Local Average of Nearest Neighbors: Univariate Time Series Imputation

Anibal Flores[1], Hugo Tito[2], Carlos Silva[3]
E.P. Ingeniería de Sistemas e Informática, Universidad Nacional de Moquegua, Moquegua, Peru

***Abstract*—The imputation of time series is one of the most important tasks in the homogenization process, the quality and precision of this process will directly influence the accuracy of the time series predictions. This paper proposes two simple algorithms, but quite powerful for univariate time series imputation process, which are based on the means of the nearest neighbors for the imputation of missing data. The first of them Local Average of Neighbors Neighbors (LANN) calculates the missing value from the average of the previous neighbor and the following neighbor to the missing value. The second Local Average of Neighbors Neighbors+ (LANN+), considers a threshold parameter, which allows to differentiate the calculation of the missing values according to the difference between the neighbors: for the differences less than or equal to the threshold the missing value is calculated through of LANN and for major differences the missing value is calculated from the average of the four closest neighbors, two previous and two subsequent to the missing value. Imputation results on different time series are promising.**

Keywords—Univariate time series; imputation; LANN; LANN+

I. Introduction

Time series data are used in a large variety of real-world applications, and they often encounter the missing value problem due to data transmisión errors, machine malfunction, or human errors [1]. While imputation in general is a well-known problem and widely covered by different tools, finding algorithms or techniques able to fill missing values in univariate time series is more complicated [2]. The reason for this lies in fact is that the most imputation algorithms rely on inter-attribute correlations, while univariate time series imputation instead needs to employ time dependencies.

For univariate time series, the techniques that can be applied range from univariate algorithms, univariate time series algorithms and multivariate algorithms on lagged data [3].

In time series, can be find different gap sizes for NA values, a quick classification could be: short-gaps from 1 to 2 consecutive NAs; medium-gaps from 3 to 10 consecutive NAs; and big-gaps more than 10 consecutive NAs. In this paper we focus only on short-gaps.

In meteorological time series we find the three types of gaps mentioned above, we could even add a new category very big-gaps, since in some time series, there are gaps of NAs that range between approximately 1 and 72 months. A 72-month NA gap was found in the Punta de Coles time series between 1960/01/01 and 1965/12/31 (1978 consecutive NAs).

In this paper, we propose two algorithms for short-gaps of NAs within the univariate time series algorithms category and these are based on local averages of numerical time series. The first Local Average of Nearest Neighbors (LANN) algorithm is based on the average of the two nearest neighbors to the missing value or NA, the previous neighbor and the neighbor after the missing data. The second Local Average of Neighbors Neighbors+ algorithm (LANN+) is based on the difference (d) between the previous value and the value close to the missing value, this difference is compared with a threshold parameter that allows determining the way in which the missing value is calculated. When the differences are less than or equal to the threshold value, the missing value is calculated with the LANN algorithm and when the difference is greater than the threshold value, the missing value is calculated from the 4 neighbors closest to the NA value, the two previous and the two next to the NA value or missing value.

The paper is structured as follows: In Section II, a brief review of the state of the art regarding the proposals in this work is shown; in Section III, the fundamental theoretical bases for the better understanding of the paper content are shown; in Section IV, the proposed algorithms are described; in Section V, the results with different sizes of time series are described and discussed, likewise, they are compared with similar works; in Section VI, the conclusions reached at the end of the study are described and finally in the last Section VII, the future work is shown, which can be done to improve the proposals.

II. Related Work

A review of the state of the art of imputation works in univariate time series has been carried out and the results are shown below.

Commonly-used methods for univariate time series are relatively simple and include the arithmetic mean, interpolation, and last observation carried forward (LOCF) [4].

Last Observed Carried Forward LOCF [5] is a technique for replacing each NA with the most recent non-NA prior to it. For each individual missing value are replaced by the last observed value of that variable. In this work, zoo R package was used to implement LOCF imputation.

Hot-deck [6] imputation dates back to the days when data sets were saved on punch cards, the hot-deck referring to the “hot” staple of cards (in opposite to the “cold” deck of cards from the previous period). Most of the time, hot-deck imputation refers to sequential hot-deck imputation, meaning that the data set is sorted and missing values are imputed sequentially running through the data set line (observation) by line (observation). In this work VIM R package was used to implement hot-deck imputation.

Missing Value Imputation by Weighted Moving Average [7], the mean in this implementation taken from an equal number of observations on either side of a central value. This means for an NA value at position i of a time series, the observations *i-1,i+1* and *i+1, i+2* (assuming a window size of *k=2*) are used to calculate the mean. We have three types of algorithms in this category: Simple Moving Average (SMA), Linear Weighted Moving Average (LWMA) and Exponential Weighted Moving Average (EWMA).

Simple Moving Average (SMA) [2], all observations in the window are equally weighted for calculating the mean. For gap sizes equal to 1, and the parameter k equal to 1, SMA produces the same results as LANN in other cases results are different.

Linear Weighted Moving Average (LWMA) [2], weights decrease in arithmetical progression. The observations directly next to a central value i, have weight 1/2, the observations one further away (i-2,i+2) have weight 1/3, the next (i-3,i+3) have weight 1/4.

Exponential Weighted Moving Average (EWMA) [2] [8], is an approach that imputes the missing values by calculating the exponentially weighted moving average (EWMA). Initially, the value of the moving average window is set; the mean thereafter is calculated from equal number of observations on either side of a central missing value [8]. The observations directly next to a central value i, have weight $(1/2)^1$, the observations one further away (i-2,i+2) have weight $(1/2)^2$, the next (i-3,i+3) have weight $(1/2)^3$,.

In this work, imputeTS R package is used to implement SMA, LWMA y EWMA imputations.

Kalman Smoothing [8] on the state space representation of an autoregressive integrated moving average (ARIMA) model, is usually a good approach for imputation of highly seasonal univariate data [9]. In this work, we use imputeTS R package to implement ARIMA Kalman imputation.

Datawig[1] is a Python library that learns Machine Learning models using Deep Neural Networks to impute missing values in a dataframe. This method works very well with categorical and non-numerical features, therefore, it was not considered in the comparisons made in this work.

In order to compare the accuracy of the imputation techniques proposed with multivariable imputation techniques, two well-known multiple imputation algorithms were experimented, such as MICE [10] (Multiple Imputation by Chained Equations) and KNN [11] [12] (K-Nearest Neighbor), results can be seen in Section V.

III. Theoretical Background

A. Time Series

A time series is a set of data points indexed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Some examples of time series are daily temperatures, weekly sales, customers per day, number of monthly visits, etc.

Studying the past behavior of a series will help to identify patterns and make better forecasts. When plotted, many time series exhibit one or more of the following features:

- Trends
- Seasonal and nonseasonal cycles
- Pulses and steps
- Outliers

B. Missing Data

Depending on what causes missing data, the gaps will have a certain distribution. Understanding this distribution may be helpful in two ways [3]. First, it may be employed as background knowledge for selecting an appropriate imputation algorithm. Second, this knowledge may help to design a reasonable simulator that removes missing data from a test set; such a simulator will help to generate data where the true values are known. Hence, the quality of an imputation algorithm can be tested.

Missing data mechanisms can be divided into three categories: Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). In practice, assigning data-gaps to a category can be blurry, because the underlying mechanisms are simply unknown [3]. While MAR and MNAR diagnosis needs manual analysis of the patterns in the data and application of domain knowledge, MCAR can be tested for with t-test [3]

C. Univariate Time Series

A univariate time series is a sequence of single observations $o_1,o_2,o_3,\ldots$, on at sucessive points $t_1,t_2,t_3,\ldots\ t_n$ in time. Although a univariate time series is usually considered as one column of observations, time is in fact an implicit variable [3].

D. Univariate Imputation Methods

Techniques capable of doing imputation for univariate time series can be roughly divided into three categories [3]:

- Univariate algorithms. These algorithms work with univariate inputs, but typically do not employ the time series characteristics of the dataset. Examples are: mean, mode, median, random simple.
- Univariate time series algorithms. These algorithms are also able to work with univariate inputs, but make use of the time series characteristics. Examples of simple algorithms of this category are locf (last observation carried forward), nocb (next observation carried backward), arithmetic smoothing and

[1] W. Badr, “6 different ways to compensate for missing values in a dataset (data imputation with examples),”, Towards Data Science, [Online]. Available: https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779. [Accessed 2019/07/15]

linearinterpolation. The more advanced algorithms are based on structural time series models and can handle seasonality.

- Multivariate algorithms on lagged data. Usually, multivariate algorithms cannot be applied on univariate data. But since time is an implicit variable for time series, it is possible to add time information as covariates in order to make it possible to apply multivariate imputation algorithms. This process is all about making the time information available for multivariate algorithms. The usual way to do this is via lags and leads. Lags are variables that take the value of another variable in the previous time period, whereas leads take the value of another variable in the next time period.

IV. Proposed Algorithms

A. Local Average of Nearest Neighbors (LANN)

LANN is an imputation algorithm for a univariate time series, which is fundamentally based on the average of the two closest neighbors, this according to the analysis carried out in several meteorological time series, where, it was observed that the previous neighbor v_{i-1} and the next neighbor v_{i+1} usually have approximate values at a certain value v_i. Where in an imputation problem v_i would be the NA value or the value to be imputed.

Table I shows the difference or distance between a time series value and the other values. The time series corresponds to meteorological data of maximum daily temperatures of 15 days at a weather station in the Moquegua Region, Ilo province from 2016-01-01 to 2016-01-15.

As mentioned earlier, this algorithm provides the same results as the SMA algorithm [2] when SMA is configured with the parameter k = 1 and the sizes of the gaps are equal to 1. When the size of the gaps is greater than 1 the results are different.

Then, from Table I, calculating the average of the diagonal elements that are exactly below the main diagonal or above, we will find the average difference between an element of the series and its first neighbor. Similarly, the following diagonal will give us the average difference between an element of the series and its second neighbor and so on. Table II shows the average differences for the 15-day time series.

According to Table II for the time series analyzed, we find that the closest neighbors to some value are 1st, 3rd, 6th, 9th and 2nd.

Next, we will experiment by generating random NA values in the previous time series and calculate the NA values by applying the average of the nearest neighbors (previous and next) with LANN algorithm according equation (1).

$$NA= (v_{i-1} + v_{i+1})/2 \qquad (1)$$

Table III shows the randomly generated NAs and their respective calculation using equation (1) with a percentage of missing data of 40%, 26.67%, 13.33%. The algorithm in Table IV was used in such a way that we make sure that we do not generate missing data at the beginning and at the end of the time series, likewise, the algorithm does not insert more than two NAs as gaps.

The LANN algorithm implemented in Javascript Language can be seen in Table V.

B. Local Average of Nearest Neighbors+ (LANN+)

LANN+ is based on the LANN technique, but it conditionally considers the average of the 4 closest neighbors instead of just two as in the LANN case. This algorithm uses a threshold parameter, which the higher it is, the imputation results will be very similar to the LANN algorithm. This parameter must be set according to the nature of the time series. For a temperature time series, the most appropriate is probably 1.0, in the case of an air passenger time series, the most suitable is probably 110.

TABLE. I. MATRIX OF DIFFERENCES BETWEEN THE ELEMENTS OF A TIME SERIES

	23.4	**22.8**	**22.6**	**23.4**	**24.4**	**24**	**23.6**	**25.2**	**24.4**	**23.6**	**23.8**	**24.2**	**23.8**	**24.8**	**24.8**
23.4	**0**	0.6	0.8	0	1	0.6	0.2	1.8	1	0.2	0.4	0.8	0.4	1.4	1.4
22.8	0.6	**0**	0.2	0.6	1.6	1.2	0.8	2.4	1.6	0.8	1	1.4	1	2	2
22.6	0.8	0.2	**0**	0.8	1.8	1.4	1	2.6	1.8	1	1.2	1.6	1.2	2.2	2.2
23.4	0	0.6	0.8	**0**	1	0.6	0.2	1.8	1	0.2	0.4	0.8	0.4	1.4	1.4
24.4	1	1.6	1.8	1	**0**	0.4	0.8	0.8	0	0.8	0.6	0.2	0.6	0.4	0.4
24	0.6	1.2	1.4	0.6	0.4	**0**	0.4	1.2	0.4	0.4	0.2	0.2	0.2	0.8	0.8
23.6	0.2	0.8	1	0.2	0.8	0.4	**0**	1.6	0.8	0	0.2	0.6	0.2	1.2	1.2
25.2	1.8	2.4	2.6	1.8	0.8	1.2	1.6	**0**	0.8	1.6	1.4	1	1.4	0.4	0.4
24.4	1	1.6	1.8	1	0	0.4	0.8	0.8	**0**	0.8	0.6	0.2	0.6	0.4	0.4
23.6	0.2	0.8	1	0.2	0.8	0.4	0	1.6	0.8	**0**	0.2	0.6	0.2	1.2	1.2
23.8	0.4	1	1.2	0.4	0.6	0.2	0.2	1.4	0.6	0.2	**0**	0.4	0	1	1
24.2	0.8	1.4	1.6	0.8	0.2	0.2	0.6	1	0.2	0.6	0.4	**0**	0.4	0.6	0.6
23.8	0.4	1	1.2	0.4	0.6	0.2	0.2	1.4	0.6	0.2	0	0.4	**0**	1	1
24.8	1.4	2	2.2	1.4	0.4	0.8	1.2	0.4	0.4	1.2	1	0.6	1	**0**	**0**
24.8	1.4	2	2.2	1.4	0.4	0.8	1.2	0.4	0.4	1.2	1	0.6	1	**0**	**0**

TABLE. II. AVERAGE DIFFERENCE BETWEEN A GIVEN VALUE AND ITS NEIGHBORS

Neighbour	Average Difference	Ranking
1st	0.6143	1st
2nd	0.8462	5th
3rd	0.6500	2nd
4th	0.8545	7th
5th	0.9600	9th
6th	0.7111	3rd
7th	0.8500	6th
8th	0.9143	8th
9th	0.7333	4th
10th	0.9600	10th
11th	1.3500	11th
12th	1.5333	13th
13th	1.7000	14th
14th	1.4000	12th

TABLE. III. RMSE OF THE LANN ALGORITHM (15 DAYS)

Real	NAs (40%)	LANN	NAs (26.67%)	LANN	NAs (13.33%)	LANN
23.4	23.4		23.4		23.4	
22.8	NA	23.00	22.8		22.8	
22.6	22.6		22.6		22.6	
23.4	23.4		NA	23.50	23.4	
24.4	NA	23.50	24.4		NA	23.70
24	NA	23.55	NA	24.00	24	
23.6	23.6		23.6		23.6	
25.2	25.2		25.2		25.2	
24.4	NA	24.50	24.4		24.4	24.40
23.6	NA	24.15	NA	24.10	23.6	
23.8	23.8		23.8		23.8	
24.2	NA	23.8	24.2		24.2	
23.8	23.8		NA	24.50	23.8	
24.8	24.8		24.8		24.8	
24.8	24.8		24.8		24.8	
RMSE		0.5041	RMSE	0.4330	RMSE	0.4950

TABLE. IV. RANDOM INSERTION ALGORITHM OF MISSING VALUES

```
function insertNAs(tso,k)
{     nts=tso.length;
      nn=(Math.floor(nts/k)-1);
      inf=1;sup=k;pna=0;
      for(j=0;j<nn;j++)
      {     pna=Math.floor(Math.random() * (sup - inf + 1)) + inf;
            pos.push(pna);
            tso[pna]="NA";
            inf+=k;
            sup+=k;
      }
      return tso;
}
```

TABLE. V. LANN ALGORITHM

```
function lann(tsna)
{     npos=pos.length;
      for(i=0;i<npos;i++)
      {     if(tsna[pos[i]-1]!='NA')
                  prior=parseFloat(tsna[pos[i]-1]);
            else
                  prior=parseFloat(tsna[pos[i]-2]);
            if(tsna2[pos[i]+1]!='NA')
                  next=parseFloat(tsna[pos[i]+1]);
            else
                  next=parseFloat(tsna[pos[i]+2]);
            base=(prior+next)/2;
            tsna[pos[i]]=base.toFixed(2);
      }
      return tsna;
}
```

The consideration of having a threshold is based on the fact that missing values in time series should not be imputed with the same technique, since each missing value and its neighbors have their own characteristics so there should be a technique that suits these characteristics in such a way that the imputed value has these characteristics. In that sense, although there is no exhaustive extraction of characteristics of time series with missing data, with LANN+, an effort is made to consider at least one characteristic that becomes the difference (d) between the previous neighbor and the neighbor after the missing value or NA data.

Regarding the alternation between two neighbors for differences less than or equal to the value of the threshold, and four neighbors for differences greater than the value of the threshold, it was considered so because when analyzing different time series of temperatures it was found that for small differences the average of the two closest neighbors (v_{i-1}, v_{i+1}) in most cases produced good results, while for larger differences it was more appropriate to use the average of the four nearest neighbors (v_{i-2}, v_{i-1}, v_{i+1}, v_{i+2}), something that can be seen if we compare the RMSE of Table III with those of Table VI.

TABLE. VI. RMSE OF THE LANN+ ALGORITHM (15 DAYS)

Real	NAs (40%)	LANN+ *	NAs (26.67%)	LANN+ *	NAs (13.33%)	LANN+ *
23.4	23.4		23.4		23.4	
22.8	NA	23.00	22.8		22.8	
22.6	22.6		22.6		22.6	
23.4	23.4		NA	23.35	23.4	
24.4	NA	23.50	24.4		NA	23.70
24	NA	23.55	NA	24.00	24	
23.6	23.6		23.6		23.6	
25.2	25.2		25.2		25.2	
24.4	NA	24.10	24.4		NA	24.05
23.6	NA	23.95	NA	24.10	23.6	
23.8	23.8		23.8		23.8	
24.2	NA	23.8	24.2		24.2	
23.8	23.8		NA	24.50	23.8	
24.8	24.8		24.8		24.8	
24.8	24.8		24.8		24.8	
RMSE		0.4873	RMSE	0.4308	RMSE	0.5534

*threshold=1.0

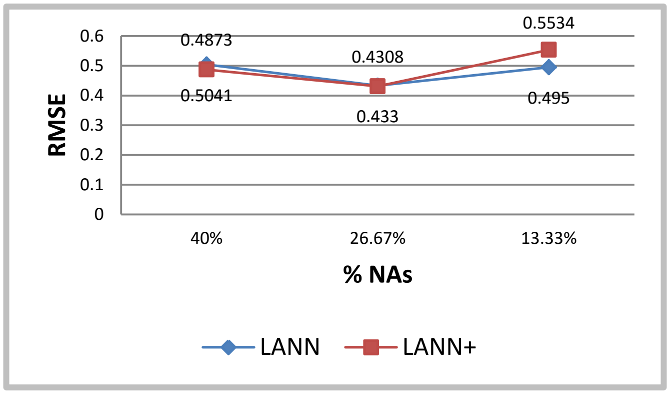

Fig. 1. RMSE Comparison between LANN and LANN+.

Fig. 1 shows a comparison between LANN and LANN+ for a time series of 15 days.

Table VII, shows the LANN+ algorithm implemented in Javascript language.

TABLE. VII. LANN+ ALGORITHM

```
function lannp(tsna)
{    npos=pos.length;
     for(i=0;i<npos;i++)
     {    if(tsna[pos[i]-1]!='NA')
               prior=parseFloat(tsna[pos[i]-1]);
          else
               prior=parseFloat(tsna[pos[i]-2]);
          if(tsna[pos[i]+1]!='NA')
               next=parseFloat(tsna[pos[i]+1]);
          else
               next=parseFloat(tsna[pos[i]+2]);
          d=Math.abs(prior-next);
          base=(prior+next)/2;
          if(d<=threshold)
               tsna[pos[i]]=base.toFixed(2);
          else
          {    mean2nn=get2nn_mean(tsna,pos[i]);
               tsna[pos[i]]=mean2nn.toFixed(2);
          }
     }
     return tsna;
}
```

V. RESULTS AND DISCUSSION

This section shows the results of comparing the proposed algorithms with other algorithms mentioned in section II. Likewise, the precision is compared with other time series with different characteristics to the temperature time series seen in Section IV.

A. Comparison with other Univariate Imputation Techniques

The LANN and LANN+ algorithms are compared with other imputation techniques in a maximum temperature time series of 15 days, Table VIII shows the results.

According to Table VIII, it is appreciated that for the percentage of NAs equal to 40%, the algorithm that obtained the best precision was the LWMA (0.4572) followed by the EWMA algorithm (0.4692) and thirdly the proposed algorithm LANN+ (0.4873). For the percentage of NAs equal to 26.67%, the algorithm with the best performance was the proposed algorithm LANN+ (0.4308) followed by LANN, SMA and ARIMA Kalman with the same RMSE (0.4330). For a percentage of NAs equal to 13.33%, in the first place, we have matched the LANN, SMA and ARIMA-Kalman algorithms with the same RMSE (0.4950).

Also, the performance of the same algorithms was evaluated with a time series with more data, in this case instead of 15 days, it is considered 90 days of maximum daily temperatures, from 2016-01-01 to 2016-03-30. Table IX shows the results.

According to Table IX, it can be seen that for a percentage of NAs of 48.89%, the algorithm with better precision is LANN (0.6059), secondly, we have the LANN+ algorithm (0.6196) and thirdly the SMA algorithm (0.6211). For a percentage of NAs of 32.22%, again the best precision was obtained by the LANN algorithm (0.5099), followed by the LANN+ algorithm (0.5296) and thirdly by the SMA algorithm (0.5451). For a percentage of NAs of 23.33%, the best algorithm was EWMA (0.4765), followed by LWMA (0.4970) and thirdly we have two, LANN and SMA with a RMSE equal to 0.5085.

The proposed algorithms were also compared with the precision of two well-known multiple imputation algorithms such as MICE and KNN and the results shown in Table X were obtained. In this case, it's used the data from the same previous data range of the nearest meteorological station to the Punta de Coles station, which is the Ilo Station. Ilo station is located in the El Algarrobal district of the province of Ilo.

TABLE. VIII. COMPARISON WITH OTHER UNIVARIATE IMPUTATION TECHNIQUES – 15 DAYS

Technique	RMSE (NAs 40%)	RMSE (NAs 26.67%)	RMSE (NAs 13.33%)
LANN	0.5041	**0.4330**	**0.4950**
LANN+ (treshold=1.0)	**0.4873**	**0.4308**	0.5534
LOCF	0.8869	0.6324	0.9055
Hotdeck	0.9201	1.0295	0.8000
SMA	0.6448	**0.4330**	**0.4950**
LWMA	**0.4572**	0.4721	0.6275
EWMA	**0.4692**	0.4613	0.6170
ARIMA Kalman	0.5482	**0.4330**	**0.4950**

TABLE. IX. COMPARISON WITH OTHER UNIVARIATE IMPUTATION TECHNIQUES – 90 DAYS

Technique	RMSE (NAs 48.89%)	RMSE (NAs 32.22%)	RMSE (NAs 23.33%)
LANN	**0.6059**	**0.5099**	**0.5085**
LANN+ (1.0)*	**0.6196**	**0.5296**	0.5210
LOCF	0.8382	0.7485	0.7461
Hotdeck	1.322	1.5349	0.5778
SMA (k=1)	**0.6211**	**0.5451**	**0.5085**
LWMA (k=4)	0.6428	0.6299	**0.4970**
EWMA (k=4)	0.6266	0.5878	**0.4765**
ARIMA Kalman	0.7447	0.5964	0.5191

* threshold=1.0

TABLE. X. COMPARING WITH MICE AND KNN

Technique	RMSE (NAs 48.89%)	RMSE (NAs 32.22%)	RMSE (NAs 23.33%)
LANN	0.6059	0.5099	0.5084
LANN+*	0.6196	0.5296	0.5210
MICE	1.4208	1.0993	0.8632
KNN	1.0842	0.9762	0.9646

*Threshold: 1.0

According to Table X, the accuracy of the proposed LANN and LANN+ algorithms greatly outperform MICE and KNN.

The proposed algorithms were also evaluated with time series with other characteristics:

- Airpass: Monthly total international airline passengers from 01/1960-12/1971 [13] Characteristics: trend, seasonality.
- Beersales: Monthly beer sales in millions of barrels, 01/1975-12/1990 [13] Characteristics: no trend, seasonality.

Table XI shows the results achieved with the Airpass time series.

Table XI shows that for time series with different characteristics than maximum temperatures, the proposed algorithms also offered good performance.

Table XII shows the results with the Beersales time series, where the LANN algorithm showed the best accuracy in the imputation process of missing data.

TABLE. XI. COMPARISON ON AIRPASS TIME SERIES

Technique	**RMSE**
LANN	**22.0368**
LANN+*	**20.9122**
LOCF	43.6041
Hotdeck	164.6075
SMA	**21.7995**
LWMA	28.9395
EWMA	24.4703
Kalman-ARIMA	**20.8952**

*Threshold: 110

TABLE. XII. COMPARISON ON BEERSALES TIME SERIES

Technique	**RMSE**
LANN	**0.8738**
LANN+ *	0.9738
LOCF	1.6869
Hotdeck	2.6295
SMA	**0.9246**
LWMA	1.1915
EWMA	1.0772
Kalman-ARIMA	**0.9283**

*Threshold: 0.02

VI. CONCLUSIONS

The proposed algorithms showed a very good performance in the imputation process of NAs short-gaps in different time series in which they were analyzed. They outperformed many well-known imputation algorithms such as ARIMA-Kalman, Hotdeck, LOCF, MICE, KNN in different percentages of missing data.

For meteorological time series such as maximum temperature series, LANN and LANN+ are highly recommended due to the good accuracy achieved.

For the time series with high trend and seasonality, the use of the LANN+ algorithm is recommended and for time series with low trend and high seasonality, the use of LANN is recommended.

VII. FUTURE WORK

The algorithms proposed in the present work have been analysed and evaluated in short-gaps of NAs, it is important in future works to configure them for larger gaps, three or more data and evaluate the corresponding accuracy.

The proposed algorithms can be improved by combining with forecast models such as Deep Learning, especially Recurrent Neural Networks [14] especially Long Short Term Memory (LSTM) or Gate Recurrent Unit (GRU) that allow improving the accuracy of the estimates reached.

REFERENCES

[1] Chang, C. Wang, S. Lee, "Novel Imputation for Time Series Data," in International Conference on Machine Learning and Cybernetics, Guangzhou, 2015.

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

[3] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer, J. Stork, "Comparison of different Methods for Univariate Time Series Imputation in R," arxiv.org, 2015.

[4] N. Bokde, M. Beck, F. Martinez, K. Kulat, "A novel imputation methodology for time series based on pattern sequence forecasting," Pattern Recognition Letters, 2018.

[5] A. Zeileis, G. Grothendieck, "zoo: S3 infrastructure for regular and irregular time series,"Journal of Statistical Software, vol.14, no. 6, 2005.

[6] A. Kowarick, M. Templ, "Imputation with the R package VIM," Journa of Statistical Software, vol. 74, no. 7, 2016.

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

[8] E. Rantou, "Missing Data in Time Series and Imputation Methods," University of the Aegean, Samos, 2017.

[9] A. Chaudhry, W. Li, A. Basri, F. Patenaude, "On improving imputation accuracy of LTE spectrum measurements data," in Wireless Telecommunications Symposium, Phoenix, AZ, USA, 2018.

[10] S. Van Buuren, K. Groothuis-Oudshoorn, "mice: multivariate imputation by chained equations in R," Journal of Statistical Software, vol. 45, no. 3, 2011.

[11] G. Chang, T. Ge, "Comparison of missing data imputation methods for traffic flow," in International Conference of Transportation, Mechanical, and Electrical Engineering (TMEE), Chanchung, China, 2011.

[12] B. Sun, L. Ma, W. Cheng, "An improved k-nearest neighbours method for traffic time series imputation," in Chinese Automation Congress (CAC), 2017.

[13] K. Chan, B. Ripley, "TSA: Time series analysis," CRAN. R-project.org, 2012.

[14] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," Scientific Reports, 2018.