Improving Long Short-Term Memory Predictions with Local Average of Nearest Neighbors

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 study presented in this paper aims to improve the accuracy of meteorological time series predictions made with the recurrent neural network known as Long Short-Term Memory (LSTM). To reach this, instead of just making adjustments to the architecture of LSTM as seen in different related works, it is proposed to adjust the LSTM results using the univariate time series imputation algorithm known as Local Average of Nearest Neighbors (LANN) and LANNc which is a variation of LANN, that allows to avoid the bias towards the left of the synthetic data generated by LANN. The results obtained show that both LANN and LANNc allow to improve the accuracy of the predictions generated by LSTM, with LANN being superior to LANNc. Likewise, on average the best LANN and LANNc configurations make it possible to outperform the predictions reached by another recurrent neural network known as Gated Recurrent Unit (GRU).**

***Keywords*—Long Short-Term Memory; Local Average of Nearest Neighbors; univariate time series prediction; LANN; LANNc; Gated Recurrent Unit; GRU**

I. Introduction

Forecasting is one of the most exciting subfields in the field of time series. Since the beginning, forecasting techniques have evolved greatly from simple linear regressions, passing for moving averages, autoregressive models, machine learning models, until reach Deep Learning [1] techniques.

Within Deep learning, for forecasting activities, recurrent neural networks are very common, and within them Long Short-Term Memory [2] and the Gated Recurrent Unit [3].

Long Short-Term Memory in many forecasting works has been used successfully, and the changes implemented to improve or reduce the error rate mainly includes input adjustments, tunning of parameters, number of layers, training epochs, etc.

After analyzing and evaluate the prediction results of an LSTM model in a 4-year meteorological time series corresponding to maximum temperatures, it was observed that various synthetic values could better approximate their real values through imputation processes. Fig. 1 shows in most cases how the imputation of a single value can improve the error rate of an estimated value with LSTM.

Thus, in this paper, it is proposed to improve the error rate of LSTM predictions through univariate time series imputation algorithms, such as: Local Average of Nearest Neighbors (LANN) [4] and a variation of this that will be called LANNc. LANNc tries to solve the problem of bias to the left of the synthetic data generated by LANN, just as CBRm [5] does with CBRi "in press" [6].

For the experimentation of the proposals, gap-sizes in the range of 1 to 11 consecutive NA values are evaluated, also different amounts of predicted values 15, 30, 60, 90, 120 and 150 days are considered. The results achieved show that the algorithms used allow to improve the error rate of LSTM.

It is important to highlight that the study is limited to the prediction of highly seasonal meteorological time series as it is the case of maximum temperature time series.

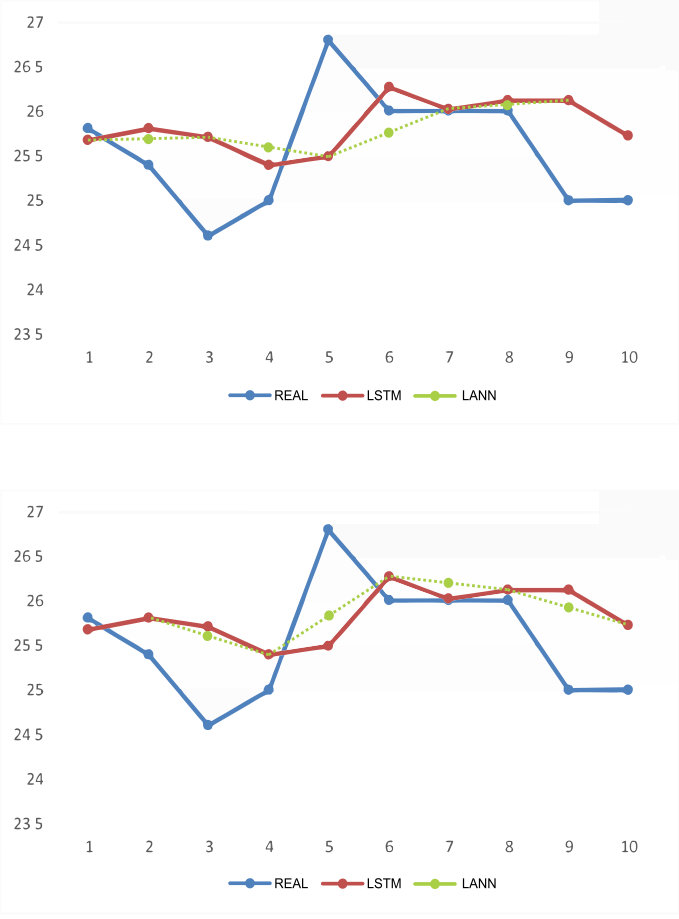

Fig. 1. How LANN can Improve LSTM Predictions.

The content of this paper is organized as follows: In Section II, a brief review of the state of the art in relation to forecasting techniques is shown; in Section III, some concepts and definitions are included that will allow a better understanding of the paper content; in Section IV, the implementation of the proposal is described; then, in Section V, the results achieved are described and analyzed; in Section VI, the results achieved are compared with other proposals of the state of the art; in Section VII, the conclusions reached at the end of this work are described; and finally it shows the future work that can be implemented to improve the results achieved.

II. Related Work

A. Simple Linear Regression

In the field of regression, one of the first models used for forecasting is known as linear regression. It consists in a statistical analysis to identify the relationship between the dependent and independent variables [7]. Simple Linear Regression is given by equation (1)

$$y(t) = \beta_0 + \beta_1 x_t + \varepsilon_t \quad (1)$$

B. ARIMA

ARIMA [8] (Autoregressive Integrated Moving Average) is a statistical model that uses variations and regressions of statistical data in order to find patterns for a prediction into the future. It is a dynamic time series model, that is, future estimates are explained by past data and not by independent variables.

Some works that implement ARIMA for forecasting are briefly described below:

In [9] the authors implement the ARIMA model and the NNT Back Propagation Neural Network to forecast wind speed time series. The results show a slight superiority of the ARIMA model relative to the model based on neural networks.

In [10] the authors of the paper propose an ARIMA model to predict the number of epidemic disease for a center of disease control and prevention. The results achieved are compared with those of a model based on Simple Moving Average (SMA) showing the superiority of ARIMA over SMA.

In [11] the authors implement ARIMA and Support Vector Machine 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.

C. Prophet

Prophet [12] is a forecasting decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in equation (2).

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (2)$$

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

D. Gated Recurrent Unit (GRU)

GRUs are a gating mechanism in recurrent neural networks [3]. The GRU is like a long-term memory with a forget gate and with fewer parameters than LSTM, since it lacks an output gate. GRUs have been shown to exhibit even better performance in certain smaller datasets. However, the LSTM is "strictly stronger" than the GRU, since it can easily perform an unlimited count, while the GRU cannot [13]. Fig. 2 shows the architecture of GRU.

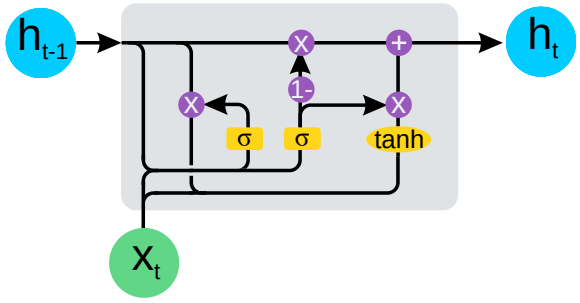

Fig. 2. Architecture of GRU.

Some works that implement GRU are briefly described below:

In [14], the authors propose the use of LSTM and GRU recurrent neural networks for electric load time series forecasting. The results achieved show the superiority of the GRU results over the results achieved by LSTM.

In [15] the authors propose the use of LSTM and GRU for traffic flow prediction comparing the results achieved with ARIMA model, demonstrating the superiority of the recurrent neural networks in this type of time series.

In [16] the authors propose the power load forecasting of residential community using Gated Recurrent Unit (GRU). The GRU results are compared with LSTM results in different configurations. GRU proved to be more efficient than LSTM for this type of time series.

In [17] the authors propose the use of a multilayer recurrent neural network called MS-GRU to forecast load electricity time series. The results are compared with the traditional recurrent neural networks such as LSTM and GRU, showing greater precision in the proposal MS-GRU.

III. Background

A. Recurrent Neural Networks (RNN)

A recurrent neural network is a neural network model for modeling time series [2]. The structure of this type of network is very similar to that of a standard multilayer perceptron (MLP), with the difference that it allows connections between hidden units associated with a time delay. Through these connections, the model can retain information from the past [18], allowing it to discover temporal correlations between events that may be very far from each other. Fig. 3 shows the architecture of a recurrent neural network.

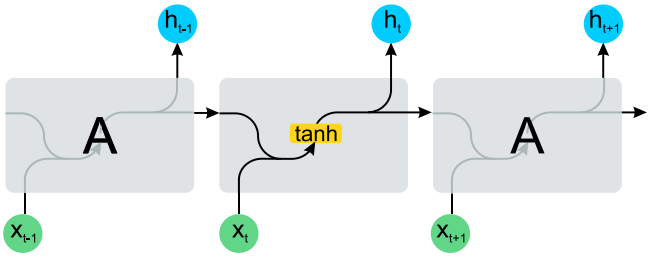

Fig. 3. Architecture of Recurrent Neural Network.

Recurrent neural networks are difficult to train [2] due to the problems of vanishing and exploding gradients, these problems resulted in the creation of LSTM networks.

B. Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) network was created with the goal of addressing the vanishing gradients problem, this is due to the unfold process of an RNN. LSTM networks use special hidden units, whose task is to remember entries for a long time [1]. LSTM networks have subsequently proved to be more effective than conventional RNNs [1], especially when they have several layers for each time step. Fig. 4 shows the LSTM architecture.

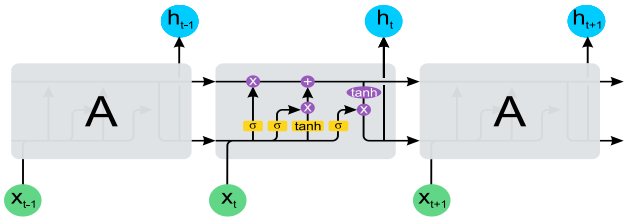

Fig. 4. Architecture of LSTM Network.

C. Local Average of Nearest Neighbors (LANN)

Local Average of Nearest Neighbors (LANN) [4] is a very simple algorithm for univariate time series imputation that uses the mean of the prior and next value of a block of NAs to replace the missing or NA values according equation (3).

$$NA=(\pi\rho\iota o\rho + \nu\varepsilon\xi\tau)/2 \quad (3)$$

LANN produces very good results in the imputation processes because, according to the analysis carried out in [4] the values closest to the missing values correspond to prior and next.

IV. Process

A. Selection of Time Series

The selected time series corresponds to maximum daily temperatures at the SENAMHI[1] meteorological station known as Punta de Coles[2] in the city of Ilo-Peru. The data that will be used for the training correspond to 4 years (2012-2015) and the data that will be used for testing correspond to the year 2016.

B. LSTM Model

The architecture that will be used to implement LSTM is shown in Fig. 5.

[1] https://www.senamhi.gob.pe/
[2] Lat.: 17°41'55.2"S Long.:71°22'25"W Alt.: 25 msnm

```
lstm_model = Sequential()

lstm_model.add(LSTM(units=50, return_sequences=True, input_shape=(lstm_ftrs.shape[1], 1)))
lstm_model.add(Dropout(0.2))

lstm_model.add(LSTM(units=50, return_sequences=True))
lstm_model.add(Dropout(0.2))

lstm_model.add(LSTM(units=50, return_sequences=True))
lstm_model.add(Dropout(0.2))

lstm_model.add(LSTM(units=50))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(units = 1))

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

Fig. 5. Architecture for LSTM Model in Python.

C. Inserting NAs

Once the LSTM model is built and compiled predicting a certain number of days, NA values are inserted, which will be calculated in the next stage using the LANN and LANNc algorithms. The NA values blocks were inserted uniformly in the predicted time series as shown in Fig. 6.

D. Applying LANN/LANNc

LANN is the first algorithm that is used to impute the predicted time series adapted from [4] for gap-sizes over 2 NA values.

LANNc: is an adaptation of LANN algorithm that solves the problem of bias to the left in imputation processes with gap-sizes over 1 NA value.

Fig. 7 shows the difference between the LANN and LANNc imputation.

The adapted LANN algorithm is shown in Table I.

E. Evaluation

The evaluation of the results is performed through Root Mean Squared Error (RMSE) that is calculated with the equation (4).

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n-1}(Pi-Ri)^2}{n}} \quad (4)$$

The results achieved are shown in the Results section.

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. 6. Distribution of NA Values in Predicted Time Series.

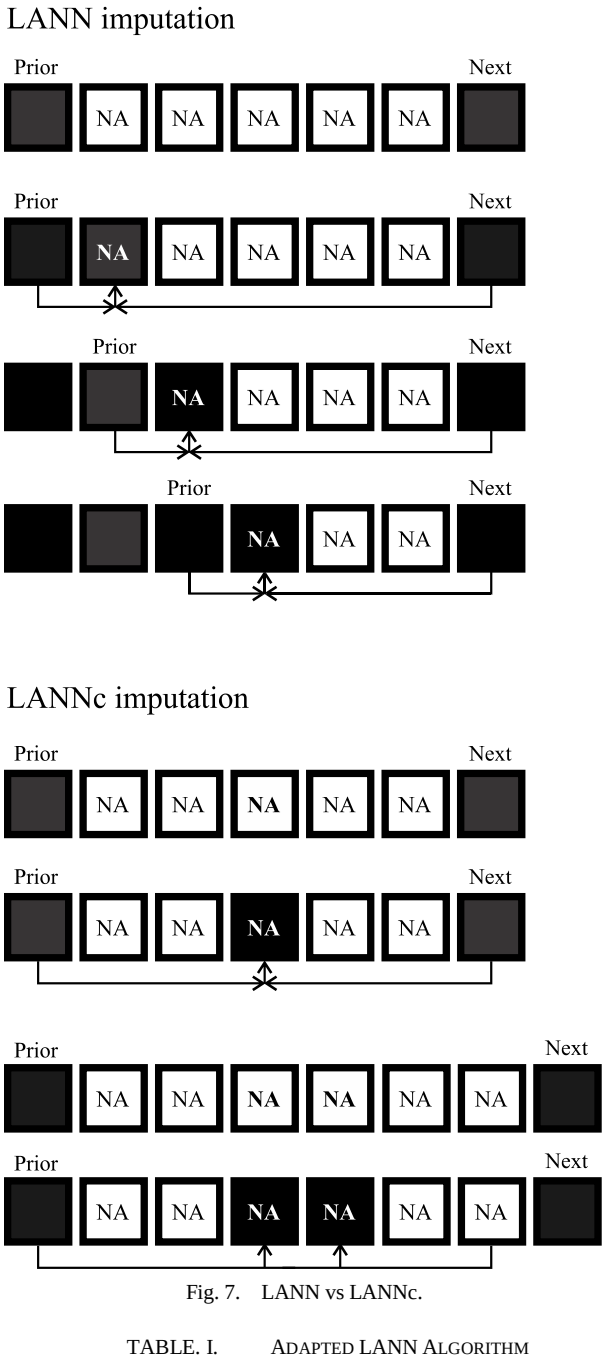

Fig. 7. LANN vs LANNc.

TABLE. I. ADAPTED LANN ALGORITHM

```
function lann(tsna,pos)
{   npos=pos.length;
    for(i=0;i<npos;i++)
    {   res=getPriorNext();
        prior=res[0];
        next=res[1];
        base=(prior+next)/2;
        tsna[pos[i]]=base.toFixed(2);
    }
    return tsna;
}
```

V. RESULTS

This section shows the results achieved after experimentation. The LANN and LANNc algorithms were implemented with different configurations of NA values between 1 and 11 as it shown in Table II and Table III with the respective RMSE values.

According to Table II and Fig. 8, it is appreciated that on average, from the eleven (11) configurations experienced for LANN, all of them improve the RMSE of LSTM. The best configuration corresponds to six consecutive NA values (RMSE 0.6577).

According to Table III and Fig. 9, it is appreciated that on average, from the eleven (11) configurations experienced for LANNc seven (7) allow to improve the RMSE of LSTM, so the best configuration corresponds to two consecutive NA values (RMSE 0.6606).

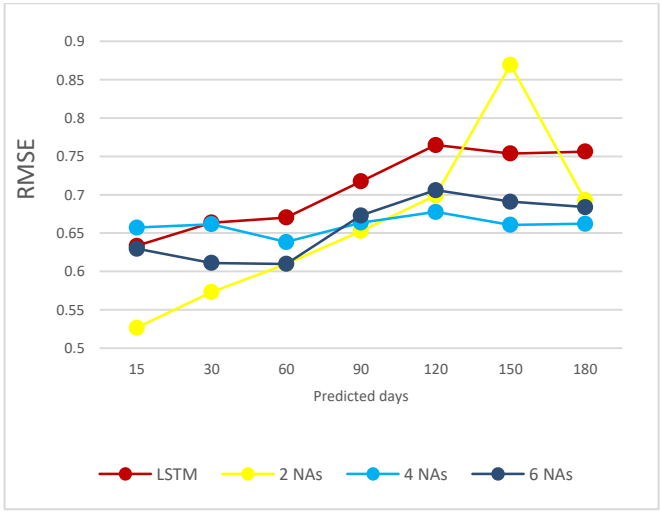

Fig. 8. LSTM vs Top 3 LANN.

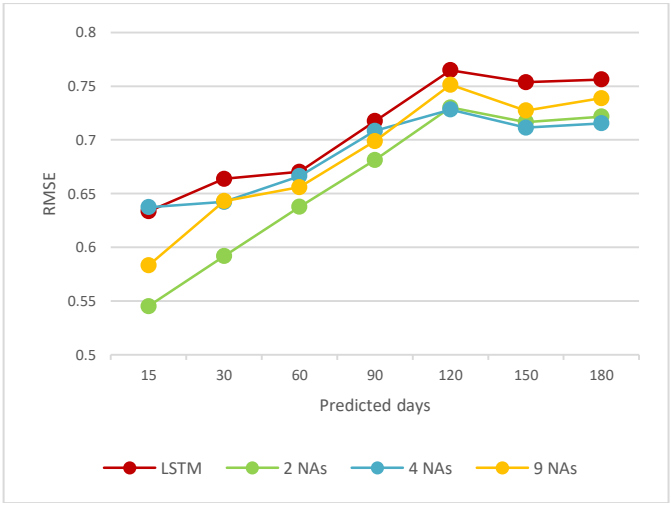

Fig. 9. LSTM vs Top 3 LANN.

TABLE. II. LSTM VS LANN

Technique		RMSE of Predicted Days							Avg
		15	30	60	90	120	150	180	
LSTM		0.6334	0.6637	0.6702	0.7175	0.7649	0.7537	0.7562	0.7085
LANN	1	0.6515	0.6523	0.6428	0.7003	0.7575	0.7421	0.7384	0.6978
	2	**0.5265**	**0.5730**	**0.6096**	**0.6527**	0.6990	0.8695	0.6929	0.6604
	3	0.6377	0.6352	0.6328	0.6686	0.6961	0.6858	0.6877	0.6634
	4	0.6572	0.6616	0.6383	0.6636	0.6776	0.**6606**	0.6622	0.6601
	5	0.6389	0.6550	0.6636	0.6691	0.**6880**	0.6697	**0.6553**	0.6628
	6	0.6296	0.6111	0.6097	0.6730	0.7059	0.6910	0.6838	**0.6577**
	7	0.6649	0.7564	0.7054	0.7045	0.7258	0.7011	0.6954	0.7076
	8	0.5896	0.6610	0.6706	0.7122	0.7059	0.6813	0.6723	0.6704
	9	0.5831	0.6612	0.7201	0.7373	0.7294	0.7046	0.6749	0.6872
	10	0.6396	0.6810	0.7189	0.7131	0.7457	0.7234	0.7149	0.7052
	11	0.6185	0.6484	0.7180	0.7050	0.6992	0.6837	0.6735	0.6780

TABLE. III. LSTM VS LANNC

Technique		RMSE of Predicted Days							Avg
		15	30	60	90	120	150	180	
LSTM		0.6334	0.6637	0.6702	0.7175	0.7649	0.7537	0.7562	0.7085
LANNc	1	0.6515	0.6523	0.6428	0.7003	0.7575	0.7421	0.7384	0.6978
	2	**0.5452**	**0.5918**	**0.6377**	0.**6813**	0.7302	0.7166	0.7216	**0.6606**
	3	0.6586	0.6360	0.6753	0.7143	0.7435	0.7294	0.7357	0.6989
	4	0.6374	0.6422	0.6663	0.7084	0.**7282**	0.**7113**	0.**7154**	0.6870
	5	0.6186	0.6430	0.7073	0.7164	0.7673	0.7475	0.7362	0.7051
	6	0.6388	0.6414	0.7101	0.7352	0.8015	0.7833	0.7681	0.7254
	7	0.6197	0.7018	0.6638	0.7424	0.7949	0.7815	0.7840	0.7268
	8	0.5876	0.6081	0.7546	0.7708	0.7935	0.7621	0.7552	0.7188
	9	0.5831	0.6430	0.6559	0.6989	0.7513	0.7273	0.7388	0.6854
	10	0.6090	0.6617	0.7010	0.7351	0.7623	0.7322	0.7226	0.7034
	11	0.5926	0.6531	0.7653	0.7616	0.8357	0.7914	0.7681	0.7382

VI. DISCUSSION

According to Table IV and Fig. 10, it can be seen that, on average, the best configuration of the proposed LANN and LANNc algorithms allowed to overcome the techniques mentioned in the Related Work section. It should be noted that GRU in forecasting of time series, in general, in most cases outperforms LSTM, it can be seen in [15] [14] [19] [20]. However, by applying any of the two imputation techniques mentioned in this work, on average it was possible to improve LSTM predictions and overcome the results achieved by GRU.

TABLE. IV. COMPARISON WITH ANOTHER TECHNIQUES

Technique	RMSE of Predicted Days							Avg
	15	30	60	90	120	150	180	
LSTM	0.6334	0.6637	0.6702	0.7175	0.7649	0.7537	0.7562	0.7085
LANN*	0.6296	0.6111	0.6097	0.6730	0.7059	0.6910	0.6838	**0.6577**
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
GRU	0.5953	0.6917	0.6678	0.6689	0.7076	0.6751	0.6727	0.6684
ARIMA	0.6134	1.2988	2.2932	2.5240	2.2320	2.2320	2.0440	2.1639

* 6 NAs ** 2 NAs

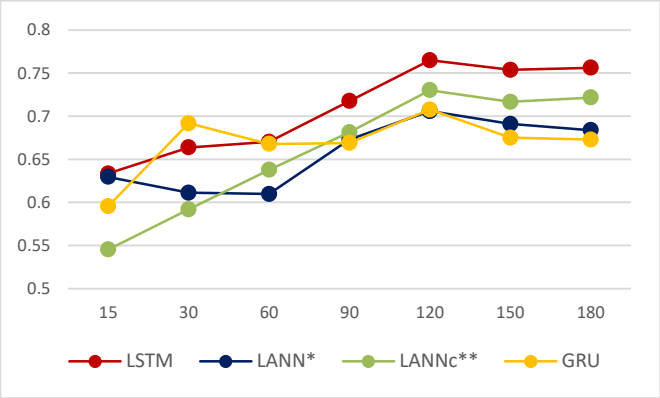

Fig. 10. Comparison of LSTM, LANN, LANNc and GRU.

VII. Conclusions

The use of Imputation techniques based on Local Average of Nearest Neighbors allowed to improve the prediction results of LSTM by exceeding on average the prediction results of GRU and other state of the art techniques.

Despite the risk of inserting bias to the left of the gap of NA values, LANN obtained a better performance than LANNc and in all NA cases it outperformed or improved LSTM predictions.

VIII. Future Work

The use of imputation techniques on the results of prediction techniques to improve their accuracy, opens a new research line that will improve current techniques. At this point, it is possible to experiment with other imputation techniques such as SMA [21], LWMA [21], EWMA [21], ARIMA-Kalman [22] [23], CBRi [6], CBRm [5], HSV [24], LANNf [24], etc.

In this work the results of LSTM were improved, it would be important to analyze how much the results of GRU or other prediction techniques can be improved through imputation techniques.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] S. Moritz, "Package ImputeTS," cran.r-project.org, 2019.

[22] P. Zarchan, H. Musoff, Fundamentals of kalman filtering, American Institute of Aeronautics and astronautics, 2000.

[23] S. Moritz, T. Bartz-Beielstein, "imputeTS: Time Series Missing Value Imputation in R," The R Journal, vol. 9, nº 1, pp. 207-2018, 2017.

[24] A. Flores, H. Tito & D. Centty, "Model for time series imputation based on average of historical vectors, fitting and smoothing," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 10, nº 10, pp. 346-352, 2019.