25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-0789
MODELING THE THERMAL PERFORMANCE OF A WINDOW TYPE AIR-CONDITIONING SYSTEM WITH ARTIFICIAL NEURAL NETWORKS

João Victor Fabri
Pablo Andretta Jaskowiak
Diogo Lôndero da Silva
ReVe - Vehicular Refrigeration Laboratory
Departament of Mobility Engineering
Federal University of Santa Catarina (UFSC)
89219-600, Joinville, SC, Brazil
joao.fabri@grad.ufsc.br, pablo.andretta@ufsc.br, diogo.londero@ufsc.br

Abstract. *Mathematical simulations of air-conditioning systems based on physical principles usually produce high precision results. Such simulations, however, tend to be time consuming and / or computationally expensive, turning out to be prohibitive in applications where a quick response is needed. A promising alternative to tackle this issue is the use of Artificial Neural Networks (ANNs), data driven, bio-inspired models that mimic (roughly) human cognition to solve predefined, specific problems. A remarkable characteristic of ANNs is their ability to generalize, that is, to learn based on a set of a priory labeled data and produce predictions for new (unseen) data quickly. In this work we employ ANNs to model the steady-state operation of a window type air-conditioning system. More specifically, based on experimental data, we build / induce models to predict the cooling capacity of the system. Ten different architectures of multilayer feedforward neural networks are considered in our empirical evaluation. Our results suggest that several configurations are capable of modeling the system with considerable accuracy (R^2 up to 0.97) in relation to the experimental results, considering only non-invasive measurements of temperature, humidity, and air flow levels from the device. Given that such measurements are readily available and can be obtained with a considerably low cost, our results suggest that these models are appealing for applications where low latency predictions are needed, such as, Internet of Things (IoT) platforms, Smart Buildings, and Control Systems.*

Keywords: *HVAC systems, air-conditioning systems, artificial neural networks, machine learning*

1. INTRODUCTION

The use of Heating, Ventilation, and Air Conditioning (HVAC) systems has increased significantly in past decades, with some authors even claiming that HVAC systems are no more a luxury but an essential item in everyday life (Pérez-Lombard *et al.*, 2008). On the other hand, such systems account for as much as 50% of energy consumption in a building, according to Pérez-Lombard *et al.* (2008). Other estimates suggest that, in countries with a tropical climate, consumption figures for HVAC systems can go even higher than 50% (Chua *et al.*, 2013). Given their increasingly adoption and high energy consumption, there is clear demand for the development of more efficient devices.

Models of Heating, Ventilation, and Air Conditioning systems play an important role in their design, development and performance analysis (Castilla *et al.*, 2013; Afram and Janabi-Sharifi, 2015; Afroz *et al.*, 2018). A better understanding of energy consumption behavior can, in turn, allow the development and adoption of efficient control strategies (Afroz *et al.*, 2018). In brief, according to Afram and Janabi-Sharifi (2014), who provide an in depth review of models in this particular reserch area, HVAC systems can be modeled by techniques that fall into three major classes or categories, namely: physics based (or white-box), data driven (or black-box), and hybrid models (or gray-box). Techniques from latter category are usually built by a combination of the former two, in which model structure comes from a physics based model and model parameters are estimated based on experimental data (Homod, 2013).

Techniques that fall within the first category, that is, physics based models, build an explicit mathematical model of the system. Simulations of air-conditioning systems are performed taking into account the conservation of mass, energy and momentum, requiring the simultaneous solution of non-linear equations sets (Homod, 2013). The iterative procedures employed to obtain such solutions have a relatively high computational cost (Hermes and Melo, 2009; Zeng *et al.*, 2015). Nevertheless, they are widely adopted due to their sound physical foundations and accurate results (Ploug-

Sørensen *et al.*, 1997; Li *et al.*, 2017). It is worth noting that if results are needed with low latency (as in real-time control applications), such mathematical simulations can turn out to be prohibitive or unfeasible due to their high computational requirements (Afroz *et al.*, 2018).

Data driven models build a mapping (relationship) between input and output variables of the air conditioning system (Afroz *et al.*, 2018). As suggested by their name, such mappings are obtained from experimental data, containing both the input and desired output, the so-called training data. Once a model is trained, it should (ideally) be able to generalize, that is, produce consistent results (output values) for inputs not seen during its training phase (new data). Regarding data driven models, various methods from different fields, such as Statistics, Machine Learning, and Data Mining, have been broadly adopted (Afram and Janabi-Sharifi, 2014). Indeed, methods such as Artificial Neural Networks (ANNs) have received considerable attention in the HVAC literature (Zeng *et al.*, 2015). Data driven methods are usually characterized by their simplicity and ability to meet requirements for real-time operation and control. They rely, however, on experimental measurements (Afroz *et al.*, 2018). These models are deemed as black box given that little knowledge about the dynamics of the system is needed for their application. Conversely, these models alone, provide limited or no additional understanding regarding the system behavior.

In this paper we model the steady-state thermal performance of a window type air-conditioning system using ANNs, which are data driven / black box models. More specifically, we evaluate ten different configurations (architectures) of multilayer feedforward neural networks (Haykin, 2009) with different numbers of neurons and layers. These models are applied to predict the cooling capacity of the HVAC system under study. Our results suggest that several configurations are capable of modeling the system. In order to train the different models we collect experimental data from a commercial window type air-conditioning system considering solely measurements of temperature, humidity, and air flow levels from the device. Given that such measurements are readily available at relatively low cost, these models are appealing for applications such as Internet of Things (IoT) platforms, Smart Buildings, and Control Systems.

The remainder of the paper is organized as follows. In Section 2, Materials and Methods, we discuss how experimental data was collected, briefly review Artificial Neural Networks (ANNs) and discuss the evaluation procedure (experimental setup) adopted in this paper. Results from an empirical evaluation of the ten different models are presented in Section 3, alongside discussion of the results. Finally, in Section 4 we draw the main conclusions of our work and discuss future venues of research.

2. MATERIALS AND METHODS

In this section we detail how data collection was performed and provide a brief review of Artificial Neural Networks (ANNs), the data driven / black box model considered in our empirical evaluation. We conclude the section by discussing the evaluation methodology (experimental setup).

2.1 Data Collection

In order to train, evaluate and validate the different ANN models, experimental data was collected from a commercially available window-type air-conditioning system. The air-conditioning system has a nominal cooling capacity of 7500 Btu/h and is equipped with a rotary type compressor and two capillary tubes as expansion device. The air-conditioning system was instrumented with T-type thermocouples, capacitive humidity transducers and an ammeter with uncertainties equivalent to $\pm 0.5^o$C, $\pm 3\%$, and ± 0.1A, respectively. The air temperature was measured in the inlet and outlet positions of the evaporator and the condenser. The refrigerant temperature was measured in the evaporator, condenser, compressor and expansion device inlet and outlet positions. Moreover, the air humidity change in the evaporator and the electrical current supplied to the air-conditioning system were continually measured.

The experiments were carried out under steady-state and different environmental conditions, considering dry bulb temperatures and relative humidity between 24 — 37^{o}C and 34 — 73%, respectively. In order to identify a steady-state condition we fit a linear regression model considering the last 50 points measured. If the absolute value of the angular coefficient was smaller than 10^{-3}, then a steady-state condition was identified (Balbinot and Brusamarello, 2007). After identifyig that the system was operating in steady-state condition 50 data points were collected, considering all variables under observation. The air-conditioning system cooling capacity was evaluated based on the compressor manufacturer performance curves, as a function of the operating conditions. Moreover, two levels of air-flow (the two settings available for the device) were tested on the evaporator. A total of 14 experiments were carried out, resulting in the collection of 700 points during the experimental activity, all of them considering the steady-state operation of the air-conditioning system.

2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are mathematical / computational models inspired in the brain and its underlying capacity of solving complex problems. These models, which are also referred to as bio-inspired, aim to mimic (roughly) the brain's ability to solve well defined and specific problems (Haykin, 2009). Research in Artificial Neural Networks has a long history of development, marked by periods of great excitement and skepticism. Currently, the field is receiving great attention mostly due to recent advances in Deep Learning (Schmidhuber, 2015; Goodfellow *et al.*, 2016). ANNs have been applied to a wide range of problems including, naturally, HVAC systems.

The basic processing unit (building block) of an ANN is the artificial neuron, hereafter simply referred to as neuron. A neuron receives as inputs values that come from the ambient (values that are external to the neural network), that is, from the problem under consideration (in our case, values of humidity, temperature and air-flow level) or other neurons. Each one of these values is multiplied by weights (which roughly correspond to the delays in information transmission in biological synapses). Once combined (added), these go through an activation function, that produces the output of the neuron. The basic model of a neuron is given in Figure 1, where the bias is another degree of freedom (parameter) that the network will learn. There are several different types of artificial neurons described in the literature (Haykin, 2009). In our work, we consider the ReLU (Rectified Linear Unit) Activation Function (Goodfellow *et al.*, 2016), given by $f(u) = y = \max(0, u)$, with $u = \sum_i^n x_i w_i + \text{Bias}$, where n is the number of inputs provided to the neuron.

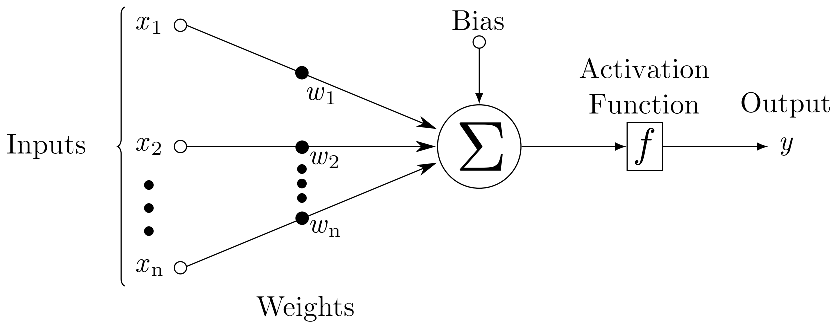

Figure 1: An Artificial Neuron and its basic building blocks.

A single neuron, by itself, has very limited computational power. In order to solve complex problems, neurons are often combined to form a so called neural network. A well-known and widely adopted configuration of neurons is the feedforward neural network (the architecture we employ in this work), for which an example diagram is shown in Figure 2. In this particular neural network architecture the values are propagated from one layer to the next, without loops, cycles or any kind of recurrency. It is worth noting that no processing is performed in the nodes in the input layer (these are not actually neurons, hence the different representation), they solely propagate the input value to the neurons of the next layer.

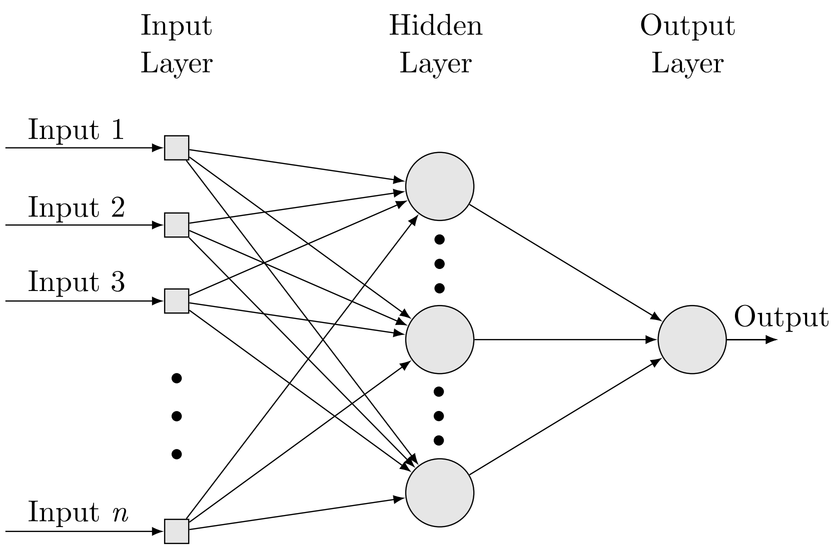

Figure 2: Diagram of a multilayer feedforward artificial neural network. For simplicity, the weights and bias of each neuron are not shown in this figure.

Note that although the ANN shown in Figure 2 has only one hidden layer, theoretically, there is actually no limit on the number of hidden layers a feedforward neural network can have. Including layers and / or neurons in each layer results in an increase in the number of free parameters of the model which, in turn, will have to be adjusted during the training phase of the model. Determining the number of free parameters (number of layers and neurons per layer) is not a trivial task (Haykin, 2009). On one hand, a small number of free parameters can result in underfitting, that is, the model is unable to learn the complexities of the problem. On the other hand, a large number of free parameters can result in overfitting during the training procedure, that is, the model, becomes overspecialized for the training data, losing its ability to handle new data (generalization) (Haykin, 2009). In order to circumvent such possibilities, we considered and evaluated different configurations regarding number of layers and neurons.

Regarding architecture, in our specific case, each neural network has six inputs, namely: external air humidity, fan speed, and four inputs related to temperature, two from the condenser and two from the evaporator (collected in a non intrusive manner). All networks have a single output, which corresponds to an estimate of the cooling capacity for the given input configuration. All networks were implemented and evaluated in Python (van Rossum, 1995), using Keras (Chollet *et al.*, 2015) and Tensorflow (Abadi *et al.*, 2015) frameworks / libraries, which provide functionalities for building and training different ANN models. Finally, in the training phase of the model we employed the RMSProp algorithm (Tieleman and Hinton, 2012) with its default parameters, as provided by the Keras library. For a given model, this algorithm creates a mapping between the inputs and desired output, by adjusting the weights of the neural network. In this work, the inputs are external air humidity, fan speed, and four inputs related to temperature, two from the condenser and two from the evaporator, whereas the output corresponds to an estimate of the cooling capacity for the given input configuration. After obtaining the model it is then evaluated as detailed in the next section.

2.3 Evaluation

In order to evaluate each network model, we simulate a real application where the ANN is trained in part of the data (hereafter referred to as training data) and evaluated it in data not used for training, that is, data that was not previously seen by the model (i.e., test data). This process is systematically performed through a procedure called cross validation (Haykin, 2009; Tan *et al.*, 2018). More specifically, we employed the 10 fold cross validation procedure, as detailed in the sequel. In brief, data is split into 10 equally sized folds (parts) and the following process is performed: one fold of the data is set apart for testing, and the model is trained in the remaining 9 folds. This process is then repeated 10 times and, at each step, the fold used for test changes. As a result, each test fold (and consequently each data point) is used exactly once for testing. At the end of the process, different measures of quality can be computed to evaluate the generalization capacity of the model, that is, how it performs on new data, as simulated by the test folds.

In this work we compute two measures in order to evaluate the models. For the definition of the measures, let us consider a dataset with n objects (data points). The first measure we adopt is the Mean Absolute Percentage Error (MAPE), given by Equation (1), where E_i and P_i account for the *expected* and *predicted* values for the ith data point.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{E_i - P_i}{E_i} \right| \tag{1}$$

The second one is the well-known Coefficient of Determination (R^2), which is related to the Pearson correlation (Tan *et al.*, 2018). Its values are bounded in the [0,1] interval, with higher values indicating a better model. Its formulation is given by Equation (2), where E_i and P_i, once again, account for the *expected* and *predicted* values for the ith data point. The term $\bar{E}$ accounts for the mean of the expected values.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (E_i - P_i)^2}{\sum_{i=1}^{n} (E_i - \bar{E})^2} \tag{2}$$

Both metrics were computed using the Scikit-learn library (Pedregosa *et al.*, 2011).

3. RESULTS

We evaluated a total of 10 different ANN architectures, varying their number of layers and neurons per layer. Each model is specified by its particular configuration. For example, a $6 \times 14 \times 12 \times 1$ architecture has 6 inputs, 2 hidden layers, with 14 and 12 neurons, respectively, and one output. It is worth noticing that all ANN architectures under evaluation have 6 inputs and 1 single output. The inputs are external air humidity, fan speed, and four inputs related to temperature, two from the condenser and two from the evaporator (collected in a non intrusive manner). The output corresponds to an estimate of the cooling capacity for the given input configuration. Although there is some guidance on how to select the number of layers and neurons, this number is, in practice, usually obtained from empirical evaluations and / or experience (Haykin, 2009; Goodfellow *et al.*, 2016), procedure which we adopt here. In our experiments we considered 1 and 2 hidden layers and different numbers of neurons per layer (defined empirically). Each model was trained for a maximum of 500 epochs (iterations of the RMSProp training algorithm) given that, in general, models converged fast to a low MSE (Mean Squared Error), which is the loss / objective function optimized during the training phase. This is exemplified by Figure 3, which relates epochs and MSE values during training of one of the models.

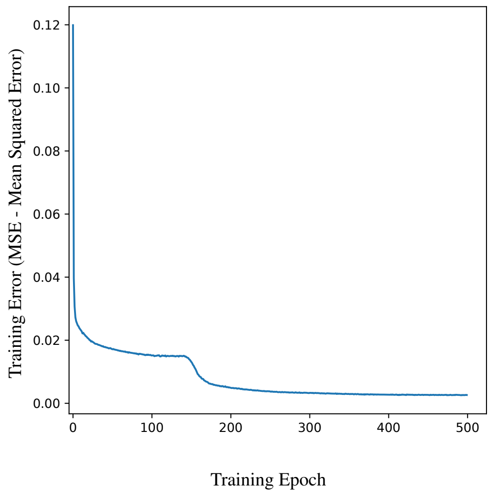

Figure 3: Convergence plot obtained during the training of model 6x14x6x1.

Results for the ten different architectures under evaluation are shown in Table 1. We report the mean and standard deviation (referred to as Std) regarding both R^2 and MAPE. These values were obtained by the 10 fold cross validation.

As expected, one of the worst results was obtained with one of the simplest models under evaluation (6x3x1). In terms of R^2, the model scored on average 0.89 ± 0.02 with a MAPE close to 25%. In view of such results it is likely that the model was not complex enough (small number of free parameters) to capture the system behavior. For the remainder models, values of R^2 were higher than 0.90, indicating a good fit and generalization when evaluated on new data (test data). Indeed, models with two hidden layers achieved R^2 values of 0.97 when evaluated on the test data. It is actually possible to observe that increasing model complexity led to better results, in general. Apart from the underfitting observed for the 6x3x1 model, there appears to be no overfitting in the results. If that was the case, one would expect to observe a decrease in generalization performance as model complexity increases. That is, the model would become so much overspecialized for the training data that its performance on unseen data would be impaired.

The results regarding MAPE are consistent to those observed with R^2. The top scoring models are the ones with two hidden layers (more free parameters). These models achieved a Mean Absolute Precision Error (MAPE) of approximately 10% which is a high prediction accuracy. It is also worth noticing that the standard deviation for these particular models is quite low across the different test folds (smaller than 5%), indicating that the models are quite stable to data disturbances and variations.

In Figure 4 we present scatter plots relating predictions and desired values (expected values) for some of the models under consideration (we draw a diagonal line to show what would be the behavior of an “optimal” model).

Table 1: Evaluation metrics for each model

Model Configuration	R^2		MAPE (%)	
	Mean	Std	Mean	Std
$6 \times 3 \times 1$	0.89	0.02	24.51	08.88
$6 \times 6 \times 1$	0.91	0.02	22.68	09.03
$6 \times 9 \times 1$	0.90	0.02	24.45	10.97
$6 \times 14 \times 1$	0.95	0.01	16.55	06.41
$6 \times 20 \times 1$	0.95	0.01	15.87	06.75
$6 \times 40 \times 1$	0.95	0.02	13.63	06.23
$6 \times 6 \times 6 \times 1$	0.93	0.06	17.33	12.01
$6 \times 6 \times 12 \times 1$	0.97	0.01	12.77	05.17
$6 \times 14 \times 6 \times 1$	0.97	0.01	10.97	04.44
$6 \times 14 \times 12 \times 1$	0.97	0.01	10.17	04.88

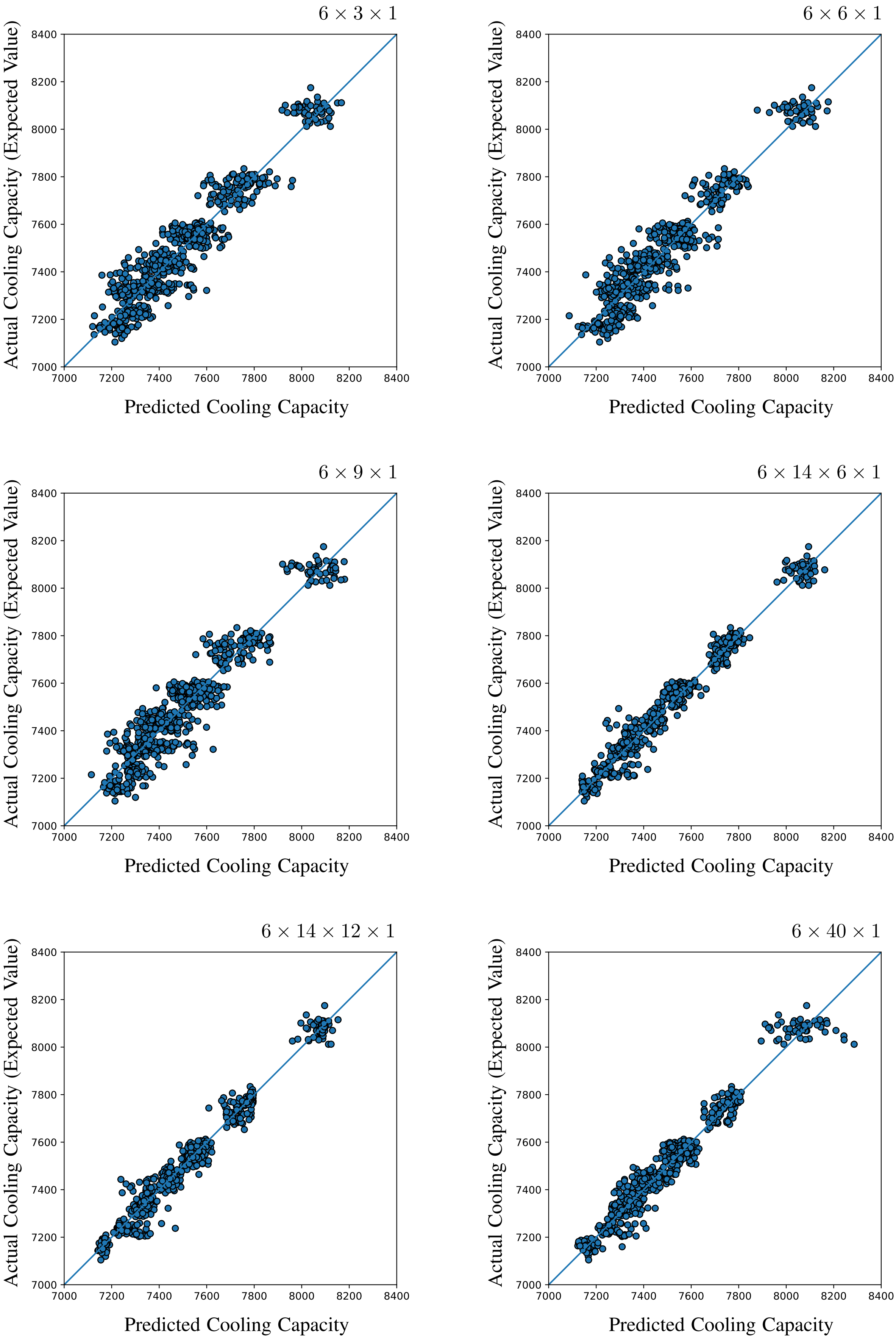

Figure 4: Scatter plots relating actual cooling capacity (y axis) and predicted cooling capacity (x axis) — both presented in Btu/h. Model architecture is specified in each plot (top right). Increasing the model complexity, usualy led to better predictions results.

One can see that as model complexity increases[1] there is also an increase in the performance of the model with respect to its predictions.

4. CONCLUSIONS

In this work we employed and evaluated Artificial Neural Networks to model the thermal performance of a window type air-conditioning system. Experimental data from a HVAC system was collected and used to induce (train) different data driven (black box) models. We evaluated 10 models (which translate in different configurations of ANNs) regarding their ability to predict the cooling capacity of the HVAC system under study. Our results suggest that several ANN architectures can be successfully applied to predict the cooling capacity of the system. In fact, the best model overall ($6 \times 14 \times 12 \times 1$) obtained a R^2 coefficient of 0.97 and a MAPE of 10.17%.

Although the training process of each model is somehow time consuming, it is usually performed offline, that is, prior to the model application. Once trained, the models respond quickly to the inputs. Indeed, our models took less than 1 millisecond to provide each prediction in a personal laptop. Therefore, we argue that ANNs not only provide a good prediction, but also a low latency to provide such results, making them an appealing alternative to implementations in control, for example. Moreover, applications in embedded systems can also be considered, even when low computational power is available. Given this results, we intend to investigate the use of such models in intelligent control strategies in our next works.

5. REFERENCES

Abadi, M., Agarwal, A., Barham, P. *et al.*, 2015. "TensorFlow: Large-scale machine learning on heterogeneous systems". URL `http://tensorflow.org/`. Software available from tensorflow.org.

Afram, A. and Janabi-Sharifi, F., 2014. "Review of modeling methods for hvac systems". *Applied Thermal Engineering*, Vol. 67, No. 1, pp. 507 – 519.

Afram, A. and Janabi-Sharifi, F., 2015. "Gray-box modeling and validation of residential hvac system for control system design". *Applied Energy*, Vol. 137, pp. 134 – 150. ISSN 0306-2619.

Afroz, Z., Shafiullah, G., Urmee, T. and Higgins, G., 2018. "Modeling techniques used in building hvac control systems: A review". *Renewable and Sustainable Energy Reviews*, Vol. 83, pp. 64 – 84.

Balbinot, A. and Brusamarello, V.J., 2007. *Instrumentação e Fundamentos de Medida*. Editora LTC.

Castilla, M., Álvarez, J., Ortega, M. and Arahal, M., 2013. "Neural network and polynomial approximated thermal comfort models for hvac systems". *Building and Environment*, Vol. 59, pp. 107 – 115. ISSN 0360-1323.

Chollet, F. *et al.*, 2015. "Keras". `https://github.com/fchollet/keras`.

Chua, K., Chou, S., Yang, W. and Yan, J., 2013. "Achieving better energy-efficient air conditioning - a review of technologies and strategies". *Applied Energy*, Vol. 104, pp. 87 – 104. ISSN 0306-2619.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Haykin, S., 2009. *Neural networks and learning machines*. Pearson Education, 3rd edition.

Hermes, C.J. and Melo, C., 2009. "Assessment of the energy performance of household refrigerators via dynamic simulation". *Applied Thermal Engineering*, Vol. 29, No. 5, pp. 1153 – 1165.

Homod, R.Z., 2013. "Review on the HVAC System Modeling Types and the Shortcomings of Their Application". *Journal of Energy*, Vol. 2013, p. 10.

Li, Z., Wang, B., Li, X., Shi, W., Zhang, S. and Liu, Y., 2017. "Simulation of recombined household multi-split variable refrigerant flow system with split-type air conditioners". *Applied Thermal Engineering*, Vol. 117, pp. 343 – 354.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., 2011. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.

Pérez-Lombard, L., Ortiz, J. and Pout, C., 2008. "A review on buildings energy consumption information". *Energy and Buildings*, Vol. 40, No. 3, pp. 394 – 398. ISSN 0378-7788.

Ploug-Sørensen, L., Fredsted, J.P. and Willatzen, M., 1997. "Improvements in the modeling and simulation of refrigeration systems: Aerospace tools applied to a domestic refrigerator". *HVAC&R Research*, Vol. 3, No. 4, pp. 387–403.

Schmidhuber, J., 2015. "Deep learning in neural networks: An overview". *Neural Networks*, Vol. 61, pp. 85 – 117.

[1] In this case, an increase in number of neurons and/or number of layers results in an increase of model complexity, as more parameters must be adjusted during the training phase.

ISSN 0893-6080.

Tan, P.N., Steinbach, M., Karpatne, A. and Kumar, V., 2018. *Introduction to Data Mining*. Pearson, 2nd edition.

Tieleman, T. and Hinton, G., 2012. "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude". COURSERA: Neural Networks for Machine Learning.

van Rossum, G., 1995. "Python tutorial". Technical Report Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Zeng, Y., Zhang, Z. and Kusiak, A., 2015. "Predictive modeling and optimization of a multi-zone hvac system with data mining and firefly algorithms". *Energy*, Vol. 86, pp. 393 – 402. ISSN 0360-5442.

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.