

Document downloaded from:

<http://hdl.handle.net/10251/170274>

This paper must be cited as:

Guillén-Navarro, MA.; Martínez-España, R.; Llanes, A.; Bueno-Crespo, A.; Cecilia-Canales, JM. (2020). A deep learning model to predict lower temperatures in agriculture. *Journal of Ambient Intelligence and Smart Environments*. 12(1):21-34. <https://doi.org/10.3233/AIS-200546>



The final publication is available at
<https://doi.org/10.3233/AIS-200546>

Copyright IOS Press

Additional Information

A deep learning model to predict lower temperatures in agriculture

Miguel A. Guillén-Navarro^a, Raquel Martínez-España^{a,*}, Antonio Llanes^a, Andrés Bueno-Crespo^a and José M. Cecilia^a

^a *Computer Science Department, Universidad Católica de Murcia, Murcia, Spain*

E-mails: maguillen@ucam.edu, rmartinez@ucam.edu, allanes@ucam.edu, abueno@ucam.edu, jmcecilia@ucam.edu

Abstract. Deep learning techniques provide a novel framework for prediction and classification in decision-making procedures that are widely applied in different fields. Precision agriculture is one of these fields where the use of decision-making technologies provides better production with better costs and a greater benefit for farmers. This paper develops an intelligent framework based on a deep learning model for early prediction of crop frost to help farmers activate anti-frost techniques to save the crop. This model is based on a long short-term memory (LSTM) model and it is designed to predict low temperatures. The model is based on information from an IoT infrastructure deployed on two plots in Murcia (Southeast of Spain). Three experiments are performed; a cross validation to validate the model from the most pessimistic point of view, a validation of 24 consecutive hours of temperatures, in order to know 24 hours before the possible temperature drop and a comparison with two traditional time series prediction techniques, namely Auto Regressive Integrated Moving Average and the Gaussian process. The results obtained are satisfactory, being better the results of the LSTM, obtaining an average quadratic error of less than a Celsius degree and a determination coefficient R^2 greater than 0.95.

Keywords: Deep Learning, LSTM, Precision agriculture, IoT

1. Introduction

Smart agriculture is extrapolating and applying the concepts, techniques and systems of Industry 4.0 to the agrarian world. With the incorporation of the advantages offered by new technologies, smart agriculture builds approaches to reorganize the entire agricultural system towards sustainable, high-efficiency, low-input agriculture [1, 2]. This new approach benefits mainly from the emergence and convergence of new technologies, including the global positioning system, the geographic information system, sensors, automatic control, remote sensing, mobile computing, advanced information processing and telecommunications. All these technologies unified under the paradigm of Cloud Computing and the Internet of Things (IoT) enable the efficient deployment of new

technologies to deal with several issues that affect agriculture [3–5]. Therefore, smart agriculture provides a real benefit to society in different ways, such as facilitating the stabilization and increase of agricultural production and improving the environmental impact of this activity [6]. Some of the activities involved within precision agriculture are the disease detection [7, 8], the prediction of weather conditions [9, 10], yield prediction [11, 12], water saving through irrigation monitoring [13, 14], just to mention a few.

In this study we focus on the paradigm of smart agriculture by dealing with the problem of predicting weather conditions. In particular, we focus on avoiding frost in crops in extensive cultivation. Low temperatures at certain times of the agricultural cycle are a major problem that can result in the loss of millions of

*Corresponding author. E-mail: rmartinez@ucam.edu.

euros ¹. This can be avoided by predicting in advance a significant drop in temperature in a certain area to activate antifreeze techniques such as the connection of windmills and stoves or the connection of heating in a greenhouse. However, if no action is taken early enough, the crop can be lost completely, leading to great economic losses for the farmer.

The issue of frost prevention is not simple, as it depends on several factors, such as temperature, humidity, wind speed, etc., but also on the location of the plot. The global weather forecast provides coarse-grained information that is not sufficient to predict frost at the plot level. Fine grain hydrometeorological information from the particular area or plot should be taken into account to create more accurate models. The different geographical conditions in which the plantations can be found within the same plot determine the climatic conditions of the crops. In order to monitor the climatic conditions of the crops automatically, an IoT system is used to measure the wind speed, temperature and humidity of a given area within a plot [15]. However, the IoT system needs an intelligent component that helps farmers take decisions to prevent damage to crops from low temperatures. This article proposes a deep learning model based on time series to predict the possibility of frost in crops, taking as input data those provided by an IoT system. Thus, a temperature prediction approach proposed allows the farmer to be able to know the possibility of a drop in temperatures and thus will be able to activate and/or prepare all the necessary resources to apply the anti-frost technique. The latter will be possible thanks to the integration of the proposed approach to predict temperature within the IoT system, as well as helping the farmer to make decisions this system can be programmed to automatically activate the anti-frost techniques.

Deep Learning represents a set of machine learning algorithms based on a set of artificial neural networks composed of complex hierarchical levels [16]. Deep learning models are beginning to be used in the world of agriculture to solve complex problems such as the classification of diseases and/or plants through images or yield predictions in crops [17]. In this study a type of recurrent neural network is proposed to create the temperature prediction model, specifically Long short-term memory (LSTM) neural network is used. This type of neural networks obtain very satisfactory results

when the data have a temporal tendency, as is the case of the temperature data of a plot [18]. Therefore, the main objective of this work is to perform a preliminary analysis and design of an LSTM neural network to create a temperature prediction model to be integrated into an IoT system deployed in several agricultural plots. To achieve the best model, we compare local temperature models with a global model containing information from several plots to determine which is more accurate.

This study is organized as follow. In section 2 a brief background review on aspects related to deep learning and precision agriculture is presented. Section 3 describes the data and techniques used for this study of temperature prediction. Finally, Section 4 presents the results and an analysis of them and Section 5 presents the conclusions and future work.

2. Background

Deep learning techniques have begun to be introduced in the field of precision agriculture to help complete and realize the challenges presented by agriculture [17].

Among the fields of application deep learning has been applied to find the classification of plant species, identification of plant diseases, identification of soil cover, classification of crop type, estimation of yields, identification of weeds, predictions on climatology, etc.

For each of these fields of application, different types of deep learning techniques are used such as convolutional neural networks (CNN), which are deep neural networks traditionally used to classify images. For instance, a new approach for detecting plant diseases is described in [19] using a trained CNN and adjusted to fit accurately into the database of the leaves of a plant that was collected independently for various plant diseases. Another work where a CNN is used is presented in [20]. The authors introduce a new approach to classify and recognize the health status of the plantation and immediately generate treatment solutions on the fly. They use a CNN to classify and recognize different kinds of plant images, detect plant diseases, and determine the growth rate of plantations. The authors of [21] propose a novel approach that uses deep learning to count the fruits of a tree. An initial CNN quickly tags large spot-based datasets. Then, a counting algorithm based on a second CNN estimates the number of fruits in each tree. Finally, a linear re-

¹<https://www.laverdad.es/murcia/ultimas-heladas-region-20190404113101-nt.html>

gression model verifies that the estimated number of fruits coincides with the actual number of fruits. In [22] a CNN is proposed to create a system to obtain a high quality classification of field vegetation in valuable crops and weeds. Specifically this system has been applied to sugar beet fields, seeking to accurately identify weeds in the field.

The LSTM are neural networks are applied to problems where there is a dependence of temporality in the data. In the case of agriculture they are applied to multiple areas. Some examples are described below. In [23], the authors design a new LSTM model as an alternative to computationally expensive physical models to predict the long-term depth of groundwater in agricultural areas. The proposed model consists of a LSTM layer with another layer completely connected above it, with a drop method applied to the first LSTM layer. In this study, the proposed model was applied and evaluated in five sub-areas of the Hetao Irrigation District in the arid northwest of China using 14-year data. In [24] an accurate prediction model of wheat production for agriculture in Pakistan is presented. The model uses a data pre-processing smoothing mechanism, together with an LSTM-based model, to further improve the accuracy of the predictions. In [25], the authors design a high-precision identification model for haploid maize seeds from diploid seeds by applying optimal data from hyperspectral images selected by the combination of an algorithm based on LSTM and CNN.

Focusing on the problem of the prediction of weather conditions, several works have designed LSTM-based models to predict the climatic variables using different climatic variables as input, and they have shown positive results. In [26], the construction of a robust statistical model is proposed for predicting meteorological visibility based on other intermediate variables (temperature, pressure, humidity and dew point). Two single-layer and four-layer LSTM networks are used. The data have been preprocessed by means of normalization, rescaling to the range [0, 1] and using a moving average. The multilayered LSTM model proves to be the most effective. Another work that uses an LSTM to predict climate variables is presented in [27]. The variables used are temperature, humidity and wind speed. In this case the network architecture consists of two LSTM layers. The activation function chosen for the output of the dense layer is the RELU. The optimizer used is RMSProp. The data have been normalized and rescaled to the range [-1, 1] and the results obtained have been satisfactory. In [28], the authors

intend to model rain and runoff using LSTM network, which predict discharge for a variety of watersheds. The authors aim to demonstrate the potential of this method. Some variables used are day length, rainfall, temperature or humidity. The network is composed of 2 LSTM layers and between them a Dropout layer to avoid overtraining of the network. The difference between the existing techniques and the proposal made in this work is that here we try to predict the temperature value in order to be able to incorporate an intelligent component into an IoT system and also that only the temperature value is used.

Analyzing conventional machine learning techniques that predict climatic variables, in [29] a extreme learning machine technique is used to predict daily dew point temperature. This study proposes an algorithm based on Extreme-Machine learning and compares it against machine-learning techniques to those that it surpasses in quality. Similar results were to be expected, as they are comparing against algorithms that do not maintain temporality in data such as SVN (Support Vector Machine) and ANN (Artificial Neural Networks). In [30], a study is presented where they use a distributed algorithm based on MapReduce to predict temperature and precipitation variables using a linear regression-based algorithm. Other related paper is [31], where authors estimate and map daily mean air temperature using daytime and nighttime land surface temperatures. In this paper authors use a linear regression to estimate mean temperature, we differ with the authors that they estimate an average value of the day, however we work with values taken every ten minutes to forecast 24 hours in a row. In comparison Deep Learning techniques provide better performance when dealing with temporal data, obtaining models with a better fit and more satisfactory results.

3. Material and method

3.1. Data Collecting

The data used to train and validate the model are real data obtained from a IoT system that has deployed 3 nodes in the towns of Cieza and Moratalla (Region of Murcia, Spain). A map with the location of both cities is shown in Figure 1. The distance between the two cities is approximately 50 km. The goal is to analyze whether a global model is better than individual local models for each area.

The deployed IoT system consists of several nodes located in different areas of several plots with sensors of temperature, humidity and wind speed. A sample of each row collected is shown in table 1. Each day will have 144 rows of values, 1 value every 10 minutes, 6 measurements per hour, during 24 hours. We can collect the *date* in dd/mm/aaaa format, *hour* in hh:mm format, and values obtained by sensors for temperature *Celsius degrees*, wind speed (m/s) and relative air humidity(%). In this work, the temperature values are used to analyse the behaviour of the IoT system considering the need for a temperature sensor, in order to reduce costs for farmers.

date (Cieza)	Temperature	Wind speed	Humidity
14/11/2018 7:10	7.76	4.21	3.25

Table 1

The format of each row collected every 10 minutes.

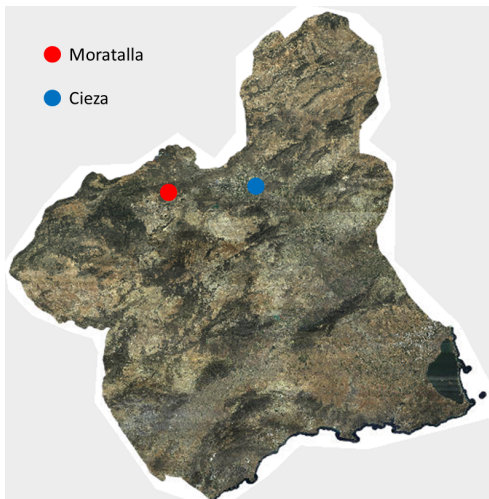


Fig. 1. Map of the region of Murcia with the location of the cities of Cieza and Moratalla.

The IoT system, employed to collect data, consists on a 4H Demeter control unit and a wind speed, humidity and temperature sensor, configured as follows: Demeter 4H is a remote management equipment designed and implemented by Hidroconta². It is a modular equipment and adaptable to the most of installations that is capable of transmitting information through different communication interfaces. These can be with other Demeter through LoRa or with a server in the cloud through GPRS. One of the most outstand-

²<https://www.hidroconta.com/>

ing features is that it can work uninterruptedly for 4 months without communication and without loss of information. Being a totally autonomous equipment that is powered by batteries, which are recharged by a small solar panel. Different sensors can be connected to the equipment through its digital input or through its two analogicals (0-20/4-20 mA of 10 bits of resolution). This connectivity can be extended with expansion cards. Demeter 4H is operated by a micro-controller that has 256 KB of storage for the firmware developed by the same company. It also has 96 KB of volatile memory for program data. It also has a non-volatile external memory with 244 KB for storing history and configuration. Enough to store more than 20,000 records. Having 144 records per day in our work, give a total of the 4 months mentioned above. A picture of this equipment is shown in 2. Power consumption: 126uA in low power mode (without communications), 42uA additional for each expansion and 19 mA with GPRS connection.

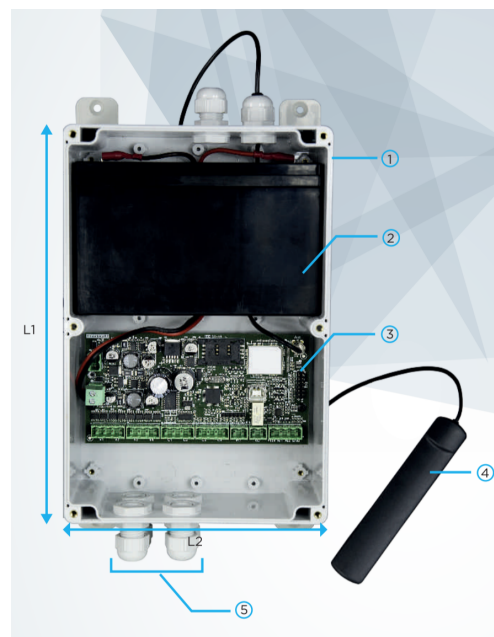


Fig. 2. Demeter 4H by Hidroconta. 1. Outer case. 2. Battery 3. Main Board 4. Antenna 5. Closures 6. L1: length (26 cm) 7. L2: width (21 cm) 8. Depth: 10,5 cm

The temperature and humidity sensor is a low-cost transmitter of ambient temperature and relative humidity of the Comet brand, specifically the model P3110E. Among its most outstanding features are: Relative humidity range: 0-100%. Accuracy of relative humidity measurement: $\pm 3\%$. Accuracy of temperature output:

$\pm 0.6^{\circ}\text{C}$. Temperature range: -30 to 80°C . The transmitter contains a microprocessor based control circuit included in a plastic case with connection terminals and sensors in a stainless steel mesh filter. The humidity transmitters are equipped with two isolated 4-20 mA outputs. Speed wind sensor is the model PCE-WS. This sensor has a range between 3 and 180 Km/h with an error of $\pm 1\text{Km/h}$.

The IoT nodes use Lora technology to communicate and send the data via GPRS to a data visualization application. In addition, this data is preprocessed to avoid the errors that can be proved by the sensors. The errors have been eliminated using the Kmeans algorithm for outlier detection and correction presented in [32]. The data used correspond to the period from 1/11/2018 to 28/02/2019, having temperature values (measured in degrees Celsius [$^{\circ}\text{C}$]) with a frequency of 10 minutes Table 2 shows the number of instances of the datasets used for the creation and validation of the LSTM model. The datasets ‘‘All’’ is formed by the Cieza and Moratalla data in order to analyze the global behavior of the model using both locations. The difference between the number of instances of Cieza and Moratalla within the same period is due to problems with the information collection instruments, as sometimes the data were not recorded correctly. This deficiency was later corrected by the introduction of an outliers correction technique [15].

Datasets	Cieza	Moratalla	All
N.Instances	16739	17070	33809

Table 2

Description of the datasets

3.2. Deep Learning LSTM

LSTM networks were introduced by [33] as a model of Recurrent Neural Networks (RNN) series capable of learning long-term dependencies [33, 34]. While the RNNs have a structure of chains of neural networks, the LSTM follow a similar structure, however, each part of the chain, instead of being a layer of neural network, are multiple layers, which interact between them. LSTM networks overcome the previously inherent problems and memorize temporal patterns over a long period of time. This is a problem that it presents the RNN, since it works very well with short-term dependencies, however, if RNN have to recognize a long-term dependency, they are not as efficient. These drawbacks, presented by RNN, are overcome by LSTM us-

ing memory cells and door units [33]. The door units regulate the information that can be added or removed from the memory cells. A memory cell with its different door units is shown in Figure 3. In this figure, it can be seen that a LSTM blocks contains four layers of neural networks that interact with each other, through operations of additions, concatenations and multiplications. These four neuronal layers are made up of three sigmoids (σ) and a hyperbolic tangent (\tanh). First sigmoid layer is in charge of deciding which information will be thrown away from the cell state, which is a way of forgetting unnecessary information. The next sigmoid, is responsible for deciding what information is to be stored in the cell state. To do this, it also uses a \tanh layer. Finally, the last sigmoid decides which parts of the cell state will be passed to the output. In this way, and thanks to this combination of neural networks layers, it makes the LSTM block decide what information it will store in the state of the cell.

This combination of neural networks makes the LSTM block decide what information it will store in the state of the cell.

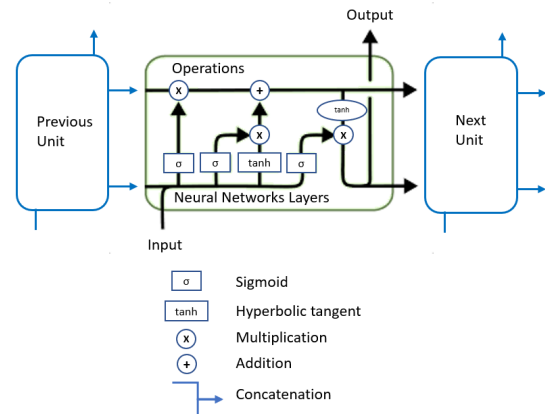


Fig. 3. A memory cell for LSTM Model. Unlike RNNs, LSTMs use different door units to recognize long-term dependency.

This makes the LSTMs very efficient in univariate time series models for forecasting problems. These types of problems are based on learning from the series of past observations to predict the next value in the sequence. Therefore, LSTMs are designed to remember information for long periods of time. It has been verified that these networks are especially useful in solving problems based on learning sequences of past observations, with the aim of predicting their next value [34]. The LSTM used for this study is designed with TensorFlow and Keras, and a detailed explanation of the

input parameters used for the experiments are shown in the next section.

3.3. Parameters configuration

In this section parameters configuration is described, showing different tests made over all the parameters of the algorithm. This tests are made to discriminate the best parameters in the LSTM algorithm. The specific values of the different parameters are shown in table 3, in section 3.4, and here we show several tests previously run that justify the value selection for those parameters. In next lines, we will show the experiments for:

- Optimizer - figure 4
- Learning Rate - figure 5
- Delay Sequence - figure 6
- Activation Function - figure 7
- Number of Neurons - figure 8
- Batch Size - figure 9
- Loss Function - figure 10

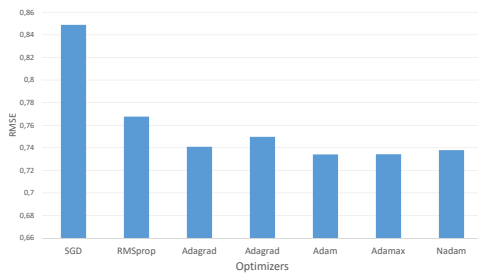


Fig. 4. Selecting the best LSTM Optimizer parameter.

In figure 4, the results obtained with different optimizers are shown. The optimizer with more accuracy was Adam (*Adaptive Moment Estimation*) optimizer, so it was the optimizer selected. This optimizer is just an extension to stochastic gradient descent and is being used as benchmarks in deep learning problems, i.e. Kelvin xu et. al. [35] and Karol Gregor et. al. [36]. Other optimizers studied are:

- SGD. Stochastic gradient descent, this is the classic algorithm used in deep learning.
- RMSProp. Root Mean Square Propagation, this algorithm does usually well on non-stationary problems.

- Adagrad. Adaptive Gradient Algorithm, that usually fits better in problems like computer vision or natural language.
- Adadelata. Adadelata seeks to reduce AdaGrad's aggressiveness by monotonously slowing down the rate of learning.
- Adamax. Adamax is a variant of Adam, but in this case is based on the infinity norm.
- Nadam. Nesterov Adam optimizer. Basically, this method is Adam RMSProp with Nesterov momentum.

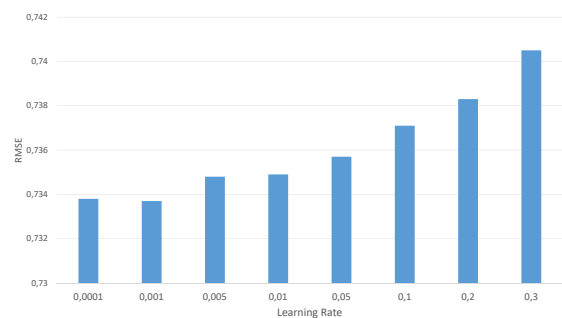


Fig. 5. Selecting the best LSTM Learning Rate parameter.

In figure 5, Learning Rate is studied. This is a factor to control the learning variation adjusting the weights of the model. Very low values of learning rate make the learning slow, taking more time to converge. In the model was selected the learning rate with value of 0.001.

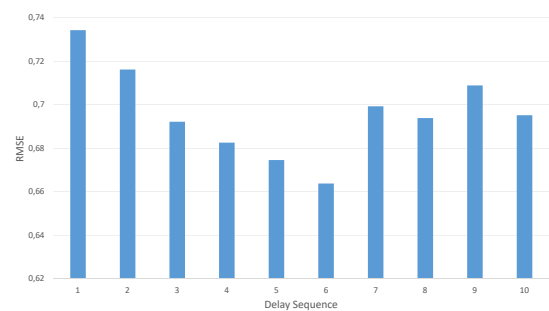


Fig. 6. Selecting the best LSTM Delay Sequence parameter.

In figure 6, we study the number of temporal series stored in each sequence will allow the LSTM to remember more or fewer series. This hyperparameter is in problems predicting temporal series, not being com-

mon in the rest of the areas. The algorithm is configured to remember 6 delay sequences.

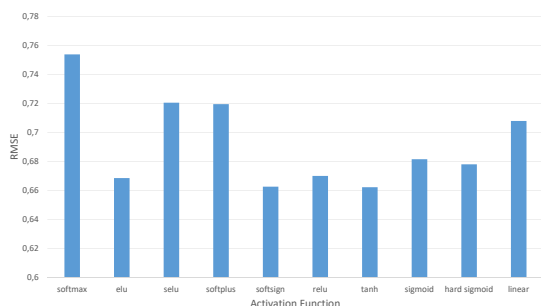


Fig. 7. Selecting the best LSTM Activation Function parameter.

In figure 7 we show different activation function studied, in neurons, weighted sums are calculated from inputs passing through activation functions where non-linear deformations are introduced. In this way, the computation of several neurons can be effectively chained. The hyperbolic tangent *tanh*, is selected as activation function in our model.

In figure 8 tests performed to find the optimal number of neurons that the LSTM network should have are shown. The number of neurons is established at 100 for the application.

Training data can be partitioned into small batches to train the network. Less memory is required when training the network with fewer samples and, generally, the batch size is inversely proportional to the number of times needed to train the network, since the weights of the neurons are updated after each forward-propagation. As figure 9 show, different values have been tested and a size of 32 is chosen for the algorithm.

After the forwardpropagation phase, an error is calculated in order to measure the effectiveness of the prediction according to the result to be expected. There are several functions to calculate the error and to be able to adjust the weights of the neurons in the back-propagation phase. After analysing the most relevant ones in the field of linear regression, results shown in figure 10 are obtained. The loss function of the quadratic mean error is chosen.

Finally, epochs is the last parameter studied. It indicates the number of times the training data are passed through the neural network, i.e., the iterations performed during the training process. If the number of times is very large, the model can be overtrained (over-

fitting), as it learns very specific characteristics of the training set without being able to generalize well. It is also possible not to train the model too much (underfitting), without converging on a point. 1000 Epochs is selected to our model. As we mentioned at the beginning of this section, the summarized values of these parameters are listed in the 3.4, in table 3.

3.4. Experiment configuration

The objective of this work, as already indicated, is to carry out a temperature prediction model, analysing whether the best option is a global model for the two areas or a local model for each of them. To evaluate the models created by the LSTM, we will follow a conservative methodology. First we will perform a 3-cross validation for each of the indicated datasets. With this evaluation we intend to analyze the performance of the models from the point of view of robustness, without analyzing the type of error produced [37]. This first evaluation will indicate the quality and robustness of the models created, both local and global (using the “All” dataset) considering time data series. Then a second experiment is performed, specifically 90% of the data fro train models and the remaining 10% has been used as a test, choosing whole days of 24 hours. In this second experiment we analyze not only the robustness and prediction efficiency of the models but also the type of error made by the model. It must be taken into account that the same as for training the models, for the test the prediction of the temperature values is made in degree Celsius and every 10 minutes. Therefore the farmer will have updated information and predictions every 10 minutes. Finally a comparison with two traditional techniques, ARIMA and the Gaussian process is carried out, to validate and check the bias of LSTM with other methods of time series prediction.

The results of this experiment are shown and discussed in the section 4.2. The two experiments are executed for the 3 datasets previously described. The optimal parameters are shown in the Table 3. In order to achieve the final configuration of the network, an empirical experimentation of the different parameters involved has been carried out. A brief summary of this parameterization study is shown in Appendix 3.3.

The quality evaluation of the model proposed is performed by measuring the goodness of the prediction by the following metrics:

- the Root Mean Square Error (RMSE)
- the Mean Absolute Error (MAE)

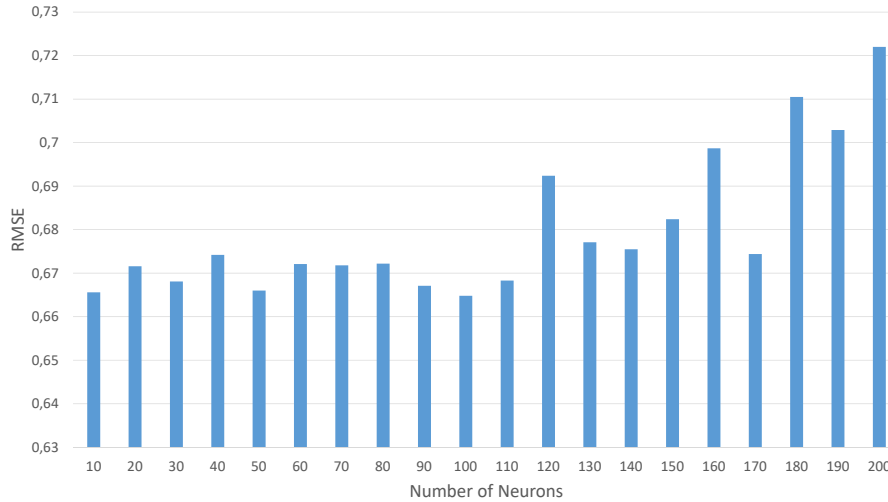


Fig. 8. Selecting the best LSTM Number of Neurons parameter.

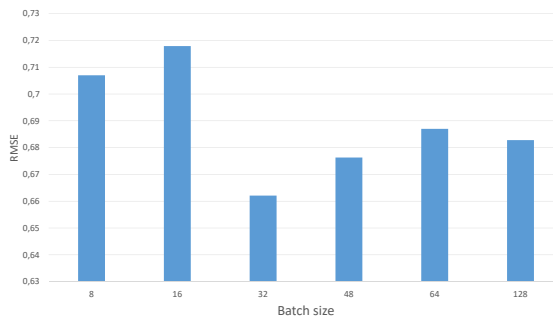


Fig. 9. Selecting the best LSTM Batch Size parameter.

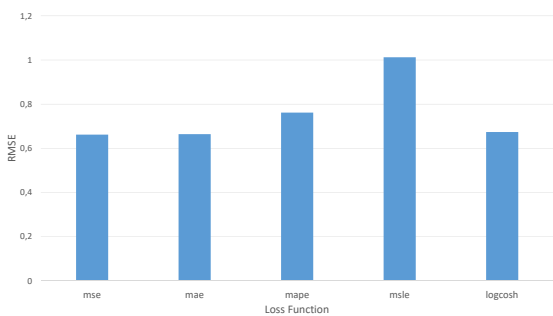


Fig. 10. Selecting the best LSTM Loss Function parameter.

- the Pearson Correlation Coefficient (PCC)
- Determination coefficient (R^2)

Parameter	Value
Number of input neurons	100
Batch size	32
Number of epochs	1000
Learning factor	0.001
Optimizer	Adam
Activation function	hyperbolic tangent
Loss Function	quadratic mean error
Delay Sequence	6

Table 3

Optimal parameters for LSTM execution experiments

To predict temperature, we perform a classification task to evaluate false positives and false negatives. A prediction where the real temperature is negative and the prediction is positive (false positive) is much more serious than the opposite, where the prediction indicates that there is no risk of frost and, in reality, it freezes, being able to lose the harvest (false negative). In what follows, false positives are categorized into Error type 1 ($Error_1$) and false negatives are categorized as Error type 2 ($Error_2$).

Experiments have been carried out in a GPU-based platform. This platform is composed of:

- 2 hexa-core Intel Xeon E5-2650 at 2.20 GHz.
- 128 GB of RAM.

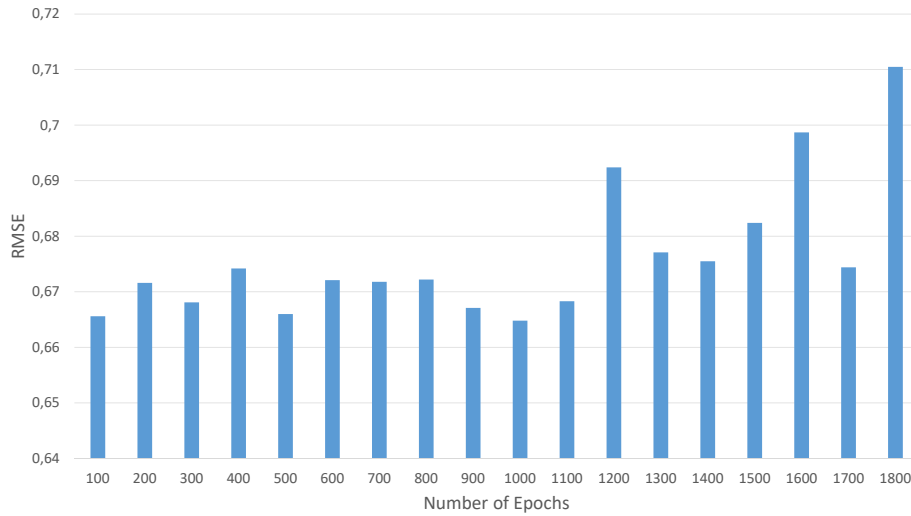


Fig. 11. Selecting the best LSTM Number of Epochs parameter.

- private L1 and L2 caches of 32 KB and 256 KB per node, and a L3 cache of 32 MB shared by all the cores of a socket.
- It includes an NVidia GTX 1080 Ti(Pascal), with 12 GB and 3584 cores (28 SM and 128 SP per SM).

The software environment is based on:

- gcc 7.4.0.
- Nvidia cuda 10.
- Python 3.6.5.
- The design of our LSTM model is based on Tensorflow 1.12.0 and Keras 2.2.4.

4. Results and discussion

This section shows the results obtained from the LSTM model designed to predict the temperature from IoT information. First, we show and analyze the results of the 3-cross validation. Then, we show the results of the test of 90% for train and 10% for test (considering 24 consecutive hours) for the LSTM technique. Finally, a comparison with two techniques for time series (ARIMA and Gaussian Process) is then made to evaluate and compare the performance of LSTM with other techniques.

4.1. Cross Validation test

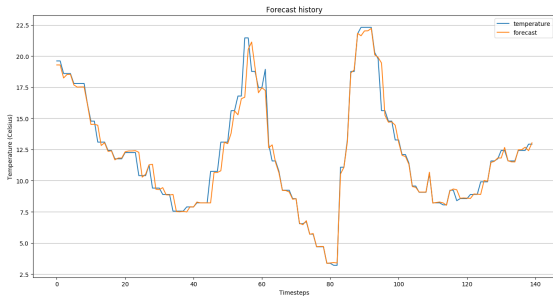
In this experiment the adjustment of the LSTM is shown by means of a 3-fold cross validation. Thus, the

16739 registers in Cieza, 17070 registers in Moratalla, and the 33809 registers in Cieza+Moratalla Dataset (“All Datasets”) are randomly divided into 3 subsets of which two of them are trained and the third is performed the test, this is repeated until you have tested with the 3 subsets. Table 4 shows the mean results of the 3-fold cross validation experiment for the 3 Datasets. The results in terms of fit of the models are satisfactory, the value of R^2 indicates that the model created for each dataset, fits the real behavior of the data. Analyzing the models from the point of view of the RMSE and the MAE, we obtain an RMSE lower than a Celsius degree for the local models of Cieza and Moratalla. However, the global model obtains an RMSE higher than 1 degree Celsius. Analyzing the MAE, the results are similar to the RMSE, obtaining a better error the results created by the individual models, than by the global model. At a general level the results of all the models are satisfactory and we can affirm that the technique obtains a good behaviour for the prediction of the temperature.

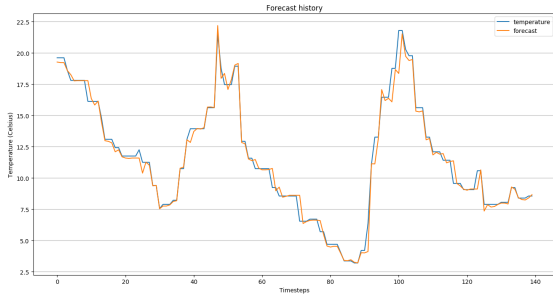
Dataset	Cieza	Moratalla	All
RMSE	0.9782	0.8133	1.0283
MAE	0.4067	0.4528	0.5287
PCC	0.9863	0.9912	0.9869
R²	0.9725	0.9824	0.9740

Table 4

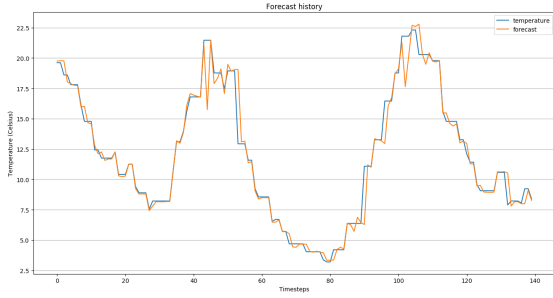
Mean results obtained after the execution of experiment of 3-fold cross validation. Being **RMSE** the Root Mean Square Error, **MAE** the Mean Absolute Error, **PCC** the Pearson Correlation Coefficient, and **R²**, the determination coefficient.



e) Fold 1- Cieza cross validation experiments



b) Fold 2- Cieza cross validation experiments

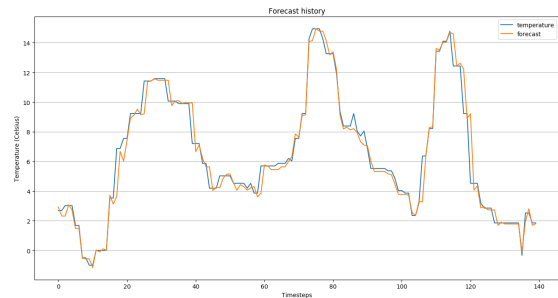


c) Fold 3- Cieza cross validation experiments

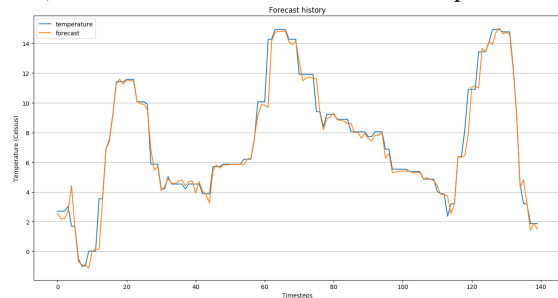
Fig. 12. Comparison between the prediction of the LSTM and the actual temperature of the first 200 instances for the Cieza and Moratalla datasets

If in addition to analyzing the error, the trend of the models is analyzed, the goodness and fit of the model can be appreciated. Specifically, Figures 12 a), 12 b) and 12 c), show the first 200 instances of the test for each fold of the Cieza dataset. Only 200 instances are shown to be able to appreciate, the small errors that occur. The blue line is the actual temperature and the orange line is predicted by the LSTM model. It stands out as the higher the temperature the more errors occur, however the lower the temperature is, the better the prediction of the model is. This behavior for the frost problem is beneficial because the error in the estimation of low temperatures is more adjusted.

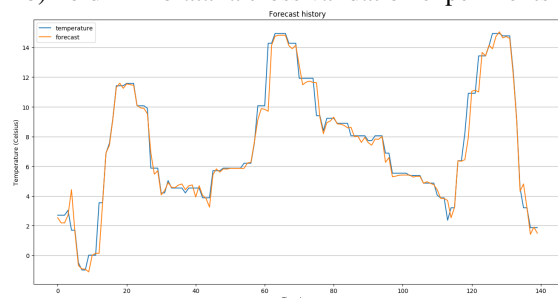
Same way, Figures 13 a), 13 b) and 13 c), show the first 200 instances of the test for each fold of the Moratalla dataset. The behavior is much more uniform



e) Fold 1- Moratalla cross validation experiments



b) Fold 2- Moratalla cross validation experiments



c) Fold 3- Moratalla cross validation experiments

Fig. 13. Comparison between the prediction of the LSTM and the actual temperature of the first 200 instances for the Cieza and Moratalla datasets

in this dataset than in the Cieza datasets. Hence RMSE shown in table X is lower. However, this translates into a greater error of the MAE, since it has a greater deviation in all temperature bands.

The global model of the “All” dataset is not shown graphically because it has a larger error and it is not interesting in terms of results.

4.2. 24-hours temperature prediction

Once the robustness and reliability of the model have been analysed by means of cross validation, the model’s capacity to predict 24 consecutive hours is analysed in this experiment. Specifically, to make the prediction we use full days of 24 hours, predicting values every 10 minutes, selecting the last available days

of each dataset. Thus, the aim of this experiment is to measure the quality of the LSTM prediction whenever it tries to predict the temperature in a long period (i.e. 24 hours). To assess the LSTM model to predict 24 consecutive hours we use the RMSE, MAE, PCC and R^2 . The values of this experiment are shown in the Table 5.

Datasets	Cieza	Moratalla	All
RMSE	0.6524	1.1147	1.2058
MAE	0.4089	0.6321	0.6703
PCC	0.9911	0.9869	0.9827
R²	0.9820	0.9732	0.9648

Table 5

Results obtained for the experiments of predicting 24 hours for different days. In the table **RMSE** is the Root Mean Square Error, **MAE** the Mean Absolute Error, **PCC** the Pearson Correlation Coefficient, and **R²**, the determination coefficient.

The results obtained for this experiment are similar to the previous general level. The RMSE obtains an average error of 0.64 Celsius degrees in Cieza dataset, and a little above a grade for Moratalla and "All" datasets. The fit of the model (R^2) is similar to the previous experiment. It is worth mentioning that the best results have been obtained with the models created by local temperatures. Although the differences are not too large, but in the Cieza area, the error difference is almost double, while having a local model the prediction is much more accurate. Figures 14 show the prediction of a full day with 144 records for the Cieza and Moratalla datasets, specifically the records correspond to January 3, 2019.

In addition, this experiment carries out an in-depth analysis of the different types of errors produced. From the point of view of the frost problem, a system error predicting that it will not freeze and then that if the thermometer is actually lowered from zero, is a very serious error, as the farmer loses his/her entire crop. On the contrary, an error in the system indicating that it freezes and then in reality there is no such frost is less serious, because the farmer does not lose the crop simply loses the resources to apply the anti-freeze technique that economically these resources cost much less than losing the crop. To perform this in-depth analysis of the different types of errors, we will perform the classification of values to analyze false positives and false negatives. In the classification we consider cold ('C') when the temperature is lower than or equal to 0 and No-Cold ('NC') when the temperature is higher than 0. Using the prediction values made

by the LSTM, we make the comparison between the real value and the predicted value and the classification comparison. Table 6 shows the classification results for the 3 datasets under study. The columns of Cieza, Moratalla and All show the percentage with respect to the total of the data, taking into account the comparison between the predicted by the model and the real value. Prediction column refers to the predictions made by our model, and the Real column correspond to the real data. For these last two columns, the different combinations have the following meaning:

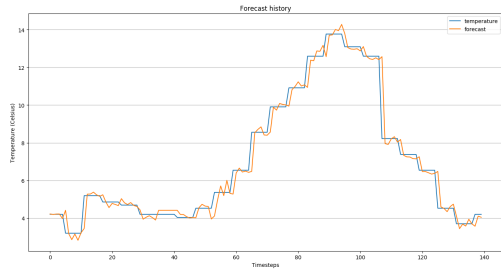
1. **NC&NC** means that the prediction was *No Cold*, and the real data was *No Cold* too, this is considered as a success/hit in the prediction.
2. **NC&C** means that the prediction was *No Cold*, but the real value was *Cold*, therefore the model failed, and the type of error as defined at the beginning of this section is error type 2, "*Error₂*".
3. **C&NC** means that the prediction was *Cold*, but the real value was *No Cold*, thus, the prediction was wrong. This is the "*Error₁*" kind of error.
4. **C&C** means that the prediction was *Cold* and the real data was the same, this is also considered as a success in the model.
5. **Successes** row contains the sum of rows considered as success in the model, "NC&NC" and "C&C" rows where the forecast matches with the actual weather.
6. **Errors** contains the sum of rows considered as failures in the model, "C&NC" and "NC&C", that is, "*Error₁*" and "*Error₂*" respectively.

Dataset		Cieza	Moratalla	All
Predition	Real			
NC	NC	91.33%	52.41%	70.57%
NC	C	0.78%	1.80%	2.39%
C	NC	0.61%	1.13%	1.13%
C	C	7.26%	44.64%	25.89%
Successes		98.60%	97.06%	96.46%
Error		1.39%	2.93%	3.53%

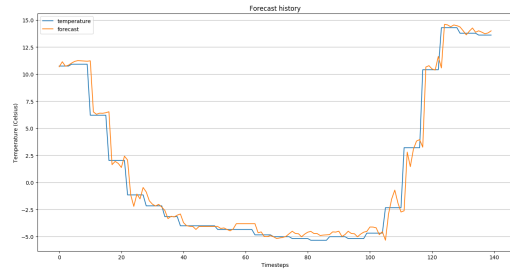
Table 6

Results obtained for the experiments of predicting 24 hours for different days, for the three datasets.

The results of Table 6 show the two types of errors for the Cieza dataset are very low (less than 1%). For the Moratalla dataset the error rises slightly. However, it is worth mentioning the *Error₂* percentage, considered NC&C of the dataset "All". This type of error is very serious and the percentage of errors should be as



a) Cieza Dataset



b) Moratalla Dataset

Fig. 14. Comparison between the LSTM prediction and the actual temperature of the 144 temperature measurements obtained on 03 January 2019 in Cieza (a) and in Moratalla (b), this corresponds to a full 24-hour day.

low as possible since it causes the farmer to lose his/her crop, since the system indicates that there is no frost and finally there is a frost, so the farmer has not activated the anti-freeze techniques and the crop is lost. To analyze more in detail the types of errors, we show the confusion matrices in the Tables 7, 8, and 9. Table 7 shows the result for Cieza datasets, Table 8 shows the Moratalla and Table 9 shows “All” dataset respectively.

Cieza	C	NC
C	83	9
NC	7	1044

Table 7

Confusion Matrix for Cieza Dataset. Tested with 1143 registers.

Moratalla	C	NC
C	471	19
NC	12	553

Table 8

Confusion Matrix for Moratalla Dataset. Tested with 1055 registers

Cieza+Moratalla	C	NC
C	594	26
NC	55	1619

Table 9

Confusion Matrix for Cieza+Moratalla Dataset. Tested with 2294 registers

Table 6, in Cieza datasets, successes are measured with 98.60%, this is the sum of "NC&NC" and "C&C", $91.33\%+7.26\%=98.60\%$. This value is detailed in table 7, with 1044 (from NC&NC cell), and 83 (from C & C cell), $(1044+83)/1143=98.60\%$ too. This is important because with table 7 now we can know the kind

of success, in other words, we know that almost all the hits in Cieza were from the "NC&NC" side, maybe due to a warmer weather, 1044 against 83. For Moratalla dataset, the successes are 97.06%, this is the sum of "NC" and "C&C", $52.41\%+44.64\%$. The successes are more distributed between C&C and NC&NC, since it is a colder area and there are more instances with values equal to or less than zero. It should be noted that Error₂ percentage is twice as high as for Cieza datasets, but less than 2%. Finally, the datasets All has a very high percentage of errors and especially in Error₂. In addition to the Error percentage, we are going to analyze each of the confusion matrices of the 3 datasets, this matrices are shown in Tables 7, 8 and 9. In the first two tables Error₁ and Error₂, has almost the same probability, and in any case, the one that appears a little more is type one, which, as we already know is a bit less dangerous. The difference in the errors between Cieza and Moratalla is that Moratalla is a city with more hours of cold than Cieza hence the best accuracy obtained since the hours of cold are the most difficult to estimate because datasets are unbalanced in terms of number of cold hours. Finally, in the table 9, we can observe that although there are some differences that we can see in table 6, the kind of error is much more sensible in this case, because the most common type of error in this case is type 1, 55 cases versus 26, which is much more dangerous than in the previous datasets. Therefore, this global model gets a more negative result for the constraints of the proposed low temperature prediction problem.

As a summary of results, the LSTM technique works well and obtains satisfactory models for both local and global areas. However, taking into account the limitation of the data given by the unbalancing of the cold hours, the local models obtain more positive results, from the economic point of view for the

farmer. This is given because the global models get a higher percentage of $Error_2$ which are the errors, where the model indicates that there is no frost and finally a frost occurs, this causes the farmer to lose his/her crop, while the other type of error, the farmer only loses the resources of the antifrost techniques that are much smaller. Thus, after analyzing the results, it is recommended to deploy an IoT system per area with a single temperature sensor, lowering the cost of that system.

4.3. Comparing LSTM with other techniques

This section compares our LSTM approach with two traditional methods that have been widely used in the literature to predict values in a time series; they are Auto Regressive Integrated Moving Average (ARIMA) [38] and Gaussian process [39]. In ARIMA, the autoregressive part indicates that the variable of interest in evolution is regressed based on its previous values. The moving average part shows that the regression error is actually a linear combination of error terms whose values occurred simultaneously and at various times in the past. Integrated means that the data values have been replaced with the difference between their values and the previous values, and this process could be performed more than once. ARIMA is usually applied in cases where the information does not have stationarity. The Gaussian process, however, is a stochastic process based on a collection of random variables indexed by time or space, such that each finite collection of those random variables has a normal multivariate distribution, that is, each finite linear combination of them is normally distributed.

The validation carried out for all the experiments has been a 90% of instances for train and 10% of instances for test was used, as in the previous experiment.

Table 10 the comparison of the results between the proposed LSTM model, the ARIMA and Gaussian Process techniques for air temperature. Three metrics have been used to perform this comparison, specifically root-mean-square error (RMSE), Mean Absolute Error (MAE) and the coefficient of determination (R^2). Analyzing these results from the point of view of model fit, (R^2), LSTM and ARIMA techniques have satisfactory results and obtain competent models for the 3 datasets. However the Gaussian process has a less stable behaviour and its model is worse than the other two. On the other hand, if we analyze the result from the point of view of the error, both of the RMSE and of the MAE, the best result is obtained by the LSTM

Techniques	Datasets	Cieza	Moratalla	All
LSTM	RMSE	0,6524	1,1147	1,2
	MAE	0,4089	0,6321	0,6703
	R^2	0,982	0,9732	0,9648
ARIMA	RMSE	1,0352	1,3512	1,5631
	MAE	0,5935	0,6985	0,8521
	R^2	0,9436	0,9423	0,9321
G. Process	RMSE	2,3622	2,6859	3,0123
	MAE	1,2305	1,3625	1,6589
	R^2	0,7236	0,7365	0,7025

Table 10

Comparison of the LSTM technique with the ARIMA time series techniques and the Gaussian Process (G.Process), showing for each value **RMSE** is the Root Mean Square Error, **MAE** the Mean Absolute Error and **R^2** , the determination coefficient.

model for the 3 datasets, being the difference almost of 0.5°C for some datasets in the RMSE. Therefore, after the comparison, we can conclude that the model that best fits the problem of predicting temperature, using a univariate model, is the LSTM model.

5. Conclusions and Future Work

The prediction of low temperatures is a latent problem for farmers, as climate change is causing abrupt temperature changes even in warm areas. Knowing in advance whether or not low temperatures will occur helps the farmer to foresee resources and apply antifrost control techniques early enough to ensure maximum effectiveness. In this work an a predictive model for temperature has been analyzed and constructed using a Long short-term memory neural network. The LSTM has been tested using actual temperature data provided by an IoT system deployed on two areas of Region of Murcia (Spain), also a global model has been tested. The models obtained by the LSTM for temperature prediction will be part of the intelligent component of the IoT system distinguishing the type of zone, as it has been found that local models are more accurate than the global model. The two experiments carried out to test the proposed model have obtained similar results and with quite satisfactory goodness obtaining about 96-98% model fit and an RMSE of less than 1 Celsius degrees. However, from the point of view of the errors made, specifically the $Error_2$, which means that the system predicts no frost and in reality there is frost, the global model gets a higher percentage of them and local models less error in general and less $Error_2$. Therefore, the best model to implement the

IoT system prediction is the local model in each area. This model also indicates that the IoT system can be composed only of the temperature sensor, as the results obtained are acceptable. Furthermore, in the comparison with the two traditional techniques, it can be seen that the proposed LSTM model obtains a better performance, followed by the ARIMA technique, although the latter obtains a good adjustment of the model but a higher temperature prediction error.

As future work, new variables will be incorporated into the LSTM to create a multivariate LSTM with our local IoT infrastructure and study the influence of other variables on temperature prediction. Moreover, the computational side of deep learning models here developed will be analyzed in terms of performance and power-consumption. This will enable edge computing platforms to execute our local models. Finally, a new machine learning technique based on K nearest neighbours, that maintains the temporality of the data, will be developed to compare with the current results.

Acknowledgment

This work was partially supported by the Fundación Séneca del Centro de Coordinación de la Investigación de la Región de Murcia under Project 20813/PI/18, and by Spanish Ministry of Science, Innovation and Universities under grants TIN2016-78799-P (AEI/FEDER, UE) and RTC-2017-6389-5.

References

- [1] E. Pierpaoli, G. Carli, E. Pignatti and M. Canavari, Drivers of precision agriculture technologies adoption: a literature review, *Procedia Technology* **8** (2013), 61–69.
- [2] N. Zhang, M. Wang and N. Wang, Precision agriculture a worldwide overview, *Computers and electronics in agriculture* **36**(2–3) (2002), 113–132.
- [3] S. Wolfert, L. Ge, C. Verdouw and M.-J. Bogaardt, Big data in smart farming—a review, *Agricultural Systems* **153** (2017), 69–80.
- [4] Z. Shareef and S. Reddy, Wireless sensor network for aquaculture: Review, survey, and case study of aquaculture practices in western Godavari region, *Journal of Ambient Intelligence and Smart Environments* **10**(5) (2018), 409–423.
- [5] P.P. Ray, Internet of things for smart agriculture: Technologies, practices and future direction, *Journal of Ambient Intelligence and Smart Environments* **9**(4) (2017), 395–420.
- [6] C. Zhang and J.M. Kovacs, The application of small unmanned aerial systems for precision agriculture: a review, *Precision agriculture* **13**(6) (2012), 693–712.
- [7] A.-K. Mahlein, Plant disease detection by imaging sensors—parallels and specific demands for precision agriculture and plant phenotyping, *Plant Disease* **100**(2) (2016), 241–251.
- [8] J.D. Pujari, R. Yakkundimath and A.S. Byadgi, Image processing based detection of fungal diseases in plants, *Procedia Computer Science* **46** (2015), 1802–1808.
- [9] K. Barlow, B. Christy, G. O’leary, P. Riffkin and J. Nuttall, Simulating the impact of extreme heat and frost events on wheat crop production: A review, *Field Crops Research* **171** (2015), 109–119.
- [10] M.Á. Guillén-Navarro, F. Pereñíguez-García and R. Martínez-España, IoT-based system to forecast crop frost, in: *2017 International Conference on Intelligent Environments (IE)*, IEEE, 2017, pp. 28–35.
- [11] K.A. Al-Gaadi, A.A. Hassaballa, E. Tola, A.G. Kayad, R. Madugundu, B. Alblewi and F. Assiri, Prediction of potato crop yield using precision agriculture techniques, *PLoS one* **11**(9) (2016), e0162219.
- [12] X.E. Pantazi, D. Moshou, T. Alexandridis, R. Whetton and A.M. Mouazen, Wheat yield prediction using machine learning and advanced sensing techniques, *Computers and Electronics in Agriculture* **121** (2016), 57–65.
- [13] D. Bretreger, J. Quijano, J. Awad et al., Monitoring irrigation volumes using climate data and remote sensing observations, in: *Hydrology and Water Resources Symposium (HWRS 2018): Water and Communities*, Engineers Australia, 2018, p. 112.
- [14] A.N. Harun, M.R.M. Kassim, I. Mat and S.S. Ramli, Precision irrigation using wireless sensor network, in: *2015 International Conference on Smart Sensors and Application (ICSSA)*, IEEE, 2015, pp. 71–75.
- [15] M.A. Guillén-Navarro, R. Martínez-España, B. López and J.M. Cecilia, A high-performance IoT solution to reduce frost damages in stone fruits, *Concurrency and Computation: Practice and Experience*, e5299.
- [16] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, MIT press, 2016.
- [17] A. Kamilaris and F.X. Prenafeta-Boldu, Deep learning in agriculture: A survey, *Computers and Electronics in Agriculture* **147** (2018), 70–90.
- [18] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural networks* **61** (2015), 85–117.
- [19] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, *Computational intelligence and neuroscience* **2016** (2016).
- [20] H.S. Abdullahi, R. Sheriff and F. Mahieddine, Convolution neural network in precision agriculture for plant image recognition and classification, in: *2017 Seventh International Conference on Innovative Computing Technology (Intech), Ieee, Londrés*, 2017, pp. 1–3.
- [21] S.W. Chen, S.S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C.J. Taylor and V. Kumar, Counting apples and oranges with deep learning: A data-driven approach, *IEEE Robotics and Automation Letters* **2**(2) (2017), 781–788.
- [22] A. Milioto, P. Lottes and C. Stachniss, Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **4** (2017), 41.

- [23] J. Zhang, Y. Zhu, X. Zhang, M. Ye and J. Yang, Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas, *Journal of hydrology* **561** (2018), 918–929.
- [24] S.A. Haider, S.R. Naqvi, T. Akram, G.A. Umar, A. Shahzad, M.R. Sial, S. Khaliq and M. Kamran, LSTM neural network based forecasting model for wheat production in Pakistan, *Agronomy* **9**(2) (2019), 72.
- [25] X.-Y. Wang, W.-X. Liao, D. An and Y.-G. Wei, Maize Haploid Identification via LSTM-CNN and Hyperspectral Imaging Technology, *arXiv preprint arXiv:1805.09105* (2018).
- [26] A.G. Salman, Y. Heryadi, E. Abdurahman and W. Suparta, Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting, *Procedia Computer Science* **135** (2018), 89–98.
- [27] M.A. Zaytar and C. El Amrani, Sequence to sequence weather forecasting with long short-term memory recurrent neural networks, *International Journal of Computer Applications* **143**(11) (2016), 7–11.
- [28] F. Kratzert, D. Klotz, C. Brenner, K. Schulz and M. Hrnegger, Rainfall-Runoff modelling using Long-Short-Term-Memory (LSTM) networks, *Hydrol. Earth Syst. Sci. Discuss.*, <https://doi.org/10.5194/hess-2018-247>, in review (2018).
- [29] K. Mohammadi, S. Shamshirband, S. Motamedi, D. Petković, R. Hashim and M. Gocic, Extreme learning machine based prediction of daily dew point temperature, *Computers and Electronics in Agriculture* **117** (2015), 214–225.
- [30] M. Bendre, R. Thool and V. Thool, Big data in precision agriculture: Weather forecasting for future farming, in: *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, 2015, pp. 744–750.
- [31] R. Huang, C. Zhang, J. Huang, D. Zhu, L. Wang and J. Liu, Mapping of daily mean air temperature in agricultural regions using daytime and nighttime land surface temperatures derived from TERRA and AQUA MODIS data, *Remote Sensing* **7**(7) (2015), 8728–8756.
- [32] M.Á. Guillén-Navarro, R. Martínez-España, B. López and J.M. Cecilia, A high performance IoT solution to reduce frost damages in stone fruits, *Concurrency and Computation: Practice and Experience* (2019). doi:10.1002/cpe.5299.
- [33] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural computation* **9**(8) (1997), 1735–1780.
- [34] S. Hochreiter and J. Schmidhuber, LSTM can solve hard long time lag problems, in: *Advances in neural information processing systems*, 1997, pp. 473–479.
- [35] K. Xu, J. Ba, R. Kiros, K. Cho, A.C. Courville, R. Salakhutdinov, R.S. Zemel and Y. Bengio, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, *CoRR abs/1502.03044* (2015). <http://arxiv.org/abs/1502.03044>.
- [36] K. Gregor, I. Danihelka, A. Graves and D. Wierstra, DRAW: A Recurrent Neural Network For Image Generation, *CoRR abs/1502.04623* (2015). <http://arxiv.org/abs/1502.04623>.
- [37] C. Bergmeir and J.M. Benítez, On the use of cross-validation for time series predictor evaluation, *Information Sciences* **191** (2012), 192–213.
- [38] P.J. Brockwell and R.A. Davis, *Introduction to time series and forecasting*, Springer, 2016.
- [39] S. Brahim-Belhouari and A. Bermak, Gaussian process for nonstationary time series prediction, *Computational Statistics & Data Analysis* **47**(4) (2004), 705–712.
- [40] M. Roopaei, P. Rad and K.-K.R. Choo, Cloud of things in smart agriculture: Intelligent irrigation monitoring by thermal imaging, *IEEE Cloud computing* **4**(1) (2017), 10–15.