In Giannetti, B.F.; Almeida, C.M.V.B.; Agostinho, F. (editors): Advances in Cleaner Production, Proceedings of the 12h International Workshop, Stellenbosch, South Africa. November 23nd and 24nd, 2023

Deep Learning-Enabled Temperature Simulation of a Greenhouse Tunnel

JOGUNOLA, O.[A], HULL, K.J.[B], MABITSELA, M. M.[B], PHIRI, E.E [B], ADEBISI, B.[A], BOOYSEN, M.J.[B*]

a. Manchester Metropolitan University, United Kingdom

b. Stellenbosch University, South Africa

**Corresponding author, mjbooysen@sun.ac.za*

Abstract

Agriculture is poised to suffer greatly from the effects of climate change. Prediction models, using deep learning, have been developed that can simulate and predict conditions in open field farming to combat the climate variability from climate change. However, deep learning used in precision agriculture, specifically greenhouse tunnels, is under-researched despite also being affected by this variability. Utilising tunnel data collected over 42 days, two hybrid deep learning models were designed. Specifically, a hybrid of convolutional neural network (CNN) and Long Short-Term Memory (LSTM), and a hybrid of CNN and Bidirectional LSTM (BLSTM). The models are designed to forecast the internal temperature of the tunnel to support its management. The cooling wet wall state, solar irradiance, inside and outside temperature of the tunnel are input variables to the developed deep-learning models. Two scenarios are discussed with the results, the first scenario includes all the external variables as input, while the second scenario only considers the internal temperature as input. Results show a performance improvement of 48% and 14% computation time for the CNN-LSTM compared to the CNN-BLSTM model for the two scenarios, respectively. In terms of the measured loss metrics, both models had varied performance and model fitness, with an average mean square error of 0.025 across the models and scenarios.

Keywords: Internet of Things, Deep Learning, Precision Agriculture, Climate Change, Thermal Modelling

1. Introduction

Due to climate change, farmers have adapted their growing practices to accommodate sudden changes in rainfall and temperature patterns that can directly affect their growing cycles. This change, however, has come at the expense of productivity and, potentially, yield (Malhi et al., 2021; Yang et al., 2022). At a global scale, any climate variability is a risk to food security (Fujimori et al., 2022). Although this variability is unique to each country, Africa as a continent is poised to suffer the most as crop irrigation is largely dependent on rainfall (Tadese et al., 2022). As the effects of climate change are generally focused on a global scale, Weber et al., 2018 presented the potential effects of climate change and global warming in an African context. Weber et al. showed that in the Western Cape (in South Africa), a 2°C increase in a global temperature average would lead to 9% fewer rainy days, 35 more hot nights (temperatures in the 90th percentile of minimum temperatures experienced at night), and nine more days considered to be heat waves per year. This combined with a variability in climate could be devastating for farmers.

To combat this variability, prediction algorithms using deep learning can be used. According to (Lecun et al., 2015), deep learning is a layered approach to machine learning, with each layer being a higher abstraction level with new features and information to be found. Extending this knowledge of deep learning, Emmert-Streib et al. (2020) then describes the ability of using deep learning algorithms for predictions and forecasting. For example, using Long Short-Term Memory (LSTM) for predictions uses this layered approach, with the output of the final layer being the predicted value. The main benefit of using an LSTM is the feedback connections to previous layers it creates which helps the algorithm remember important information within the applied data.

Although there are many applications for deep learning for prediction (Emmert-Streib et al., 2020), most are aimed at image or video processing. For agriculture, however, the use of deep

learning has lagged behind. Zhao et al., 2021, however, derive a novel deep learning algorithm for temperature prediction specifically for agricultural purposes, specifically extreme weather forecasting for farmers. Their model, a combination of a Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU), and a Relative Position-based Self Attention Mechanism (RPASM), provided accurate results compared to their independent counterparts. They were able to predict 24-hours ahead to a mean absolute error (MAE) of 1.95°C.

Despite this being used in open field farming, precision agriculture cannot benefit from this advancement. Deep learning models specifically predicting conditions within greenhouses with minimal input data are under-researched. In this paper, a deep learning method for temperature simulations in a greenhouse tunnel is proposed to address the variability of external conditions to African greenhouses and understand how this affects conditions inside these greenhouses.

1.1 Related Works

Climate forecasting techniques can assist farmers in preparing their crop for specific seasonal conditions. However, due to climate change causing variability in these conditions, forecasting techniques trained on historical data are leading to inaccurate models. Han et al., 2019 developed a tool that aims to avoid these inaccuracies by combining historical data with probabilistic models to not only predict the climate, but also the probability of its variability. This tool was developed specifically for Uruguay but can easily be adapted for any climate. Machine learning models, mainly regressor models, were investigated by Nyasulu et al., 2022 to determine the best model for different climate (minimum and maximum temperature, relative humidity, and rainfall) predictions. They used an ensemble learning algorithm that trained and tested multiple models sequentially to determine the model best for each climatic parameter. The CatBoost Regressor, Gradient Boosting Regressor and Light Gradient Boosting machine learning algorithms were stacked and used to forecast the climatic parameters as a sliding window of one-time step ahead. Although useful, both previous examples (Han et al., 2019; Nyasulu et al., 2022) are particularly focused on external conditions. PA systems, as the one that is studied in this paper, can generally form part of protected agriculture where the crop is covered by plastic tunnels (Shi et al., 2019).

For this paper, related work focussing on greenhouse tunnel temperature forecasting or simulation are more valuable as a baseline. Petrakis et al., 2022 developed a neural network model for temperature forecasting in a greenhouse, in particular a multilayer perceptron neural network (MLP-NN). Their input feature selection included wind speed, outside temperature, solar irradiance, outside relative humidity, inside relative humidity (at different previous timesteps), and inside temperature (at different previous timesteps). For their temperature prediction, they achieved a mean absolute error (MAE), root mean squared error (RMSE), and R^2 of 0.218°C, 0.271°C, and 0.999 respectively. Although the results appear accurate, each simulation iteration did not use previous predicted results, but measured previous results, thereby aligning the model with the measured results after each timestep. Codeluppi et al., 2020 used artificial neural networks (ANN) in a similar manner to Petrakis et al., however, the model is executed on an edge device in the greenhouse in real-time. Further, instead of arranging their dataset as a time series, the authors use only the model for prediction which eliminates errors that can occur if data is corrupted or missing. Along with their ability to predict other input features, the authors can predict and forecast any temperature in time. Their results are more indicative of a true forecasting model with an RMSE, MAPE, and R^2 of 1.50°C, 4.91%, and 0.965 respectively. The study in this paper hopes to fill the gap of temperature forecasting methodologies in a South African context and extend the use of deep learning in agriculture for improved temperature simulations in a greenhouse tunnel.

2. Method

2.1 Experimental Setup

This study occurred in a closed tunnel at the Welgevallen Experimental Farm (Stellenbosch University, South Africa) and data was captured between 18 November and 30 December 2022 (42 days). The tunnel is dome-shaped and is 28 meters in length, 9 meters in width, and 3 meters

in height.

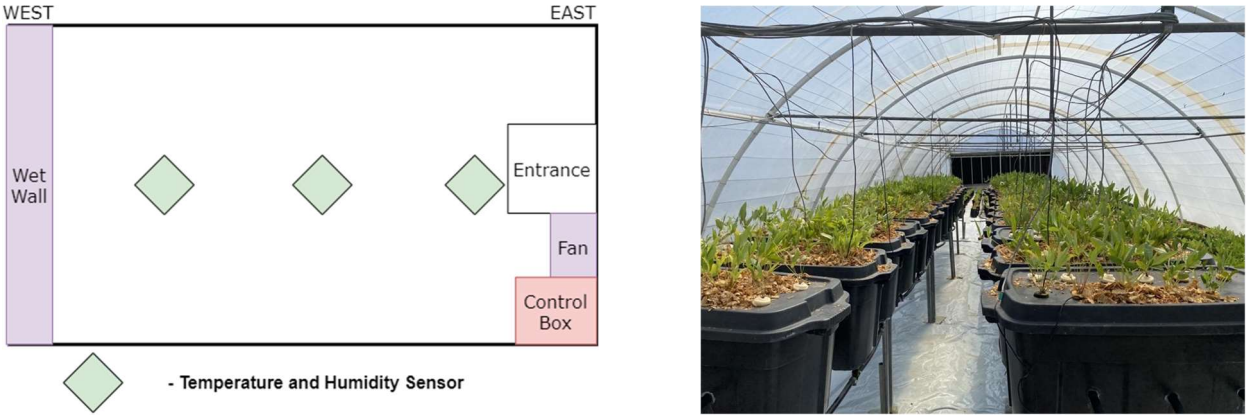


Fig. 1 A system diagram of the tunnel and the actual tunnel. A.) Sensors are placed equidistant apart beginning at the entrance of the tunnel; B) Containers occupy the first

Three Sensors were placed twelve meters apart, starting from the entrance of the tunnel to the wet wall on the opposite end. The fan occupies approximately one-third of the wall space next to the entrance of the tunnel and a control box housing the electronics is next to the distribution board (DB) (Fig. 1A). The system is designed to capture sensor data in three sections of the 28-meter-long tunnel: the front, the middle, and the back (Fig. 1B). The DHT22 temperature and humidity sensor was chosen for its low-power capabilities and low price and is compatible with most microcontrollers. It has the added benefit of simultaneous temperature (-40°C to 80°C) and humidity (0% to 100%) sensing. The sensor's accuracy is 0.5°C for temperature and 2 to 5% for humidity, which was acceptable for the purposes of the study. A Raspberry Pi Model 3B was chosen for sensor data aggregation, internet connectivity, CSV (Comma Separated Value) file formulation, and multiple General Purpose Input Output (GPIO) pins for future expansions and interfacing with control mechanisms. Each sensor captured humidity and temperature reading every minute for five minutes. After five minutes, an average was calculated for each parameter and stored in a CSV file on the Raspberry Pi. Further, the middle sensor was chosen for fan and wet wall control, as the front and back sensors were biased by both apparatuses respectively. Ten readings were made every minute and the median of those readings was chosen to control the fan and wet wall. Along with the environmental readings, the time that the reading was recorded, and a binary value (0 or 1) was stored representing whether the fan and wet wall were turned on during that period. At the end of each day, the CSV file was uploaded to Dropbox™ (Dropbox Inc., 2023) for remote access to the day's data. Furthermore, each sensor's temperature and humidity and the fan and wet wall's state were broadcasted to a Telegram™ (Telegram FZ LLC. & Telegram Messenger Inc., n.d.) channel for real-time monitoring.

2.2 Deep Learning Modelling

The deep learning algorithms used for the forecasting of the internal temperature of the tunnel are described in this subsection. Each of the deep learning algorithms are first explained, followed by the developed hybrid model for the tunnel temperature forecasting.

2.3.1 Convolutional Neural Network (CNN)

Data feature extraction is a preliminary step that ensures the reduction of the parameters needed for forecasting, this reduces the network computation and orchestrates the prediction accuracy. Convolutional Neural Network (CNN) is mostly skilful at extracting complex features in a dataset while storing varied irregular trends. CNN has hidden layers consisting of a pooling layer, a convolutional layer, and an activation function. The convolutional layer is the first input layer that converts the input data into features map to be sampled by the pooling layer in a bid to further reduce its dimensionality.

Given an input vector ${x_1}^m = \{x_1, x_2, \ldots, x_n\}$, , where $x^m, m \in M$ is the input vectors, including the solar irradiance, outside and inside temperatures, and the wet wall state (fan off or on). n is the normalised half-hourly unit per window of observation. Feeding these input variables into the first convolutional layer of the CNN framework will result in the output expressed in (*1*).

$$ {y_{ij}}^{m(k)} = \sigma({b_j}^{m(k)} + \sum_{m=1}^{M} {w_{m,j}}^{m(k)} \; {x^0}_{i+m-1,j}) \quad (1) $$

Where σ is the activation function, ${b_j}^{m(k)}$ is the bias for the j^{th} feature map and k^{th} convolutional layer, w is the weight of the kernel for the CNN framework. This output is fed into the next layer of the CNN, the pooling layer to downsample the activation from the feature maps, thereby reducing the parameters and the computation costs. The output of this layer is expressed in (*2*).

$$ {P_{ij}}^{m(k)} = \max_{r \in R} {y^{k-1}}_{i \times T + rj} \quad (2) $$

Where T is the stride to decide input data length. The output ${P_{ij}}^{m(k)}$ is fed to the input of the next deep learning architecture in the hybrid model.

2.3.2 Long Short Term Memory (LSTM)

LSTM was proposed by Hochreiter and Schmidhuber, 1997 to address the vanishing gradient of recurrent neural networks (RNN) to preserve long-term dependencies. LSTM is skilful in learning temporal dependencies in a sequence of information. Utilisation of memory cells and gates like the input, forget and output gates in LSTM addresses the RNN vanishing gradient challenge. In LSTM, the input gate preserves the input data, forget gate determines the unused data, memory cells store the processing states, and the output gate delivers the LSTM operation output. This process is expressed in the following equations.

$$ i_t = \sigma(W_i, [h_{t-1,} x_t] + b_i \quad (3) $$
$$ f_t = \sigma(W_f, [h_{t-1,} x_t] + b_f \quad (4) $$
$$ o_t = \sigma(W_o, [h_{t-1,} x_t] + b_o \quad (5) $$
$$ \bar{C}_t = tanh(W_c, [h_{t-1,} x_t] + b_c \quad (6) $$
$$ C_t = f_t \times C_{t-1} + i_t \times \bar{C}_t \quad (7) $$
$$ h_t = o + t \times tanh(C_t) \quad (8) $$

In the equations, i_t, f_t , and o_t represent the input, forget and output gates respectively, while W_i , W_f , W_o are their respective weights. W_c is the weight for the memory cell. The bias of each gates are b_i , b_f , b_o , b_c . Other variables are, x_t , the input vector at time t ; h_t , the hidden state; $\bar{C}_t$, C_{t-1} , and C_t , the candidate memory, previous cell state and new cell state, respectively.

2.3.3 Bidirectional Long Short Term Memory (BLSTM)

Utilisation of the cells and gates in the LSTM enables it to address the vanishing gradient of RNN. However, LSTM only considers the previous state of information and losses the next state information. To address this challenge, BLSTM combines information in both forward and backward directions. Architecture of BLSTM is similar to LSTM gates and cells represented in (3) through (5), but in both directions. The hidden state, cell state and the output of the BLSTM concatenate for both direction are expressed in (9), (10) and (11), respectively.

$$ c_t = f_t . c_{t-1} + i_t \, . \, \sigma(W_i, [h_{t-1,} x_t] + b_i \quad (9) $$
$$ h_t = o_t . \sigma(c_t) \quad (10) $$
$$ \bar{y} = \sigma(W_y h_{t,} + b_y) \quad (11) $$

2.3.4 The Architecture of the Hybrid Models

In forecasting the internal temperature of the tunnel to support its management, two hybrid learning models are developed, comprising of a CNN-LSTM architecture and a CNN-BLSTM architecture. The CNN-LSTM and CNN-BLSTM models are presented in Table 1 and Table 2,

respectively. The output of the CNN layer is fed into the input of the LSTM and/or the BLSTM input gates. The CNN extracts important features from the dataset and the LSTM and BLSTM layers are for information analysis and sequence predictions.

The hyperparameter values, detailing the number of neurons, parameters and the layers for the models are derived following an extensive experimentation of varied values to achieve the optimal performance for our requirements.

Table 1 CNN-LSTM and its definition

No	Layer Type	Neurons	Param
1	Input	8	8
2	Convolution1D	64	1 600
3	Convolution1D	64	12 352
4	MaxPooling1D	64	0
5	Time Distributed (Dense)	64	0
6	LSTM	100	66 000
7	Dense	100	10 100
8	Dense	7	707

Table 2 CNN-Bidirectional LSTM and its definition

No	*Layer Type*	*Neurons*	*Param*
1	*Input*	*8*	*8*
2	*Convolution1D*	*64*	*1 600*
3	*Convolution1D*	*64*	*12 352*
4	*MaxPooling1D*	*64*	*0*
5	*Bidirectional*	*128*	*66 048*
6	*Dense*	*100*	*12 900*
7	*Dense*	*7*	*707*

2.3.5 Evaluation Metrics

To evaluate the deep learning models for the internal temperature of tunnel forecasting, several error metrics are utilised. Such as, mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and the computation time, which includes the training and testing time of each model. The MSE, expressed in (12), represents the mean of the squares of the difference of the predicted and the actual values.

$$MSE = 1/n \sum_{1}^{n} (y - \bar{y})^2 \quad (12)$$

where y is the observed value over the number of n observed dataset, and $\bar{y}$ is the predicted output. Similarly, the RMSE, the root mean square error and MAE, the measure of the absolute difference of the actual and predicted values, are expressed in (13) and (14), respectively.

$$RMSE = \sqrt{1/n \sum_{1}^{n} (y - \bar{y})^2} \quad (13)$$

$$MAE = 1/n \sum_{1}^{n} |y - \bar{y}| \quad (14)$$

In Giannetti, B.F.; Almeida, C.M.V.B.; Agostinho, F. (editors): Advances in Cleaner Production, Proceedings of the 12h International Workshop, Stellenbosch, South Africa. November 23nd and 24nd, 2023

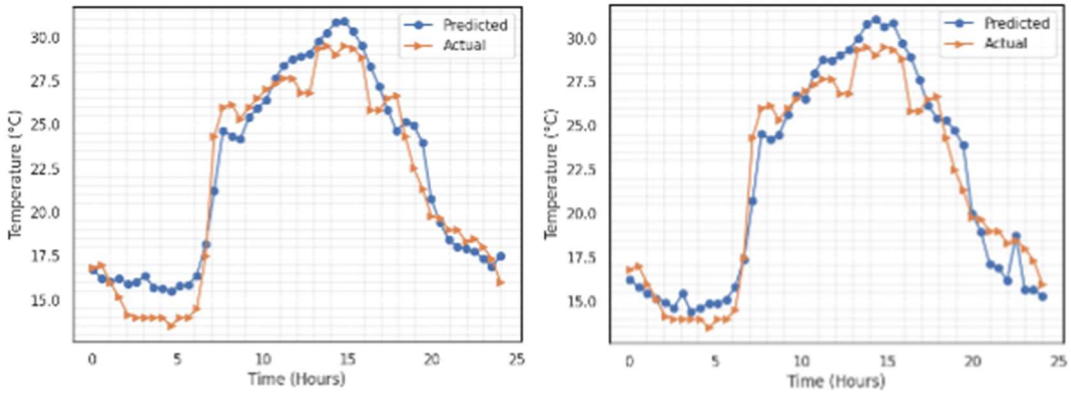
3 Results

3.1 Simulation environment

All computations were developed on Google Colaboratory (https://colab.research.google.com) using an Apple M1 Pro Laptop with 32GB Memory. After extensive experiments, and in addition to the hyperparameter values in Tables 1 and 2, the system optimisation parameters selected were a learning rate of 0.01, 80 epoch, 160 batch-size, 0.33 validation split, and a ReLU activation function.

3.2 Model Comparison

To evaluate the two models, two scenarios were created. The first scenario integrated all the input variables, the solar irradiance, internal temperature, outside temperature and the fan and wet wall state. The second scenario only uses the internal temperature as an input. This scenario is specifically relevant when all contributing factors to a particular data of interest are not available. Fig. 3 illustrates the output from the CNN-BLSTM model for the two scenarios.



A) B)

Fig.3 The 24-hour actual vs. predicted internal temperature output. A) CNN-BLSTM utilising all the input variables; B) CNN-BLSTM utilising only the internal temperature variable.

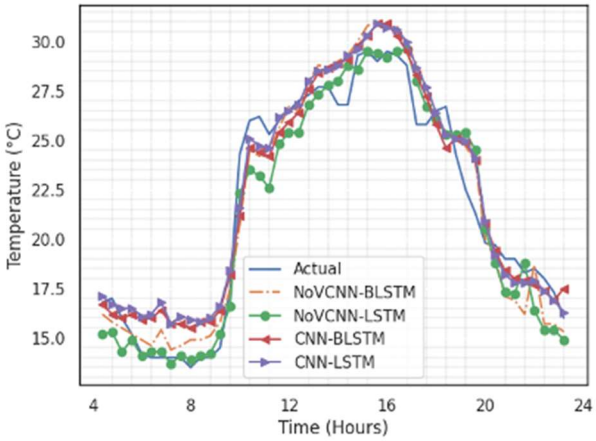


Fig.4 The 24-hour actual vs. predicted internal temperature output for the two hybrid models for the two scenarios. CNN-BLSTM and CNN-LSTM are the two models with all external variables.

NoVCNN-BLSTM and NoVCNN-LSTM are the models with only the internal temperature as input.

Surprisingly, in the 0:00am and 6:00am period, the model with the internal temperature achieves a greater predicted accuracy, while between 6:00am and 23:59pm the case with all the external variables achieves better accuracy. Fig. 4 illustrates the predicted and actual summary of the two

models and the two scenarios models. This result illustrates that the predicted values for all models are very narrow with a good fitting.

3.3 Model Evaluation Metrics

An example representation of the training and validation loss function of the tunnel data is shown in Fig. 5. This is illustrated for the MSE and RMSE of the CNN-LSTM model for the second scenario. To verify the effectiveness of the deep learning models. a comparative summary of the MSE, RMSE and MAE for the two models under the two scenarios are shown in Fig. 6.

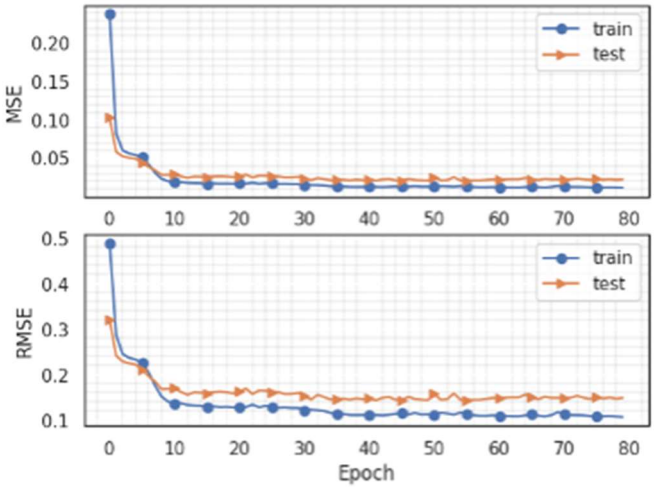


Fig.5 The training and validation loss during training for the CNN-LSTM model utilising the internal temperature as an input.

Interestingly, the performance of the models for the two scenarios varies. For MSE, the CNN-LSTM utilising all the external variables achieves the lowest MSE but a high MAE. While the CNN-LSTM model for the second scenario archives the lowest RMSE and MAE.

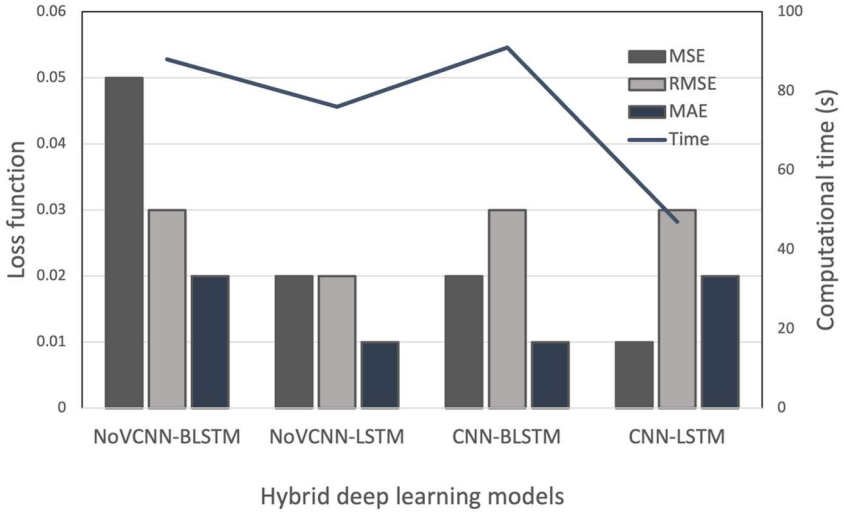


Fig.6 The performance comparison of the evaluation metrics for the two hybrid models. The left axis is the 'Column' Loss function, while the right axis is the 'Line' Computation time (s).

Comparing the computation time for the case of all the external variables, as expected the CNN-LSTM model has lower computation time compared to the CNN-BLSTM model. This is also similar for the second scenario. Specifically, the CNN-LSTM achieves a 48.0% performance increase to CNN-BLSTM for the first scenario and a 13.6% for the second scenario. The computation performance is particularly of interest in memory constrained IoT devices in tunnel modelling.

4 Conclusion

This work developed two hybrid deep learning models, a CNN-LSTM and a CNN-BLSTM to forecast

the internal temperature of a greenhouse tunnel utilising the tunnel data collected over 42 days, The effectiveness of the models were analysed based on error metrics, MSE, RMSE and MAE, as well as their computation time. Results show performance improvement of 48% and 13.6% computation time for the CNN-LSTM compared to the CNN-BLSTM model for the two scenarios created, respectively. Also, an average mean square error of 0.025 was observed across the models and scenarios. Through this research, two deep learning algorithms have been developed that will directly benefit precision agriculture in an effort to combat climate variability. This will allow better understanding of the effects of external conditions on internal variables in a greenhouse tunnel and will allow for better decision making in crop production in the future.

5 References

Codeluppi, G., Cilfone, A., Davoli, L., & Ferrari, G. (2020). AI at the Edge: a Smart Gateway for Greenhouse Air Temperature Forecasting. 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), 348–353. https://doi.org/10.1109/MetroAgriFor50201.2020.9277553

Dropbox Inc. (2023). Dropbox (179.4.4985.). Dropbox, Inc. https://www.dropbox.com/

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., & Dehmer, M. (2020). An Introductory Review of Deep Learning for Prediction Models With Big Data. Frontiers in Artificial Intelligence, 3, 507091. https://doi.org/10.3389/FRAI.2020.00004/BIBTEX

Fujimori, S., Wu, W., Doelman, J., Frank, S., Hristov, J., Kyle, P., Sands, R., Van Zeist, W.-J., Havlik, P., Dom\'\inguez, I. P., & others. (2022). Land-based climate change mitigation measures can affect agricultural markets and food security. Nature Food, 3(2), 110–121.

Han, E., Baethgen, W. E., Ines, A. V. M., Mer, F., Souza, J. S., Berterretche, M., Atunez, G., & Barreira, C. (2019). SIMAGRI: An agro-climate decision support tool. Computers and Electronics in Agriculture, 161, 241–251. https://doi.org/10.1016/J.COMPAG.2018.06.034

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature 2015 521:7553, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation 9, 8 (1997), 1735–178.

Malhi, G. S., Kaur, M., & Kaushik, P. (2021). Impact of climate change on agriculture and its mitigation strategies: A review. Sustainability, 13(3), 1318.

Nyasulu, C., Diattara, A., Traore, A., Deme, A., & Ba, C. (2022). Towards Resilient Agriculture to Hostile Climate Change in the Sahel Region: A Case Study of Machine Learning-Based Weather Prediction in Senegal. Agriculture, 12(9), 1473.

Petrakis, T., Kavga, A., Thomopoulos, V., & Argiriou, A. A. (2022). Neural Network Model for Greenhouse Microclimate Predictions. Agriculture, 12(6). https://doi.org/10.3390/agriculture12060780

Shi, X., An, X., Zhao, Q., Liu, H., Xia, L., Sun, X., & Guo, Y. (2019). State-of-the-art internet of things in protected agriculture. Sensors, 19(8), 1833.

Tadese, M. T., Kumar, L., Koech, R., & Kogo, B. K. (2022). Perception of the impacts of climate and environmental variability on water availability, irrigation and farming systems: a study in rural households of Awash River Basin, Ethiopia. International Journal of Agricultural Sustainability, 20(2), 231–246.

Telegram FZ LLC., & Telegram Messenger Inc. (n.d.). Telegram (v4.8.1). Retrieved 7 August 2023, from https://telegram.org/apps

Weber, T., Haensler, A., Rechid, D., Pfeifer, S., Eggert, B., & Jacob, D. (2018). Analyzing regional climate change in Africa in a 1.5, 2, and 3 C global warming world. Earth's Future, 6(4), 643–655.

Yang, H., He, J., Su, Y., & Xu, J. (2022). Adaptation to climate change: ethnic groups in Southwest China. Environmental Hazards, 21(2), 117–136.

Zhao, X., Huang, L., & Nie, Y. (2021). Temperature Prediction Based on Integrated Deep Learning and Attention Mechanism. 2021 IEEE 6th International Conference on Computer and Communication Systems, ICCCS 2021, 163–167. https://doi.org/10.1109/ICCCS52626.2021.9449176