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

COB-2019-0123
PREDICTION OF ENVIRONMENT PARAMETERS INSIDE A GREENHOUSE USING AN LSTM MODEL

Luísa Castello Branco de Sá
Universidade Federal de Minas Gerais – Undergraduate Program of Mechanical Engineering
Avenida Antônio Carlos, 6627, Pampulha – Belo Horizonte – MG - Brasil
luisacbsa@gmail.com

Thiago Nunes Coelho Cardoso
Hekima – Avenida Álvares Cabral 1315, Lourdes – Belo Horizonte – MG - Brasil
thiago.cardoso@hekima.com

Ricardo Mortara Batistic
BeGreen – Rua Professor Otaviano Almeida 44, Santa Efigênia – Belo Horizonte – MG – Brasil
ricardo@begreen.farm

Antônio Augusto Torres Maia
Universidade Federal de Minas Gerais – Avenida Antônio Carlos, 6627, Pampulha – Belo Horizonte – MG - Brasil
aamaia@ufmg.br

Abstract. *Greenhouses production depends on controlling its environmental conditions so that they are suitable for vegetables growth. In this context, greenhouses temperature and humidity forecast models have been developed to help farmers mitigate production losses, as well as to support the development of optimal climate control strategies. Among these models are the ones based on Artificial Neural Networks (ANNs), which have the ability to capture linear and non-linear relationships of temporal series. The current literature is mostly focused on Multilayer Perceptrons (MLP), although the architecture of Recurrent Neural Networks (RNNs) is specific to handle sequential data and forecast models. Therefore, the present study aims to develop a temperature and humidity forecast model of a greenhouse, with a type of RNN denominated Long Short-Term Memory (LSTM). Environmental data from a commercial greenhouse in Belo Horizonte, Brazil, was collected with six internal and one external meteorological station and used to train the network. Performances of the LSTM model and a baseline multiple linear regression (MLR) model were compared, and the predictions of the LSTM method were better according to smaller values of RMSE and MAPE. On temperature test data, LSTM average RMSE and MAPE scores were respectively 1.232ºC and 3.349%, while MLR average scores were respectively 2.354ºC and 6.759%. On humidity test data, LSTM achieved average RMSE and MAPE of 6.051% and 7.442%, while MLR RMSE and MAPE scores were 10.313% and 12.676%.*

Keywords: *Long Short-Term Memory Networks, Multiple Linear Regression, Greenhouse, Temperature Model, Humidity Model*

1. INTRODUCTION

Greenhouses are indoor environments surrounded by a thin and transparent cover that provide suitable conditions for plant growth. Their complex and dynamic microclimate is a nonlinear, multi-input multi-output (MIMO) system, whose time-varying behaviors are impacted by weather disturbances. Global radiation, external air temperature, humidity, wind speed, and direction have a strong influence on its dynamics (He et al., 2010; Wang et al., 2009).

Classic feedback control approaches provide poor control faced to the disturbances subjecting the greenhouse environment, and model-based feed-forward control represents a more robust alternative (Wang et al., 2009). It depends on a precise environment model, and many methods have been proposed to solve this modeling challenge. Mechanism static and dynamic models offer a physical comprehension of the greenhouse microclimate. They are a function of meteorological conditions, energy balance and heat storage capacities of greenhouse components, i.e., cover, inside air, plant, and soil (Abdel-Ghani et al., 2011; Bot, 1991). Transfer function models have a simple structure and have been applied, although they are suitable only for linear systems (Nielsen et al., 1995). Black-box models are based on input and output data and are applicable for both linear and nonlinear problems (Dariouchy et al., 2009)

Support Vector Machines (SVMs) are based on the statistical learning theory and the principle of structural risk minimization (Haykin, 2009). The temperature prediction model based on a least square support vector machine (LS-SVM) and Support Vector Machine Regression (SVMR) models have already been proposed (Wang et al., 2009; Yu et al., 2016)

Artificial Neural Networks (ANNs) have also exhibited a strong power to predict greenhouses internal parameters, once they have the ability to capture linear and nonlinear relationships of temporal series (Dreyfus et al., 2004). Recent work (Taki et al., 2016) applied Multilayer Perceptron (MLP) and Multiple Linear Regression (MLR) models to predict a greenhouse inside air and roof temperatures, and the MLP model achieved better results according to small values of RMSE and MAPE. Another research (He and Ma, 2010) found that Back Propagation ANNs models could predict the internal humidity of a greenhouse with higher accuracy than a stepwise regression method. MLP and other direct ANNs architectures, such as Radial Basis Function Networks (RBFs), have been widely used in further works (Dariouchy et al., 2009; Ferreira et al., 2002; Taki et al., 2018)

Recurrent Neural Networks (RNNs) were less often used in such greenhouse environment modeling tasks. One of the attempts used an Elman ANN, a kind of ANN that saves the output state of the hidden layer in an extra context layer. It outperformed the experiments with a direct ANN algorithm (Hongkang et al., 2018). A previous work (Fourati, 2014) also built an Elman Network greenhouse temperature and humidity model, which supported the development of a greenhouse climate neural control.

To the best of authors' knowledge, no greenhouse climate models with Long-Short-Term-Memory Recurrent Networks (LSTMs) have been developed, although they have become state-of-the-art networks for a variety of machine learning models. LSTMs provide effective and scalable models to deal with sequential data, capturing long temporal dependencies, and avoiding optimization hurdles that affect simple RNNs (Greff et al., 2017 Hochreiter et al., 2001). Therefore, this work aimed to develop an LSTM model to predict greenhouse temperature and relative humidity. The LSTM model performance was compared to a baseline multiple linear regression (MLR) model, as in previous work presented by Taki et al. (2016).

2. EXPERIMENTAL MEASUREMENTS

2.1 Greenhouse structure

Figure 1 presents the greenhouse structure installed in BeGreen Urban Farm in Belo Horizonte – Brazil, that produces hydroponic lettuce in a 1440 m² area. Its temperature, humidity, and luminosity are regulated with twelve 1CV exhaust fans, a cellulose evaporative panel, and curtains near the ceiling. These actuators are driven by on-off control laws, whose inputs are real-time temperature, humidity, and luminosity collected by six weather stations inside the greenhouse. An additional weather station was installed outside the greenhouse to provide external meteorological data.

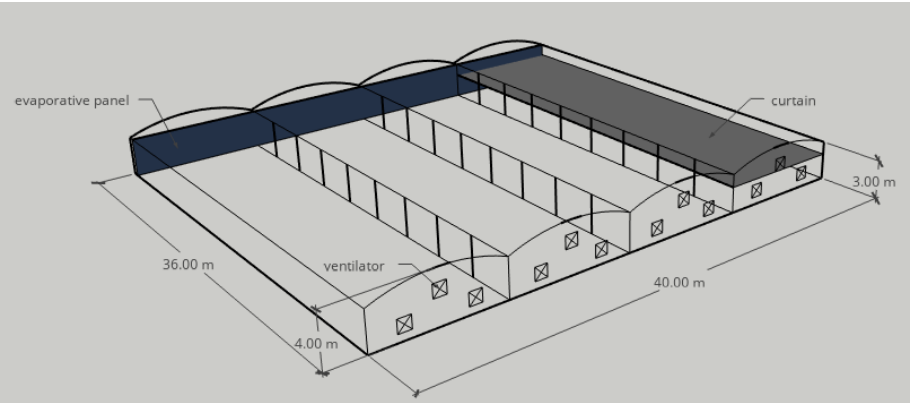


Figure 1. Design picture of the experimental greenhouse

2.2 Climate data acquisition and pre-treatment

Data from the weather stations were used to construct the neural networks models. Each station consists of a BME280 sensor, which collects measurements of temperature and relative humidity, a photoresistor to detect light intensity and an ESP8266 microcontroller that sends environmental data to the greenhouse central computer via JSON. At 25ºC, BME280 exhibits accuracies of ±1.0 ºC for temperature and ±3% for relative humidity. The resistance range of the photoresistor is converted to a range between 0 and 1024 bits, and luminosity is registered with a resolution of 20 bits.

The data set consists of 40-day data points, acquired with a sampling rate of approximately 1 minute, and averaged every 5 minutes. Only measurements from 8 AM to 6 PM were used, once the greenhouse microclimate suffers small changes during the night, and a control strategy improvement is not demanded for this period. Daytime missing values were estimated using linear extrapolation. This approach considers only past observations to calculate the missing values, once an interpolation with future probes is not appropriate for a forecasting model.

Data were normalized over the range of [0, 1], once variables different scales would decrease the neural networks ability to produce meaningful results. The transformation was performed using Eq. (1):

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \times (r_{max} - r_{min}) + r_{min} \tag{1}$$

where x_n are the normalized samples, x is the original data, x_{max} and x_{min} are the maximum and minimums values of the concerned variable and $r_{max}$and r_{min}correspond to the limits of the desired range.

3. THE ARTIFICIAL NEURAL NETWORKS APPROACH

The brain and nervous system behavior inspire ANN technology. It is a computational algorithm that describes several simple and highly interconnected neurons, which process information in parallel and by a dynamic response to their connection to external inputs (Dreyfus et al., 2004; Hsieh and Lu, 2008).

Direct ANNs comprise an input layer, one or more hidden layers, and an output layer. Each neuron calculates a weighted sum of its inputs and takes this output through a non-decreasing and differentiable activation function, as described in Eq. (2):

$$y_j = f\left(\sum_{i=1}^{n} x_i w_{ij} + b_j\right) \tag{2}$$

In which y_j is the output of the j^{th}neuron, $f(\,)$ is the activation function, x_i corresponds to every incoming signal that is weighted by w_{ij}, and b_j is the neuron threshold value.

3.1 LSTM Model

RNNs are a class of ANNs that process time series one step at a time to predict the next point in the sequence. They iteratively feed the output samples as an input to the next step, making a high-dimensional interpolation between the training examples (Graves, 2014). Figure 2 elucidates the difference between direct and recurrent ANNs architectures.

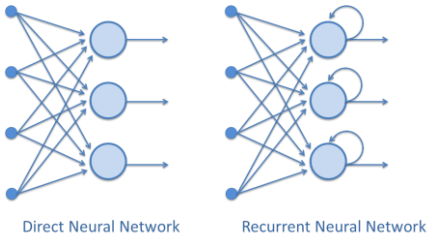


Figure 2. Difference between direct and recurrent ANNs

LSTM is a kind of RNN with an improved architecture for better storing and accessing of information. Its neurons, called memory cells, feature three gates (input, forget and output), a single cell (the Constant Error Carousel), an activation function and peephole connections, as shown in Figure 3. The gates regulate the information flow in and out of the cell (Graves, 2014; Greff et al., 2017).

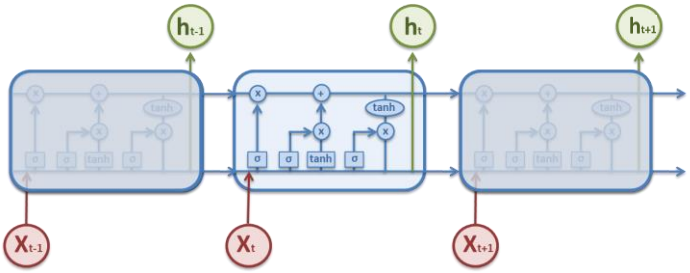


Figure 3. LSTM cell architecture

From the architecture diagram, the relationship between the input and output of LSTM cells is given by the following composite function:

$$\sigma(\varphi) = \frac{1}{1+e^{-\varphi}} \tag{3}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4}$$

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

$$c_t = f_t c_{t-1} + i_t \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{6}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t tanh(c_t) \tag{8}$$

In which σ is the logistic sigmoid function, i_t, f_t, c_t and o_t are respectively the input gate, forget gate, cell state and output gate, h_t is the cell intern state and W and b are the respective weight matrix and threshold values. In Eq. 3, φ represents the terms transformed by the logistic sigmoid function in Eq. 4, 5 and 7.

In this work, the LSTM architecture was built with one input layer, one hidden layer and one output layer. The input layer consists of a 3-dimensional input tensor ([*samples*, *time steps*, *features*]) with samples containing 2-hour past observations of all the computed features. Each time-step corresponds to 5 minutes once data were resampled. The hidden layer contains the recurrent units and received a rectified linear unit (ReLU) activation function at its output. ReLU is one of the most common activation functions nowadays due to its low computational expense:

$$f(x) = max(0, x) \tag{9}$$

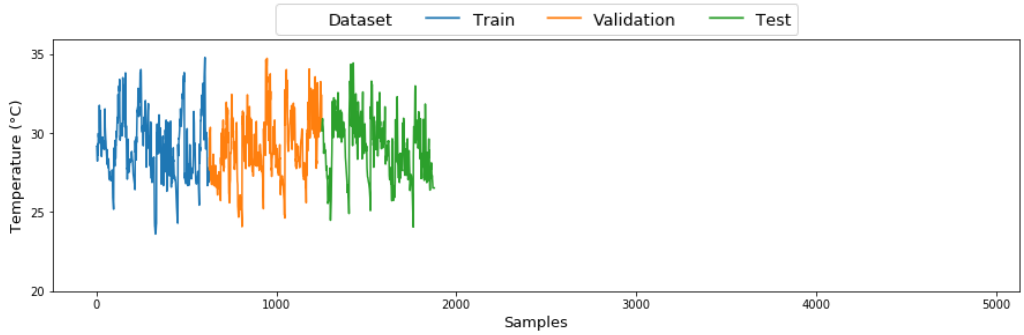
The output layer delivers the target 15-minute temperature and humidity forecasts and has a linear activation, as desired for estimating problems. LSTM model is represented by Eq. (10):

$$[(\mathrm{T}, \mathrm{H})_{t+15}] = \mathrm{LSTM}\begin{bmatrix} [(\mathrm{T_{ext}}), (\mathrm{U_{ext}}), (\mathrm{L_{ext}}), (\mathrm{T_{int}}), (\mathrm{U_{int}}), (\mathrm{L_{int}})]_{t}, \\ [(\mathrm{T_{ext}}), (\mathrm{U_{ext}}), (\mathrm{L_{ext}}), (\mathrm{T_{int}}), (\mathrm{U_{int}}), (\mathrm{L_{int}})]_{t-5}, \\ \dots, \\ [(\mathrm{T_{ext}}), (\mathrm{U_{ext}}), (\mathrm{L_{ext}}), (\mathrm{T_{int}}), (\mathrm{U_{int}}), (\mathrm{L_{int}})]_{t-115}, \\ [(\mathrm{T_{ext}}), (\mathrm{U_{ext}}), (\mathrm{L_{ext}}), (\mathrm{T_{int}}), (\mathrm{U_{int}}), (\mathrm{L_{int}})]_{t-120} \end{bmatrix} \tag{10}$$

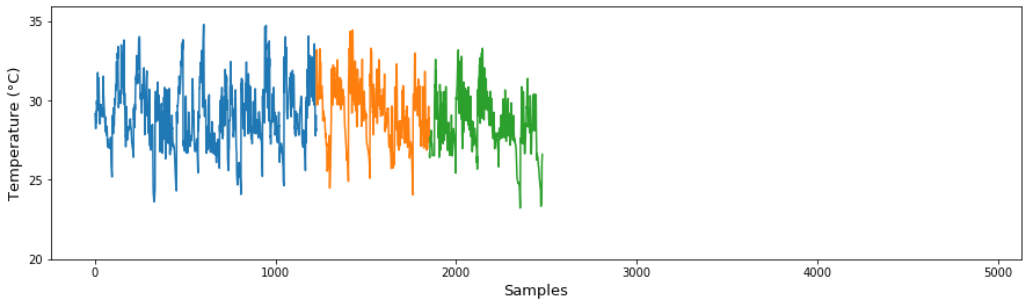
3.2 Model Training

Among the supervised learning algorithms, Back Propagation (BP) has a wide application in complex nonlinear systems identification. The training and learning processes of BP models happen in two steps. The first one, forward-propagation, is the flow of input data from the input layer to the hidden and then output layers. In the second step, back-propagation, error (loss) between expected and actual outputs is computed and fed through the opposite ANN path. Weights between layers are updated, and the process of forward and back-propagation is repeated until the error is decreased to an expected level (He and Ma, 2010).

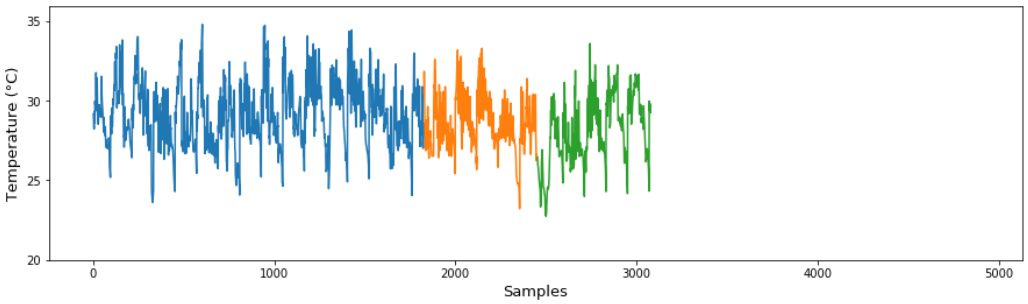
Backtesting was performed to achieve a more robust estimate of the model performance, once traditional N-fold-cross validation would not be appropriate for sequential data. Data were split into five time-windows, and each window was divided into training, validating, and testing sets. The ANN sees and learns from the training set; the validation set provides an unbiased evaluation of the model fit while the ANN hyperparameters are still being tuned and the test set is used to evaluate the final model. Figure 4 is an example of how backtest approach was applied to internal temperature data and graphs *a*, *b*, *c*, *d* and *e* present respectively the first, second, third, fourth and fifth time-window. As shown, the number of training samples increases from one time-window to the next. Validating and testing sets slide along the time-windows, and their sizes were always the same so that models performance statistics were consistent and comparable.



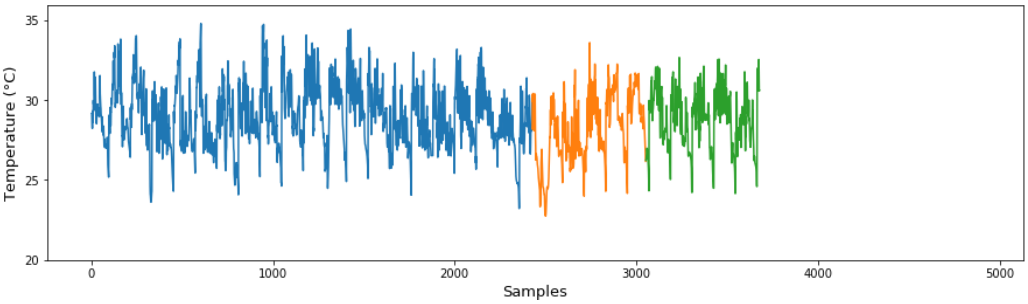
a) First time-window



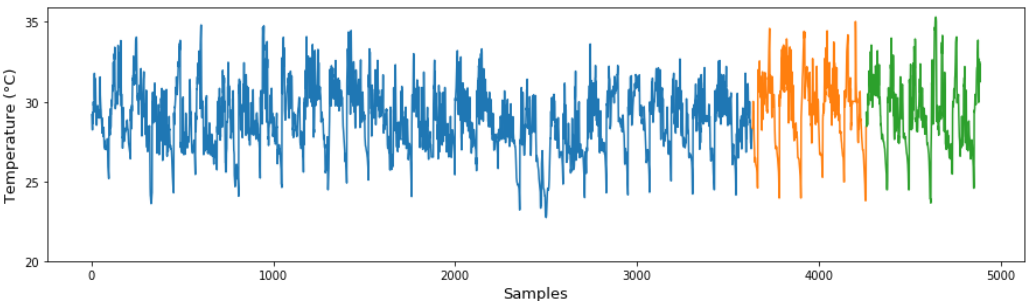
b) Second time-window



c) Third time-window



d) Fourth time-window



e) Fifth time-window

Figure 4. Backtesting with five time-windows

3.3 Performance Evaluation Criteria

Several hyperparameters influence ANNs performance, and it is necessary to search the most suitable combinations between them. Among these hyperparameters are the number of hidden layers, number of units on each layer, type of activation function, number of training epochs, learning rate and batch size (number of training samples seen by the network before its weights are updated). Regularization strategies such as the usage of dropout (random deactivation of some neurons) or the addition of penalty parameters to the model loss function are also important to avoid overfitting.

A hyperparameter grid search was performed, and the developed LSTM model presented the best compromise between training performance and the ability of generalization on validation and test datasets, in accordance to the bias-variance dilemma (Geman, 1992). Its selection required comparing different model configurations errors, calculated with performance criteria defined in the literature. Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) criteria were used in this study:

$$RMSE = \sqrt{\frac{\sum_i^n (d_i - p_i)^2}{n}} \tag{11}$$

$$MAPE = \frac{1}{n}\sum_i^n \frac{|d_i - \hat{p}_i|}{d_i} \tag{12}$$

where n is the number of data, d_i is the desired value, p_i is the predicted value, and $\bar{d}$ and $\bar{p}$ are the respective averages considering the number of variable outputs.

4. RESULTS AND DISCUSSION

Model tuning through hyperparameters grid search found 30 units as an optimal number of neurons on the hidden layer of the LSTM model. L2 regularization was used to impose a penalty to the sizes of the weights and therefore avoid overfitting, giving the final loss function to be minimized:

$$\min J(w) = RMSE + \alpha|w|^2 \tag{13}$$

In which α is the regularization parameter, chosen as 0.01 for the weights of the input connections and 0.0005 for bias and recurrent connections.

Figure 5 demonstrates error versus the number of training epochs of the LSTM model. The loss function was computed for training and validation sets on every learning runs until this loss reached a minimum value after over 2000 epochs. Experiments with larger α values produced smoother error curves. However, the extra regularization effect reduced the model ability to make predictions on daily temperature and humidity peaks.

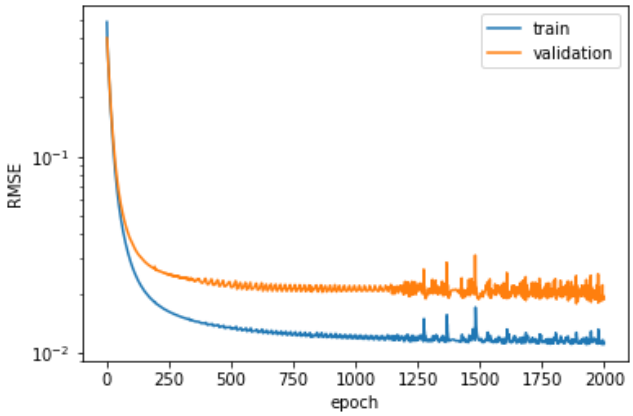


Figure 5. Training and validation error trends vs. number of epochs

RMSE and MAPE scores between LSTM and MLR models predictions and actual temperature and humidity values were computed. LSTM training was repeated five times for each time-window to estimate the mean and standard deviation of errors. The mean between scores of each time-window was also calculated, providing an average expected performance of the designed LSTM and MLR approaches. Table 1 and Table 2 summarize the results.

Table 1. Evaluation of the LSTM and MLR models on temperature training, validation, and test datasets.

Time-Window	Model	LSTM		MLR	
		RMSE	MAPE	RMSE	MAPE
1st	Train	1.420 ± 0.000	3.757 ± 0.003	1.048	3.182
	Val	1.371 ± 0.000	3.724 ± 0.002	4.963	13.069
	Test	1.284 ± 0.000	3.481 ± 0.001	5.717	17.169
2nd	Train	1.933 ± 0.002	5.322 ± 0.014	1.724	5.010
	Val	1.763 ± 0.001	4.845 ± 0.012	2.184	6.124
	Test	1.253 ± 0.001	3.439 ± 0.003	1.495	4.174
3rd	Train	1.710 ± 0.001	4.661 ± 0.012	1.562	4.482
	Val	1.113 ± 0.000	3.132 ± 0.000	1.404	4.022
	Test	1.134 ± 0.000	3.110 ± 0.003	1.687	4.999
4th	Train	1.710 ± 0.000	4.545 ± 0.001	1.550	4.161
	Val	1.352 ± 0.000	3.859 ± 0.004	1.400	4.103
	Test	1.323 ± 0.000	3.564 ± 0.001	1.595	4.135
5th	Train	1.252 ± 0.000	3.495 ± 0.001	1.098	3.065
	Val	1.121 ± 0.000	3.287 ± 0.000	1.145	3.217
	Test	1.165 ± 0.000	3.150 ± 0.000	1.275	3.316
Average	Train	1.605 ± 0.001	4.356 ± 0.00	1.396	3.980
	Val	1.344 ± 0.000	3.769 ± 0.004	2.219	6.215
	Test	1.232 ± 0.000	3.349 ± 0.002	2.354	6.759

Table 2. Evaluation of the LSTM and MLR models on humidity training, validation, and test datasets.

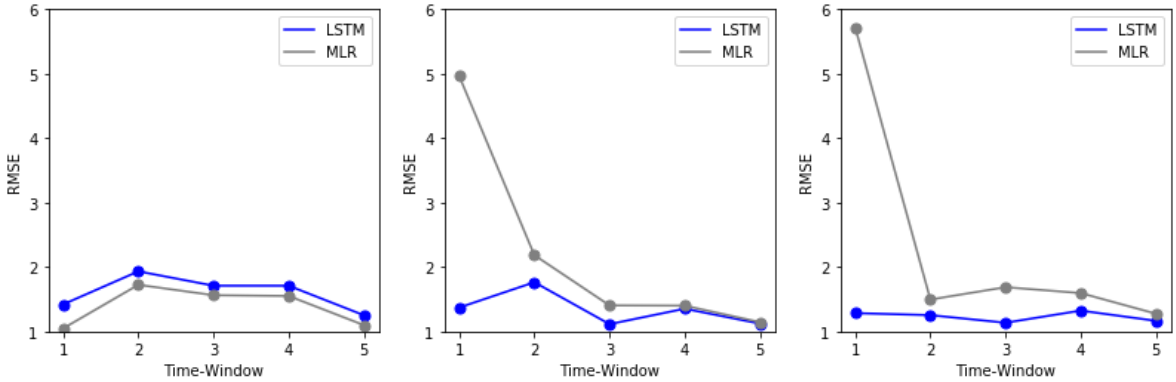
Time-Window	Model	LSTM		MLR	
		RMSE	MAPE	RMSE	MAPE
1st	Train	10.963 ± 0.002	17.576 ± 0.057	9.666	3.069
	Val	7.438 ± 0.004	8.774 ± 0.011	21.681	27.188
	Test	6.502 ± 0.003	7.666 ± 0.005	24.053	30.353
2nd	Train	9.436 ± 0.030	13.433 ± 0.077	8.222	11.885
	Val	6.785 ± 0.002	8.330 ± 0.005	9.877	12.017
	Test	5.721 ± 0.005	6.832 ± 0.012	7.757	8.828
3rd	Train	6.687 ± 0.036	8.561 ± 0.113	5.514	6.991
	Val	5.920 ± 0.069	6.834 ± 0.101	8.702	10.215
	Test	7.245 ± 0.026	9.576 ± 0.096	8.110	10.285
4th	Train	7.048 ± 0.000	9.136 ± 0.002	6.097	7.845
	Val	8.442 ± 0.123	11.640 ± 0.174	6.896	9.019
	Test	5.332 ± 0.002	6.515 ± 0.004	5.960	7.403
5th	Train	6.212 ± 0.003	7.793 ± 0.009	5.287	6.512
	Val	4.963 ± 0.005	5.795 ± 0.006	6.258	7.248
	Test	5.457 ± 0.005	6.622 ± 0.014	5.683	6.513
Average	Train	8.069 ± 0.018	11.300 ± 0.052	6.957	7.260
	Val	6.709 ± 0.040	8.274 ± 0.059	10.683	13.138
	Test	6.051 ±0.008	7.442 ± 0.026	10.313	12.676

Figure 6 and Figure 7 help elucidate the results presenting the RMSE scores of the LSTM and MLR models along the time-windows. Graphs *a*, *b* and *c* of Figure 7, related to humidity forecasts, gives a clearer view of the expected tendency – once models performance depend on the amount of samples from which they learn, errors were smaller for time-windows with a larger training set.

Predictions on training data were less accurate than on validation and test sets for the LSTM model and also for some time-windows of the MLR. For instance, LSTM average temperature RMSE and MAPE were respectively 1.605ºC and 4.356% for training data, 1.344ºC and 3.769% for validating data and 1.232ºC and 3.349% for testing data. This behavior is related to challenges of environmental data acquisition at the beginning of the temperature and

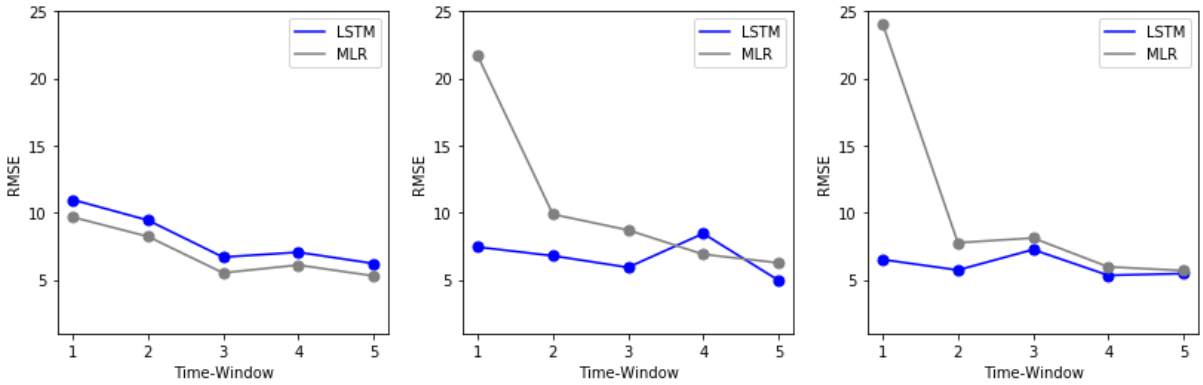
humidity monitoring period. Internet connection problems at the greenhouse resulted in missing data points that needed to be estimated through linear extrapolation. The lack of input from the weather stations also had a negative impact on the on-off control performance, partially disturbing the weather conditions inside the greenhouse.

Another evidence of this challenge is that the MLR model presented considerably bigger validation and test errors on the first and second time-windows. In the first time-window, the most critical one, temperature RMSE were respectively 4.963ºC and 5.717º for validation and test sets, and humidity RMSE were respectively 21.681% and 24.053%. These results show that the MLR was not able to make reasonable predictions after learning from data with poorer quality. On the other hand, the LSTM model was able to keep smaller RMSE scores (1.371ºC and 1.284ºC for training and validating temperature sets, 7.438% and 6.502% for training and validating humidity sets) on the same time-windows despite the training data quality.



a) Train b) Validation c) Test

Figure 6. Temperature error trends vs. time-windows



a) Train b) Validation c) Test

Figure 7. Humidity error trends vs. time-windows

The LSTM model had a worse average performance than the MLR technique on training data, according to higher RMSE and MAPE scores and which can be seen from Figure 6, graph *a* and figure 7, graph *a*. This is not a bad result considering that the LSTM model performed better than the MRL on validation and test datasets, achieving lower RMSE and MAPE values on almost every time-windows. On temperature test data, LSTM average RMSE and MAPE scores were respectively 1.232ºC and 3.349%, while MLR average scores were respectively 2.354ºC and 6.759%. On humidity test data, LSTM achieved average RMSE and MAPE of respectively 6.051% and 7.442%, while MLR scores were respectively 10.313% and 12.676%. That is, the LSTM model ability to make more accurate predictions on data not seen during the training stage is the measure of its higher effectiveness and generalization power.

Figure 8 and Figure 9 show LSTM predictions on the fifth time-window test set, showing that the model was able to capture daily temperature and humidity patterns.

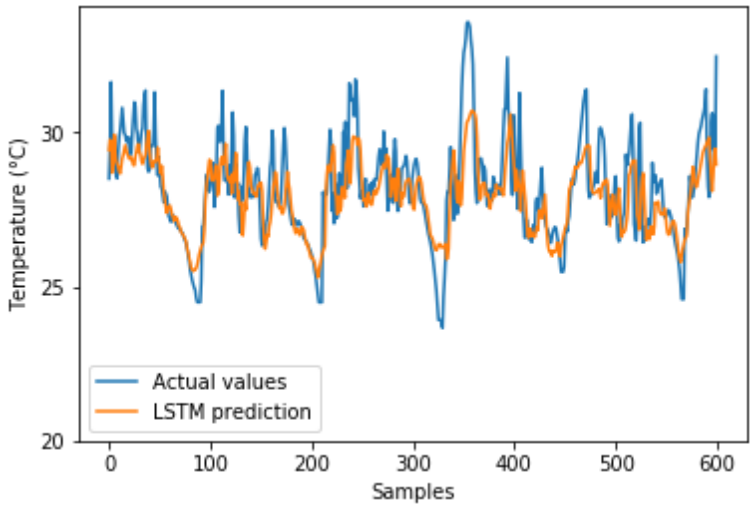


Figure 8. Comparison between real temperature and the prediction by the LSTM network

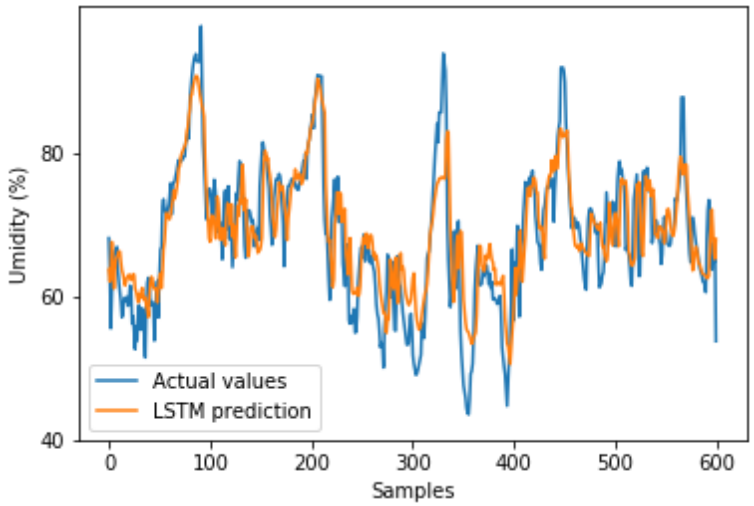


Figure 9. Comparison between real relative humidity and the prediction by the LSTM network

LSTM technique outperformed the traditional MLR method. Further tuning of the presented LTSM model with temperature and humidity data from subsequent time-windows may increase its prediction ability, making it suitable for the development of a feed-forward control strategy.

5. CONCLUSION

In this work, a temperature and humidity forecast model of a greenhouse located in Belo Horizonte, Brazil was developed using LSTM Neural Networks. Inside and outside meteorological stations collected environmental measurements used as input for the LSTM predictions. Backtesting with five time-windows was implemented to calculate the model average prediction performance, and a comparison was made with a baseline MLR. The LSTM approach achieved better temperature and humidity predictions, according to smaller values of RMSE and MAPE. On temperature test data, LSTM average RMSE and MAPE scores were respectively 1.232ºC and 3.349%, while MLR average scores were respectively 2.354ºC and 6.759%. On humidity test data, LSTM achieved average RMSE and MAPE of 6.051% and 7.442%, while MLR RMSE and MAPE scores were 10.313% and 12.676%. Further tuning of the model with data collected subsequently may improve its prediction ability, and it can be used on the development of a feed-forward controller for the greenhouse.

6. ACKNOWLEDGEMENTS

The authors would like to thank CNPq (Conselho Nacional de Desenvolvimento Cientifico e Tecnologico) for the support obtained during the period of elaboration of this work.

7. REFERENCES

Abdel-Ghani, A. M. and Al Helal, I. M., 2011. "Solar energy utilization by a greenhouse: general relations". *Renewable Energy*, Vol. 36, pp. 189-196.

Bot, G. P. A., 1991. "Physical modelling of greenhouse climate". *Proceedings of the IFAC/ISHS Workshop*, pp. 7-12.

Dariouchy, A. Assif, E., Lekouch, K. Bouirden, L. and Maze, G., 2018. "Prediction of the intern parameters tomato greenhouse in a semi-arid area using a time-series model of artificial neural networks". *Measurement*, Vol. 42, pp. 456-463.

Dreyfus G., Martinez, J.M., Samuelides, M., Gorden, M.B., Badran, F., Thiria, S. and Hérault, L., 2004. *Réseaux de neurones méthodologie et applications*. 2nd edition.

Ferreira, P.M., Faria, E.A. and Ruano, A.E., 2002. "Neural network models in greenhouse air temperature prediction". *Neurocomputing*, Vol. 43, pp. 51-75.

Fourati, F, 2014. "Multiple neural control of a greenhouse". *Neurocomputing*, Vol. 139, pp .138-144.

Geman, S, Bienenstock, E., Doursat and R., 1992. "Neural networks and the bias/variance dilemma". *Neural Computation*, Vol. 4, pp. 1-58.

Graves, A., 2014. "Generating Sequences With Recurrent Neural Networks". In arXiv:1308.0850v5 [cs.NE]

Greff, K., Srivastava, R.K, Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2017. "*LSTM: A Search Space Odyssey*". In arXiv:1503.04069v2 [cs.NE]

Haykin, S., 2009. "Neural networks and learning machines. *Prentice Hall*. 3rd edition.

He, F. and Ma, C., 2010. "Modeling greenhouse air humidity by means of artificial neural network and principal component analysis". *Computers And Electronics In Agriculture*, Vol. 71, pp. 19-23.

Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J., 2001. "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". In n S. C. Kremer and J. F. Kolen , editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press.

Hsieh, K.L., Lu, Y.S., 2008. "Model construction and parameter effect for TFT-LCD process based on yield analysis by using ANNs and stepwise regression". *Expert Systems with Applications*, Vol. 34, pp. 717-724.

Hongkang, W., Li, L., Yong, W., Fanjia, M., Haihua, W. and Sigrimis, N.A., 2018. " Recurrent Neural Network Model for Prediction of Microclimate in Solar Greenhouse". *Ifac-papersonline*, Vol. 51, pp. 790-795.

Nielsen, B. and Madsen, H., 1995. "Identification of transfer functions for control of greenhouse air temperature". *Journal Of Agricultural Engineering*, Vol. 60, pp. 25-34.

Taki, M., Ajabshirchi, Y., Ranjbar, S.F., Rohani, A. and Matloobi, M., 2016. "Heat transfer and MLP neural network models to predict inside environment variables and energy lost in a semi-solar greenhouse". *Energy And Buildings*, Vol. 110, pp. 314-329.

Taki, M., Mehdizadeh, S.A, Rohani, A., Rahnama, M. and Joneidabad, M.R. 2018. "Applied machine learning in greenhouse simulation; new application and analysis". *Information Processing In Agriculture*, Vol. 5, pp. 253-268.

Wang D., Wang M. and Qiao X., 2009. "Support vector machines regression and modelling of greenhouse environment". *Computers and Electronics in Agriculture*, Vol. 66 pp. 46–52

Yu, H., Chen, Y. and Gul H. S., 2016. "Prediction of the temperature in a Chinese solar greenhouse based on LS-SVM optimized by improved PSO". *Computers and Electronics in Agriculture*, Vol. 122, pp. 94-102.

Wang D., Wang M. and Qiao X., 2009. "Support vector machines regression and modelling of greenhouse environment". Computers and Electronics in Agriculture, Vol. 66 pp. 46–52

8. RESPONSIBILITY NOTICE

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