buildings 146 (2017) 141–151.

[7] S. Paudel, M. Elmitri, S. Couturier, P. H. Nguyen, R. Kamphuis, B. Lacarrière, O. Le Corre, A relevant data selection method for energy consumption prediction of low energy building based on support vector machine, Energy and Buildings 138 (2017) 240–256.

[8] F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, G. P. Vanoli, Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach, Energy 118 (2017) 999–1017.

[9] C. Deb, F. Zhang, J. Yang, S. E. Lee, K. W. Shah, A review on time series forecasting techniques for building energy consumption, Renewable and Sustainable Energy Reviews 74 (2017) 902–924.

[10] Q. Dong, K. Xing, H. Zhang, Artificial neural network for assessment of energy consumption and cost for cross laminated timber office building in severe cold regions, Sustainability 10 (1) (2017) 84.

[11] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for iot big data and streaming analytics: A survey, IEEE Communications Surveys Tutorials 20 (4) (2018) 2923–2960.

[12] K. Muralitharan, R. Sakthivel, R. Vishnuvarthan, Neural network based

optimization approach for energy demand prediction in smart grid, Neurocomputing 273 (2018) 199 – 208.

[13] M. W. Ahmad, M. Mourshed, Y. Rezgui, Trees vs neurons: Comparison between random forest and ann for high-resolution prediction of building energy consumption, Energy and Buildings 147 (2017) 77–89.

[14] A. Ahmad, M. Hassan, M. Abdullah, H. Rahman, F. Hussin, H. Abdullah, R. Saidur, A review on applications of ann and svm for building electrical energy consumption forecasting, Renewable and Sustainable Energy Reviews 33 (2014) 102–109.

[15] M. A. M. Daut, M. Y. Hassan, H. Abdullah, H. A. Rahman, M. P. Abdullah, F. Hussin, Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review, Renewable and Sustainable Energy Reviews 70 (2017) 1108–1118.

[16] N. G. Paterakis, E. Mocanu, M. Gibescu, B. Stappers, W. van Alst, Deep learning versus traditional machine learning methods for aggregated energy demand prediction, in: Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2017 IEEE PES, IEEE, 2017, pp. 1–6.

[17] C. Fan, F. Xiao, S. Wang, Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques, Applied Energy 127 (2014) 1–10.

[18] P. A. Mynhoff, E. Mocanu, M. Gibescu, Statistical learning versus deep learning: Performance comparison for building energy prediction methods, in: 8th IEEE PES Innovative Smart Grid Technologies Conference Europe, 2018, p. 1.

[19] P. Willett, Dissimilarity-based algorithms for selecting structurally diverse

sets of compounds, Journal of Computational Biology 6 (3-4) (1999) 447–457.

[20] M. B. Kursa, W. R. Rudnicki, et al., Feature selection with the boruta package, J Stat Softw 36 (11) (2010) 1–13.

[21] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.

[22] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognition 77 (2018) 354 – 377.

[23] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feed-forward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[24] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning 4 (2) (2012) 26–31.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

Michel Chammas received his M.S. degree in Computer Science - Networking and Communication from the University of Balamand, Lebanon, in 2009. Since 2010, he has been a Researcher at the Digital Humanities Center at the University of Balamand. His research interests include Data Mining, Machine Learning, Internet of Things, Digital Humanities and CyberSecurity.

Abdallah Makhoul received the M.S. degree in computer science from INSA

Lyon, France, in 2005, and the Ph.D. degree from the University of Franche-Comté, Belfort, France, in 2008. Since 2009, he has been an Associate Professor with the University of Franche-Comté. His research interests include Internet of Things, structural health monitoring, and real-time issues in wireless sensor networks.

Jacques Demerjian received his PhD degree in Network and Computer Science from TELECOM ParisTech (Ecole Nationale Supérieure des Télécommunications - Paris) in 2004. Dr. Demerjian is a Professor at the Faculty of Sciences at the Lebanese University. His main research interests include Body Sensor Network, Mobile Cloud Computing and Recommender Systems. He is an IEEE Senior Member.