The International Journal
ENTREPRENEURSHIP AND SUSTAINABILITY ISSUES
ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2017 Volume 4 Number 3 (March)

ENTREPRENEURSHIP AND SUSTAINABILITY ISSUES
ISSN 2345-0282 (online) http://jssidoi.org/jesi/

INNOVATIVE TIME SERIES FORECASTING: AUTO REGRESSIVE MOVING AVERAGE VS DEEP NETWORKS[1]

Anthony Mouraud

CEA Tech PdlL, 5 rue de l'Halbrane, 44340 Bouguenais, France

E-mails:anthony.mouraud@cea.fr;

Received 20 September 2016; accepted 29 November 2016

Abstract. Growing interest in meaningful indicators extraction from the huge amounts of data generated by energy efficient buildings instrumentations has led to focusing on so called smart analysis algorithms. This work proposes to focus on statistical and machine learning approaches that make use only of available data to learn relationships, correlations and dependencies between signals. In particular, time series forecasting is a key indication to anticipate, prevent and detect anomalies or unexpected behaviors.
We propose to compare performances of a classical Auto Regressive Moving Average (ARMA) approach to a Deep Highway Network on time serie forecasting only making use of past values of the serie. In recent years, Deep Learning has been extensively used for many classification or detection tasks. The complexity of such models is often an argument to discard such approaches for time serie prediction with regard to more common approaches performances. Here we give a first attempt to evaluate benefits of one of the most up to date Deep Learning model in the literature for time serie prediction.

Keywords: Sustainability, Buildings, Time Series Forecasting, Auto Regressive Moving Average (ARMA), Deep Networks

Reference to this paper should be made as follows: Mouraud, A. 2016. Innovative Time Series Forecasting: Auto Regressive Moving Average vs Deep Networks, *Entrepreneurship and Sustainability Issues*, *Entrepreneurship and Sustainability Issues* 4(3): 282-293. http://dx.doi.org/10.9770/jesi.2017.4.3S(4)

JEL Classifications: C45, C53

Additional disciplines: Mathematics, Data Science, Computer Science

[1] *This research was supported by the PERFORMER project. Project funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 609154.*

1. Introduction

The growing interest in building efficiency in terms of energy and comfort leads to critical needs in modelling tools to design them and efficient instrumentations to be aware of their real behaviour (Foucquier, Robert, Suard, Stephan, & Arnaud, 2013). Furthermore the huge amount of data generated during building lifetime hardens the extraction of useful information about real building behaviour, sensors health and to anticipate future needs.
The latter issue has seen lots of interesting work achieved in the last years. Several kinds of approaches are used to help in understanding data produced. Physical models (white boxes) are among the preferred methods due to their accuracy in describing the building studied. Another category builds analyses upon a mix between physical considerations and statistical methods (grey boxes).

In this work based on the PERFORMER project (FP7, N°609154), we focus on studies based on statistical and machine learning approaches (black boxes) that make use only of available data to learn relationships, correlations and dependencies between data. These studies often introduce exogenous information such as weather information to help models to converge to better predictions. The main drawbacks of such methods is to apply the suited pre-processing on data, find the best exogenous information to add to models inputs and make use of a sufficiently large and reliable dataset.

The present work proposes to use a well-known class of models in signal processing, namely: an ARMA model to generate predictions from the signal past values (Box, Jenkins, & Reinsel, 1994). In that way, signal is considered as a time series and the objective is to predict future values of this series without any other information as input, a so called 'data-driven' approach. An ARMA model makes no physical assumption about the underlying process but still has some assumption about the process itself. Aside from classical prediction models, recent years have seen Deep Learning (DL) models win lots of open contests on various application domains ranging from handwritten characters classification (LeCun, 2015) to breast cancer mitosis detection (Ciresan, Giusti, Gambardella, & Schmidhuber, 2013). Despite their performances on classification/selection, the use of DL models for time series prediction is mainly seen as an overshoot considering both performance of common models and complexity of DL models. In this work, our aim is to give an insight into the comparative performances of a recent model of Deep Neural Network (DNN) to the more common ARMA model for time series prediction.

This work both proposes to give useful forecasting information to time series data producers, further looking for anomalies comparing forecasting and actual values, and to give an insight of the ability of DNNs in time series prediction. This is part of a work that aims at generating online forecasting for each signal of instrumented building that upload their recordings to a database. This allow automatic forecasting production of several hundreds of signals without human interaction at each building. Each prediction model generates new forecasting at regular periods of time from the data uploaded by the building sensors. The provided forecasting for each individual signal is then used as a criterion for anomaly detection (not detailed here).
Furthermore, such massive forecasting can then be used as input to other indicators such as expert systems, for part of the control for local energy storage/consumption, or to allow anticipation with regard to cooling/heating materials inertia …

2. Data time series

As a part of the PERFORMER Eurpoean project this work benefits from the project's pilot sites buildings settings. The project is based on four pilot sites:

- *Baltic Plaza Hotel (PL)*
- *Las Letras Hotel (ES)*
- *Saint Teilo's High Scool (UK)*
- *Woopa office (FR)*

In the scope of the project are the instrumentations of each building. Buildings' sensors transmit their recordings to an online data warehouse that can be queried via a REST API. Datasets considered in this paper come from Woopa building (Lyon, France) and Saint Teilo's High Scool (Cardif, UK). This section presents some of the data used to achieve analyses presented in the following sections. Datasets are presented in two subsets each: a training set and a test set.

Saint Teilo High School

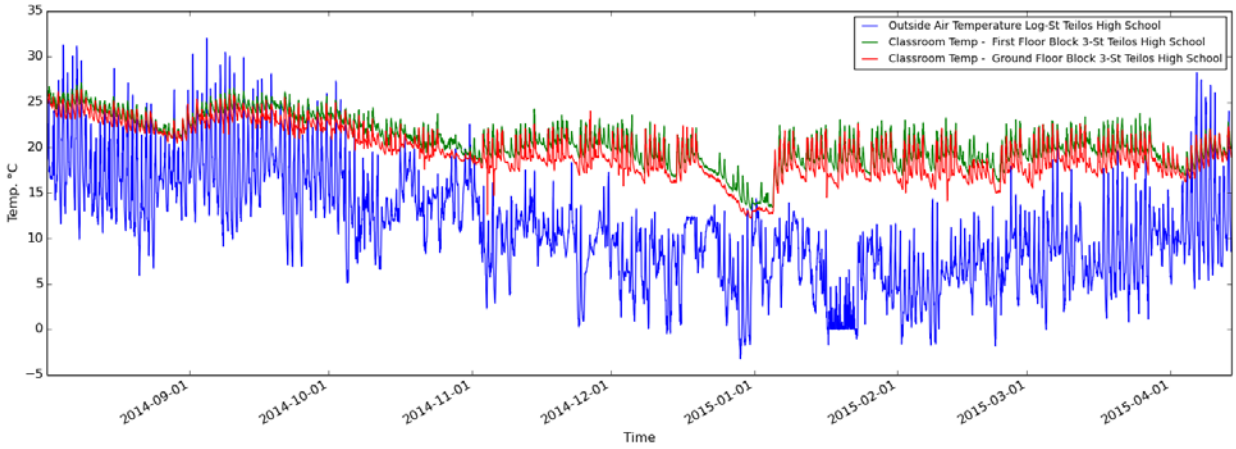

Fig. 1 Raw data hourly resampled

The Saint Teilo's high school data available for analysis are temperature values coming from a block of classrooms in the building. Figure 1 shows a resampled set of data. It consists in a first floor temperatures dataset and a ground floor temperatures dataset. The third set is the outside temperature recorded during the acquisition phase.
These data have been used in preliminary analyses (not shown) and have allowed to explore the periodicities and cluster properties of such building data and helped to set the parameters of the models detailed in the following sections.

These data have not been used thereafter for the forecasting because not enough different types of data were available at Saint Teilo for a common period of time. Thus, models presented in section 3 focus on data from the Woopa building introduced below.

Woopa

The first set of data in Woopa's building is the global gas consumption of the building (Figure 2).

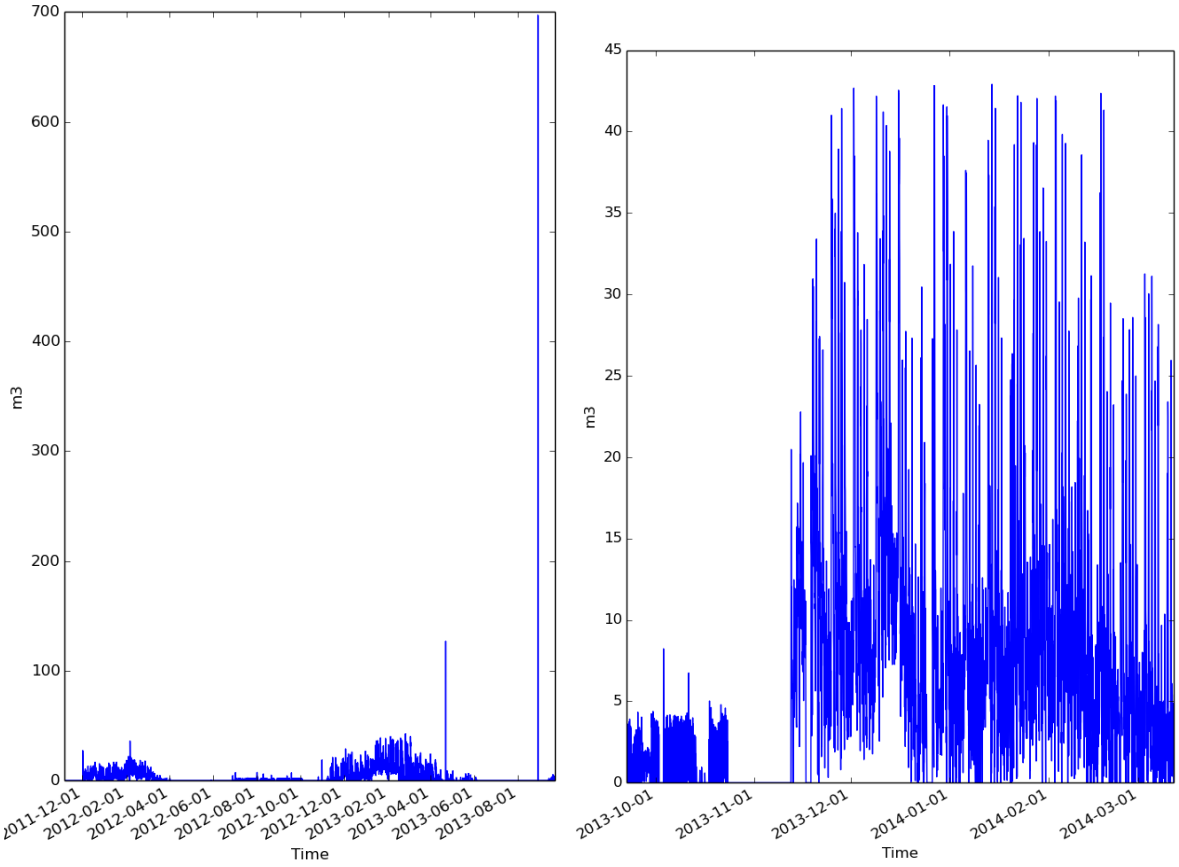

Fig. 2 Woopa Total Gas Consumption (m3). Training set (left), test set (right).

Data is acquired at a sample rate of 15 minutes and ranges from 2011 November 6th to 2014 March 12th. Data is first resampled hourly keeping track of the summed consumption in that interval. Data contain some erroneous values, baseline data should be validated by pilot sites specialists before considering that no abnormal values are present. The dataset is given as is to the models.

A second set of data is the Cold Sanitary water consumption in the Woopa building. Acquisition period and sample rate remain identical to previous set. Again some suspicious spikes are easily detected in the training set, the highest spike has been removed from dataset and remaining is given as is to the models.

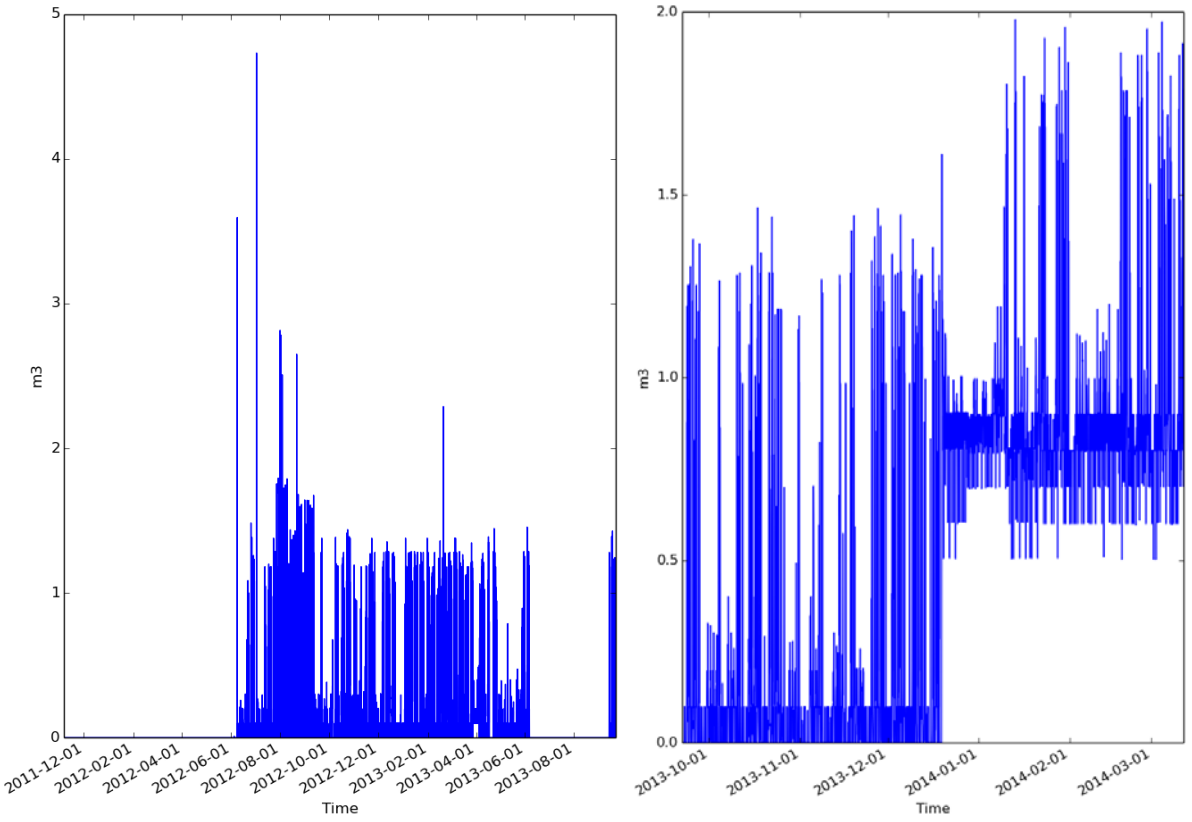

Fig. 3 Woopa Total Water Consumption (m3). Training set (left), test set (right). (m3)

However, in this dataset, some slippage offset is also clearly visible in test set. Test set starts in October 2013 until the end of the period (see Figure 3). This may/will have an impact on models performances as it is not included in models training set. A huge part of the training set is also composed of zero values. It seems that consumption has either not been correctly recorded during the acquisition period or greatly changed over time.

The third dataset used in this work is an electrical consumption one (Figure 4). We chose to use the global lighting consumption of a subzone in the first level of the Woopa building. This third dataset allows us to test the models on an occupancy signal of the building which constitutes a third category of data.

The training set contains some abnormal spikes, especially one raising up to about 750kWh that has been removed from training set (0 instead). There remain suspicious ones that are fed as is to the models. Normal data range is more visible in the test set (< 1kWh per hour). For Deep Networks models, total value range of the data has great influence as being normalized before feeding network. A quick test showed that there is a factor 10 between test mse with the 750kwh spike and without.

3. Algorithms

ARMA model

Auto-Regressive and Moving Average (ARMA) (Box, Jenkins, & Reinsel, 1994) is a common class of models for time series prediction composed of two parameterized parts for describing a stationary process. If raw data do not reflect a stationary process, an Integrated model (ARIMA) can be used to get rid of non-stationarity behaviour by using a given order of process differentiation.

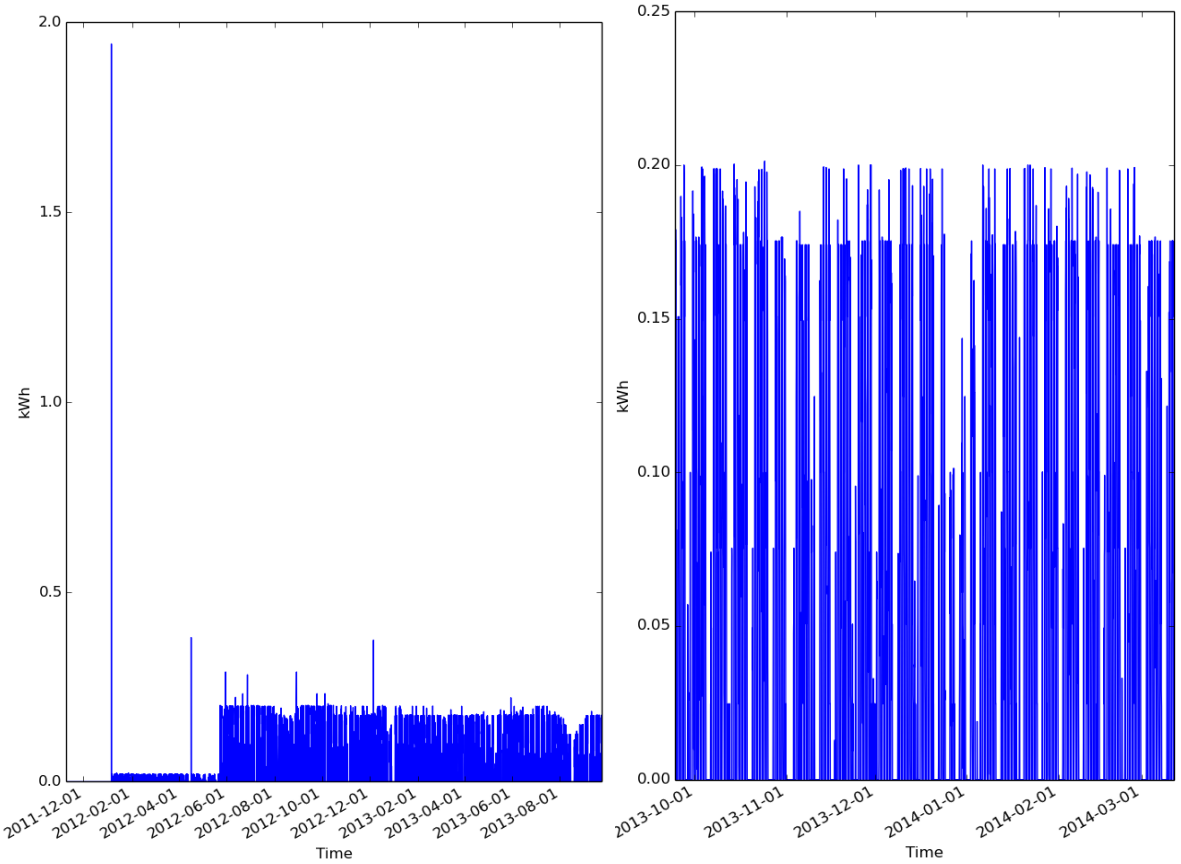

Fig. 4 Woopa lighting consumption of a subzone of a storey (kWh). Training set (left), test set (right)

An ARMA model is a combination of an auto-regressive AR and moving average model.
An AR model can be expressed as follows:

$$X_t = C + \sum_{i=1}^{n} \propto_i X_{t-i} + \varepsilon_t$$

where C is a constant, ε is a residual (random noise), n is the model's order, and $\propto$ is the coefficients used for the weighted sum of the past values used in the model. X_t is the time serie value at time t. The order of the model can be a consecutive past values set, where n will then reflect the number of past values used. The order can also be a list enumerating indices of past values to be used in the combination.

An MA moving average model is expressed as follows:

$$X_t = \mu + \sum_{i=1}^{p} \beta_i \varepsilon_{t-i} + \varepsilon_t$$

Where μ is mean of the series modelled and ε is random noise.
From these definitions, an ARMA model is the combination leading to the following expression;

$$X_t = C + \mu + \sum_{i=1}^{n} \propto_i X_{t-i} + \sum_{i=1}^{p} \beta_i \varepsilon_{t-i} + \varepsilon_t$$

The model implementation used in the present work comes from (Perktold, Seabold, & Taylor, 2016). The model is encapsulated in a python library for online data acquisition and predictions updates.

Deep Highway Networks

DNN has gained popularity in recent years (LeCun, 2015). They have been extensively used in different applications for classification purposes (He, Zhang, Ren, & Sun, 2015) (Ilya Sutskever, 2014) but their application for sequence/times series prediction is quite new (Busseti, Osband, & Wong, 2012) (Dalto, 2015) (Li, Bai, & Zeng, 2016) (Wang & al. 2016). A recent review (Schmidhuber, 2014) retraces progress and advances in DL which encompasses a wide scope of applications and contests. The learning phase of these models can be a challenging task. In particular, learning algorithms hardly perform when increasing the networks depth. A recent work (Srivastava, Greff, & Schmidhuber, 2015) proposes a Deep so-called Highway Network (DHN) model that shows to be more stable in learning when increasing the number of hidden layers. A Highway Network can be defined in a simplified form as follows;

$$y = H(x, W_H).T(x, W_T) + x.C(x, W_C)$$

Where y is the output of the network, H is the non-linear transformation applied to the input x weighted by a parameters matrix W_H. This corresponds to a classical feedforward neural network architecture. In the H is generally composed of several layers of non-linear transformations (and their corresponding weights matrices), where each layer receives its inputs from the preceding layer's outputs and outputs to the next layer.
The Highway property of this model compared to a more classical DNN lies in the two classes of linear transformations added; the *transform gate* $T(x, W_T)$ and *carry gate* (x, W_C) . The first gate transforms the input and the second allows to carry input in a possibly unchanged form through the different layers of the network, depending on the weights applied. In this work, a simplified model is used as proposed in the original work where $C = 1 - T$.

The regular hidden layers are populated with Rectifier Linear Units (ReLu) (Glorot, Bordes, & Bengio, 2011) and gate layers use a sigmoidal activation function. The model used in the present work is adapted from (Dieleman, 2015) and encapsulated in a python library to allow online data acquisition and predictions updates.

4. Results

Preliminary analyses have shown that daily, weekly and yearly periods were among the most powerful periodicities contained in time series. It also showed that clear discrimination between week days and week end/vacations existed in data.
From these preliminary analyses, the models parameters have been chosen. ARMA models orders were also slightly optimized minimizing the Bayesian Information Criterion (BIC). A best moving average order was found at 20 (with a paremeter space between 0 and 50). The first derivative order, making the ARMA an ARIMA model, of time series was used to make the data stationary. For the auto-regressive order, two models were kept for comparison. The first one uses values of the same hour the day before and the same hour on same day the week before to predict a value. The second model uses every values of the past 24 hours and every values of the same day the week before (24 hours starting 168 hours in the past). ARMA models parameters are tuned through maximum likelihood.

This produces two ARMA models to compare.

Table 1. Root mean squared error obtained on Woopa Total Gas consumption. Forecast 1hour

Woopa Total Gas consumption (m3)	
Model	**RMSE**
Best ARMA	5.49
Best DHN	**5.04**
Naïve	7.86

Highway networks parameters have been selected after preliminary tests on subparts of data. The selected set of parameters showed the best prediction behaviour compromise between parameters space size and performance. Networks are fed with the same inputs as the second ARMA model: every hour of last 24 hours and the 24 hours of the last week's same day. A 48 input vector is thus built from signals to feed the network. Thus both kinds of models received comparable inputs.

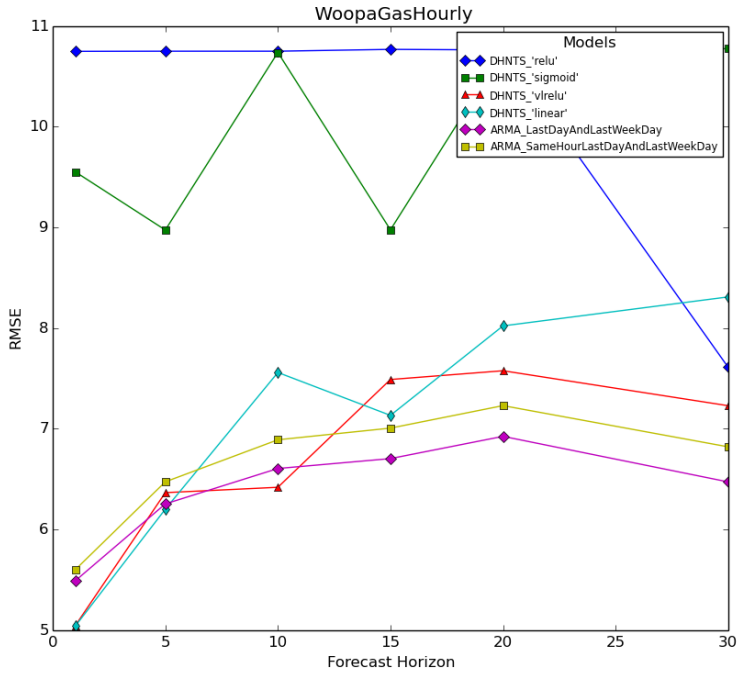

Fig. 5. Performances of models w.r.t the forecast horizon, Gas consumption.

The networks are chosen with 150 hidden units in each of the 20 hidden layers. An output layer is composed of only one cell, to reproduce a one dimensional time serie. The network is thus trained to achieve a regression on the input time series. Four types of activation function of the output layer are used for sake of comparison: ReLu, VeryLeaky Relu (with 0.3 as slope of the rectified part), Sigmoid and Linear. 150 training epochs with a learning rate set to 0.3 (momentum at 0.95) and a bias initialized to -4.0 ends the set of training parameters. Networks are trained through SGD (Stochastic Gradient Descent) with batches of 64 samples and some optimization as described in (Srivastava, Greff, & Schmidhuber, 2015).

Each model is trained with the same part of data and tested with another same part to make results comparable. Models are trained and tested with a forecast horizon of 1, 5, 10, 15, 20 and 30 steps ahead to compare their ability to produce accurate predictions on different horizons.

Networks outputs consist in only 1 value predicted for the next step. To achieve several steps ahead forecasting predictions are reused as input for next step prediction. Thus the same model is used, as for ARMA models, to generate forecasted values whatever the the forecast horizon is.

Table 2. Root mean squared error obtained on Woopa sanitary cold water consumption. Forecast 1hour

Woopa Total Sanitary cold water consumption (m3)	
Model	**RMSE**
Best ARMA	**0.22**
Best DHN	0.53
Naïve	0.3

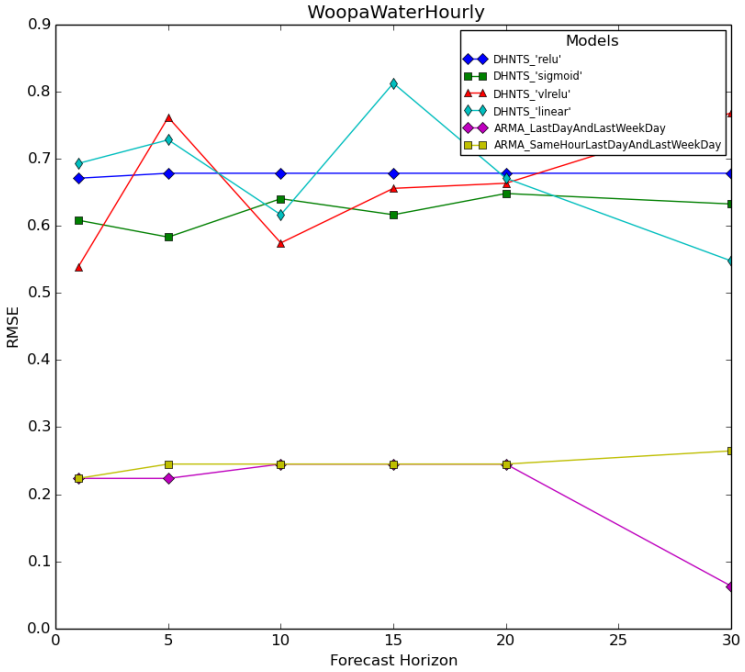

Fig. 6. Performances of models w.r.t the forecast horizon, Water consumption

Tables 1, 2 and 3 shows the performances of the best models for a forecast horizon of 1 sample for each data type. Models performances are compared to a so-called "naïve" model which only reproduces last sample's value as the predicted future value.

Figure 5, 6 and 7 shows performances of all models w.r.t. the forecast horizon duration from 1 to 30 steps ahead. Performance is expressed in terms of root mean squared error.

Table 3. Root mean squared error obtained on Woopa First floor zone RH5 Global Lighting consumption. Forecast 1hour

Woopa Light consumption zone RH5 – R+1 (kWh)	
Model	**RMSE**
Best ARMA	0.03
Best DHN	**0.02**
Naïve	0.22

Labels in Figure 5, 6 and 7 legends indicate the corresponding model: four different output activation functions for DHN models and the two different Auto Regressive orders for ARMA models. From this plot and the tables presented above, we can observe several indications of comparative performances of each model.

The first observation concerns one of the three datasets; namely the water consumption one. In the Figure 5 presented in previous sections a clear offset arises during the test set. This offset shifts the average consumption values until the end of the dataset. Such an offset is not observable in the training set. When we focus on models performances on this dataset, the weak performance of DHN models (worse than naïve, see Table 2) can be explained by this difference of behaviour in training and test sets. The ARMA models on the contrary handles this difficulty with an interesting performance (0.22 as RMSE for an average dataset value between 0.0 and 2.0 m^3, see Table 2 and Figure 5).

This difference in performances is explainable by the fact that DHN focuses on training set to build a model of the data when ARMA considers a moving average and an auto-regressive part, thus whatever the changes in dataset, ARMA will adapt its predictions in this way.

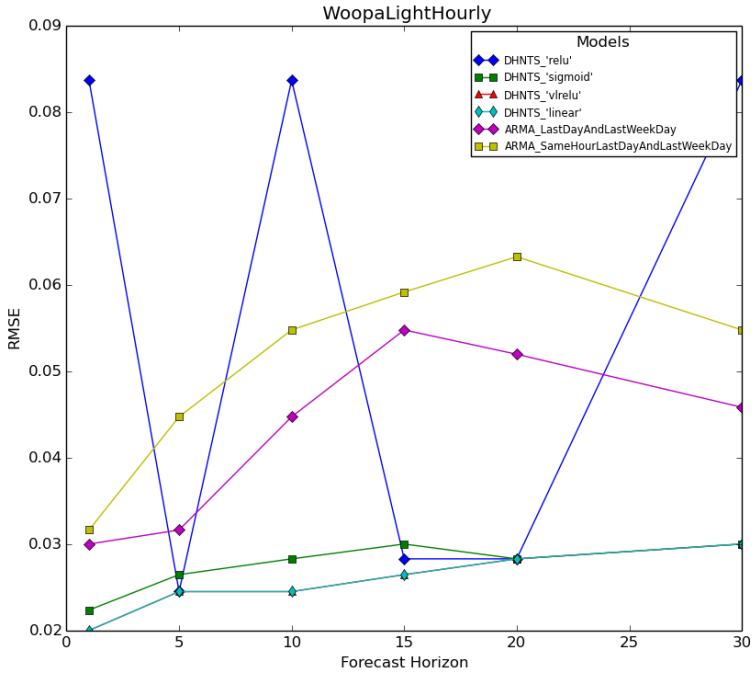

Fig. 7. Performances of models w.r.t the forecast horizon, Light consumption

A second observation is that, if we exclude the water dataset for the reasons explained above, DHN models perform better than ARMA models. This is particularly true on short term prediction as shown in table 1 and 3. Figure 7 also shows that it remains true in the light consumption dataset even for longer forecast horizons.
However, the gas consumption dataset shows that long horizons benefits to ARMA models (Figure 7). The fact that, in this study, DHN models are built with only one output value for next step can be a limitating property in this case (Models with *n* output values could have been created for each of the forecast horizons studied, with *n* the considered horizon). Output function of DHN models shows great influence on model performance depending on dataset. Across datasets and forecast horizons, linear and very leaky rectified linear units (vlReLu) show better performance than sigmoid and rectified linear units (ReLu).

The complexity of the models often being used as a criterion against the use of deep networks models, it is interesting to have an idea of the computational time needed for each model type. In the present work, the the models implementations used lead to a faster execution time for DHN models training and test w.r.t. ARMA models.

Conclusions

This work as part of the Performer European project has focused on comparing performances of ARMA and Deep Highway Networks models for the forecasting of monodimensional signals solely based on past values. Data used have been produced by the project pilot sites, mainly Woopa building, and represent different types of real data acquired over a long period of time. Models parameters have been chosen based on preliminary anylses conducted on project's buildings data. Models performances have been compared across data types and different forecast horizons.

Performance results show that both model types achieve accurate predictions with regard to a naïve prediction, even for uncleaned datasets. Results also show that Deep Networks can improve forecast accuracy. As a counter part they are less adaptative to changes in data behaviors. As is known, networks need to be trained on highly representative datasets to improve accuracy.
It has also been shown that computation load is heavier for the ARMA models than for the DHNs when considering the implmentations used in this work.
Thus, even in a regression scheme with unclean datasets, Deep Networks seem to be able to perform beyond ARMA models, for the present data tested, with less computational load. The main drawback lie in the need for a representative training set and an adapted output dimension.

The work presented in this paper was supported by the European Community's Seventh Framework Programme under Grant Agreement N° 609154 (Project PERFORMER). Special thanks also goes to Sylvain Robert (CEA) and Frederic Suard (CEA) for their valuable comments and to the other partners of the project for their help in data gathering.

References

Box, G. E., Jenkins, G., & Reinsel, G. C. 1994. *Time Series Analysis: Forecasting and Control*. Prentice-Hall. http://dx.doi.org/10.1002/9781118619193

Busseti, E., I. Osband, and S. Wong, 2012. *Deep Learning for Time Series Modeling.* CS 2009, Final Project Report: p. 1-5.

Ciresan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. 2013. *Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks*. MICCAI, 411-418. DOI: 10.1007/978-3-642-40763-5_51

Dalto, M., 2015. *Deep Neural networks for ultra-short-term wind forecasting.* ICIT: p. 1657-1663. DOI**:** 10.1109/ICIT.2015.7125335

Dieleman, S., 2015. *Highway Networks*. From Github.com: http://github.com/Lasagne/blob/highway_example/examples/Highway Networks.ipynb

Foucquier, A., Robert, S., Suard, F., Stephan, L., & Arnaud, J. 2013. *State of the art in building modelling and energy performances prediction: A review*. Renewable and Sustainable Energy Reviews, 272-288. http://dx.doi.org/10.1016/j.rser.2013.03.004

Glorot, X., A. Bordes, and Y. Bengio, 2011. *Deep Sparse Rectifier Neural Networks.* AISTATS: p. 315-323

He, K., et al., 2015. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.* Arxiv**:**1502.01852.

Ilya Sutskever, O.V., Quoc V. Le, 2014. *Sequence to Sequence Learning.* NIPS.

LeCun, Y. B. 2015. *Deep learning*. Nature, 436–444. http://dx.doi.org/10.1038/nature14539

Li, C., Bai, Y., Zeng, B. 2016. *Deep Feature Learning Architectures for Daily Reservoir Inflow Forecasting.* Water Resources Management, 30 (14), pp. 5145-5161.

Perktold, J., Seabold, S., & Taylor, J., 2016. *Statsmodels SARIMAX*. From www.statsmodels.org: http://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html

Schmidhuber, J., 2014. *Deep Learning in Neural Networks: An Overview.* arxiv: Technical Report.

Srivastava, R., Kumar, K. Greff, and J. Schmidhuber, 2015. *Training Very Deep Networks.* NIPS. arXiv:1507.06228

Wang, H.Z., Wang, G.B., Li, G.Q., Peng, J.C., Liu, Y.T. 2016. *Deep belief network based deterministic and probabilistic wind speed forecasting approach.* Applied Energy, 182, pp. 80-93.

Aknowledgements

This research was supported by the PERFORMER project. Project funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 609154.

The article reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability

Author

Anthony MOURAUD is a Research Engineer and Project Manager at CEA Tech PdlL, Bouguenais, FR. He has been working on computational neuroscience as a Doctoral and Post-Doctoral fellow at Louis Lumière University, Lyon, FR, Antilles-Guyane University, Pointe-à-Pitre, FR, Luminy University, Marseille, FR, Computer Science Research Laboratory (LRI), Orsay, FR and CEA, Saclay, FR. He also has been a Software Engineer at INCKA, Arcueil, FR. Research interests: Machine Learning; Computational Neuroscience; Signal Processing.
ORCID ID: 0000-0003-3692-7310
