CBRm: Case based Reasoning Approach for Imputation of Medium Gaps

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

***Abstract*—This paper presents a new algorithm called CBRm for univariate time series imputation of medium-gaps inspired by the algorithm called Case Based Reasoning Imputation (CBRi) for short-gaps. The performance of the proposed algorithm is analyzed in meteorological time series corresponding to maximum temperatures; also it was compared with several similar techniques. Although the algorithm failed to overcome in some cases to other proposals regarding precision, the results achieved are encouraging considering that some weaknesses of other proposals with which it was compared were outperformed.**

Keywords—Case Based Reasoning; CBR; CBRm; univariate time series imputation; medium-gaps

I. Introduction

Time series data exist in nearly every scientific field, where data are measured, recorded and monitored, so it is understandable that missing values may occur [1]. The imputation or completeness of missing values in time series is a very important task, since if it is not performed it is very complicated or impossible to be able to successfully carry out a prediction or forecasting process.

In the research field of imputation, univariate time series are a special challenge, most of the standard algorithms rely on inter-attribute correlations to estimate values for the missing data [2]. In the univariate case no additional attributes can be employed directly, so effective univariate algorithms instead need to make use of the time series characteristics.

In time series, different gaps sizes of NA values can be found: 1 or 2 consecutive NAs (short-gaps), from 3 to 10 consecutive NAs (medium-gaps) and more than 10 consecutive NAs (big-gaps) [3]. In this paper, a new algorithm for univariate time series imputation of medium-gaps is proposed, which is based on Case Based Reasoning (CBR) in such a way that the historical data of the time series can be used to improve the estimation of NA values. This algorithm is called CBRm and is implemented very similarly to CBRi "unpublished" [4] algorithm.

CBRm uses the same case base that was implemented for CBRi "unpublished" [4], this case base was built from maximum daily temperatures of 9 years (2007-01-01 to 2015-12-31) recorded at the Punta de Coles weather station located in the Moquegua region - Peru. The fundamental difference respect to CBRi lies in the operation of both techniques. Fig. 1 shows in summary the CBRi imputation process. As it's appreciated, this operation for medium-gaps can introduce bias to the left of the gap, this because CBRi was designed to impute time series for short-gaps, between 1 and 2 consecutive NAs. Something similar happens with the LANN and LANN+ algorithms that were also designed for short-gaps.

The CBRm imputation process is shown in Fig. 2. As can be seen when a value between prior and next is calculated, it is not assigned immediately after prior, but is assigned to the center of the NA series by doubling in the case that the total of NAs is an even number.

Additionally, this work also presents the results achieved by the algorithms called Local Average Nearest Neighbors LANN [3] and LANN+ [3] in medium-gaps imputation processes. So, a small adaptation for these algorithms was done, specifically in the part corresponding to the determination of the prior and next values.

The present work has been organized as follows: in the second section, a brief description of the work related to univariate time series imputation is shown. The third section shows the theoretical bases necessary for a better understanding of the content of the work. The fourth section describes the proposed algorithm and its implementation. The fifth section describes the results achieved, which are compared with different univariate time series imputation techniques. The sixth section shows the conclusions reached in the present work and finally in the seventh section, it is indicated, the works that can be carried out based on the results of the work presented.

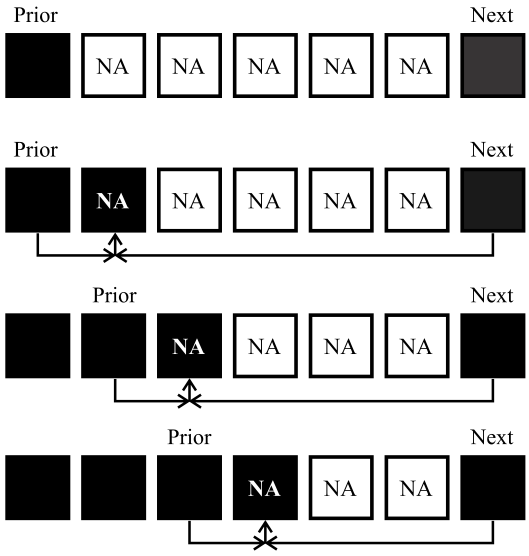

Fig. 1. CBRi Imputation Process.

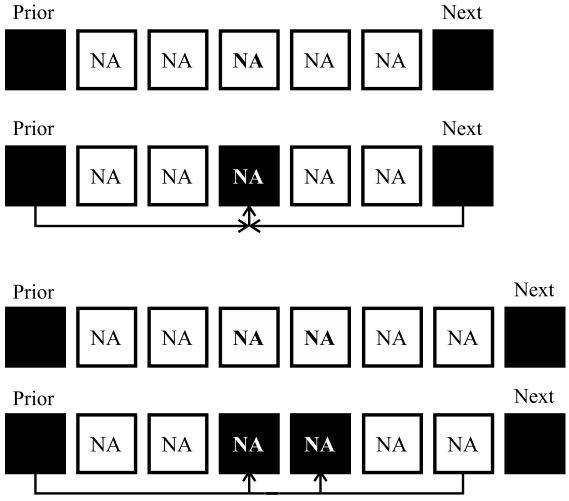

Fig. 2. CBRm Imputation Process.

II. Related Work

This section shows the results of the review of different techniques or algorithms for univariate time series imputation, from the oldest to the newest.

The first techniques for univariate time series imputation were quite simple and consisted of using arithmetic mean, median, mode, interpolation and Last Observation Carried Forward (LOCF) [5].

Last Observed Carried Forward (LOCF) is a technique for filling a NA value with the closest non-NA value prior to it [6]. Each individual NA value is replaced by the last observed value of that variable.

Baseline Observation Carried Forward (BOCF) [7] is similar to the LOCF; it replaces NA values with the non-missing baseline observation of the time series.

Hot-deck [8] [9], an NA value is replaced with an observed value that is closer in terms of distance. The Hot-deck algorithm randomly selects a value from a set of non-NA values and replaces the NA value. For comparative analysis, in this work, the hot-deck algorithm implemented in VIM R package is used.

Missing Value Imputation by Weighted Moving Average [10], is a set of algorithms that use the average or mean of the non-NA elements around an NA value. For an NA value at position *i* of a time series and assuming a window size of k=2, the observations *i-1*, *i+1* and *i+1*, *i+2* are used to calculate the mean.

There are three algorithms for univariate time series imputation in this category such as: Simple Moving Average (SMA) [10], Linear Weighted Moving Average (LWMA) [10] and Exponential Weighted Moving Average (EWMA) [10].

Simple Moving Average (SMA) [10] [11]: This algorithm for calculating the mean use all observations in the window which are equally weighted.

Linear Weighted Moving Average (LWMA) [10] [11]: In this algorithm weights decrease in arithmetical progression. The observations directly next to an NA value in position *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*, and so on.

Exponential Weighted Moving Average (EWMA) [1] [10] [11]: it is an approach that allows imputing NA values by calculating the exponentially weighted moving average. Initially, the value of the window for the moving average is established, and then the average is calculated from the same number of observations on each side of the central missing value or NA value. 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 $(1/2)^3$, and so on. In this work, the algorithms SMA, LWMA and EWMA are implemented for comparative analysis using the imputeTS R package.

The Kalman filter [12], also known as LQE (linear quadratic estimation), is an algorithm that uses a series of measurements observed over time, which contains statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement. Kalman filter integrated with ARIMA produces very good results in regression processes. In this work imputeTS package in R is used for implementing Kalman ARIMA imputation, imputeTS implements auto.arima [11] for better results.

LANN and LANN+ [3] are two fairly simple algorithms based on moving averages that show very good results in the short-gaps imputation process. As mentioned earlier, these techniques were adapted for the respective evaluation in medium-gaps. This adaptation only consisted of modifying the way in which these algorithms obtained the prior and next values.

For a comparative analysis of the results achieved by the imputation algorithm (CBRm) proposed in the present work, two well-known multivariate imputation algorithms were also implemented, such as KNN (K-Nearest Neighbor) [13] and MICE (Multiple Imputation by Chained Equations) [14] [15], these algorithms were implemented using the R VIM package for KNN and the mice package for MICE. In section V of this work, the achieved results can be seen.

III. Theoretical Background

A. Time Series

A time series is a sequence of data, observations or values, measured at certain time periods and sorted chronologically. The data can be spaced at equal intervals or uneven. For the analysis of the time series, different methods are used that help to interpret them and that allow extracting representative information about the underlying relationships between the data of the series.

One of the most common uses of time series is its analysis for prediction and forecasting. Time series are studied in different areas such as statistics, signal processing, econometrics, etc. Some features or characteristics of time series are: trends, cycles of seasonality and non-seasonality, pulses and steps, and 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 [16]. First, this knowledge can be used to select the most appropriate imputation algorithm to complete the NA values. Secondly, this knowledge can help design an imputation model, which allows the elimination of the NA values from a set of test data. This model will help generate the NA values where the true values are known. Therefore, the quality of the model can be tested through different regression metrics such as RMSE, MAPE, etc.

Mechanisms of missing data can be classified into three categories: Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). The process of completing NA-gaps in time series is sometimes complicated, since the underlying mechanisms are unknown [16]. The diagnosis of MAR and NMAR requires a manual analysis of data patterns and the application of domain knowledge, while MCAR can be tested with the t-test or Little's test [17].

C. Univariate Time Series

This term refers to a time series that consists of single observations recorded sequentially over successive time periods. Although a univariate time series is usually considered as one column of observations, time is in fact an implicit variable [16].

D. Univariate Imputation Methods

Techniques capable of doing imputation for univariate time series can be roughly divided into three categories [16]:

- Univariate algorithms. These algorithms work with univariate inputs and commonly do not employ the time series features. Some of them are: mean, mode, median, random simple, last observed carried forward, etc.
- Univariate time series algorithms. Most of these algorithms are developed in section II, and some of them are: Missing Value Imputation by Weighted Moving Average [3] (SMA, LWMA and EWMA), Kalman, ARIMA, ARIMA-Kalman, Local Average of Nearest Neighbors [3] (LANN y LANN+), and Case Based Reasoning Imputation (CBRi) among others no cited in this work.
- Multivariate algorithms on lagged data. Commonly, multivariate algorithms cannot be used for univariate time series. However, using lags and leads it is possible to apply multivariate time series algorithms to a univariate time series and thus take advantage of features offered by multivariate algorithms.

E. Case Based Reasoning (CBR)

CBR is a nature inspired problem solving methodology [18]. It uses a solution that worked for a problem to solve a similar new problem, it's called reasoning by remembering.

The first principle of the CBR approach is: similar problems have alike solutions i.e. to solve a new problem [18], the existing problems and their solutions from the case base are retrieved and re-used.

The second principle is that the type of problems which an agent faces tends to repeat [18]. Thus, there is similarity between past and current problems or current and future problems. Therefore, it is worth to remember and reuse. This leads to construction of the case base which contains completely resolved problems and their respective solutions.

The complete Case Based Reasoning process is shown in Fig. 3.

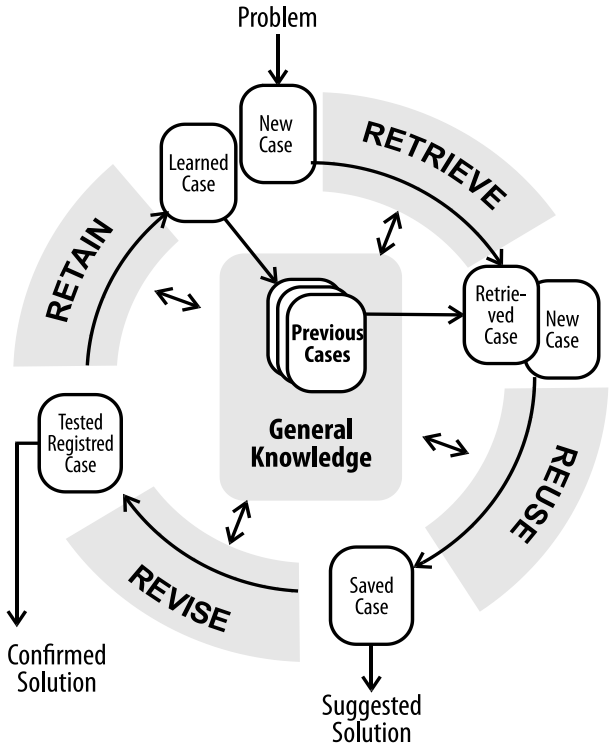

Fig. 3. CBR Process.

IV. CBRm

CBRm is inspired by the CBRi "unpublished" [4] algorithm that was designed for short-gaps imputation processes and that when applied to medium-gaps imputation processes can present problems of bias towards the prior value. Taking this assessment into consideration, CBRm begins the imputation from the middle of the series of consecutive NAs as shown in Fig. 2.

Fig. 4 shows the proposed CBR system within which implements the CBRm algorithm. Implementation process for CBRm is quite similar to CBRi "unpublished" [4], below are the required steps to implement it.

A. Time Series Selection

A time series of maximum daily temperatures corresponding to 9 years was chosen, from 2017-01-01 to 2015-12-31. These data correspond to the Punta de Coles weather station in Moquegua region (Peru) and were retrieved from the SENAMHI institutional repository at the following web link: https://www.senamhi.gob.pe/?&p = download-hydrometeorological-data.

B. Case Base Implementation

An algorithm was implemented to build the case base. The case base matrix consists of something similar to what is shown in Table I.

The algorithm in Javascript language to build the case base is shown in Table II "unpublished" [4]. This algorithm aims to create the matrix or case base (Q). It receives as arguments the empty Q matrix and a temperature vector, and returns as a result the matrix of cases Q.

TABLE. I. CASE BASE FROM 9-YEAR TIME SERIES

	..	17.2	17.3	17.4	17.5	17.6	17.7	17.8	17.9	18.0	18.1	18.2	18.3	18.4	…
…		16.8*16.4*17.2 *17.4*17.5*17		16.6*17*16.6 *17.2		17.4*17.4		17.4*17.6*19.4		16.6*18.6		18*18.2*17.6		17*18.8	
17.2															
17.3		15.8*17*17.2* 18*17.6*17.1		17.6*17.1		17.2	19.2	17.6*19.4		18.2*17.6		17.8*17.2*17 .6		18	
17.4		18.8						17.6							
17.5				18.2*17.4*17 .8*17.8*17.8		18*17.8*1 7.2		16.8*19*19*17. 8*19.1		17.8*17*17.8*1 8.2		18*18.8*18.4		16.6*18	
17.6												18			
17.7		15*17.4*17.8		18*17.9		17.4*18.8* 17.4*17.8* 17.5		18.6*17.4*17.6 *17.6*17.6*17. 8*17.2	17.8	17.4*17.6		18*18.2*18.4 *18*18.8*18		17.4*18.8	
17.8					18.6										
17.9		16.8*18.2*17.2		17.2*18.2*17 .8	17.8	17.8		17.6*18.2*17.8		18.2*18.2		17.8*19.2*18 .4*18.4*18.4 *18.2*18.4*1 8.1		18.2*18.2*1 8.8*18.7	
18.0														18.2	
18.1		18*17.4		17*17.8		17.4*18.2* 18.4*17.2		16.4*18*18.6*1 8*18*18.2*18.4		17.8*17.6*18.8 *17.8*17.8*18. 2*18.2		18*18		17.8*18.6*1 9.4*19.4*18. 8*18*	
18.2															
18.3		18.8		18.8*18.6		17.4		18*17.8		18.6*19.2*18.2 *18.2		17.6*18*19.8 *18.6*18.2		17.6*18.2	
…															

TABLE. II. ALGORITHM TO BUILD THE CASE BASE (Q)

```
function fillMatrix(Q,temv)
{    nQ=Q.length;
     for(i=0; i<nQ; i++)
     {    prior=temv[i];
          for(j=0; j<nQ; j++)
          {     next=temv[j];
                res=look4cases(prior,next);
                if(res!="")
                     Q[i][j]=res;
          }
     }
     return Q;
}
```

C. CBRm Implementation

According to Fig. 4, four blocks of code can be seen in the CBRm algorithm, and their detail can be seen in the code shown in Table III. The CBRm algorithm receives as inputs the time series with NA values and an array with the positions of each NA value.

As it shows in Fig. 4, for the first block of code that corresponds to the determination of the prior and next values that are required by the getMoreSimilar() function to extract the most similar case from the case base; these values are determined through the code between line 4 and line 18 using for this task the array of positions of the NA values.

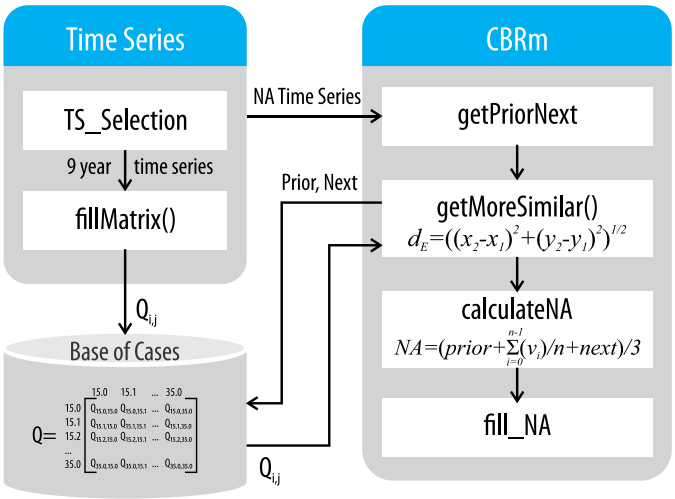

Fig. 4. CBR System.

TABLE. III. CBRM ALGORITHM

```
1.  function CBRm(tsna,pos)
2.  {   npos=pos.length;
3.      while(npos>0)
4.      {   nna=0;
5.          ini1=pos[0];
6.          fin1=pos[0];
7.          pini=0;
8.          pfin=pini;
9.          prior=parseFloat(tsna[pos[0]-1]);
10.         nav=tsna[ini1];
11.         while(nav=="NA")
12.         {   nna++;
13.             fin1++;
14.             pfin++;
15.             nav=tsna[fin1];
16.         }
17.         next=parseFloat(nav);
18.         fin1--;
19.         data=getMoreSimilar(prior,next);
20.         dat=data.split("*");
21.         ndat=dat.length;
22.         s=0;
23.         for(k=0;k<ndat;k++)
24.             s+=parseFloat(dat[k]);
25.         NA=(prior+(s/ndat)+next)/3;
26.         sNA=NA.toFixed(1);
27.         rna=nna%2;
28.         pna=Math.floor((ini1+fin1)/2);
29.         del=Math.floor((pini+pfin)/2);
30.         if(rna==0)
31.         {   m1=pna;
32.             m2=pna+1;
33.             tsna[m1]=smed;
34.             tsna[m2]=smed;
35.             pos.splice(del-1,2);
36.         }
37.         else
38.         {   tsna[pna]=sNA;
39.             pos.splice(del,1);
40.         }
41.         npos=pos.length;
42.     }
43.     return tsna;
44. }
```

In the second block of code (line 19) the getMoreSimilar () function is called, this function implements a similarity search in the base of cases (Q) using the prior and next values determined in the previous code block, returning a string containing the values that will be used in the next code block. The getMoreSimilar() function implements Euclidean Distance according to equation (1) to determine the similarity between two points.

$$d_E = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

In the third block of code between lines 20 and 26, the string returned by the getMoreSimilar () function is used and the NA value is calculated according to equation (2).

$$NA = \frac{(prior + \frac{\sum_{i=0}^{n-1}(Vi)}{n} + next)}{3} \quad (2)$$

In the last block of code from line 27 to line 40, the NA value is filled with the value calculated according to the imputation process shown in Fig. 2. The process is repeated until the time series does not contain NAs values.

V. RESULTS AND DISCUSSION

In this section, the performance of the proposed algorithm CBRm is compared with different techniques described in Section II, the comparative results are shown below.

According to Table IV, for a 15-days maximum temperature time series with 73.33% of NA values, the best techniques were LWMA and EWMA in the first place (RMSE 0.6941); second is LANN+ (RMSE 0.7077); and thirdly very close to the previous one is CBRm (RMSE 0.7083). For a percentage of 60% of NAs, the best technique was LANN+ (RMSE 0.6616); secondly LANN (RMSE 0.7187); and thirdly CBRi (RMSE 0.7461). For a percentage of 46.67% of NAs, the best technique was CBRi (RMSE 0.4140); second is LANN (RMSE 0.4423); and finally, thirdly EWMA (RMSE 0.4780).

It is important to highlight that for the first two cases, ARIMA Kalman produced no results.

A graphical comparison of CBRm RMSE with other techniques can be seen in Fig. 5.

TABLE. IV. COMPARISON WITH OTHER UNIVARIATE IMPUTATION TECHNIQUES (15 DAYS)

Technique	RMSE (NAs 73.33%)	RMSE (NAs 60%)	RMSE (NAs 46.67%)
CBRm	0.7083	0.8164	0.6152
CBRi	0.8575	0.7461	0.4140
LANN	0.8050	0.7187	0.4423
LANN+	0.7077	0.6616	0.6175
Hotdeck	0.9534	0.9189	1.0823
SMA	0.7323	0.8432	0.4928
LWMA	0.6941	0.8096	0.5209
EWMA	0.6941	0.7958	0.4780
ARIMA Kalman	NA	NA	0.5976

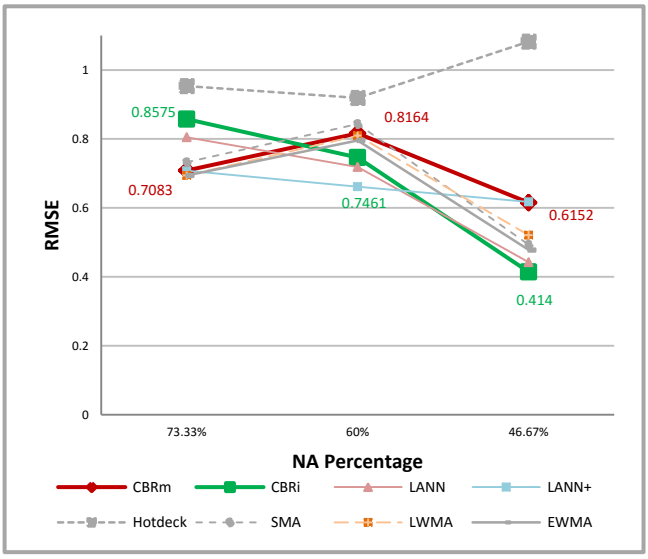

Fig. 5. Comparison with other Techniques (15 Days).

According to Table V, for 90-day time series with a percentage of 80% of NA values, the best technique was CBRm (RMSE 0.6844); second is LWMA (RMSE 0.7673); and thirdly EWMA (RMSE 0.7682). For a percentage of 65.55% of NAs, the best technique was SMA (RMSE 0.7035), followed by LWMA (RMSE 0.7083), and thirdly LANN+ (RMSE 0.7339). For a percentage of NAs of 54.44%, the best technique was LWMA (RMSE 0.8106), followed by SMA (RMSE 0.8403), and thirdly EWMA (RMSE 0.8535)

A graphical comparison of CBRm RMSE with other techniques can be seen in Fig. 6.

Also, CBRm was compared with two multivariate imputation techniques such as MICE and KNN. For this task, the data from the closest weather station to Punta de Coles, the Ilo station was used. In Table VI, the results are shown.

TABLE. V. COMPARISON WITH OTHER UNIVARIATE IMPUTATION TECHNIQUES (90 DAYS)

Technique	RMSE (NAs 80%)	RMSE (NAs 65.55%)	RMSE (NAs 54.44%)
CBRm	**0.6844**	0.8050	0.8968
CBRi	0.8086	0.8112	0.8905
LANN	0.8422	0.8198	0.9053
LANN+	0.8276	**0.7339**	0.8608
Hotdeck	1.4337	1.6323	1.4996
SMA (k=1)	0.8324	**0.7035**	**0.8403**
LWMA (k=4)	0.7673	0.7083	**0.8106**
EWMA (k=4)	**0.7682**	0.7456	**0.8535**
ARIMA Kalman	5.4275	6.7383	2.6836

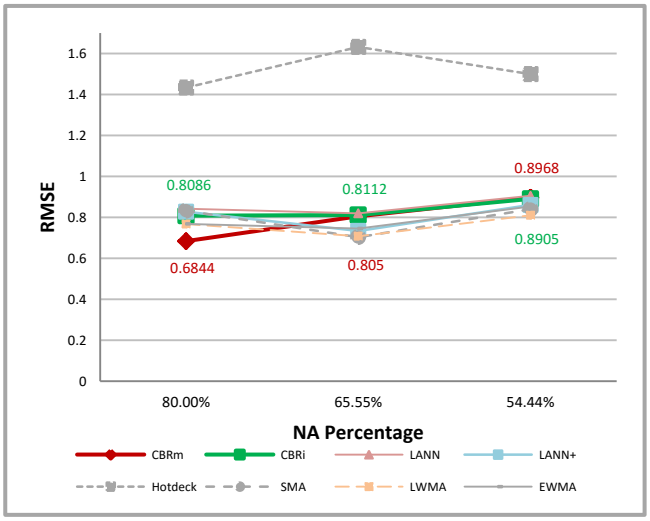

Fig. 6. Comparison with other Techniques (90 Days).

TABLE. VI. COMPARING WITH MICE AND KNN (90 DAYS)

Technique	RMSE (NAs 80%)	RMSE (NAs 65.55%)	RMSE (NAs 54.44%)
CBRm	0.6844	0.8050	0.8968
MICE	1.4063	1.3900	1.4714
KNN	1.0807	1.0751	1.2388

According to Table VI, the proposed CBRm outperformed the multivariate imputation algorithms KNN and MICE.

CBRi "unpublished" [4] despite the weaknesses mentioned, of the six problems proposed, as CBRm in two of them was among the best: in one of them it reached first place and in another it achieved third place.

LANN [3] for medium-gaps from 6 problems, in 2 of them he was among the best, it got second place twice. LANN+ [3] was a little better tan LANN, in 3 cases it was among the best getting the first, second and third place.

VI. CONCLUSION

In imputation processes of meteorological time series with medium-gaps (from 3 to 10 consecutive NAs), corresponding to time series of maximum temperatures, of the six proposed problems, in two of them CBRm was among the best: in one of them it reached the best performance and in another it achieved the third place.

Comparing CBRm with CBRi "unpublished" [4], of the 6 problems proposed in 3 cases CBRm outperformed CBRi "unpublished" [4] and in three other cases CBRi outperformed CBRm, so it is concluded that both techniques are good alternatives for the medium-gaps imputation process.

VII. FUTURE WORK

In this section, it is important to highlight the main limitation of CBRm, since it is based on historical data from the time series; it requires large amounts of historical data, something not always present in the known time series. A solution to this problem could be the complementation of the technique with other techniques until the case base has enough cases.

In the present study a medium-gaps imputation algorithm was proposed and analyzed, it would be interesting and important for future work to use gaps of 11 or more NA values (big-gaps). Also, it would be important to analyze the CBRm performance in other time series, for example in time series with no trend and no seasonality.

REFERENCES

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

[2] S. Moritz, A.Sardá, T. Bartz-Beielstein, M. Zaeffer, J, Stork, "Comparison of different methods for univariate time series imputation in R," arxiv.org, 2015.

[3] A. Flores, H. Tito, C. Silva, "Local average of nearest neighbors: Univariate time series imputation," International Journal of Advanced Computer Science and Applications, vol. 10, nº 8, 2019.

[4] A. Flores, H. Tito, C. Silva, "CBRi: A Case Based Reasoning-Inspired Approach for Univariate Time Series Imputation. Unpublished," de IEEE Latin American Conference on Computational Intelligence, Guayaquil, Ecuador, 2019.

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

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

[7] K. Kaiser, O. Affuso, T, Beasley, D. Allison, "Getting carried away: A note showing baseline observation carried forward (BOCF) results can be calculated from published complete-cases results," PMC US National Library of Medicine, 2012.

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

[9] T. Aljuaid, S. Sasi, “Proper imputation techniques for missing values in data sets,” de International Conference on Data Science and Engineering (ICDSE), Cochin, India, 2016.

[10] S. Moritz, “Package ImputeTS,” cran.r-project.org, 2019.

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

[12] A. Chaudhry, W. Li, A. Basri, F. Patenaude, “On improving imputation accuracy of LTE spectrum measurements data,” de Wireless Telecommunications Symposium, Phoenix, AZ, USA, 2018.

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

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

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

[16] 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.

[17] R. Little, “A test of missing completely at random for multivariate data with missing values,” Journal of the American Statistical Association, vol. 83, nº 404, pp. 1198-1202, 1988.

[18] M. Khan, H. Hayat, I, Awan, “Hybrid case-base maintenance approach for modeling large scale case-based reasoning systems,” Human-centric Computing and Information Sciences, vol. 9, nº 9, 2019.