Time Series Forecasting to Support Irrigation Management

Dieinison J. F. Braga[1], Ticiana L. Coelho da Silva[2], Atslands Rocha[2], Gustavo Coutinho[2], Regis P. Magalhães[2], Paulo T. Guerra[2], Jose A. F. de Macêdo[2], Simone D. J. Barbosa[1]

[1] Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro – Brazil
dbraga@inf.puc-rio.br, simone@inf.puc-rio.br
[2] Federal University of Ceará (UFC)
Ceará – Brazil
{ticianalc, regis, gustavo.coutinho, jose.macedo}@insightlab.ufc.br,
{atslands, paulodetarso}@ufc.br

Abstract. Irrigated agriculture is the most water-consuming sector in Brazil, representing one of the main challenges for the sustainable use of water. This study has investigated and evaluated popular machine learning techniques like Gradient Boosting and Random Forest, deep learning models and univariate time series models to predict the value of reference evapotranspiration, a metric of water loss from the crop to the environment. The reference evapotranspiration, ET_0, plays an essential role in irrigation management since it can be used to reduce the amount of water that will not be absorbed by the crop. We performed the experiments with two real datasets generated by weather stations. The results show that the deep learning models are data-hungry, even when we increased the training set it was not enough to outperform multivariate models like Random Forest, Gradient Boosting and M5' which indeed execute faster than the deep learning models during the training phase. However, the univariate time series model as the evaluated deep learning models (stacked LSTM and BLSTM) is a viable and lower-cost solution for predicting ET_0, since we need to monitor only one variable.

Categories and Subject Descriptors: G.3 [**Time series analysis**]: Miscellaneous; H.2.8 [**Data Mining**]: Miscellaneous; H.2 [**Applied computing**]: Miscellaneous

Keywords: Time Series Forecasting, Deep Learning, Reference Evapotranspiration

1. INTRODUCTION

Population growth and changes in climate have a direct impact on global food security. One of the primary goals of agricultural research is to find improved ways to produce food. Irrigation consumes 72% of freshwater in Brazil according to [Fontenelle et al. 2017]. A massive portion of this amount is wasted due to poorly executed irrigation and lack of control of the exact amount of water to use in the irrigation process.

The evapotranspiration value (ET_m) plays a crucial role in supporting decision making in irrigation management. Evapotranspiration is the simultaneous occurrence of evaporation and transpiration processes in a crop, measured in millimeters per unit of time. We use the following equation to compute it: $ET_m = K_c \times ET_0$, where K_c is the crop coefficient c, given on INMET website[1], and ET_0 is the reference crop evapotranspiration, which corresponds to the evapotranspiration rate of a grass surface. The value of ET_0 is relevant to management and scaling in irrigation since it gives information about how much water the crop losses to the environment [Fontenelle et al. 2017].

[1] http://sisdagro.inmet.gov.br/sisdagro/app/monitoramento/bhc

The traditional *Penman-Monteith* method [Allen et al. 1998] used to compute ET_0 is complex and does not tolerate the unavailability of some of its variables, which makes its use unfeasible. The paper [Caminha et al. 2017] proposes a Machine Learning-based approach to forecast ET_0 based on Linear Regression [James et al. 2013] and M5' [Wang and Witten 1996]. Despite the good results obtained in both techniques, they are multivariate models, which means that they require a weather station with many sensors to capture all the required variables, and there is no guarantee that the models would perform well in the absence of some variables.

Experiments performed in [Siami-Namini and Namin 2018] with a univariate time series model showed that the Autoregressive Integrated Moving Average (ARIMA) [Box et al. 2015] model is a promising technique to achieve good accuracy performance in the forecast of financial time series. ARIMA models aim at describing the correlations in the data with each other. An improvement over ARIMA is Seasonal ARIMA (SARIMA) [Box et al. 2015], which takes into account the seasonality of the dataset and is useful in short-term forecast [Tseng and Tzeng 2002]. We use both approaches in our experiments presented in this paper.

Besides the aforementioned techniques, a new one is recently gaining momentum: deep learning, which is an evolution of artificial neural networks and generally used to deal with non-linear models. Such models are about "deeper" neural networks and allow more considerable learning capabilities and thus higher performance and precision. The Recurrent Neural Network (RNN) or, more specifically, the Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber 1997] has been extensively used in regression and classification tasks since it processes variable-length input and can learn highly non-trivial long-distance dependencies. LSTM models may prove feasible to forecast ET_0 since the amount of water needed to irrigate the soil depends on the soil conditions in the previous days. All in all, we need a model that can learn long-distance even if it is hours or days. Several approaches propose models derived from LSTM for different prediction tasks [Laptev et al. 2017; Kamilaris and Prenafeta-Boldu 2018]. Our experiments were conducted on two real datasets and show the comparison of popular machine learning techniques like Random Forest and M5', deep learning models, and univariate time series models as ARIMA and SARIMA to build prediction models.

The paper [Braga et al. 2018] proposes the following contributions: (i) it offers an accurate and lower cost solution to estimate ET_0, since only one variable needs monitoring, and (ii) compares the performance of ARIMA, SARIMA, Linear Regression and M5' with respect to minimization achieved in the error rates in prediction. This work substantially extends the work conducted in [Braga et al. 2018] in the following ways: (i) we expand the related work section; (ii) we improve the theoretical presentation of our approach; (iii) we include deep learning models based on stacked LSTM and Bidirectional LSTM in the performance comparison with ARIMA, SARIMA, Linear Regression, and M5'; (iv) we include efficient and widely used regression techniques Random Forest and Gradient Boosting; and (v) we further expand the experimental evaluation to better assess the quality of the forecasting models for ET_0 in terms of analyzing another dataset and finding the proper values for the algorithm parameters.

The remaining sections of this article are organized as follows. Section 2 explains the preliminary concepts required to understand this work. Section 3 shows the steps necessary to accomplish our goals. Section 4 presents our experiments and their analysis. Section 5 presents related work. Finally, Section 6 summarizes this work and proposes future developments.

2. PRELIMINARY NOTIONS

In this section, we introduce the main notions and algorithms to create the prediction models used throughout this work.

2.1 Time Series

A time series (TS) is a series of data records indexed by dates. A time series model supposes that a series Z_t could be defined as $Z_t = T_t + S_t + \alpha_t$, being T the tendency, S the seasonality and α the white noise, at the moment t [Brockwell and Davis 2016]. Most of the TS models work on the assumption that the TS is stationary, i.e., its statistical properties such as mean and standard deviation remain constant over time. As many real-time series are non-stationary, statisticians have figured out ways to make TS stationary [Box et al. 2015].

In particular, a differencing operator (∇) is a simple and efficient operator to transform a non-stationary TS into stationary. The equation defines it: $\nabla Z_t = Z_t - Z_{t-1}$, where Z is a TS at the moment t [Brockwell and Davis 2016]. In other words, we take the difference between the observations at a particular instant t and at the previous instant $t-1$. In our experiments, both datasets are time series and were transformed into stationary.

2.2 Forecasting Models

This section introduces briefly the forecasting models investigated in this work: univariate models as ARIMA and SARIMA, machine learning techniques such as Linear Regression and M5, and a deep learning model based on stacked LSTM.

The idea behind the ARIMA technique is to report the correlations within the data. It is a widely used algorithm to forecast time series. The ARIMA model takes three hyper-parameters p, d, q, which capture the key elements of the model, which are: (i) Autoregression (AR), a regression model that uses the relationship between an observation and a number (p) of lagged observations; (ii) Integrated (I), the number (d) of differentiations required to obtain stationarity; (iii) Moving Average (MA), an approach that takes into account the dependency between the observations and the residual error terms when a moving average model is used for the lagged observations (q). For further details, we refer the reader to [Box et al. 2015].

An improvement over ARIMA is SARIMA, which has been successfully used in short-term forecasts. The SARIMA model incorporates both seasonal and non-seasonal factors in TS data. Its signature is $SARIMA(p, d, q) \times (P, D, Q)S$, where p and P are the non-seasonal and seasonal AR order; d and D are the non-seasonal and seasonal differencing; q and Q are the non-seasonal and seasonal MA order; and S is the time span of repeating seasonal pattern, respectively. For further details, we refer the reader to [Box et al. 2015] and [Tseng and Tzeng 2002].

Regression analysis is a mathematical method for investigating and modeling the relation between variables [Montgomery et al. 2012]. It is broadly used for knowledge discovery in databases with continuous variables. According to [Montgomery et al. 2012], a multiple linear regression model can be represented by $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon$, where y corresponds to the variable to be predicted and $\beta_j, \forall j, j \in \{0, 1, ..., k\}$ the regression coefficients, which represent the change in the response value of y for each unit of x_j. The ϵ corresponds to the statistical error to map the data according to the model.

Another approach for creating predictive models is the application of decision trees. Traditional decision trees and their learning rules have been developed to use attributes whose values are discrete [Wang and Witten 1996]. However, in real-world data, continuous values are very common and improvements in this approach have been necessary. From this need, [Quinlan et al. 1992] proposed the algorithm M5, whose main characteristic is the use of a decision tree in which, unlike the traditional one, offers the possibility of using multivariate linear regression models in its leaves.

Later, [Wang and Witten 1996] realized that M5 had no treatment of enumerated attributes or missing values and proposed the M5' algorithm, which implemented these improvements. In M5', the solution for attributes with missing values is made from the process of partitioning them, which is a

method used by several attribute selection algorithms. Because of this feature, it is possible that the linear models generated in the leaves of the tree do not present all attributes given as input.

Ensemble-based algorithms usually achieve high accuracy in many machine learning tasks [Caruana and Niculescu-Mizil 2006]. Recently they have been widely used with great success in solving prediction and classification problems [Zhou 2012]. Two widely used ensemble methods are bagging [Breiman 1996] and boosting [Schapire et al. 1998] of trees. They are general techniques that can be applied to tree-based methods in order to increase the accuracy of the resulting predictions. Bagging of trees, or bootstrap aggregation, combines several tree predictors trained on bootstrap samples of the training data. Successive trees used in bagging do not depend on earlier trees. The results are combined, by simple averaging for regression and simple voting for classification, to obtain the overall prediction, with the variance being reduced due to the averaging.

Differences between bagging and boosting are described by [Sutton 2005]: Unlike bagging, which uses a simple averaging of results to obtain an overall prediction, boosting uses a weighted average of results obtained from applying a prediction method to various samples. Also, with boosting, the samples used at each step are not all drawn in the same way from the same population, but rather the incorrectly predicted cases from a given step are given increased weight during the next step. Thus boosting is an iterative procedure, incorporating weights, as opposed to being based on a simple averaging of predictions, as is the case with bagging. In addition, boosting is often applied to weak learners (e.g., a simple classifier such as a two-node decision tree), whereas this is not the case with bagging.

Random forests (RF) proposed by [Breiman 2001] add an additional layer of randomness to bagging. It uses random feature selection at each node for the set of splitting variables. RF are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large. The generalization error of a forest depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields low error rates and more robustness with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance.

Gradient Boost Machine (GBM) [Friedman 2001], also known as gradient tree boosting or gradient boosted regression tree, constructs additive regression models by sequentially fitting a simple parameterized function (base learner) to current pseudo-residuals by least squares at each iteration. The pseudo-residuals are the gradient of the loss functional being minimized, with respect to the model values at each training data point evaluated at the current step. GBM is a Machine Learning (ML) technique that produces competitive, highly robust state-of-the-art results in a wide range of regression, classification, and ranking applications. Boosting is a numerical optimization problem where the goal is to minimize the loss of the model by adding weak learners using a gradient-descent-like procedure. The original implementation of this algorithm was called Multiple Additive Regression Trees (MART). The approximation accuracy and execution speed of gradient boosting can be substantially improved by incorporating randomization into the procedure [Friedman 2002]. At each iteration, a subsample of the training data is drawn at random from the full training data set. This randomly selected subsample is then used in place of the full sample to fit the base learner and compute the model update for the current iteration. This randomized approach is called Stochastic Gradient Boosting and it is frequently used to increase the robustness against overcapacity of the base learner.

Another approach investigated in this work is LSTM was proposed in [Hochreiter and Schmidhuber 1997], and it is a variant of the recurrent neural network. LSTM was the culmination of researches on the vanishing gradient problem. An LSTM unit is composed of three multiplicative gates, which control the proportions of information to forget and to pass on to the next time step. For many

prediction tasks, it is beneficial to have access to both past (left) and future (right) contexts. However, LSTM's hidden state takes information only from the past, knowing nothing about the future. An elegant solution is a bi-directional LSTM (BLSTM), which consists of using two regular LSTM layers, each of which processes the input sequence in one direction (chronologically and anti-chronologically), and then merges their representations. By treating a sequence both ways, a BLSTM can catch patterns that may be missed by the chronological-order version alone. In the experiments, the number of LSTM and BLSTM layers in the neural network model is a tuned hyper-parameter.

Figure 1 shows the best achieved neural network architecture using stacked BLSTM found in the experiments. Section 4 provides further details.

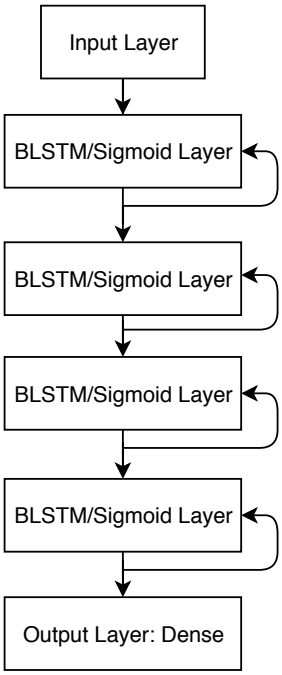

Fig. 1: BLSTM Architecture

3. METHODOLOGY

This section explains the methodology adopted to create the time series forecasting models.

3.1 Data Collection and Cleaning

In this paper, we use two real datasets. The first one (**dataset I**) is the climatic data collected by a weather station, in the period from January, 1st to November, 29th of 2017 in the city of Quixada, Ceara, Brazil. The original dataset contains 7,941 hourly records, and it is composed of the features described in Table I. The dataset is available. [2]

We aggregated the original hourly data on a daily basis. Furthermore, we detected outlier observations through the *Proximity-Based Outlier Detection* technique [Tan et al. 2006] and removed them. The tuples that contain the values described in Table II were removed. Only 333 tuples remained on the dataset I after removing the outliers.

[2]https://github.com/Dieinison/ProjectET0/blob/master/dataset.csv

Table I: Samples from dataset I collected in Quixada, Ceara, Brazil

Date	Atmospheric pressure		Air temperature			Relative humidity			Solar radiation		Temperature		Precipitation	Wind Speed	ET_0
	Max.	Min.	Max.	Min.	Mean	Max.	Min.	Mean	Total	Mean	Max.	Min.			
2017-11-29	620.5	599.7	32	19.6	21.4	55.2	45.3	50.1	1610	12.7	21.3	20.2	0.0	1.58	0.095
2017-11-30	620.2	599.7	32	19.4	21.7	52.3	41.9	46.9	1638	11.9	20.7	19.4	0.0	1.73	0.109
2017-11-31	620.4	599.6	34	19.1	20.9	45.8	39.7	42.3	1620	19	22.6	21.5	0.0	2.10	0.147
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Table II: Conditions to remove instances in dataset I

Precipitation ≥ 60
Minimum temperature ≤ 0
Minimum relative humidity ≤ 20

The second real dataset (**dataset II**) was collected by a weather station maintained by INMET[3], in the period from January, 1st of 2016 until February, 26th of 2019, in the city of Quixeramobim, Ceara, Brazil. The original dataset contains 1,153 daily records, and its features are described in Table III.

Table III: Samples from dataset II collected in Quixeramobim, Ceará, Brazil

DATE	ARM	SURPLUS	ET_0	PRECIPITATION	TEMPERATURE
01/01/2016	2.94818e−07	0	6.95434	0	28.9
02/01/2016	2.61196e−07	0	9.081503	0	29.6
03/01/2016	2.33433e−07	0	9.028169	0.6	29.2
...	...	...	...	...	...

The same analysis to detect outliers was performed by applying *Proximity-Based Outlier Detection* technique on dataset II. However, no outliers were found.

3.2 Analysis of time series stationarity

The stationary aspect of dataset I was checked by plotting rolling average and rolling standard deviation, as shown in Figure 2. The evaluated mean and standard deviation show significant instability over time, suggesting the data is non-stationary. Another technique to evaluate the non-stationary is the Dickey-Fuller (DF) test. The DF is a unit root test that evaluates the strength of a trend in a time series component [P. Avishek 2017]. The output of the DF test is shown in Table IV. As we can see, the DF is higher than the critical values, so dataset I is a non-stationary time series. We transform such dataset in a stationary series using the *PMDARIMA* API.

The stationarity of dataset II was checked by the DF test. Table V shows the output of the DF test for dataset II. In this case, the p-value of 0.08734 is greater than our 0.05 (5%) significance level, therefore we fail to reject the null hypothesis that unit root does exist, which means that dataset II is a non-stationary time series. We performed the same steps (using the *PMDARIMA* API) as in dataset I to transform the dataset II into a stationary series.

3.3 Prediction models

For the dataset I. To create the prediction model for ET_0 from dataset I, we split the dataset into 80% for training and 20% for testing. As dataset I is small, we chose not to create LSMT and BLSTM models, since deep learning techniques demand a large training dataset. Thus, we generated six distinct models, Linear Regression and M5', both implemented using WEKA [4]. The ARIMA and

[3]For further information visit: `www.inmet.gov.br/`
[4]https://www.cs.waikato.ac.nz/ml/weka/

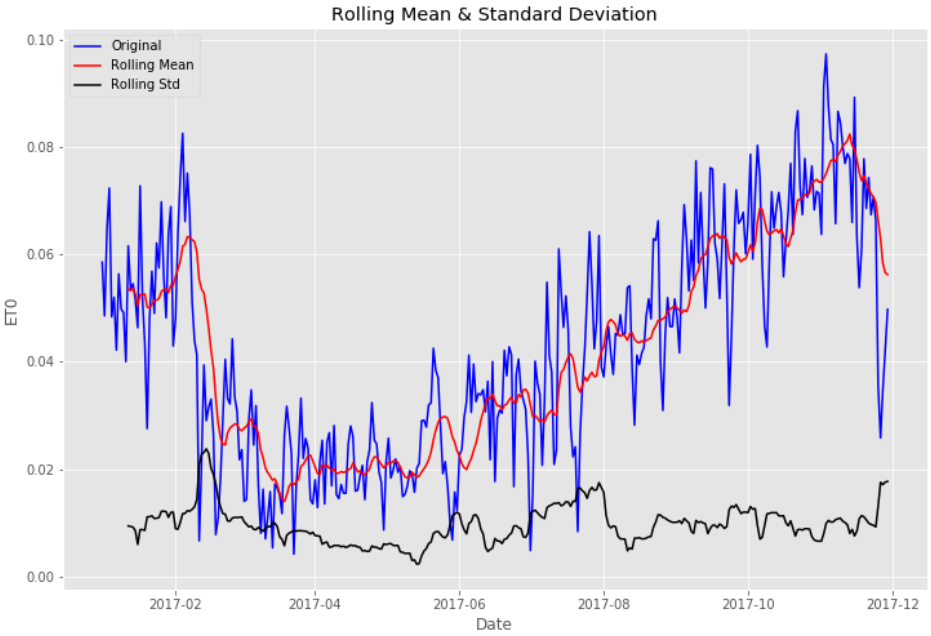

Fig. 2: Rolling average and standard deviation of ET_0 for dataset I

Table IV: Results of the DF Test for dataset I

DF Statistic	-1.695411
P-Value	0.619480
Critical Value 1%	-3.450695
Critical Value 5%	-2.870502
Critical Value 10%	-2.571545

Table V: Results of the DF Test for dataset II

DF Statistic	-3.472054
P-Value	0.08734
Critical Value 1%	-3.436089
Critical Value 5%	-2.864074
Critical Value 10%	-2.568119

SARIMA, both implemented with the *PMDARIMA* API under an MIT License[5]. The other two models experimented were Gradient Boosting and Random Forest, both implemented in scikit-learn[6].

Linear Regression, M5', Gradient Boosting and Random Forest are multivariate models. To build such models, all the features described in Table I (except the Date column) were used to predict ET_0. By contrast, ARIMA and SARIMA are univariate models, so the past values of ET_0 were used to predict the next one.

The Gradient Boosting technique presents some parameters, the most sensitive ones are (i) the

[5] `https://pypi.org/project/pmdarima/`
[6] `https://scikit-learn.org/`

number of trees in the forest, called the number of estimators; (ii) the maximum depth of the tree, called here max depth; and finally, the (iii) learning rate. Random Forest also presents as the most sensitive parameters: the number of trees and the maximum depth of the tree. In the experiments, we perform a grid search to find the proper values for these parameters for both techniques. We used 30% of the training set from the dataset as a validation set.

Grid search [Bergstra and Bengio 2012] is a widely used method to choose proper values for parameters. Grid search is simple to implement and parallelization is trivial. Basically, we predefined a range of values for each parameter. In grid search, the set of trials is formed by assembling every possible combination of values. Next, we rank the trials according to some quality measures (in this paper, we use RMSE and MAE as explained next). Then, we start the fine search by looking at the range where the best-ranked values for all parameters fall within.

In order to forecast through ARIMA and SARIMA, we adopted the Box-Jenkins methodology [Box et al. 2015], defined as (i) identification of the model, i.e., finding the appropriate orders for p, d, q, P, D, Q, S; (ii) estimation of the unknown parameters; (iii) validation of the model; and (iv) forecast future outcomes based on the known data. All these steps are implemented in the *PMDARIMA* API, so there is no need to consider a validation dataset in order to find the proper parameters for ARIMA and SARIMA. The *PMDARIMA* API performs these steps automatically.

For the dataset II. To create the prediction model for ET_0 from the dataset II, we applied the same idea explained for dataset I by splitting the data into 80% for training and 20% for testing to build the forecasting models using Linear Regression, M5', Random Forest, Gradient Boosting, ARIMA, SARIMA, and the deep learning LSTM and BLSTM models. We adopt LSTM and BLSTM models to build a prediction model for this dataset since it is larger than the previous one.

However, the LSTM and BLSTM models requires five different parameters: (i) the window length, which means how many ET_0 past values to predict a new ET_0, (ii) the number of epochs, (iii) the dropout rate, (iv) the hidden units, and finally (v) the number of stacked layers. To discover the proper values for each parameter during the experimentation of the LSTM and BLSTM model, 15% of the training set from the dataset II was used as a validation set. Section 4 provides details about the optimization of hyperparameters and the achieved results.

To find the proper parameter values for ARIMA, SARIMA, Random Forest and Gradient Boosting methods for dataset II, we adopted the same idea previously explained for dataset I.

3.4 Models Evaluations

To measure the quality of the forecasting models, the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are calculated as the evaluation metrics of the performance, defined by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad , \quad MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where i is the sample index, n is the total number of observations, y is the expected attribute value and $\hat{y}$ is the value output by the algorithm used [James et al. 2013]. Both metrics can range from 0 to ∞. They are negatively-oriented scores, which means lower values are better. RMSE has the benefit of penalizing large errors, while MAE is a measure of average error.

4. EXPERIMENTAL EVALUATION

As stated earlier, these experiments used two real datasets with observations collected from a weather station located in Quixada (dataset I) and Quixeramobim, Ceara, Brazil (dataset II). Then, we split the experiment results in the following subsections.

4.1 Analysis of the Results for dataset I

The Linear Regression and M5' models obtained using Weka tool can be found under these links: [7] and [8], respectively.

In order to obtain the optimal hyper-parameters for ARIMA and SARIMA models, we used a function called *auto arima* implemented in the *PMDARIMA* API. It provides a systematic approach to find the best hyperparameters, based on given information criteria. In these experiments, we used the Corrected Akaike Information Criterion (AIC_c), as recommended in [Brockwell and Davis 2016]. This criterion includes a penalty term to discourage the fitting of too many parameters, i.e., the fitted model with a smaller value of AIC_c is the best choice [Smith 2017; P. Avishek 2017]. Tables VI and VII present the parameters output by the *auto arima* function for the ARIMA and SARIMA algorithms, respectively.

Table VI: Proper values of ARIMA parameters found for dataset I

Parameter	**Value**
AR order p	1
Difference order d	1
MA order q	1

Table VII: Proper values of SARIMA parameters found for dataset I

Parameter	**Value**
AR order p	1
Difference order d	1
MA order q	1
Seasonal AR order P	0
Seasonal difference D	1
Seasonal MA order Q	2
S	12

Table VIII and IX show the parameters used by Random Forest and Gradient Boosting, the set of values used for each parameter and the proper value chosen by the grid search.

Table VIII: Parameters used by Random Forest, together with the associated ranges of values used at run-time for dataset I.

Parameter	**Set of values**	**Chosen Value**
number of estimators	[10,100], step 10	90
max_depth	[3,9], step 2	7

Table X shows the RMSEs and MAEs achieved by the models. As we can see, all the evaluated models achieved error values very low, almost close to zero. A value of RMSE or MAE equal to zero means that the estimator is predicting observations with perfect accuracy. Besides, Table XI shows statistical properties of our label variable, ET_0. As the error rates (RMSE and MAE) are less than the standard deviation, our results indeed show great accuracy [Legates and McCabe 1999].

The results show the outperformance of the multivariate model created using the M5' algorithm, according to RMSE and MAE metrics, over the univariate time series models. Nevertheless, the univariate time series models indeed fit well the data, since the difference between predictions and expected values is small. Regarding the *TS* models, ARIMA outperformed SARIMA in both metrics, indicating that our data is well-fitting by a nonseasonal model.

[7] `http://bit.ly/result\_linear\_regression`
[8] `http://bit.ly/resul\t_m5p`

Table IX: Parameters used by Gradient Boosting, together with the associated ranges of values used at run-time for dataset I.

Parameter	Set of values	Chosen Value
number of estimators	[10,100], step 10	60
max_depth	[3,9], step 2	5
learning rate	[0.05, 0.2], step 0.05	0.15

Table X: Comparison of the forecasting models using dataset I

Model	RMSE	MAE
ARIMA	0.0196	0.0173
Linear Regression	0.0072	0.0056
M5P	**0.0070**	**0.0056**
SARIMA	0.0225	0.0201
Gradient Boosting	0.0072	0.0059
Random Forest	0.0077	0.0062

Table XI: Mean and Standard Deviation of ET_0 for dataset I

Statistic	Value
Mean	0.0430
Standard deviation	0.0462

Due to the costs of owning a weather station with many sensors, capturing all the variables required for multivariate models may not be affordable for low-income farmers. In contrast, the results show that an ARIMA model is an affordable solution for predicting ET_0, since only one variable needs to be monitored, with no need for multiple sensors.

4.2 Analysis of the Results for dataset II

The Linear Regression and M5' models trained over dataset II on Weka are available at [9] and [10], respectively.

The proper hyper-parameter values for the ARIMA and SARIMA models for dataset II were also found using *PMDARIMA*. Tables XII and XIII present the parameter values found for ARIMA and SARIMA algorithms, respectively.

Table XII: Proper values of ARIMA parameters found for dataset II.

Parameter	Value
AR order p	2
Difference order d	1
MA order q	2

Tables XIV and XV show the parameters used by Random Forest and Gradient Boosting, the set of values used for each parameter and the proper value chosen by the grid search for the validation set of dataset II. As we mentioned before, for stacked LSTM and BLSTM models we should provide appropriate values for five parameters. This means the trial of dropout-rate, number of epochs, stacked layers, window length, and hidden units which provides the highest RMSE and MAE is the best configuration. Tables XVI and XVII show the range of values for each parameter and the final values found by the grid search to create the stacked BLSTM and LSTM models, respectively.

Table XVIII shows RMSEs and MAEs achieved by the models using the testing set of dataset II. Table XIX shows the statistical properties of our label variable, ET_0, for dataset II. Almost all evaluated models, except ARIMA and SARIMA, present RMSE and MAE values, less than the

[9] https://github.com/Dieinison/ProjectET0-v2/blob/master/results/result_linear_regression.txt
[10] https://github.com/Dieinison/ProjectET0-v2/blob/master/results/result_m5p.txt

Table XIII: Proper values of SARIMA parameters found for dataset II.

Parameter	Value
AR order p	2
Difference order d	1
MA order q	2
Seasonal AR order P	0
Seasonal difference D	1
Seasonal MA order Q	1
S	12

Table XIV: Parameters used by Random Forest, together with the associated ranges of values used at run-time for dataset II.

Parameter	Set of values	Chosen Value
number of estimators	[10,100], step 10	30
max_depth	[3,9], step 2	9

Table XV: Parameters used by Gradient Boosting, together with the associated ranges of values used at run-time for dataset II.

Parameter	Set of values	Chosen Value
number of estimators	[10,100], step 10	100
max_depth	[3,9], step 2	5
learning rate	[0.05, 0.2], step 0.05	0.1

standard deviation. In summary, we can notice that these evaluated models show great accuracy [Legates and McCabe 1999] for predicting ET_0 for dataset II.

Notice that, according to Table XVIII, the Gradient Boosting model outperforms all the other approaches by achieving the lowest RMSE and MAE for the testing set. We should also notice that Random Forest achieved almost the same quality of Gradient Boosting. As we can see, these multivariate models still outperform univariate models like ARIMA, SARIMA, and the deep learning models as well. However, it is worth mentioning that multivariate models might not be affordable for low-income farmers, as we discussed before since the weather stations with several sensors are very costly. In contrast, univariate models predict ET_0 by looking at only one variable. A very compromising solution, in this case, is to use one of the deep learning models since only the past ET_0 values were used to train the BLSTM and LSTM models instead of monitoring a set of climate variables. Besides, the deep learning models are very promising solutions as univariate models since they outperform ARIMA and SARIMA.

5. RELATED WORK

In the literature, many works have dealt with the problem of time series forecasting. In the context of this work, we give an overview on two main classes of works that are of interest: the first class of approaches uses traditional approaches for time series forecasting (including machine learning techniques, ARIMA, and SARIMA techniques), while the second class employs deep learning models.

Traditional Approaches. [Goldstein et al. 2017] uses Linear Regression, Decision Tree, Random Forest, and Gradient Boosting Regression Tree (GBRT) to predict the best weekly irrigation plan for a jojoba crop by using weather and soil data (moisture and drought stress). The GBRT model and Boosted Tree Classifier outperformed the other evaluated models. [Goap and Sharma 2018] investigated a model based on support vector regression and k-means techniques to predict the soil moisture by using weather data, rain forecast, and soil temperature. In [Zhang et al. 2013], the authors investigate the soil water deficit by analyzing the temporal and spatial variability of moisture sensors data to decide whether irrigation was needed. The soil moisture data is used to build a variogram

Table XVI: BLSTM parameters.

(Hyper-) parameter	Range	Chosen Value
dropout-rate	[0 - 0.6], step 0,2	0
number of epochs	[10 - 30], step 5	30
stacked layers	[2 - 5], step 1	4
window length	[1 - 3], step 1	1
hidden units	[32 - 512], step ×2	256

Table XVII: LSTM parameters.

(Hyper-) parameter	Range	Chosen Value
dropout-rate	[0 - 0.6], step 0,2	0
number of epochs	[10 - 30], step 5	30
stacked layers	[2 - 5], step 1	2
window length	[1 - 3], step 1	3
hidden units	[32 - 512], step × 2	256

Table XVIII: Comparison of the forecasting models using dataset II.

Model	RMSE	MAE
ARIMA	3.3221	2.8538
Linear Regression	1.8583	1.4608
M5P	1.8455	2.3349
SARIMA	4.2703	3.7076
Gradient Boosting	**0.1001**	**0.0681**
Random Forest	0.1248	0.0761
LSTM	1.7019	1.3175
BLSTM	1.6990	1.3142

Table XIX: Mean and Standard Deviation of observed ET_0 for dataset II.

Statistic	Value
Mean	6.871
Standard deviation	2.565

with Kriging interpolation that estimates the soil water deficit in an area. Sensors gather moisture data at different places in a field and, then, the known data is used to estimate the moisture in areas not monitored by the sensors.

Experiments performed by [Feng et al. 2017] demonstrated the Random Forest and Generalized Regression Neural Networks models as promising techniques for estimating daily ET_0 with complete or incomplete weather data in China. The incomplete data includes only temperature and extraterrestrial radiation. The results obtained using the Random Forest algorithm was slightly better than using Generalized Regression Neural Networks. In [Gocić et al. 2015], four models are analyzed to forecast ET_0 in Serbia: Genetic Programming, Support Vector Machine-firefly Algorithm, Artificial Neural Network, and Support Vector Machine-Wavelet. Experiments demonstrate the later algorithm is the best predictor of ET_0 for the analyzed dataset.

The papers [Caminha et al. 2017] and [Xavier 2016] are very similar to this work. They propose Machine Learning-based approaches to forecast ET_0 and potential evapotranspiration, respectively. They concluded that the models found out have great accuracy and were simpler than the Penman-Monteith method. [Hernandez et al. 2018] presents a comparative evaluation of the traditional models for time series: ARIMA and SARIMA. They address the problem of forecasting the occupancy time of the primary user in a Wi-Fi frequency band. They evaluated the performance of these two models with seven evaluation metrics. Overall, SARIMA models outperformed ARIMA.

Deep Learning Approaches. According to [Reichstein et al. 2019], deep learning approaches can automatically extract features to gain further process understanding of Earth system science problems, improving the predictive ability of seasonal forecasting and modeling of connections across multiple features. [Song et al. 2016] proposes a macroscopic cellular automata (MCA) model by combining

deep belief network (DBN) to predict the soil moisture content (SMC) over an irrigated corn field (an area of $22km^2$) in the Zhangye oasis, Northwest China. The widely used neural network, multi-layer perceptron (MLP), was utilized for comparison to DBN. The hybrid models (MLP-MCA and DBN-MCA) were calibrated and validated on SMC data within four months, i.e. June to September 2012. Compared with MLP-MCA, the DBN-MCA model led to a decrease in root mean squared error (RMSE) by 18%. The current results showed that the DBN-MCA model performs better than the MLP-MCA model, and the DBN-MCA model provides a powerful tool for predicting SMC in highly non-linear forms.

The paper [Saggi and Jain 2019] uses different prediction models to compare predictions of daily ET_0 at Hoshiarpur and Patiala Districts of Punjab (India). The performance of Deep Learning (DL)-Multilayer Perceptron model is compared to Generalized Linear Model (GLM), Random Forest (RF), and Gradient-Boosting Machine (GBM). The DL model showed high capabilities for estimation and performed much better than the original and calibrated RF, GLM, and GBM models. Moreover, the DL model has avoided the overfitting issue by giving more accuracy on training, validation, and testing, respectively, showing higher robustness than conventional approaches. Moreover, because modeling soil moisture by using environmental variables is gaining increasing popularity, DBN techniques could contribute a lot to enhancing the calibration of MCA-based SMC estimations and hence provide an alternative approach for SMC monitoring in irrigation systems on the basis of canals.

The paper [Cai et al. 2019] proposes a deep learning regression network (DNNR) with big data fitting capability to construct a soil moisture prediction model in Beijing area (China). The DNNR network consists of an input layer, multiple hidden layers, and an output layer. The number of layers can be adjusted according to the data scale. Corresponding hidden node and output layer activation functions can also be flexibly selected. The nodes are fully connected. Test results prove that the deep learning model is feasible and effective for soil moisture prediction. Its generalization capability can enrich the input characteristics while ensuring high accuracy in predicting the trends and values of soil moisture data providing an effective theoretical basis for water-saving irrigation and drought control.

The paper [Karim et al. 2019] proposes two deep learning models Multivariate LSTM-FCN (MLSTM-FCN) and Multivariate Attention LSTM-FCN (MALSTM-FCN) for multivariate time series classification. The proposed models require minimal preprocessing and feature extraction. The paper also uses a squeeze-and-excitation block to improves the accuracy of the models. In the paper [Souto et al. 2018], the authors propose an ensemble model to improve rainfall forecast over previous ensembles methods. The ensemble model is spatiotemporal aware based on deep neural networks (Convolutional Neural Networks (CNN) and LSTM).

[Patel et al. 2018] explored two deep learning models (BLSTM and 1D CNN) for precipitation nowcasting. Both models captured the temporal features over the time series datasets evaluated. However, the accuracy of BLSTM surpasses 1D CNN and confirms the promising performance of BLSTM in capturing time series aspects. The paper [Alhirmizy and Qader 2019] proposed an approach based on LSTM architecture, which aims to process multivariate environmental time-series data. The experiments were conducted in a different context, for forecasting Spain capital Madrid Air Quality.

Our work is different from these approaches, we have assessed different prediction models to forecast ET_0 and different datasets.

6. CONCLUSION

This paper compared the accuracy of univariate ARIMA and SARIMA models, and the deep learning stacked LSTM and BLSTM models with multivariate Machine Learning-based algorithms, Linear Regression, M5', Gradient Boosting and Random Forest. The results show that the multivariate models outperform the univariate techniques. However, it is worth to mention that multivariate

models might not be affordable for low-income farmers since the weather stations with several sensors are very costly and necessary to train the multivariate models. So, univariate models may be an affordable solution for low-income farmers, since only one variable needs to be monitored. And when there is enough data to train deep learning models, they can be a promising solution to predict ET_0 as shown in the experiments for dataset II.

As a future line of research, we plan to investigate other deep learning-based approaches as deep belief networks for big datasets from INMET. We will also compare the evaluated methods according to the running time, not only comparing the quality measures (RMSE and MAE).

7. ACKNOWLEDGMENT

The authors acknowledge FUNCAP for the research supported (process 178543144). The authors also acknowledge the financial support of CNPq (processes #432820/2016-7 and #311316/2018-2).

REFERENCES

Alhirmizy, S. and Qader, B. Multivariate time series forecasting with lstm for madrid, spain pollution. In *2019 International Conference on Computing and Information Science and Technology and Their Applications (ICCISTA)*. IEEE, pp. 1–5, 2019.

Allen, R. G., Pereira, L. S., Raes, D., Smith, M., et al. Crop evapotranspiration-guidelines for computing crop water requirements-fao irrigation and drainage paper 56. *FAO, Rome* 300 (9): D05109, 1998.

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13 (Feb): 281–305, 2012.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, New Jersey, USA, 2015.

Braga, D., da Silva, T. L. C., Rocha, A., Coutinho, G., Magalhaes, R. P., Guerra, P. T., and de Macêdo, J. A. Time series forecasting for purposes of irrigation management process, 2018.

Breiman, L. Bagging predictors. *Machine Learning* 24 (2): 123–140, 1996.

Breiman, L. Random forests. *Machine Learning* 45 (1): 5–32, 2001.

Brockwell, P. J. and Davis, R. A. *Introduction to time series and forecasting*. Springer, Switzerland, 2016.

Cai, Y., Zheng, W., Zhang, X., Zhangzhong, L., and Xue, X. Research on soil moisture prediction model based on deep learning. *PloS one* 14 (4): e0214508, 2019.

Caminha, H., Silva, T., Rocha, A., and Lima, S. Estimating reference evapotranspiration using data mining prediction models and feature selection. *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017)* vol. 1, pp. 272–279, 2017.

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 161–168, 2006.

Feng, Y., Cui, N., Gong, D., Zhang, Q., and Zhao, L. Evaluation of random forests and generalized regression neural networks for daily reference evapotranspiration modelling. *Agricultural Water Management* vol. 193, pp. 163 – 173, 2017.

Fontenelle, T., da Cunha, W., and Fuckner, M. *Atlas irrigação: uso da água na agricultura irrigada*. Agência Nacional de Águas, Brasília, DF, Brasil, 2017.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 2001.

Friedman, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38 (4): 367–378, 2002.

Goap, A. and Sharma, D. An iot based smart irrigation management system using machine learning and open source technologies. *Computers and Electronics in Agriculture* vol. 155, pp. 41–49, 11, 2018.

Gocić, M., Motamedi, S., Shamshirband, S., Petković, D., Ch, S., Hashim, R., and Arif, M. Soft computing approaches for forecasting reference evapotranspiration. *Comput. Electron. Agric.* 113 (C): 164–173, Apr., 2015.

Goldstein, A., Fink, L., Meitin, A., Bohadana, S., Lutenberg, O., and Ravid, G. Applying machine learning on sensor data for irrigation recommendations: Revealing the agronomistâĂŹs tacit knowledge. *Precision Agriculture* vol. 47, pp. 1–24, 05, 2017.

Hernandez, C., Giral, D., and Martinez, F. Radioelectric spectrum prediction based in arima and sarima time series models. *International Journal of Applied Engineering Research* 13 (22): 15688–15695, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation* 9 (8): 1735–1780, 1997.

James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*. Vol. 112. Springer, New York, USA, 2013.

Kamilaris, A. and Prenafeta-Boldu, F. X. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture* vol. 147, pp. 70–90, 2018.

Karim, F., Majumdar, S., Darabi, H., and Harford, S. Multivariate lstm-fcns for time series classification. *Neural Networks* vol. 116, pp. 237–245, 2019.

Laptev, N., Yosinski, J., Li, L. E., and Smyl, S. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*. Number 34. pp. 1–5, 2017.

Legates, D. R. and McCabe, G. J. Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water resources research* 35 (1): 233–241, 1999.

Montgomery, D. C., Peck, E. A., and Vining, G. G. *Introduction to linear regression analysis*. Vol. 821. John Wiley & Sons, 2012.

P. Avishek, P. P. *Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. Packt, Birmingham, UK, 2017.

Patel, M., Patel, A., Ghosh, D., et al. Precipitation nowcasting: Leveraging bidirectional lstm and 1d cnn. *arXiv preprint arXiv:1810.10485*, 2018.

Quinlan, J. R. et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*. Vol. 92. World Scientific, pp. 343–348, 1992.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. Deep learning and process understanding for data-driven earth system science. *Nature* 566 (7743): 195, 2019.

Saggi, M. K. and Jain, S. Reference evapotranspiration estimation and modeling of the punjab northern india using deep learning. *Computers and Electronics in Agriculture* vol. 156, pp. 387–398, 2019.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, 1998.

Siami-Namini, S. and Namin, A. S. Forecasting economics and financial time series: Arima vs. lstm. *arXiv preprint arXiv:1803.06386*, 2018.

Smith, T. G. *pmdarima: ARIMA estimators for Python*. MIT, USA, 2017.

Song, X., Zhang, G., Liu, F., Li, D., Zhao, Y., and Yang, J. Modeling spatio-temporal distribution of soil moisture by deep learning-based cellular automata model. *Journal of Arid Land* 8 (5): 734–748, 2016.

Souto, Y. M., Porto, F., Moura, A. M., and Bezerra, E. A spatiotemporal ensemble approach to rainfall forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8, 2018.

Sutton, C. D. Classification and regression trees, bagging, and boosting. *Handbook of statistics* vol. 24, pp. 303–329, 2005.

Tan, P.-N. et al. *Introduction to data mining*. Pearson Education India, India, 2006.

Tseng, F.-M. and Tzeng, G.-H. A fuzzy seasonal arima model for forecasting. *Fuzzy Sets and Systems* 126 (3): 367–376, 2002.

Wang, Y. and Witten, I. H. Induction of model trees for predicting continuous classes, 1996.

Xavier, F. Application of data science techniques in evapotranspiration estimation, 2016.

Zhang, M., Li, M., Wang, W., Liu, C., and Gao, H. Temporal and spatial variability of soil moisture based on WSN. *Mathematical and Computer Modelling* 58 (3-4): 826–833, 2013.

Zhou, Z.-H. *Ensemble methods: foundations and algorithms*. CRC press, 2012.