Characteristics of Machine Learning-based Univariate Time Series Imputation Method

Dini Ramadhani[1*], Agus Mohamad Soleh[2], Erfiani[3]

[1,2,3,] *Department of Statistic, Faculty of Mathematics and Natural Science, IPB University, Bogor, Indonesia*

*corr_author: diniramadhani@apps.ipb.ac.id

Abstract - Handling missing values in univariate time series analysis poses a challenge, potentially leading to inaccurate conclusions, especially with frequently occurring consecutive missing values. Machine Learning-based Univariate Time Series Imputation (MLBUI) methods, utilizing Random Forest Regression (RFR) and Support Vector Regression (SVR), aim to address this challenge. Considering factors such as time series patterns, missing data patterns, and volume, this study explores the performance of MLBUI in simulated Autoregressive Integrated Moving Average (ARIMA) datasets. Various missing data scenarios (6%, 10%, and 14%) and model scenarios (Autoregressive (AR) models: AR(1) and AR(2); Moving Average (MA) models: MA(1) and MA(2); Autoregressive Moving Average (ARMA) models: ARMA(1,1) and ARMA(2,2); and Autoregressive Integrated Moving Average (ARIMA) models: ARIMA(1,1,1) and ARIMA(1,2,1)) with different standard deviations (0.5, 1, and 2) were examined. Five comparative methods were also used in this research, including Kalman StructTS, Kalman Auto-ARIMA, Spline Interpolation, Stine Interpolation, and Moving Average. The research findings indicate that MLBUI performs exceptionally well in imputing successive missing values. The results of this study indicate that the performance of MLBUI in imputing consecutive missing values, based on MAPE, yielded values of less than 10% across all scenarios used.

Keywords: characteristics, imputation, machine learning, missing data, univariate time series

I. INTRODUCTION

Univariate time series data refers to a collection of data consisting of single measurements or observations of a variable or phenomenon at specific time intervals. This data is typically arranged in chronological order, allowing for the analysis of the development or patterns of changes in the variable over time. Examples of univariate time series data include daily stock prices [1], temperature [2], or retail store sales [3]. Time series data becomes problematic when there are missing values. The issue of missing values in time series data is a common challenge in time series analysis, as it can lead to uncertainty in understanding patterns and trends in the data over time. Missing values can occur for various reasons, such as measurement errors, unavailability of data for certain time periods, or technical issues [4]. Improper handling of missing values can result in incorrect conclusions or inaccurate models, especially when large and consecutive missing value patterns lead to efficiency loss and invalid results. This underscores the importance of selecting appropriate methods for filling in missing values. Handling consecutively missing data in univariate time series data is a significant challenge in time series analysis. One reason for the complexity of this problem is the temporal nature of time series data, where each observation depends on previous observations. Therefore, it is important to choose appropriate methods for filling in missing values.

Several conventional methods such as the Kalman method, interpolation, and moving averages have become common choices in handling missing values in time series data [5], [6]. These methods have been widely used due to their ability to estimate or fill in missing values in a reliable manner, allowing for more comprehensive and accurate analysis of time series data. However, sometimes these methods also have certain weaknesses or limitations. For example, the Kalman method may be less effective when facing high fluctuations or when the data has complex patterns. Interpolation, although it can provide smooth estimates, may introduce distortions to the original trend or pattern of the data, especially if the data has significant fluctuations. Meanwhile, moving averages can dampen extreme fluctuations but may not be sensitive to changes in trends or data patterns. However, in recent decades, rapid developments in the field of Machine Learning (ML) have transformed many domains by significantly contributing to technological advancements and scientific research [4], [7]-[9]. Machine learning methods enable deeper analysis of existing data and address complex challenges, including handling missing values. However, research focusing on the use of machine learning to handle missing values in univariate time series data remains limited. One study that attempted to use machine learning methods for this

purpose is the study introduced by Phan. Phan [10] proposed a new approach to fill missing values in univariate time series data using Machine Learning-based Univariate Time Series Imputation (MLBUI) methods. This approach transforms univariate time series data into multivariate data. The methods applied in MLBUI are Random Forest Regression (RFR) [9], [11] and Support Vector Regression (SVR) [12]. In the study, the MLBUI method was applied to data with missing values occurring at $3 \times T$, $N - 3 \times T$, and in the middle of the data. The data used in the study consisted of four time series selected from different domains with different sampling frequencies and measurement durations (short or long periods). The missing data scenarios for datasets with monthly sampling frequencies involved creating gaps of 6, 9, 12, 15, and 18 missing points, while for datasets with daily sampling frequencies, the imputation sizes ranged from 0.6% to 1.5% of the dataset size. For each level of missing data in the dataset, Phan randomly selected 10 positions to generate simulated gaps. Thus, all algorithms were performed 50 times for each dataset. The study also compared the performance of MLBUI with several other methods, such as Enhanced Dynamic Time Warping with Boundary Interpolation (eDTWBI), Kalman, interpolation, and Last Observation Carried Forward (LOCF). The results showed that the MLBUI RFR method was able to handle missing data better than the other tested methods.

Despite the use of the Machine Learning-based Univariate Time Series Imputation (MLBUI) method, a comprehensive understanding of its characteristics has not yet been fully achieved. Users of this method may encounter challenges in applying it effectively. In the context of selecting imputation methods for univariate time series, several critical factors must be carefully considered. Key considerations include the underlying time series patterns, missing data patterns, and whether the time series is stationary or non-stationary. Additionally, the presence of Autoregressive (AR), Moving Average (MA), or a combination of both in Autoregressive Moving Average (ARMA) or Autoregressive Integrated Moving Average (ARIMA) models must also be evaluated carefully. All of these factors influence the choice of the appropriate imputation method for a given situation. Furthermore, it is crucial to account for the volume of missing data. Considering all these factors holistically allows users to develop a deeper understanding of various imputation methods and apply them effectively in the appropriate context. This enables users to make more informed decisions and obtain more reliable results in their data analysis and processing. This study aims to investigate the attributes of the MLBUI technique in handling missing data, particularly in relation to simulated ARIMA datasets. It is anticipated that this research will offer valuable insights into the use of appropriate imputation methodologies for analyzing univariate time series, thereby enhancing the understanding of efficient approaches to managing data incompleteness.

II. METHOD

The research procedure consists of five stages, commencing with ARIMA simulation data generation, followed by the establishment of missing data scenarios. The subsequent stage involves the development of the MLBUI program, which is then utilized for imputation on the ARIMA simulation data. The final stage involves the study of the characteristics of the imputation method. The analysis procedure in this study is presented in Fig. 1.

A. ARIMA Simulation Data Generation

In this study, simulations were conducted to generate time series data with various model types, including AR(1), AR(2), MA(1), MA(2), ARMA(1,1), ARMA(2,2), ARIMA(1,1,1), and ARIMA(1,2,1)). In the context of time series analysis, models such as AR(1), AR(2), MA(1), MA(2), ARMA(1,1), ARMA(2,2), ARIMA(1,1,1), and ARIMA(1,2,1) represent various patterns and behaviors that can be encountered in data. The AR(1) model (Autoregressive Order 1) models the current value as linearly dependent on the previous value with one autoregressive parameter. AR(2) is an extension of AR(1) with two autoregressive parameters. Conversely, the MA(1) model (Moving Average Order 1) and MA(2) model the current value as linearly dependent on the previous error term with one and two moving average parameters, respectively. The ARMA(1,1) and ARMA(2,2) models combine autoregressive and moving average components in one model, each having one autoregressive parameter and one moving average parameter. Meanwhile, the ARIMA(1,1,1) and ARIMA(1,2,1) models are integrated ARMA models designed to handle trends or seasonality in data, with additional differencing components. In these eight types of models, parameters such as ϕ (autoregressive), θ (moving average), and their orders determine the properties and patterns of the resulting models, which are then used for simulation and analysis of time series data [13]. The simulation generation for ARIMA will utilize the arima.sim package. The models mentioned above will be generated based on the following steps:

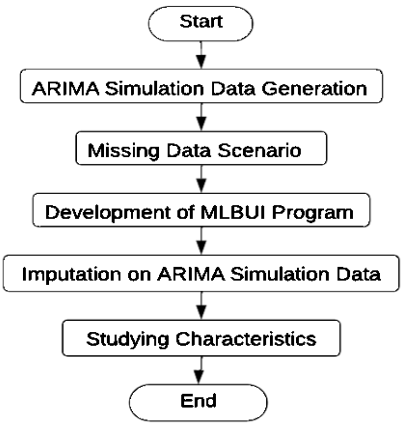


Fig. 1 Research procedure

1) Generate $e \sim N(0, \sigma_e^2)$ with $n = 100$. To generate $e \sim N(0, \sigma_e^2)$ with n=100, first specify the mean μ=0 and standard deviation σ_e^2 for the normal distribution. Then, use a random number generator to create 100 samples from this distribution.

2) The parameters used for this simulation data are $\mu = 15$ where μ represents the mean of the distribution.

3) First AR parameter (ϕ_1), Second AR parameter (ϕ_2), First MA parameter (θ_1), and Second MA parameter (θ_2) for AR(1), AR(2), MA(1), MA(2), ARMA(1,1), ARMA(2,2), ARIMA(1,1,1), and ARIMA(1,2,1) are presented in Table I. Parameters ϕ_1, ϕ_2, θ_1 and θ_2 are the coefficients for the autoregressive (AR) terms and moving average (MA) terms in the respective models.

4) Standard deviation will be created for each model scenario. The standard deviations used are: 0.5, 1, and 2.

5) This process will be repeated 100 times.

B. Missing Data Scenario

In this study, the focus is on missing values located in the middle, specifically between $3 \times T$ and $N - 3 \times T$ (N being the length of the original time series with T missing values). Three scenarios of missing data values will be applied to simulated data of AR(1), AR(2), MA(1), MA(2), ARMA(1,1), ARMA(2,2), ARIMA(1,1,1), and ARIMA(1,2,1) by removing parts of the data as follows:

Scenario 1: 6% missing data is consecutively located in the middle.
Scenario 2: 10% missing data is consecutively located in the middle.
Scenario 3: 14% missing data is consecutively located in the middle.

C. Development of MLBUI Program

Phan [10] developed the MLBUI Algorithm introduced to ensure consistent improvement in imputation outcomes. In the use of the MLBUI program, univariate data in vector Da (data after missing values) and vector Db (data before missing values) are transformed into multivariate data using matrices M Da and M Db. First, when missing data (gaps with a size of T) is at the beginning, located at $3 \times T$ from the database, machine learning methods will be applied to the remaining transformed data after the gap. If it is at the end, located at $N - 3 \times T$ from the database, this ML method is performed on the transformed data before the gap. If it is in the middle of the series, namely between $3 \times T$ and $N - 3 \times T$ (where N is the length of the original time series), the ML method is applied to the forward and backward data sets to predict the missing data. The MLBUI (Machine Learning-Based Univariate Imputation) program is designed to address the issue of incomplete data in time series, where the input $X = \{x_1, x_2, \ldots, x_N\}$ represents the incomplete time series. The program identifies t as the index of the missing data, indicating the position of the first missing value, and T as the total number of missing data points. The data will be divided into two parts: univariate data before imputation (Db) and univariate data after imputation (Da). These two datasets will then be transformed into multivariate data using the matrices M Db and M Da . Subsequently, a machine learning model will be trained, followed by a single forecasting step, which will produce the vectors $\hat{\boldsymbol{x}}\boldsymbol{b}$ dan $\hat{\boldsymbol{x}}\boldsymbol{a}$ representing the predicted results, respectively. The results will then be averaged and used to fill in the missing values. The output of this program is Y. The MLBUI program consists of five phases. The algorithm details are explained in illustrated in Fig. 2 and Fig. 3.

TABLE I
MODEL PARAMETERS

Parameters	AR(1)	AR(2)	MA(1)	MA(2)	ARMA (1,1)	ARMA (2,2)	ARMA (1,1,1)	ARMA (1,2,1)
First AR parameter (ϕ_1)	0.7	0.6	-	-	0.6	-0.5	0.6	0.3
Second AR parameter (ϕ_2)	-	0.3	-	-	-	-0.3	-	-
First MA parameter (θ_1)	-	-	0.5	0.4	0.3	0.4	0.6	0.3
Second MA parameter (θ_2)	-	-	-	0.2	-	-0.2	-	-

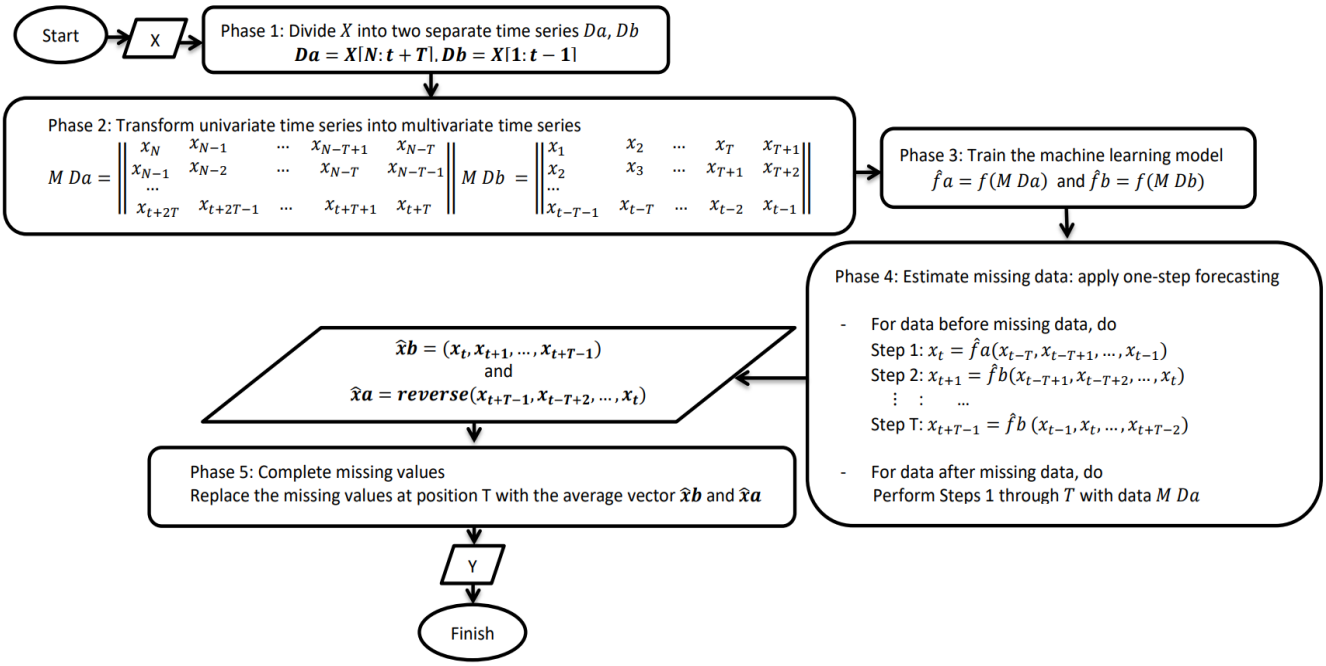
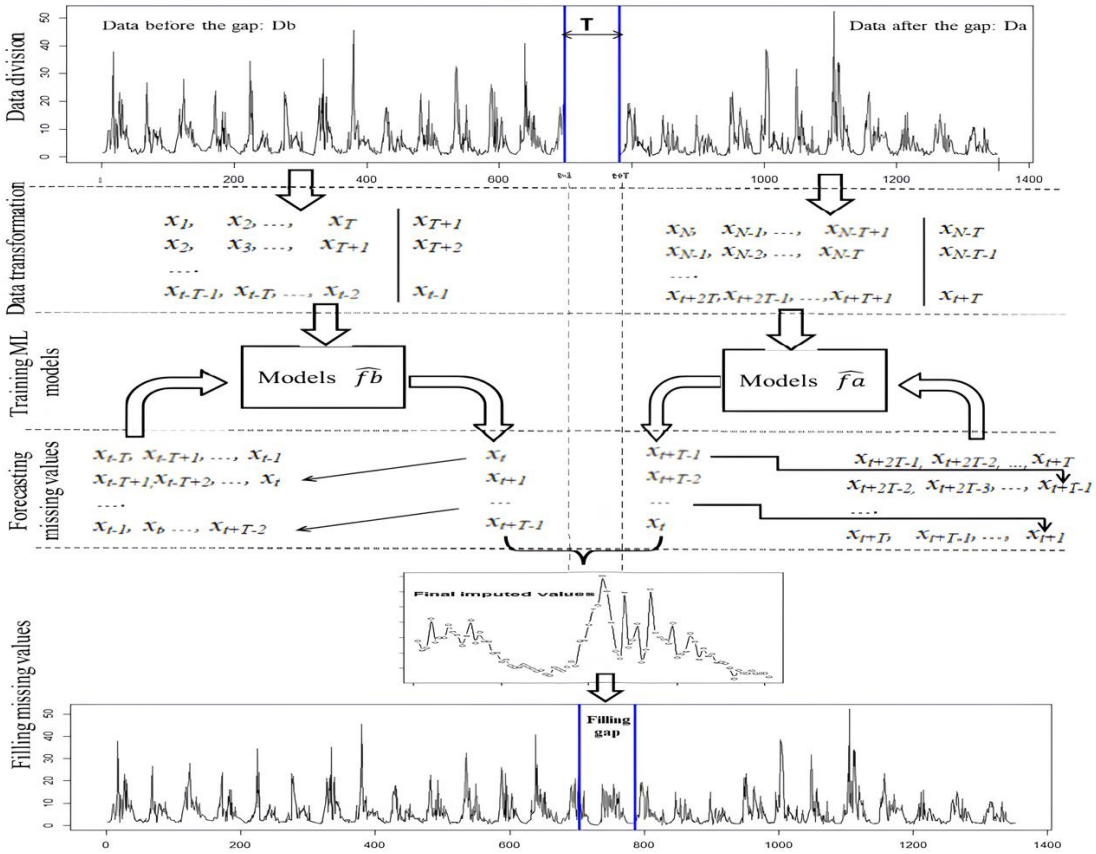


Fig. 2 Flowchart of the MLBUI algorithm



(Source: T. T. H. Phan, "Machine Learning for Univariate Time Series Imputation," in MAPR, 2020, pp. 3–4.)

Fig. 3 Scheme of the MLBUI algorithm

D. Imputation on ARIMA Simulation Data

In this study, the methods used are Kalman StructTS, Kalman Auto-Arima, Spline Interpolation, Stine Interpolation, Moving Average, MLBUI SVR, and MLBUI RFR [14]. The functions for Kalman StructTS, Kalman Auto-Arima, Spline Interpolation, Stine Interpolation, and Moving Average methods are available in the imputeTS package on CRAN.

E. Studying Characteristics

In this stage, an examination is conducted on the analyses performed to evaluate their accuracy values, aiming to understand the characteristics of each method used in the analysis. Subsequently, Kruskal-Wallis and Dunn tests are carried out to assess the significance of the observed differences between these methods. The accuracy matrix utilized in this research is the Mean Absolute Percentage Error (MAPE) matrix, commonly employed to measure how accurately a model predicts actual values. MAPE calculation computes the relative prediction errors to actual values, where the lower the MAPE value, the better the quality of the model's predictions. Eq. (1) illustrates the formula for calculating MAPE, where y_i represents the actual value of the i th data sample, x_i is the model's predicted value for the i th data sample, and n is the total number of data samples. The drawback of normalization is that MAPE becomes undefined for datasets containing values of 0 [15]. The results of the MAPE calculation will provide insights into the performance of each method in predicting the data.

$$MAPE = \frac{\sum_{t=1}^{n} \frac{|x_i - y_i|}{|y_i|}}{n} \quad (1)$$

Accuracy comparison of imputation methods is conducted using MAPE, which is calculated for each imputation method across various data scenarios. The method with the smallest MAPE value is considered the best, as it indicates the lowest prediction error. Additionally, MAPE values are categorized individually to provide a more detailed assessment. The categorization is based on predefined ranges: MAPE less than 5% is considered excellent, between 5% and 10% is considered good, between 10% and 20% is considered fair, and over 20% is considered poor.

III. RESULT AND DISCUSSION

A. Characteristics of MLBUI

In this stage, a Kruskal-Wallis Analysis will be conducted to evaluate whether there are significant differences between specific factors or groups. The reason for using this test is because the basic assumptions of normality and homogeneity are not met in the data. The data used as input for this Kruskal-Wallis test is the median of MAPE. The Kruskal-Wallis test serves as a nonparametric alternative to one-way ANOVA, commonly employed when parametric assumptions are unmet. Its primary objective is to assess whether the examined samples are derived from identical distributions. Essentially, the hypothesis evaluated in the Kruskal-Wallis test concerns whether the median values across different populations are statistically equivalent or divergent. The outcomes of this test can offer deeper insights into the presence of significant differences among the groups under scrutiny. Simply put, the hypotheses for Kruskal-Wallis in this study are:

H_0: The median of all groups is the same.

H_1: At least one group median is different from the median of other groups.

When the Kruskal-Wallis test produces significant outcomes, it indicates that there is at least one distinction among the samples. However, the test does not specify the locations or quantify the number of discrepancies. Consequently, additional examinations are required. Groups that are found to be significant in the overall Kruskal-Wallis test will then undergo further testing. In this study, the post-hoc test used is the Dunn test [16]. This test is useful for comparing all possible pairs of groups, allowing for the identification of significant differences that may not have been detected in the Kruskal-Wallis test. The hypotheses for the Dunn test are as follows:

H_0: There is no significant difference between the two groups being compared.

H_1: There is a significant difference between the two groups being compared.

The Kruskal-Wallis test and Dunn's test are conducted with a significance level of 5%. In other words, if the p-value is less than 0.05, the null hypothesis (H_0) will be rejected. If H_0 is rejected in the Kruskal-Wallis test, the conclusion is that at least one group median is different from the median of other groups. If H_0 is rejected in Dunn's test, there is a significant difference between the two groups being compared. In this study, there are several factors or groups, namely missing data scenario, standard deviation, model, and method. Some factors will be combined into one unit so they can be used for Kruskal-Wallis test. Kruskal-Wallis and Dunn tests will be conducted on factors such as method, Method-Missing Data Scenario, Method-Standard deviation, Method-Model, and Method-Model-Standard deviation. This is done to observe the characteristics of MLBUI compared to other methods. The results of the

Kruskal-Wallis and Dunn tests will be presented in a table containing the mean values and standard deviations. Each column in the table will indicate letters representing the level of significance of each group. Groups with the same letters indicate that they do not have significant differences. The results of the Kruskal-Wallis and Dunn tests on the method factor are presented in Table II.

As seen in Table II, the overall performance of MLBUI RFR is not significantly better than Stine Interpolation and Moving Average methods. However, it is noted that MLBUI RFR significantly outperforms Spline Interpolation and Kalman Auto-ARIMA. It is important to note that for the Kalman StructTS method in the model scenario, it is only used for AR, MA, and ARMA models because it cannot achieve convergence to handle missing data values in the ARIMA model, which may affect the average MAPE values. Furthermore, Table II also indicates that the MLBUI RFR method has the smallest standard deviation, indicating that MLBUI is the most consistently performing method overall compared to other methods.

Next, the characteristics of MLBUI will be examined in the missing data scenario.The results of the Kruskal-Wallis and Dunn tests on the method-missing data scenario factor are presented in Table III.

In Table III, it can be observed that increasing missing data in the MLBUI method does not show significant differences in performance. However, there is a decrease in performance in MLBUI RFR, indicated by the decreasing mean MAPE values and increasing inconsistency as seen from the increasing standard deviation. Based on Table III, it can also be concluded that the performance pattern of MLBUI SVR is not apparent with the increasing missing data. Additionally, Table III shows that for the scenarios of 6% and 14% missing data, MLBUI ranks as the 4th best method after Stine interpolation, Moving Average, and Kalman structTS methods. Meanwhile, for the scenario of 10% missing data, MLBUI RFR ranks 3rd after Stine interpolation and moving average. Both MLBUI RFR and MLBUI SVR remain competitive with other methods despite their lower performance. This is because there are no significant differences based on post-hoc tests. However, significant differences occur with the Kalman Auto-Arima method in scenarios with data loss rates of 10% and 14% when compared to other methods. This difference is quite noticeable. Next, the characteristics of MLBUI will be examined in standard deviation scenario. The results of the Kruskal-Wallis and Dunn tests on the method-standard deviation factor are presented in Table IV.

TABLE II
MEAN AND STANDARD DEVIATION OF MAPE BY METHOD

Method	Mean	Standard Deviation
Spline Interpolation	10.68	8.23[b]
Stine Interpolation	4.64	3.30[d]
Kalman Auto-Arima	73.27	41.12[a]
Kalman StructTS	6.68	12.75 [d]
Moving Average	5.06	3.90 [cd]
MLBUI RFR	5.72	2.78 [cd]
MLBUI SVR	9.32	9.22 [bc]

Numbers followed by different letters indicate significantly different results from the Dunn test at the 5% level.

TABLE III
MEAN AND STANDARD DEVIATION OF MAPE BASED ON MISSING DATA SCENARIO FOR EACH METHOD

Missing Data Scenario	Mean / Standard Deviation						
	MLBUI RFR	MLBUI SVR	Kalman StructTS	Kalman Auto-ARIMA	Spline Interpolation	Stine Interpolation	Moving Average
6%	4.93	8.16	4.92	71.96	7.91	4.58	4.46
	1.74 [d]	6.31 [bcd]	1.74 [d]	41.57 [d]	5.82 [bcd]	3.37 [d]	3.46 [d]
10%	5.75	10.27	10.1	73.85	10.7	4.58	5.05
	2.46 [cd]	10.98 [bcd]	21.96 [cd]	41.31 [a]	7.92 [abc]	3.27 [d]	3.82 [cd]
14%	6.47	9.52	5.02	74	13.44	4.76	5.66
	3.67 [bcd]	9.98 [bcd]	1.72 [cd]	42.22 [a]	9.82 [ab]	3.39 [d]	4.43 [bcd]

Numbers followed by different letters indicate significantly different results from the Dunn test at the 5% level.

TABLE IV
MEAN AND STANDARD DEVIATION OF MAPE BASED ON ERROR STANDARD DEVIATION SCENARIO FOR EACH METHOD

Standard Deviation	Mean Standard Deviation						
	MLBUI RFR	**MLBUI SVR**	**Kalman StructTS**	**Kalman Auto-ARIMA**	**Spline Interpolation**	**Stine Interpolation**	**Moving Average**
0.5	4.51 2.91^{gh}	8.08 9.85^{fgh}	3.38 0.6^{hi}	72.65 43.26^{abc}	7.18 5.04^{def}	3.09 1.92^{hi}	0.54 0.43^{i}
1	5.57 2.56^{efg}	9.21 9.32d^{ef}	9.93 21.95^{efg}	74.55 40.01^{a}	10.33 7.39^{cd}	4.51 2.87^{fgh}	6.41 2.64^{def}
2	7.07 2.31^{de}	10.66 8.67^{bcd}	6.73 1.3^{de}	72.62 41.77^{ab}	14.54 10.02^{bcd}	6.31 4.03^{def}	8.23 2.46^{bcd}

Numbers followed by different letters indicate significantly different results from the Dunn test at the 5% level.

The MLBUI RFR method ranks within the top three best methods, while MLBUI SVR does not. However, the performance of MLBUI RFR is not significantly different from MLBUI SVR. Table IV also reveals that the performance of MLBUI at an error standard deviation of 0.5 is not significantly different from the second-best method, Stine Interpolation. Additionally, MLBUI RFR's performance at standard deviations of 1 and 2 is not significantly different from the better-performing methods. This holds true for MLBUI SVR as well, as there is no significant difference in performance between MLBUI RFR and MLBUI SVR. The method that differs significantly based on standard deviation when compared to the MLBUI RFR method is the Kalman Auto-Arima method. Table IV also indicates that as error standard deviation increases, performance decreases for all methods. The increase in standard deviation of MLBUI RFR with each increment in Standard deviation is only slight. The standard deviation of MLBUI RFR at standard deviations 0.5, 1, and 2 is not significantly different from other methods. This indicates that MLBUI RFR maintains consistent and robust performance against increasing standard deviation. The next stage to be observed is the characteristics of MLBUI based on the Model. The results of the Kruskal-Wallis and Dunn tests on the Method-Model factor are presented in Table V.

In Table V, it can be observed that the MLBUI RFR method demonstrates excellent performance with the MA(2), AR(1), and MA(1) models, as does MLBUI SVR compared to other models. MLBUI RFR ranks as the second best method for the AR(1), AR(2), MA(2), and ARMA(1,1) models, and ranks third for ARMA(2,2). MLBUI SVR ranks third for the AR(1), MA(1), and ARMA(1,1) models. This indicates that the MLBUI SVR method is less effective compared to other methods for the ARIMA(1,1,1) and ARIMA(1,2,1) models. Furthermore, based on post-hoc tests, it can be seen that the performance of the MLBUI method does not significantly differ from methods with better performance on stationary models, such as AR(1), AR(2), MA(1), MA(2), ARMA(1,1), and ARMA(2,2). However, significant differences are observed for non-stationary models, namely ARIMA(1,1,1) and ARIMA(1,2,1). In the Kalman StructTS column of Table V, it is noticed that there is an empty column. This is due to the convergence limitations of Kalman StructTS in handling missing data values in the ARIMA(1,1,1) and ARIMA(1,2,1) models. As a result, the performance of Kalman StructTS cannot be evaluated for these models in the table. This suggests that Kalman StructTS is not suitable for use in cases where there are trends or seasonal components in the data, which can be found in the ARIMA(1,1,1) and ARIMA(1,2,1) models. Therefore, the Kalman Auto-ARIMA method is added. Kalman Auto-ARIMA can impute all models in this study. However, the Kalman StructTS method is still used in this study because StructTS is the default model in the function, and it still demonstrates good accuracy for the AR(1), AR(2), MA(1), MA(2), ARMA(1,1), and ARMA(2,2) models.

The next step involves examining performance across scenarios to understand MLBUI's performance in each scenario. Passive voice: The next step involves examining performance across scenarios to understand the performance of MLBUI in each scenario. Therefore, the merging of missing data, error standard deviation, model, and method groups will be conducted. This comparison provides information that MLBUI SVR is never the best-performing method in any scenario, but it ranks within the top three in some scenarios. On the other hand, MLBUI RFR consistently ranks within the top three in every scenario except for the ARIMA model in every error standard deviation and missing data scenario.

TABLE V
MEAN AND STANDARD DEVIATION OF MAPE BASED ON MODEL SCENARIO FOR EACH METHOD

Model	Mean Standard Deviation						
	MLBUI RFR	**MLBUI SVR**	**Kalman StructTS**	**Kalman Auto-ARIMA**	**Spline Interpolation**	**Stine Interpolation**	**Moving Average**
AR(1)	4.04 1.3^{qrs}	4.12 1.33^{opqrs}	4.1 1.27^{pqrs}	93.14 2.61^{abc}	11.01 4.42^{ef}	4.83 1.37^{klmnopqrs}	3.78 2.75^{nopqrs}
AR(2)	6.13 1.85^{hijklm}	6.3 1.96^{hijk}	6.21 1.99^{hijkl}	105.18 3.36^{a}	16.48 6.62^{cde}	7.7 2.52^{fgh}	5.86 4.34^{hijklmn}
MA(1)	4.49 1.29^{lmnopqrs}	4.46 1.22^{lmnopqrs}	4.45 1.19^{mnopqrs}	94.69 2.05^{abc}	11.96 4.8^{de}	5.39 1.48^{ijklmnopqr}	4.05 2.91^{lmnopqrs}
MA(2)	3.77 1.16^{s}	3.82 1.16^{rs}	3.79 1.12^{s}	86.29 5.36^{abc}	9.84 3.48^{efg}	4.32 1.30^{nopqrs}	3.4 2.45^{qrs}
ARMA(1,1)	5.53 1.66^{hijklmnopq}	5.63 1.68^{hijklmnopq}	15.86 30.78^{hijkl}	100 0.00^{ab}	18.34 6.37^{bcde}	7.31 2.22^{fghi}	5.32 3.80^{hijklmnop}
ARMA(2,2)	5.74 1.68^{hijklmno}	5.8 1.64^{hijklmn}	5.67 1.58^{hijklmnop}	100 0.00 ^{ab}	17.56 7.15^{bcde}	7.36 2.31^{fghi}	5.37 3.89^{hijklmnop}
ARIMA(1,1,1)	5.23 1.66^{jklmnopqrs}	13.07 1.52 ^{cde}	- -	0.14 0.04^{t}	0.25 0.10^{t}	0.18 0.06^{t}	4.26 2.99^{klmnopqrs}
ARIMA(1,2,1)	10.82 3.60^{ef}	31.34 6.81^{abcd}	- -	6.73 7.35^{ijklmnopqr}	0.01 0.00^{t}	0.01 0.00^{t}	8.42 5.92^{ghij}

Numbers followed by different letters indicate significantly different results from the Dunn test at the 5% level.

For AR(1) and AR(2) models, MLBUI RFR consistently performs as the best method with standard deviations of 1 and 2 across all missing data scenarios. MLBUI RFR also performs the best for the MA(2) model with an error standard deviation of 2 at 6% missing data, and with standard deviations of 1 and 2 at 10% and 14% missing data. For the ARMA(1,1) model with an error standard deviation of 1, MLBUI becomes the best method across all missing data scenarios. MLBUI RFR also becomes the best method for the ARMA(1,1) model with an error standard deviation of 2 and the ARMA(2,2) model with an error standard deviation of 1, but only in the 14% missing data scenario; otherwise, MLBUI RFR does not rank as the best method. Considering these findings, Kruskal-Wallis and post-hoc tests will be conducted to determine whether there are significant differences. The Kruskal-Wallis test results in a non-significant outcome when merging missing data, error standard deviation, model, and method groups, with a p-value of 0.4915. Therefore, the groups will be merged without the missing data group. This merging based on Kruskal-Wallis yields a test statistic of <2e-16, indicating that at least one group differs from the others. Post-hoc tests reveal no significant differences when comparing methods better than MLBUI, suggesting that the top three methods have statistically indistinguishable performance. This holds true for AR(1), AR(2), ARMA(1,1), and ARMA(2,2) models, while for ARIMA(1,1,1) and ARIMA(1,2,1) models, MLBUI exhibits significant differences from better-performing methods. In conclusion, MLBUI performs well on data with stationary models such as AR(1), AR(2), ARMA(1,1), and ARMA(2,2), but performs less effectively on data with non-stationary models like ARIMA(1,1,1) and ARIMA(1,2,1).

B. Consistency of MLBUI Performance Based on Mean Absolute Percentage Error (MAPE) Using Boxplot

In this stage, the research focuses on observing the consistency of performance among various models based on the analysis using Mean Absolute Percentage Error (MAPE) boxplots. Boxplots are compact summaries of distributions, displaying less detail than histograms or kernel density plots, but also taking up less space. Boxplots are used as a tool to assess the consistency and accuracy of the applied methods by considering outlier criteria, range, and median. Fewer outliers, as well as smaller range and median values, indicate that the method's performance is more consistent. On the other hand, lower MAPE values also indicate that the method has a higher accuracy in imputing missing data. Therefore, this stage aims to provide a comprehensive overview of how well these methods handle missing data. The data presented in the form of boxplots in Fig. 4 allows for a visual comparison of the performance among these methods, which can serve as a guide in selecting the most suitable method for imputation purposes in univariate time series.

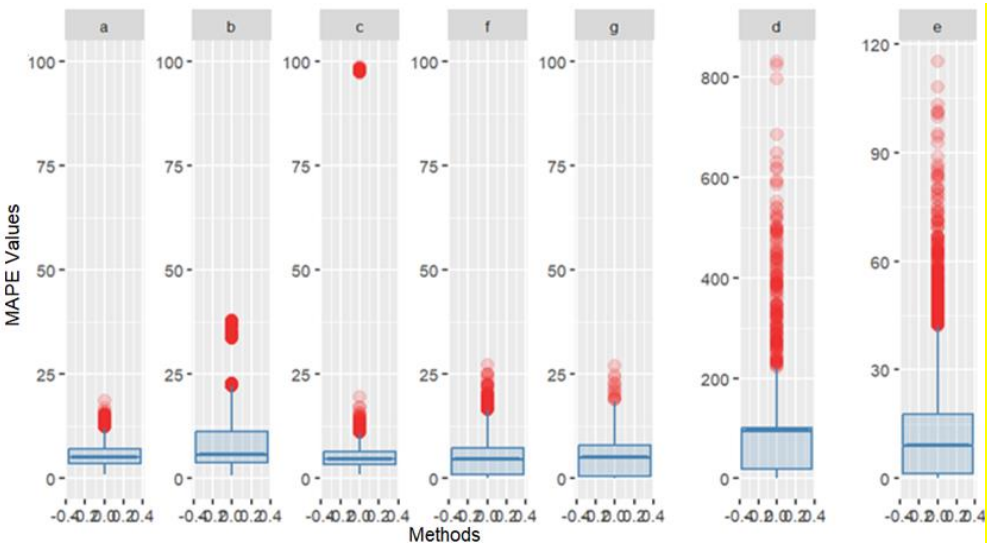


Fig. 4 Boxplot of MAPE for methods (a) MLBUI RFR, (b) MLBUI SVR, (c) Kalman StructTS, (d) Kalman Auto-ARIMA, (e) Spline Interpolation, (f) Stine Interpolation, and (g) Moving Average.

From Fig. 4, it can be seen that the MLBUI RFR method emerges as the most consistent method compared to others. Note that in the MAPE boxplot, the number of outliers in the MLBUI RFR method is fewer compared to others, except for Moving Average, although its outliers are not as significant as Moving Average. The MLBUI RFR boxplot also has the smallest range after Kalman StructTS. The median of the MLBUI RFR boxplot is also the lowest compared to other methods. On the other hand, in Fig. 4, the boxplot of the MLBUI SVR method shows a relatively wide range compared to MLBUI RFR, Kalman StructTS, and Stine Interpolation, but smaller than Kalman Auto-ARIMA and Spline Interpolation. There are many outliers in the MLBUI SVR boxplot, indicating a high level of variability, while the median value is almost close to Stine Interpolation, Moving Average, and MLBUI RFR, suggesting that MLBUI SVR is a method that is not as consistent as Stine Interpolation, Moving Average, and MLBUI RFR.

IV. CONCLUSION

In the "Results and Analysis" chapter, it is concluded that the MLBUI method is effective in handling missing values successively, as expected from the "Introduction" chapter. The MLBUI RFR method demonstrates good performance with an average MAPE of 5.06, which shows no significant difference compared to the top-performing methods (Stine Interpolation and Moving Average). Additionally, the MLBUI RFR method is the most consistent among other methods, with a standard deviation of 2.78. In this study, it was found that the performance of MLBUI RFR decreases with increasing missing data and standard deviation. Its characteristics show superiority in stationary models, such as AR, MA, ARMA, but it is less effective for non-stationary data, namely ARIMA. Therefore, parameter enhancements in MLBUI are needed to better address non-stationary data, and alternative machine learning methods such as eXtreme Gradient Boosting (XGBoost) regression can also be considered. Subsequent research can focus on trends or seasonality in the data to better understand the characteristics of MLBUI.

ACKNOWLEDGEMENT

Thank you to the Statistics and Data Science Program, IPB University, for their support and efforts in completing this research.

REFERENCES

[1] A. Syukur and A. Marjuni, "Stock price forecasting using univariate singular spectral analysis through hadamard transform," *Int. J. Intell. Eng. Syst*, vol. 13, no. 2, pp. 96–107, 2020, doi: 10.22266/ijies2020.0430.10.

[2] S. Mishra, C. Bordin, K. Taharaguchi, and I. Palu, "Comparison of deep learning models for multivariate prediction of time series wind power generation and temperature," *Energy Reports*, vol. 6, no. 3, pp. 273–286, 2020, doi: 10.1016/j.egyr.2019.11.009.

[3] Y. Ensafi, S. H. Amin, G. Zhang, and B. Shah, "Time Series Forecasting of Seasonal Items Sales using Machine Learning – A comparative analysis," *Int. J. Inf.*

Manag. Data Insights, vol. 2, no. 1, p. 100058, 2022, doi: 10.1016/j.jjimei.2022.100058.

[4] J. Park, J. Muller, B. Arora, B. Faybishenko, G. Pastorello, C. Varadharajan, R. Suhu, and D. Argarwal, "Long-term missing value imputation for time series data using deep neural networks," *Neural Comput. Appl*, vol. 35, no. 12, pp. 9071–9091, 2023, doi: 10.1007/s00521-022-08165-6.

[5] G. Chhabra, "Comparison of imputation methods for univariate time series," *Int. J. Recent Innov. Trends Comput. Commun*, vol. 11, no. 2s, pp. 286–292, 2023, doi: 10.17762/ijritcc.v11i2s.6148.

[6] M. Meggiorin, G. Passadore, S. Bertoldo, A. Sottani, and A. Rinaldo, "Comparison of three imputation methods for groundwater level timeseries," *Water (Switzerland)*, vol. 15, no. 4, p. 801, 2023, doi: 10.3390/w15040801.

[7] D. A. Gomez-Cravioto, R. E. Diaz-Ramos, F. J. Cantu-Ortiz, and H. G. Ceballos, "Data analysis and forecasting of the COVID-19 spread: A comparison of recurrent neural networks and time series models," *Cognit. Comput*, vol. 16, no. 4, pp. 1794-1805, 2021.

[8] A. Y. Yldz, E. Koc, and A. Koc, "Multivariate time series imputation with transformers," *IEEE Signal Process. Lett*, vol. 29, no. 507, pp. 2517–2521, 2022, doi: 10.1109/LSP.2022.3224880.

[9] S. Jain, N. Choudhary, and K. Jain, "Outlier detection and imputation of missing data in stock related time series mulitivariate data using LSTM autoencoder," *J. Integr. Sci. Technol*, vol. 12, no. 3, pp. 761-761, 2024, doi: 10.62110/sciencein.jist.2024.v12.761.

[10] T. T. H. Phan, "Machine Learning for Univariate Time Series Imputation," in *2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, 2020, pp. 1–6. doi: 10.1109/MAPR49794.2020.9237768.

[11] G. Shanmugasundar, M. Vanitha, R. Čep, V. Kumar, K. Kalita, and M. Ramachandran, "A comparative study of linear, random forest and adaboost regressions for modeling non-traditional machining," *Processes*, vol. 9, no. 11, p. 2015, 2021, doi: 10.3390/pr9112015.

[12] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," *Neural Comput. Appl*, vol. 35, no. 32, pp. 23295–23309, 2023, doi: 10.1007/s00521-021-05842-w.

[13] Y. Lai and D. A. Dzombak, "Use of the Autoregressive Integrated Moving Average (ARIMA) model to forecast near-term regional temperature and precipitation," *Weather Forecast*, vol. 35, no. 3, pp. 959–976, 2020, doi: 10.1175/WAF-D-19-0158.1.

[14] A. Denhard, S. Bandyopadhyay, A. Habte, and M. Sengupta, "A Comparison of Time Series Gap-Filling Methods to Impute Solar Radiation Data," in *Proceedings - ISES Solar World Congress 2021*, 2021, pp. 1–14. doi: 10.18086/swc.2021.38.03.

[15] A. A. Mir, K. J. Kearfott, F. V. Çelebi, and M. Rafique, "Imputation by Feature Importance (IBFI): A methodology to envelop machine learning method for imputing missing patterns in time series data," *PLoS One*, vol. 17, no. 1, p. e0262131, 2022, doi: 10.1371/journal.pone.0262131.

[16] D. G. da Silva, M. T. B. Geller, M. S. dos S. Moura, and A. A. de M. Meneses, "Performance evaluation of LSTM neural networks for consumption prediction," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 2, no. 47, p.100030, 2022, doi: 10.1016/j.prime.2022.100030.