Improving Gated Recurrent Unit Predictions with Univariate Time Series Imputation Techniques

Anibal Flores[1]
Grupo de Investigación en Ciencia de Datos, Universidad Nacional de Moquegua, Moquegua, Perú

Hugo Tito[2]
E.P. Ingeniería de Sistemas e Informática, Universidad Nacional de Moquegua, Moquegua, Perú

Deymor Centty[3]
E.P. Ingeniería Ambiental Universidad Nacional de Moquegua Moquegua, Perú

***Abstract*—The work presented in this paper has its main objective to improve the quality of the predictions made with the recurrent neural network known as Gated Recurrent Unit (GRU). For this, instead of making different adjustments to the architecture of the neural network in question, univariate time series imputation techniques such as Local Average of Nearest Neighbors (LANN) and Case Based Reasoning Imputation (CBRi) are used. It is experimented with different gap-sizes, from 1 to 11 consecutive NAs, resulting in the best gap-size of six consecutive NA values for LANN and for CBRi the gap-size of two NA values. The results show that both imputation techniques allow improving prediction quality of Gated Recurrent Unit, being LANN better than CBRi, thus the results of the best configurations of LANN and CBRi allowed to surpass the techniques with which they were compared.**

***Keywords*—Gated recurrent unit; local average of nearest neighbors; case based reasoning imputation; GRU+LANN; GRU+CBRi**

I. Introduction

In working with time series, forecasting is one of the most exciting and interesting [1]. Today, Deep Learning [2] techniques are used for this type of task, in particular recurrent neural networks such as Long Short-Term Memory (LSTM) [3] and Gated Recurrent Unit (GRU) [4].

Several knowledge areas work with analysis and prediction of time series for different tasks, and greater accuracy are required in the estimated values in order to provide better services or products.

LSTM and GRU has been used successfully in many forecasting works [1], and the changes implemented to improve quality or reduce the error rate in predictions mainly includes tuning of parameters, input adjustments, number of layers, training epochs, batch size, etc.

As it can be seen in [1] the insertion of different NA values in the results produced by LSTM with different gap-sizes and the subsequent recalculation of the NA values produced an improvement in the precision of the LSTM results. Thus, in this work, it is experimented with the results produced by GRU, but instead of just using the Local Average of Nearest Neighbors (LANN) [5] imputation technique, we also experiment with another technique known as Case Based Reasoning Imputation (CBRi) "in press" [6].

Similarly to the analysis performed in [1] for LSTM predictions, Fig. 1 shows a 14-day prediction analysis for Gated Recurrent Unit (GRU). In the first case, the LANN algorithm is applied to a 1 NA gap-size by rebuilding the elements 2, 4, 6, …, 12 of the GRU-predicted series in order to outperform the results. In the second case, LANN is applied for the elements 3, 5, 7, …, 13 in order to improve the results produced by GRU. How it can be appreciated GRU results are improve just in the second case.

The analysis performed in Fig. 1 with the application of the LANN algorithm and the improvement detected on GRU predictions motivate the application of another imputation algorithm known as Case Base Reasoning Imputation (CBRi) "in press" [6].

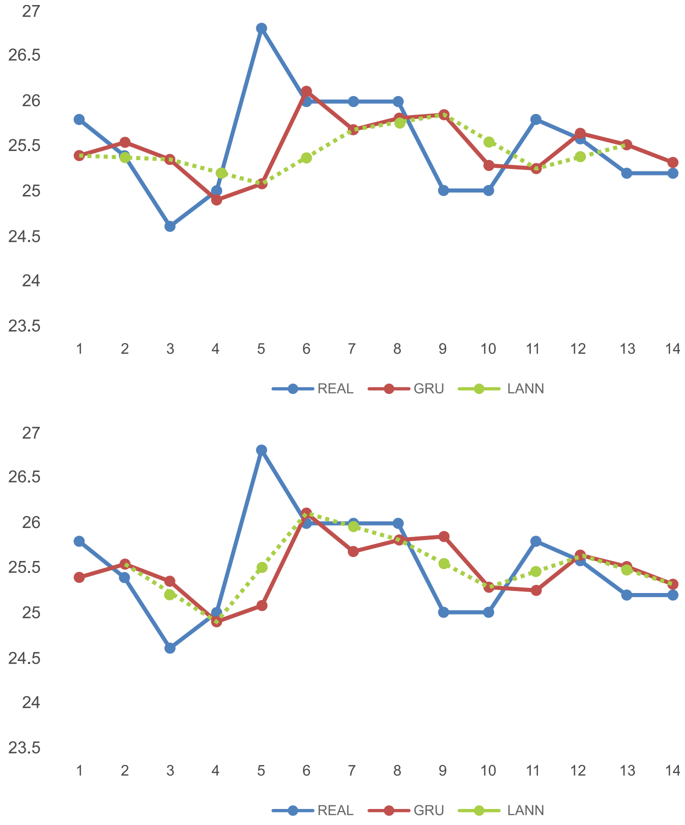


Fig. 1. Improving GRU Predictions with LANN.

The data in [1] is used for analysis and experimentation, it corresponds to maximum temperatures in a meteorological station known as Punta de Coles located in the Moquegua region in south of Peru which is highly seasonal.

The results achieved in this work show that, in the first case, LANN in [1] allowed to considerably improve the results provided by Gated Recurrent Unit (GRU). In the second case, it is experimented with CBRi, taking as a reference the results shown "in press" [6], where CBRi manages to overcome LANN results, however, in this work GRU+CBRi does not surpass GRU+LANN.

This paper has been structured as follows: The second section shows the related work that serves as a point of comparison for the results achieved in this work. In the third section, some concepts that will allow a better understanding of the content of the paper are briefly described. The fourth section shows the process followed for the experiments carried out. In the fifth section, the results achieved after the experiments are shown. In the fifth section, the results are briefly discussed compared to other state-of-the-art techniques. Then, in the sixth section, the conclusions reached at the end of this work are described. And finally, it describes the future work that can be done from the work presented in this paper.

II. Related Work

A. Simple Linear Regression

In the implementation of regression models, one of the most basic and widely used models is known as linear regression. This type of regression consists of a statistical analysis to identify the relationship between two variables, the dependent and the independent one [7]. Equation for Simple Linear Regression can be seen in (1).

$$y(t) = \beta_0 + \beta_1 x_t + \varepsilon_t \tag{1}$$

B. ARIMA

ARIMA [8] stands for Autoregressive Integrated Moving Average. It is a statistical model that works with variations and regressions of time series to find patterns that are used to predict future values. It is a dynamic model, where future estimates are explained by past data and not by independent variables.

Next, some works that implement ARIMA to predict future values in different time series are described.

In [9] the authors worked with wind speed time series and for this work they implemented an ARIMA model and a NNT Back Propagation Neural Network. The results show that the ARIMA model is slightly higher than the Back Propagation Neural Network model.

In [10] to predict the number of epidemic disease, the authors proposed an ARIMA model, which was compared with the results of a model based on Simple Moving Average (SMA). At the end, the ARIMA predictions were better than the SMA ones.

In [11] the authors implement ARIMA and Support Vector Machine (SVM) models to forecast load time series. The results show that ARIMA is better for linear type of load, while SVM is better for non-linear type of load time series.

In [11] the authors implement two models to forecast linear and non-linear load time series, these are ARIMA and Support Vector Machine (SVM). The authors concluded that the ARIMA results are better for the linear load type, while the SVM results are better for the non-linear load type.

C. Prophet

Prophet [12] is a very known forecasting technique developed by the Facebook data science team and it is a forecasting decomposable time series model, it has three main model components: trend, seasonality, and holidays which are combined in equation (2).

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{2}$$

Where: g(t) is the trend function, s(t) represents periodic changes and h(t) represents the holidays, ϵ_t represents the error.

Some works that used prophet are briefly described below:

In [13] the authors propose the forecasting of bitcoin time series using the ARIMA and Prophet techniques. The results achieved show that Prophet is superior in terms of the accuracy of the results with respect to ARIMA.

In [14] the authors propose the forecasting of groundwater-level time series using the Prophet technique. The results are compared with other techniques such as ARIMA, Linear Regression, and others showing that Prophet offers better accuracy for this type of time series.

In [15] the authors propose the forecasting of time series of microclimate temperatures corresponding to greenhouses using the Prophet technique. Unfortunately, they do not show a comparison of the results with other techniques.

D. Long Short-Term Memory (LSTM)

The LSTM recurrent networks were created in order to address the problem of vanishing gradients, due to the unfold process of a Recurrent Neural Network.

LSTM networks work with special hidden units, whose objective is to remember input data for a long time [2]. Then, it has been proved that LSTM networks are more effective than conventional RNNs [2]. This is because LSTM networks have several layers for each time step. Fig. 2 shows the LSTM architecture.

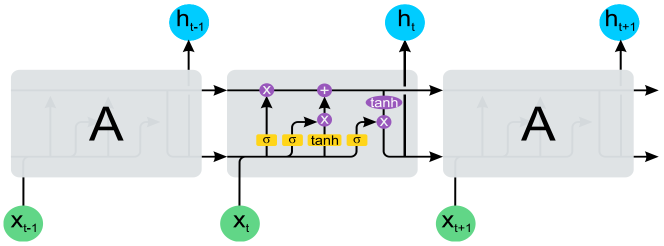


Fig. 2. Architecture of LSTM Network.

Some LSTM works are briefly described below:

In [1], the authors proposed the use of LANN and LANNc to improve the predictions of LSTM in meteorological time series corresponding to maximum temperatures. The results show that on average LANN improves the precision of the LSTM predictions in all the NA values cases: from 1 to 11 NA values.

In [16], the authors proposed the use recurrent neural networks such as LSTM and GRU to forecast electric load time series. The results of the work show that GRU is better than LSTM in terms of the accuracy of the predictions.

In [17] the authors proposed the use of LSTM and GRU networks for forecasting of traffic flow time series, comparing the results achieved with an ARIMA model. The LSTM and GRU results are better than ARIMA for this type of time series.

In [18] the authors propose the forecasting of power load time series for a residential community using Gated Recurrent Unit (GRU). The results obtained from GRU are compared with LSTM results in different settings and show that for this type of time series, the accuracy achieved by GRU is better than that of LSTM.

In [19] the use of a multilayer recurrent neural network called MS-GRU is proposed for forecasting of load electricity time series. The results are compared with LSTM and GRU networks, showing that MS-GRU has better precision than LSTM and GRU.

III. Background

A. *Recurrent Neural Networks (RNN)*

An RNN is a type of neural network that allows modeling time series [3]. The structure of this neural network is similar to that of an MLP (Multilayer Perceptron). This differs from an MLP in the sense that it allows connections between hidden units associated with a time delay. Through these connections, the RNN can retain and remember information from the past [20], allowing it to find temporary correlations between facts that may be far apart in time. Fig. 3 shows the unfolded architecture of an RNN.

The task of training an RNN is difficult to carry out [3] due to the problems of vanishing and exploding gradients. It resulted in the appearance or creation of recurrent neural networks known as LSTM that solve the problems mentioned above.

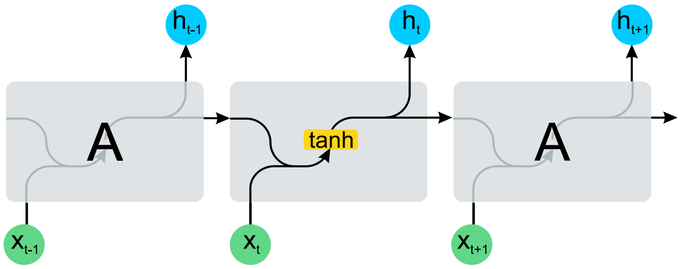


Fig. 3. Architecture of Recurrent Neural Network.

B. *Gated Recurrent Unit (GRU)*

GRUs are a gating mechanism in RNNs introduced by K. Cho et al [4] in 2014. GRU is a variation on the LSTM because both are designed similarly, this can be seen in its architectures. The Gated Recurrent Unit (GRU) has fewer parameters than LSTM, since it lacks an output gate. LSTM is stronger than the GRU, since it can easily perform an unlimited count, while the GRU cannot, that is the reason why GRU fails to learn certain languages that LSTM can [21]. However, in forecasting of univariate time series in many works such as [1], [16], [17], [22], [23], etc. GRU has shown better precision than LSTM, that is why it was chosen as the basis for the present study.

Fig. 4 shows one of the most commonly used GRU architectures.

According to Fig. 3 the following equations:

$$Z_{t=}\sigma_g(W_z x_t + U_z h_{t-1} + b_z) \quad (3)$$

$$r_{t=}\sigma_g(W_r x_t + U_r h_{t-1} + b_r) \quad (4)$$

$$h_{t=}(1 - z_t) \text{ o } h_{t-1} + z_t \text{ o } \sigma_h(W_h x_t + U_h(r_t \text{ o } h_{t-1}) + b_h \quad (5)$$

Where:

x_t : input vector

h_t : output vector

z_t : updated gate vector

r_t : reset gate vector

W,*U* and *b* : matrix parameters and vector

σ_g : sigmoid function

σ_h : hyperbolic tangent

C. *Local Average of Nearest Neighbors (LANN)*

LANN [5] is a fairly simple but very effective algorithm for univariate time series imputation. LANN consists in the calculation of the average of the prior and next values of a block of NA values in a time series according equation (6).

NA=(πριορ + νεξτ)/2 (6)

LANN produces good imputation results since according to the analysis performed in [5]; on average the closest values to a missing value are its closest neighbors, the prior and the next values, therefore the average of these two values produces a very close value to the NA value.

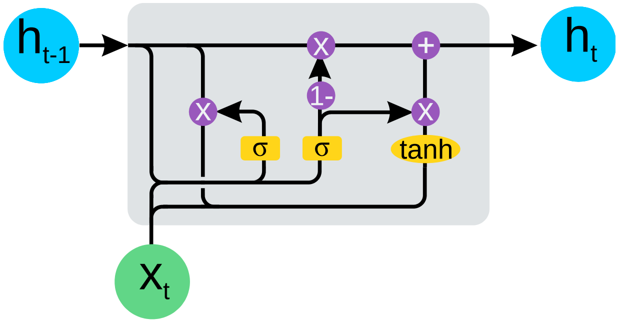


Fig. 4. Architecture of GRU.

D. Case based Reasoning Imputation (CBRi)

CBRi "in press" [6] is an imputation technique for univariate time series inspired by Case Based Reasoning that allows to calculate a NA value from a base of cases that stores historical values of a time series.

Fig. 5 graphically shows the architecture of the CBRi system. Hence, initially in the Time Series block, a base of cases is created from a historical time series; the base of cases is a matrix where the rows correspond to the prior values and the columns corresponds to the next values; every matrix cell contains values found for each prior and next value in the historical time series. The CBRi block receives as input a vector with NA values and calculates each NA value from its prior and next values using equation (7). The Testing block allows evaluating the quality of the estimated NA values.

$$NA = \frac{\sum_{i=0}^{n-1}(prior+Vi+next)}{n*3} \tag{7}$$

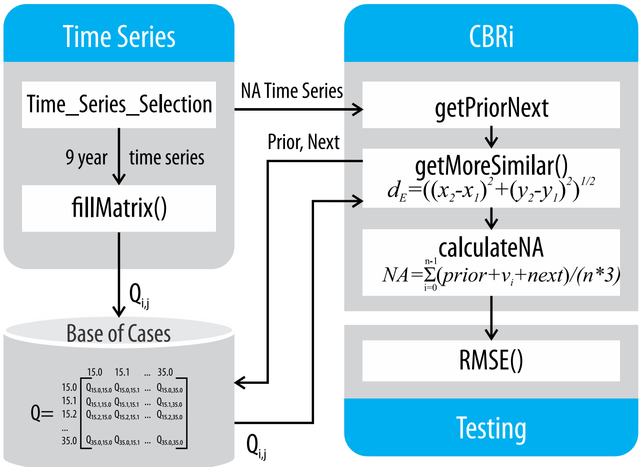

Time Series
Time_Series_Selection
9 year
time series
fillMatrix()
$Q_{i,j}$
Base of Cases
$$Q= \begin{array}{c|ccccc} & 15.0 & 15.1 & ... & 35.0 \\ \hline 15.0 & Q_{15.0,15.0} & Q_{15.0,15.1} & ... & Q_{15.0,35.0} \\ 15.1 & Q_{15.1,15.0} & Q_{15.1,15.1} & ... & Q_{15.1,35.0} \\ 15.2 & Q_{15.2,15.0} & Q_{15.2,15.1} & ... & Q_{15.2,35.0} \\ ... & & & & \\ 35.0 & Q_{35.0,15.0} & Q_{35.0,15.1} & ... & Q_{35.0,35.0} \end{array}$$
$Q_{i,j}$
NA Time Series
Prior, Next
CBRi
getPriorNext
getMoreSimilar()
$d_E=((x_2-x_1)^2+(y_2-y_1)^2)^{1/2}$
calculateNA
$NA=\sum_{i=0}^{n-1}(prior+v_i+next)/(n*3)$
RMSE()
Testing

Fig. 5. CBRi System [6].

IV. Process

A. Time Series Selection

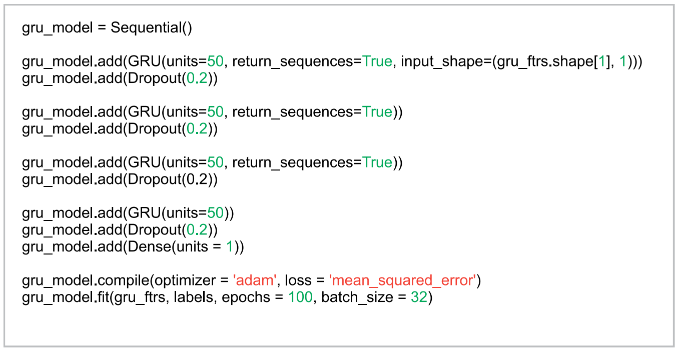
The time series chosen for experimentation corresponds to maximum daily temperatures that were recorded at the SENAMHI[1] Punta de Coles meteorological station in Ilo city in south of Peru. The data that is used for the training phase correspond to 4 years (from 2012-01-01 to 2015-12-31) and the data that is used for testing correspond to 2016.

B. GRU Model

The GRU recurrent neural network architecture used in this work is shown in Fig. 6.

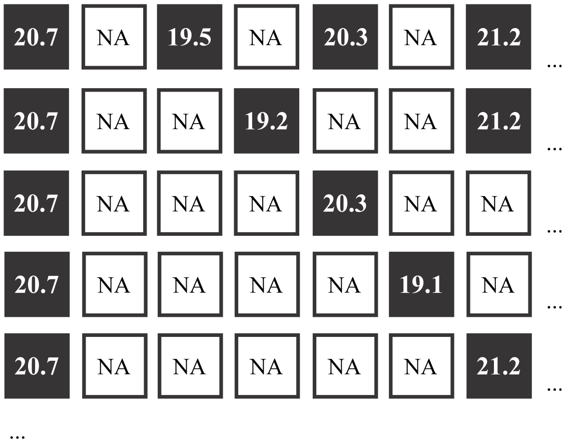
C. Inserting NAs

Once the previous phase is completed with 180 days predicted, the NA values are inserted using the same strategy that was used in [1], as shown in Fig. 7.



```
gru_model = Sequential()

gru_model.add(GRU(units=50, return_sequences=True, input_shape=(gru_ftrs.shape[1], 1)))
gru_model.add(Dropout(0.2))

gru_model.add(GRU(units=50, return_sequences=True))
gru_model.add(Dropout(0.2))

gru_model.add(GRU(units=50, return_sequences=True))
gru_model.add(Dropout(0.2))

gru_model.add(GRU(units=50))
gru_model.add(Dropout(0.2))
gru_model.add(Dense(units = 1))

gru_model.compile(optimizer = 'adam', loss = 'mean_squared_error')
gru_model.fit(gru_ftrs, labels, epochs = 100, batch_size = 32)
```

Fig. 6. Architecture for GRU Model in Python.



20.7	NA	19.5	NA	20.3	NA	21.2	...
20.7	NA	NA	19.2	NA	NA	21.2	...
20.7	NA	NA	NA	20.3	NA	NA	...
20.7	NA	NA	NA	NA	19.1	NA	...
20.7	NA	NA	NA	NA	NA	21.2	...
...							

Fig. 7. NA Values in the GRU Predicted Time Series [1].

D. Applying LANN/CBRi

LANN algorithm is used to impute the NA values in the predicted time series according adaptation made in [1] for over 2 NA values.

CBRi algorithm is used to impute NA values according adaptation made in [24] for over 2 NA values. Base of cases corresponds to 9 years (from 2007 to 2015) and it was implemented for CBRi "in press" [6].

E. Evaluation

The results of both algorithms are evaluated through Root Mean Squared Error (RMSE) according equation (4)

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n-1}(Pi-Ri)^2}{n}} \tag{8}$$

V. Results

After experimentation, this section shows and describes the results achieved. LANN and CBRi were implemented with different configurations of NA values between 1 to 11 and Table I shows the corresponding RMSE values.

According to Table I, it can be seen that on average both imputation techniques offer their best results between 2 and 6 consecutive NA values, managing to improve GRU results for all cases.

Regarding the first case of insertion of only 1 NA value, on average LANN manages to improve GRU results, however, CBRi fails to improve GRU results.

[1] https://www.senamhi.gob.pe/

TABLE. I. GRU, LANN AND CBRI

Technique		**RMSE of Predicted Days**							**Avg**
		15	**30**	**60**	**90**	**120**	**150**	**180**	
GRU		0.5953	0.6917	0.6678	0.6689	0.7076	0.6751	0.6727	0.6684
LANN	1	0.6335	0.6770	0.6340	0.6500	0.6980	0.6558	0.6421	0.6557
	2	**0.5235**	0.6169	0.6064	0.5906	0.6230	0.5880	0.5904	0.5912
	3	0.6071	0.6349	0.6049	0.6010	0.6175	0.5858	0.5961	0.6067
	4	0.5743	0.6598	0.5973	0.5855	**0.6028**	**0.5652**	**0.5891**	0.5962
	5	0.5371	0.6381	0.6174	0.6007	0.6209	0.5853	0.5941	0.5990
	6	0.5287	**0.5759**	**0.5459**	**0.5870**	0.6359	0.6016	0.6197	**0.5849**
	7	0.5553	0.7261	0.6782	0.6435	0.6703	0.6278	0.6302	0.6473
	8	0.5979	0.7271	0.6798	0.6613	0.6844	0.6493	0.6582	0.6654
	9	0.6066	0.6961	0.7451	0.7134	0.7525	0.7080	0.6973	0.7027
	10	0.5480	0.6632	0.6885	0.6903	0.7343	0.7357	0.7636	0.6890
	11	0.5520	0.6616	0.7029	0.6944	0.6968	0.6977	0.7342	0.6770
CBRi	1	0.6278	0.7049	0.6516	0.6599	0.7359	0.6887	0.6738	0.6775
	2	**0.5257**	0.6199	**0.5951**	**0.5900**	0.6386	0.6053	0.6110	**0.5979**
	3	0.6349	0.6457	0.6245	0.6185	**0.6317**	0.5972	0.6115	0.6234
	4	0.5480	0.6757	0.6081	0.6080	0.6231	**0.5870**	**0.6082**	0.6083
	5	0.5831	0.6348	0.6093	0.6107	0.6304	0.5949	0.6113	0.6106
	6	0.5494	**0.5800**	0.6731	0.6779	0.7010	0.6643	0.6746	0.6457
	7	0.8082	0.7443	0.6872	0.6377	0.6690	0.6219	0.6362	0.6863
	8	0.6949	0.7484	0.7688	0.7336	0.7417	0.7006	0.6902	0.7254
	9	0.6654	0.6837	0.7436	0.7077	0.7403	0.6981	0.7471	0.7122
	10	0.5992	0.7597	0.6744	0.6778	0.7321	0.7231	0.7471	0.7019
	11	0.5677	0.7330	0.7873	0.7406	0.7308	0.7308	0.7573	0.7210

Regarding the NA values greater than 6 LANN on average of 5 cases of NAs greater than 6 in two cases it improves the GRU results and in 3 of them it worsens them; with CBRi for the 5 cases of NAs greater than 6 in all of them worsens the GRU results.

According to Fig. 8, it can be seen how the best configuration of NA values for each imputation technique (LANN: 6 consecutive NAs and CBRi: 2 consecutive NAs) allows to improve the accuracy of GRU predictions on different amounts of predicted days.

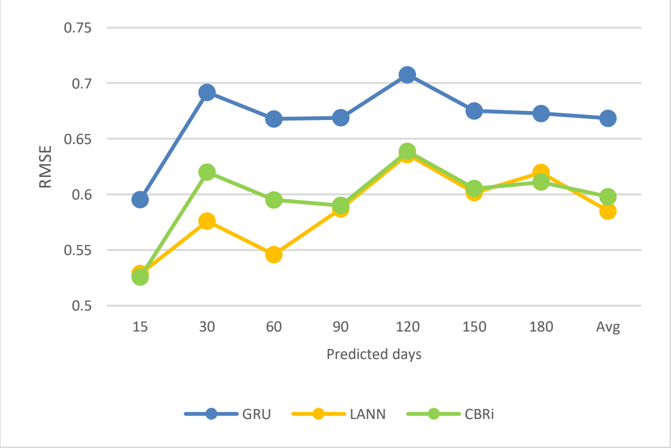


Fig. 8. GRU, GRU+LANN and GRU+CBRi.

VI. DISCUSSION

Next, the results achieved in the previous section are compared with other prediction techniques. It can be seen in Table II.

According to Table II, on average it can be seen how the GRU + LANN combination offers the most accurate results for the predicted time series, it can be seen that from 7 cases in 5 of them, GRU+LANN outperforms other techniques including GRU+CBRi, LSTM, LSTM+LANN, LSTM+LANNc among others. Only in two cases GRU+CBRi manage to surpass GRU+LANN. It demonstrates the importance of the use of univariate time series imputation techniques in the improvement of GRU results for this type of time series.

Fig. 9 clearly shows the difference between the proposals compared to other forecasting techniques.

At this point, it must be highlighted that before applying imputation techniques to GRU prediction results in order to improve them, it is important to take a prior analysis with a simple technique as Local Average of Nearest Neighbors (LANN) as it is showed in Fig. 1 to determine if imputation techniques can really improve GRU results.

TABLE. II. COMPARISON WITH ANOTHER TECHNIQUES

Technique	RMSE of Predicted Days							Avg
	15	30	60	90	120	150	180	
GRU+LANN*	0.5287	**0.5759**	**0.5459**	**0.5870**	0.6359	0.6016	0.6197	**0.5849**
GRU+CBRi**	**0.5257**	0.6199	0.5951	0.5900	0.6386	0.6053	**0.6110**	**0.5979**
GRU	0.5953	0.6917	0.6678	0.6689	0.7076	0.6751	0.6727	0.6684
LSTM	0.6334	0.6637	0.6702	0.7175	0.7649	0.7537	0.7562	0.7085
LSTM+LANN	0.6296	0.6111	0.6097	0.6730	0.7059	0.6910	0.6838	0.6577
LSTM+LANNc	0.5452	0.5918	0.6377	0.6813	0.7302	0.7166	0.7216	0.6606
PROPHET	0.5512	0.7054	1.0516	1.1637	1.1274	1.1274	1.0403	1.0279
ARIMA	0.6134	1.2988	2.2932	2.5240	2.2320	2.2320	2.0440	2.1639

* 6 NAs ** 2 NAs

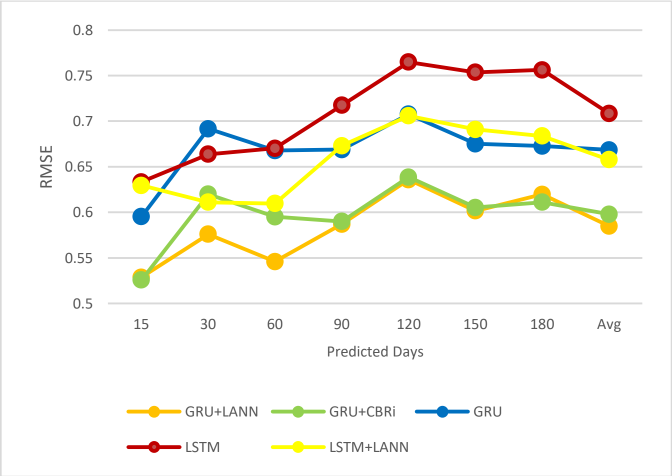


Fig. 9. Comparison with another Techniques.

VII. CONCLUSIONS

The use of univariate time series imputation techniques allows improving the accuracy of the predictions of the Gated Recurrent Unit (GRU).

Of the two imputation techniques experienced in this work, Local Average of Nearest Neighbors (LANN) showed superiority over Case Based Reasoning imputation (CBRi), of seven cases analyzed, LANN was superior to CBRi in five of them, so LANN is highly recommendable for this kind of tasks in this type of time series.

VIII. FUTURE WORK

In this work the results of GRU in time series of maximum temperatures were improved, it would be interesting to analyze if GRU predictions in another kind of time series can be improved in similar or a better way.

In addition, it would be important to determine which is the most appropriate gap-size to obtain the best results for each portion of the time series, since in this work it has been possible to appreciate that the same gap-size does not produce the best results in the entire time series.

REFERENCES

[1] Flores, H. Tito & D. Centty, “Improving Long Short-Term Memory predictions with Local Average of Nearest Neighbors,” (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 10, nº 11, pp. 392-397, 2019.

[2] Y. LeCun, Y. Bengio & G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436-444, 2015.

[3] R. Pascanu, T. Mikolov, Y. Bengio, “On the dificulty of training recurrent neural networks,” de 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.

[4] C. Kyunghyun, V. Bart, G. Caglar, B. Dzmitry, B. Fethi, S. Holger & B. Yoshua, “Learning phrase representations using RNN enconder-decoder for statistical machine traslation,” arxiv.org, pp. 1-15, 2014.

[5] A. Flores, H. Tito and C. Silva, “Local average of nearest neighbors: univariate time series imputation,” International Journal of Advanced Computer Science and Applications, vol. 10, nº 8, pp. 45-50, 2019.

[6] A. Flores, H. Tito & C. Silva, “CBRi: a case based reasoning-inspired approach for univariate time series imputation. In Press,” de 6th IEEE Latin American Conference on Computational Intelligence LA-CCI, Guayaquil, Ecuador, 2019.

[7] S. Kavitha, S. Varuna and R. Ramya, “A comparative analysis on linear regression and support vector regression,” de Online International Conference on Green Engineering and Technologies, Coimbatore, India, 2016.

[8] R. Hyndman & G. Athanasopoulos, Forecasting: principles and practice, Melbourne, Australia: OTexts, 2018.

[9] J. Palomares, J. de la Rosa, J. Ramiro, J. Melgar, A. Agüera & A. Moreno, “ARIMA vs neural networks for wind speed forecasting,” de International Conference of Computational Intelligence for Measurement Systems and Applications, Hong Kong, China, 2009.

[10] Y. Pan, M. Zhang, Z. Chen, M. Zhou, Z. Zhang, “An ARIMA based model for forecasting the patient number of epidemic disease,” de 13th International Conference on Service Systems and Service Management, Kunming, China, 2016.

[11] M. Abdullah, A. Hoque, “Comparison of ARIMA and SVM for short-term load forecasting,” de Annual Information Technology, Electromechanical Engineering and Microelectronics Conference, Jaipur, India, 2019.

[12] S. Taylor & B. Letham, “Forecasting at scale,” PeerJPreprints, pp. 1-25, 2017.

[13] I. Yenidogan, A. Cayir, O. Kozan, T. Dag & C. Arslan, “Bitcoin forecasting using ARIMA and Prophet,” de 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia-Herzegovina, 2018.

[14] H. Aguilera, C. Guardiola, N. Naranjo & C. Kohfahl, “Towards flexible groundwater-level prediction for adaptive water management: using Facebook's Prophet forecasting approach,” Hydrological Sciences Journal, 2019.

[15] Z. Oo, S. Phyu, “Microclimate prediction using cloud centric model based on IoT technology for sustainable agriculture,” de IEEE 4th International Conference on Computer and Communication Systems, Singapore, Singapore, 2019.

[16] S. Kumar, L. Hussain, S. Banarjee & M. Reza, “Energy load forecasting using deep learning approach-LSTM and GRU in spark cluster,” de Fifth

International Conference on Emerging Applications of Information Technology (EAIT), Kolkata, India, 2018.

[17] R. Fu, Z. Zhang & L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," de 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 2016.

[18] J. Zheng, X. Chen, K. Yu, L. Gan, Y. Wang & K. Wang, "Short-Term power load forecasting of residential community based on GRU neural network," de International Conference on Power System Technology, Guangzhou, China, 2018.

[19] L. Kuan, Z. Yan, W. Xin, S. Wenxue, J. Zhe, Z. Young, X. Nan, Z. Xing, "Short-term electricity load forecasting method based on multilayered self-organizing GRU network," de IEEE Conference on Energy Internet and Energy System Integration, Beijing, China, 2017.

[20] M. Paco, C. López Del Alamo & R. Alfonte, "Forecasting of meteorological weather time series through a feature vector based on correlation," de 18th International Conference Computer Analysis of Images and Patterns CAIP 2019, Salerno, Italy, 2019.

[21] W. Gail, G. Yoav & Y. Eran, "On the practical computational power of finite precision RNNs for language recognition," arxiv.org, pp. 1-9, 2018.

[22] B. Wang, W. Kong, H. Guan & N. Xiong, "Air quality forecasting based on gated recurrent long short term memory model in internet of things," IEEE Access, vol. 7, pp. 69524 - 69534, 2019.

[23] A. Tokgoz & G. Unal, "A RNN based time series approach for forecasting turkish electricity load," de 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2018.

[24] A. Flores, H. Tito & C. Silva, "CBRm: case based reasoning approach for imputation of medium gaps," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 10, nº 9, pp. 376-382, 2019.