5-2023

Effects of Missing Data Imputation Methods on Univariate Time Series Forecasting with Arima and LSTM

Nicholas Niako
The University of Texas Rio Grande Valley

EFFECTS OF MISSING DATA IMPUTATION METHODS ON UNIVARIATE TIME SERIES FORECASTING WITH ARIMA AND LSTM

A Thesis

by

NICHOLAS NIAKO

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Mathematics

The University of Texas Rio Grande Valley

May 2023

EFFECTS OF MISSING DATA IMPUTATION METHODS ON UNIVARIATE TIME SERIES FORECASTING WITH ARIMA AND LSTM

A Thesis

by

NICHOLAS NIAKO

COMMITTEE MEMBERS

Dr. Kristina Vatcheva
Chair of Committee

Dr. Oleg Musin
Committee Member

Dr. Santanu Chakraborty
Committee Member

Dr. Mrinal Kanti Roychowdhury
Committee Member

May 2023

ABSTRACT

Niako, Nicholas Effects of missing data imputation on univariate time series forecasting using ARIMA and LSTM. Master of Science (MS), May, 2023, 102 pp., 17 tables, 37 figures, references, 94 titles.

Missing data are common in real-life studies and missing observations within the univariate time series cause analytical problems in the flow of the analysis. Imputation of missing values is an inevitable step in the analysis of every incomplete univariate time series data. The reviewed literature has shown that the focus of existing studies is on comparing the distribution of imputed data. There is a gap of knowledge on how different imputation methods for univariate time series data affect the fit and prediction performance of time series models. In this work, we evaluated the predictive performance of autoregressive integrated moving average (ARIMA) and long short-term memory (LSTM) models on imputed time-series data using Kalman smoothing on ARIMA, Kalman smoothing on structural time series model, mean imputation, exponentially weighted moving average, simple moving average, linear, cubic spline, stine, and KNN interpolation techniques under missing completely at random (MCAR) mechanism. Missing values were generated at 10%, 15%, 25%, and 35% rates using complete data of 24-hour ambulatory diastolic blood pressure readings. The performance of models was compared on imputed and original data using mean absolute percentage error (MAPE) and root mean square error (RMSE). Kalman smoothing on structural time series, exponentially weighted moving average, and Kalman smoothing on ARIMA were the best missing data replacement techniques as the gap of the

missingness increased. The performance of mean imputation, cubic spline, KNN, and the other simple interpolation methods reduced significantly as the gap of missingness increased. The LSTM gave better predictions on the original training data, but the ARIMA predictions on imputed data gave consistent results across the four scenarios.

DEDICATION

I dedicate this work to my family back home in Ghana and the Ghanaian community here at the Ro Grande Valley. Your advice and support have been very punctual. I am grateful for having you all.

ACKNOWLEDGMENTS

I want to express my profound gratitude to Dr. Kristina Vatcheva for providing me with the thesis research topic and for her expert supervision throughout the development and completion of my thesis. In addition, I would like to thank Drs. Chakraborty, Musin, and Roychowdhury for serving on my thesis committee. Finally, I am grateful to Dr. Gladys Maestre for granting me access to the Maracaibo Aging Study data used as part of my thesis.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

1.1 Time series data

Time series data comprise a collection of values of a given stochastic process or phenomenon monitored over regular sampling intervals for a certain period. Time series data presents useful historical data about any process by monitoring the sequential behavior of that process in the short or long term. This process (or dependent variable) can be anything that can be conceived as a unitary entity (Wei, 2006; Cowperthwait and Metcalfe, 2009). Due to its nature, time series data can be found in every field such as traffic control management (Li et al., 2015), healthcare (Zeger, Irizarry, & Peng, 2006; Penfold and Zhang, 2013), and economics (Granger and Newbold, 2014; Yang, 2012). In healthcare, several studies have discussed the role and significance of time series data generated from blood pressure and heart rate for exploring; the progression and prognosis of disease outcomes (Li-Wei et al, 2014; Wiens, Horvitz, and Guttag, 2012; Li-Wei et al, 2013), Cardiac autonomic function in Psychiatry, panic disorders in neurological and Cardiovascular Research (Yeragani, 1995; Angelini et al., 2007; Rao and Yeragani, 2001; Yeragani et al., 2003, Chuah & Fu, 2007).

1.2 Serial dependency

Datasets recorded successively from single or multiple subjects over time exhibit serial dependency with their prior values. Thus, such observations cannot be assumed to be statistically independent of their prior or subsequent values (Shumway & Stoffer, 2019; Box et al., 2015). As

a result, standard regression inference procedures, which assume independence between subjects are invalid for the analysis of time series data. The statistical procedure used to analyze and model the nature of such dependencies within the data is called time series analysis. Often, time series analysis aims to understand the underlying stochastic behavior of the process by fitting an appropriate model, and to predict the likely future behavior of the process based on past and current data gathered from the process. This interest in time series data has led to the development several stochastic and dynamic models and methodologies in the statistical literature. Three model-based techniques have been used for forecasting in time series data (Hyndman & Athanasopoulos, 2021). The first model-based approach solely uses the historical data of the dependent variable to make forecasts. These include the Exponential smoothing-based techniques, and Autoregressive Integrated Moving Average (ARIMA) based models (Box et al., 2015). The exponential techniques make forecasts using the weighted averages of the previous values of the process, for this reason, more recent observations are assigned heavier weights. Consequentially, the weights decay in an exponential manner with time, hence the name. These techniques include Holt's linear trend method and Holt-Winters' seasonal method. See (Kalekar, 2004; Hyndman & Athanasopoulos, 2021) for a thorough overview. In the second model-based forecasting approach, a vector of predictors in the form of an explanatory model is used to forecast the likely values of a selected variable in a future time. These models include Vector autoregressive techniques (Box et al., 2015). The third approach includes Machine learning-based techniques like Recurrent Neural Network (RNN) models such as Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), which are capable of learning the behavior of the underlying stochastic process to predict

the likely future realizations of the process. These machine learning-based techniques unlike the other model-based forecasting techniques mentioned do not make any prior assumptions about the data. The method of analyzing time series data depends on the analyst, and often the type and nature of the data. Traditionally, for univariate time-series data, pure time-series models are preferred. Among traditional univariate modeling techniques, the Exponential smoothing models focus on the description of the trend and seasonality in the data, while ARIMA models aim to describe the serial dependencies within the series (Hyndman & Athanasopoulos, 2021).

1.3 Missing data

Missing data are common in real-life studies and missing observations within the series can cause analytical problems in the flow of the analysis. Missing data are unobserved values that, if observed, could be useful for statistical analysis (Little and Rubin, 2019). In practice, missing data are unavoidable in many data collection process. Particularly, when the values are taken from study unit(s) repeatedly over time (Laird, 1988; Yozgatligil et al., 2013) due to factors like human error or mechanical failure. For instance, when measurements of the blood pressure of an individual are monitored regularly over time, there is a high tendency to encounter poor internet connection for data transmission (in the case of mobile health where vitals can be monitored remotely and transmitted over the internet to a health professional), a hardware component failure, depleted batteries, cuff bladder leaks or low adherence to monitoring schedule by patients. These and other similar occurrences can lead to missing blood pressure readings which if observed would be useful for medical or clinical assessment.

Depending on the manner and reasons that lead to missing observations, missing observations are classified as being missing completely due to random phenomenon (missing completely at random, MCAR), missing at random (MAR) and missing not at random (MNAR). In MCAR, it is assumed that observations are missing due to pure chance. This assumption implies that the reason behind the missing values is unimportant because it is neither related to the values themselves nor the observed. The MAR assumption suggests that the reasons behind the missing values are not entirely due to a random phenomenon but can be tied to the observed values. For instance, if men are less likely than women to complete a survey on the severity of their depression, then a registry studying depression may have missing data that are MAR. In this scenario, the likelihood that they would complete the survey is connected to their gender (which is observed) but not the degree of their depression (Mack, Su & Westreich, 2018). Both the MCAR and MAR assumption makes it possible to estimate missing values using the observed data without violating the assumptions of missingness. In either case, the likelihood of missingness is not dependent on the missing data themselves (making the cause of missingness "ignorable"). On the other hand, the MNAR assumption suggests that the reason why the values get missing does not depend on the observed values but rather depends on the characteristics of the missing data itself (Rubin, 1976; Little & Rubin 2002, 2019).

Missing data is ubiquitous in all forms of longitudinal data collection. Longitudinal data broadly refers to any data gathered over time on a certain variable(s) of interest. The primary objective of the most longitudinal analysis is to describe the mean response as a function of time, and perhaps other covariates associated with units or individual measurements (Diggle and

Kenward, 1994; Longford, 2007). The underlying concept of collecting data on the behavior or effect of a covariate on a process "over time" is shared by both time series and traditional longitudinal analysis. The distinction, however, is in the length of the data; the data used in conventional longitudinal designs typically comes from a small number of time points (i.e a short series of time points, for instance, a one-month follow-up followed by a six-month follow-up), that can be equally or unequally spaced, whereas time series often consist of a relatively larger number of observations. Other qualities such as coming from a single unit (i.e univariate time series) and having observations that are regularly spaced (fixed or regular sampling intervals) are frequently linked to time series. Furthermore, time series can be generated from multiple units (a multiple time series) and can accommodate variations in overtime periods, these are typical of time series rather than essential qualities (Diggle et al., 1994; Cowperthwait and Metcalfe, 2009).

The consequences of missing values in cross-sectional and longitudinal data include biased estimates, reduced representativeness of a given sample (affects the sample size), and a loss of statistical power. The attrition in the longitudinal studies leads to different numbers of repeated measurements for subjects resulting in unbalanced data and to loss of information reducing the internal and external validity of the study. Internally the validity of the study is affected if, for example, in clinical-trial attrition is different between the treatment and control group which can lead to a treatment condition looking more effective than it is. The external validity of the study is affected if some participants have a higher probability of dropping than others; in this case, the generalizability of this group of participants is limited (Cox, 1972). Traditional approaches for handling missing values in statistical analysis of cross-sectional and longitudinal data are complete

case analysis (CC), available case analysis (AC), dummy variable adjustment, marginal (unconditional) mean imputation, and conditional mean imputation. More recent approaches to handling missing data are based on the maximum likelihood (ML) method, multiple imputations under normal model (MI), and Bayesian modeling with Markov chain Monte Carlo (MCMC) methods. Based on the assumptions of absence each of these methods has strengths and limitations. Poor management of missing values on the side of the analyst can result in inaccurate and misleading conclusions (Kang, 2013; van Buuren 2018; Enders, 2010; van Buuren & Groothuis-Oudshoorn, 2011).

1.3 Problems of missing data in time series

In time series analysis, the analysis cannot begin even if the missing data is a single value occurring after non-missing ones within the dataset. This is because traditional time series models like ARIMA and other learning algorithms like Long Short-Term Memory (LSTM) Recurrent Neural Network models were built for analyzing autocorrelation and patterns in sequential data with no missing values. Thus, the missing values need to be handled first before the analysis can be done. Ad hoc solutions for missing data do not generalize across all different types of data, such as longitudinal and time series data. For instance, deletion is perhaps the simplest and most common method of dealing with missing values in literature (Harel et al., 2012; Westreich, 2012; Harel & Boyko, 2013). Pairwise deletion or complete case analysis are other forms of deletion. In complete case analysis, only the rows that have no missing observations (complete rows) are considered. Thus, any observation that has a missing value is automatically taken out to avoid

complications. Theoretically, this method can be justified by the MCAR assumption, because data that are MCAR can be effectively considered as a random subsample of complete data from the overall sample. Therefore, in such instances statistical results from Complete case analysis are unbiased (Little, 1992). This method of handling missing data is the default of most statistical software programs including SAS (Kuligowski and Gharibvand, 2020), Stata, SPSS, and R. For instance, when fitting a linear regression model, rows of data with missing values are automatically removed and the complete dataset will be used by the software to analyze the data. While this technique is straightforward to implement, not every missing data are completely due to a random phenomenon. Also, by removing participants with partial data, the complete-case analysis misses potentially useful observed information and subsequent loss of statistical power (Noor et al 2015; van Buuren, 2018, Graham, 2009). This technique of deleting incomplete rows of data can be likened to "throwing the baby out with the bath water".

Due to the inherent serial dependency in time series data, handling missing data by deletion is theoretically inappropriate. Because if deletion is applied to time series data, regardless of the data characteristics, it is highly likely to fail in modeling the time dependencies. Also, this technique violates the assumption of continuity (equal time interval) between observations, required for some of the statistical methods, by causing time truncations or irregularities in the data (Velicer and Colby, 2005). Thus, appropriate values must be first imputed before conventional time series analysis can be done.

1.4 Imputation of missing values in univariate time series

Moritz et al (2015) provided an overview of several imputation techniques for univariate time series data. In their work, they evaluated imputation methods such as replacing missing values with the aggregated mean of the observed data, LOCF, and linear interpolation-based techniques. In their studies, an exponential distribution was used to replicate four distinct rates (0.1, 0.3, 0.5, and 0.7) of artificial missing values under the MCAR assumption in copies of the training data, and the above-listed techniques were used to impute the missing values. They showed that interpolation with seasonal Kalman filtering and Interpolation on seasonal loess decomposition yielded the best results. The LOCF and imputation with the mean of observed values had the best computational time but provided erroneous results. In the experiment, the performance of the imputation techniques was based on the root mean square error (RMSE) and mean absolute percentage error (MAPE) of the original and imputed data. The imputation technique which yields the smallest RMSE and MAPE was judged as the best. However, the study did not validate whether the best imputation techniques will translate into a successful time series analysis of the imputed data.

(Walter et al, 2013) compared the imputation performance of two Box Jenkins models (ARIMA and SARIMA) with a direct linear regression on non-stationary seasonal time series data. Their work focused on instances in which missing observations are encountered towards the end of the series. By only considering missing values that occur at the end of the data, ample data would be available to fit an ARIMA model and subsequently use the fitted model to impute the

missing values through prediction. To evaluate the performance of the imputation methods, five different levels of missing data were generated (5%, 7%, 10%, 12%, and 15% respectively) within each of the four different datasets that were used in the study. The rationale was to determine the most consistent of the proposed techniques as the percentage of missing data increases. The results showed that the direct linear regression technique yielded more accurate estimates after removing the seasonality within the series. However, the study only focuses on the scenario when missing values occur at the very beginning and the very end of the data. The imputation methods used will fail if a single missing value occurs within the data other than the beginning or end of the time series data. This scenario is unlikely to occur in health data and real-world scenarios where several factors lead to missing data.

Wijesekara and Liyanage (2020) aim to compare the performance of six imputation methods in univariate air quality data. The performance of these methods was evaluated across four (5%, 10%, 15%, and 20%) scenarios of missingness. The methods discussed in the study included mean imputation, spline interpolation, simple moving average, exponentially weighted moving average, Kalman smoothing on structural time series, and Kalman smoothing on ARIMA models. The study showed that Kalman smoothing on structural time series was the best method for replacing missing values that are MCAR in univariate time series data. The report showed that among the six methods, mean imputation was the worst-performing technique although it can perform considerably well when the rate of missingness is smaller. However, the study did not

examine the effects of these imputation techniques and how these changes can affect time series models.

Several other studies have demonstrated various techniques that solely rely on the features of the univariate time series data such as the periodic patterns within the observed data to impute missing the missing ones (Chaudhry et al, 2019; Bokde et al, 2018; Albano, Rocca, and Perna, 2017). Research has also been conducted to evaluate the performance of different methods of imputation techniques of missing data under different gap size scenarios. For example, (Caillault, Lefebvre, and Bigand, 2020; Albano, Rocca, and Perna, 2017) evaluated single value imputation techniques with methods for large gaps of missing data in univariate time series data. Based on the findings it is clear that different imputation methods are shown to be the best based on different scenarios of missing data patterns. Again, these large gap imputation techniques may provide a good fix for univariate time series data with large gaps of missing data. However, little is known about the reliability of the imputed data for time series modeling and prediction.

Imputation of missing values is an inevitable step in the analysis of every incomplete time series data. The reviewed literature has shown that the focus of existing studies ison comparing the distribution of imputed data and does not provide further analysis of the imputed data. There is a gap of knowledge on how different imputation methods for univariate time series data affect the model fitting and prediction performance of the analysis.

1.5 Study objectives

The objectives of our study were to investigate the effects of imputed data in univariate time series analysis on the data, modeling and prediction using ARIMA, and LSTM recurrent neural network model. Specifically, we studied the effects of mean imputation, Kalman filtering imputation, exponentially weighted moving average, simple moving average, interpolation, last observation carried forward (LOCF), and k-nearest neighbor interpolation imputation techniques, on the model fitting and prediction performance of ARIMA, and LSTM models, under Ignorable missingness (missing data satisfying either MCAR, MAR or both assumptions) at different scenarios of the rate of missingness using data points obtained from a 48-hour ambulatory blood pressure readings from the Maracaibo Aging Study.

CHAPTER II

THEORETICAL BACKGROUND

2.1 Time series

Suppose $y = X\beta + \varepsilon$ is the underlying regression model of some dependent variable y, where X is a vector of predictor variable(s) of y, β and ε are the parameter(s) and associated errors of predicting y respectively. If y_i are observed sequentially over regular time intervals, then the series of observations, $y_1, y_2, \ldots, y_t$ for $t = 1,2,3,4, \ldots$ is called a time series data.

Since y_t values are taken successively, current realizations of the dependent variable will bear similarities to its prior values. This phenomenon is known as serial dependency and the measure of the strength of this dependency is known as autocorrelation. Thus, time series data cannot be assumed to be statistically independent of their previous or future observations (Shumway & Stoffer, 2019; Box et al., 2015). As a result, standard regression inference procedures, which assume independence between units the data units are invalid for the analysis of time series data. Serial dependency within time series data are important for modelling the stochastic behavior of the process. For instance if the previous observations of the dependent variable were high, then by intuition, we can assume that the subsequent values are more likely to be high and vice versa. The analysis of Time series data makes it possible to study and predict the future behavior of a dependent variable (without necessarily having to deal with its predictor variables).

2.2 Patterns in Time series data

Often the components of time series are used interchangeably with the patterns exhibited by the series on a graph. For instance, we use terms like Seasonal, and Trend (increasing or decreasing trend), to describe the overall behavior of a given time series data on a time plot. These patterns may or may not coexist within a given time series data. We shall consider general definitions of these patterns exhibited by time series data. Trend is generally referred to as the direction of the time series, and when a trend shifts from an ascending to a descending direction. When there is a sustained rise or fall in the data, then the data is said to exhibit a trend. Trend may be linear, quadratic, parabolic etc. When seasonal elements like the time of year or the day of the week have an impact on a time series, a seasonal pattern develops in the series over time. Seasonal variations or patterns recurs at a fixed and known periods within the time frame of the data. When there are increases and dips in the data that are not periodic (e.g., does not recur on regular time intervals), then the pattern exhibited by the time series is said to cyclic. Since the time between cycles are irregular, it is usually difficult to estimate the cyclical variation (Hyndman & Athanasopoulos, 2021).

2.3 Decomposition of Time series data

Usually the trend and cycles within time series data are combined into a single trend-cycle component called the trend for simplicity when the data is decomposed into its constituent parts. We can conceptualize a time series as having three parts: a trend-cycle part, a seasonal part, and the remainder. The remainder is anything else other than the trend, and seasonal component of the time series data.

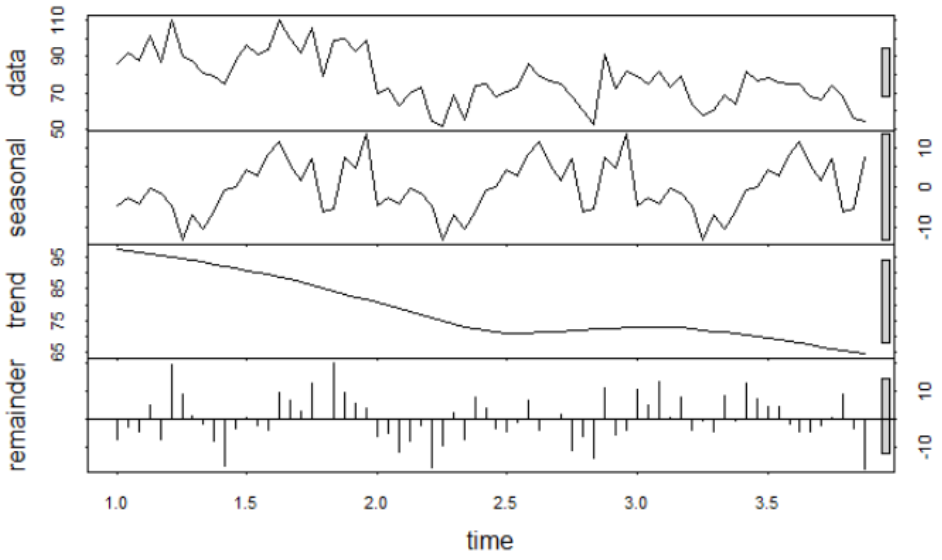

Figure 1. An illustration of Seasonal-Trend decomposition of time series data using Loess.

The graph in Figure 1 shows the additive decomposition of the Diastolic blood pressure data into its components (trend, seasonal and remainder). This decomposition plot is known as the Seasonal and Trend decomposition using Loess (or simply, STL decomposition). The first section represents the time series data, the second section represents seasonal component (i.e patterns that recur at regular time points), and the third section shows the overall direction of the series, and the last section of the plot is the remainder. If we assume an additive decomposition of the time series data, then we can write the series as a linear equation of the form;

$$y_t = \mu_t + \gamma_t + \varepsilon_t \qquad (1)$$

Where y_t is the time series data, γ_t is the seasonal component, μ_t is the trend-cycle component of the series, and ε_t is the remainder or irregular component at period t. Similarly, if we assume a multiplicative decomposition, we write

$$y_t = \mu_t \times \gamma_t \times \varepsilon_t \qquad (2)$$

When the degree of variation around the trend-cycle does not change with the level of the time series, the additive decomposition is a good fit for the series. The multiplicative decomposition, on the other hand, is more appropriate when the seasonal variation within the trend-cycle appears to be commensurate to the level of the time series. (Hynman & Athanasopoulos, 2021).

2.4 ARIMA models

These models were created by George Box and Gwilym Jenkins to mathematically describe the serial dependencies in time series data (Box & Jenkins, 1970). Since their work was published in the 1970s, the theoretical properties of the Box-Jenkins procedure for modeling time series data has made it one of the most used approaches in time series literature. The model consists of three parts; the Autoregressive (AR(p)), a differencing index (I), and the moving average (MA(q)) part. The resulting model is a compound of these procedures called ARIMA (p, d, q), the p denotes the order of the autoregressive part, d is the number of differencing steps needed to make the series stationary and, q is the order of the moving average part of the model (Box et al., 2015). The ARIMA methodology assumes that the series is stationary and the error terms are uncorrelated (homoscedastic).

2.4.1 Autoregressive models

Suppose... y_{t-1}, y_t, y_{t+1}, ... are the observations from a given process (say, the number of diastolic blood pressure readings for an individual, live births in a hospital, annual Medicare costs over a number of years, etc.) recorded at equally spaced times ... $t-1, t, t+1$, ... (e.g. yesterday, today, tomorrow).

Let a_{t-1}, a_t, a_{t+1}, ... represent a series of 'white noise' that are independent and identically distributed random variables. These white noises are approximately normally distributed with a mean of zero and variance $\boldsymbol{\sigma}_a^2$. Suppose further that $E(\mathrm{y_t}) = 0$ (otherwise the y_t may be considered as deviations from their mean). Where $\mathrm{y_t}$ is the current observation from the series and it is linearly dependent on its prior realizations (i.e., y_{i-1}) and on the white noises (also known as random shocks) a_i of the series. Let

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-\mathrm{p}} + a_t \qquad (3)$$

Where ϕ_i in (3) is a parameter of the series, and since y_t is regressed on its previous values y_{t-1}, thus, the resulting model becomes an autoregressive model with an order of p, abbreviated to AR (p) model. The model $\mathrm{y_t}$ in (3) can be expressed as a linear combination of the current and past random shocks of the series such that:

$$\mathrm{y_t} = a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \qquad (4)$$

Then (4) above is called a moving average model of order q(MA(q) model).

2.4.2 Mixed models (ARMA (p, q))

It is occasionally helpful to incorporate the autoregressive and moving average elements into one model in order to achieve greater flexibility when fitting actual time series. Thus, by combining equation (3) and equation (4) above, an autoregressive moving average model of order p and q (ARMA (p, q) model) is obtained:

$$\mathrm{y_t} = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \quad (5)$$

2.4.3 Backward shift operator

The above models expressed in Equations (3), (4) and (5) are can be written in a more concise manner using the backward shift operator B such that $By_t = y_{t-1}$, or $B^k y_t = y_{t-k}$.

Consider a special case of the AR(p) model where $\mathrm{p} = 1$. Then the model in (3) becomes:

$$\mathrm{y_t} = \phi y_{t-1} + a_t, \quad (6)$$

Called the AR(1) model: Then by using the backward shift operation, equation (6) can be written concisely as:

$$\begin{aligned} y_t &= \phi y_{t-1} + a_t \\ y_t - \phi B y_t &= a_t \\ (1 - \phi B) y_t &= a_t \end{aligned} \quad (7)$$

Similarly, the AR(p) model becomes:

$$\left(1 - \phi_1 B - \cdots - \phi_p B^p\right) y_t = a_t \quad (8)$$

and the MA(q) model:

$$y_t = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right) a_t. \tag{9}$$

The AR(p) and the MA(q) can be combined to form a ARMA(p, q) model:

$$\begin{aligned} \left(1 - \phi_1 B - \cdots - \phi_p B^p\right) y_t &= \left(1 - \theta_1 B - \cdots - \theta_q B^q\right) a_t \text{ or} \\ \phi(B) y_t &= \theta(B) a_t, \text{ where} \\ \phi(B) &= 1 - \phi_1 B - \cdots - \phi_p B^p \text{ and } \theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q. \end{aligned} \tag{10}$$

Then it follows that $\phi(B)$ is the operator of the autoregressive component, and $\theta(B)$ is operator of the moving average component. To obtain additional insight into the structure of the models, the AR(1) model mentioned earlier in (6), it can be expressed as:

$$\begin{aligned} (1 - \phi B) y_t &= a_t \\ y_t &= (1 - \phi B)^{-1} a_t \text{ or } y_t = (1 + \phi B + \phi^2 B^2 + \cdots) a_t \text{ or} \\ y_t &= a_t + \phi a_{t-1} + \phi^2 a_{t-2} + \cdots \end{aligned} \tag{11}$$

As a result, the current observation, y_t, is given by the sum of random shocks that have been(exponentially) weighted. The relationship also demonstrates that an $MA(\infty)$ model can be represented as an AR(1) model. This dichotomy between AR and MA models is constant. In particular, the white noise series is used to construct both the AR(1) and MA(1) models, although they have very different approaches to absorbing the random shocks. This distinction is reflected in the models' various predicting capabilities and diverse dependent structures. It can further be shown that the ARMA(p, q) model can expressed as a weighted sum of random shocks:

$$y_t = a_t + \Psi_1 a_{t-1} + \Psi_2 a_{t-2} + \cdots \quad (12)$$

Finding an appropriate representation of the series that has the fewest number of parameters possible is a key component of the model-building process (the principle of parsimony). The ARMA(p, q) model can usually be found using $p \leq 2$ and $q \leq 2$ in a suitable manner.

2.4.4 Stationarity

Time series are said to be stationary if their probability structure does not vary over time. As a result, a stationary time series has a constant mean and variance as well as a covariance structure that only depends on the difference between two time points. The stationary nature of an ARMA(p, q) process may be demonstrated if the roots of the polynomial $\phi(B)$ fall outside the unit circle. However, in practice many time series data are not inherently stationary. Such series can usually be made stationary by taking the first difference of the series to achieve stationarity. .

$$w_t = y_t - y_{t-1} = \nabla y_t \quad (13)$$

The symbol $\nabla = 1 - B$ is known as the ordinary differencing operator. Whenever the series has to be differenced (say once) to achieve stationarity, then the corresponding model obtained on the series becomes an integrated autoregressive moving average model or ARMA model of order $p, 1, q$ or an ARIMA $(p, 1, q)$ model. Suppose the series was differenced d times to achieve stationarity, then the model obtained becomes ARIMA(p, d, q).

2.5 Seasonal ARIMA models

Box and Jenkins expanded the aforementioned ideas to accommodate seasonal time series. It takes two steps to get the model. Take monthly statistics, for instance.

a) A single month's observation is correlated with a previous observation made 12 months earlier by

$$\begin{aligned} y_t - \Phi_1 y_{t-12} - \Phi_2 y_{t-24} - \cdots - \Phi_P y_{t-12P} &= \alpha_t - \Theta_1 \alpha_{t-12} - \Theta_2 \alpha_{t-24} - \cdots - \Theta_Q \alpha_{t-12} \text{ or} \\ \Phi(B^{12}) y_t &= \Theta(B^{12}) \alpha_t, \text{ where} \\ \Phi(B^{12}) &= (1 - \Phi_1 B^{12} - \Phi_2 B^{24} - \cdots - \Phi_P B^{12Q}) \\ \Theta(B^{12}) &= \left(1 - \Theta_1 B^{12} - \Theta_2 B^{24} - \cdots - \Theta_Q B^{12Q}\right) \end{aligned} \quad (14)$$

b) The error component α_t for a particular month is related to that for the previous month by the usual ARMA model.

$$\phi(B)\alpha_t = \theta(B)a_t \quad (15)$$

Joining the seasonal and the non-seasonal parts gives:

$$\phi(B)\Phi(B^{12})y_t = \theta(B)\Theta(B^{12})a_t \quad (16)$$

By creating seasonal differences, the idea of ordinary differencing is expanded to include seasonal differencing.

$$w_t = y_t - y_{t-s} = \nabla_s y_t, \quad (17)$$

Where $\nabla_s = 1 - B^s$ is the seasonal differencing operator and $s = 12$ for monthly data, one obtains the seasonal ARIMA model:

$$\phi(B)\Phi(B^s)\nabla^d\nabla_s^D y_t = \theta(B)\Theta(B^s)a_t \quad (18)$$

Abbreviated to the ARIMA$(p,d,q)(P,D,Q)_s$ model.

2.6 Forecasting

To obtain forecasts $\hat{y}_{t+h}$ for h time units (days, months, etc.) ahead from an ARIMA model one writes the corresponding model equation by.

1. Replacing future values of the random shocks a by zero and past values by observed residuals.
2. Future values of y_t by the corresponding forecasts; and
3. Past values of y_t by their observed values.

The following example illustrates how to obtain forecasts for a $AR(1)$ model:

$$\begin{aligned} h=1:\ & y_{t+1} = \phi y_t + a_{t+1} \\ & \hat{y}_{t+1} = \phi y_t \\ h=2:\ & y_{t+2} = \phi y_{t+1} + a_{t+2} \end{aligned} \quad (19)$$

$\hat{y}_{t+2} = \phi\hat{y}_{t+1} = \phi(\phi y_t) = \phi^2 y_t$, etc.

By continuing this procedure, one may see that the forecasts corresponding to the $AR(1)$ model follow an exponential curve.

The autocorrelation function (ACF), the dependence structure of a stationary time series is characterized by the autocorrelation function (ACF). The ACF is defined as the correlation between y_t and y_{t+k} :

$$\rho_k = \text{cor}(y_t, y_{t+k}).$$

k is called the time lag. The ACF is estimated by the empirical ACF:

$r_k = \frac{c_k}{c_0}, k = 0,1,2,$ ... Where

$$c_k = \frac{1}{n}\sum_{t=1}^{n-k} (y_t - \bar{y})(y_{t+k} - \bar{y}) \text{ and } \bar{y} = \frac{1}{n}\sum_{t=1}^{n} y_t. \quad (20)$$

c_k are the empirical autocovariance.

2.7 Major steps in ARIMA modelling:

Box and Jenkins recommend the following steps in order to get a good ARIMA model:

Step 1: Plot the time series data for visual impression of the series. This includes examining the behavior of the ACF and PACF of the series. This is an important step to identify any nonstationary behavior such as trend or seasonal patterns within the data. Since ARIMA methodology assumes that the series in question is stationary, statistical tests like the Augmented Dickey-Fuller Test and KPSS Test for level stationarity can be used to ascertain whether the series is stationary or not.

Step 2: If the results from these tests show that the series is not stationary, then one may have to 'Make the series stationary', by differencing the data and then repeat Step 1 again until the test results show that the series has become stationary.

Step 3: By examining the ACF and PACF of the differenced data, one may be able to choose a provisional model for the data, by looking at the behavior of the empirical ACF and PACF (Box et al, 2015; Helfenstein, 1996). For instance, suppose the data achieved stationarity after taking the first difference. Then one may choose an Autoregressive process of order 1, if the ACF either decays exponentially or in a damped sinusoidal manner. The corresponding PACF may have nonsignificant spikes after lag 1. In Such scenario, the tentative model would be ARIMA(1,1,0). Similarly, one may choose a Moving average process of order 1, if the ACF of the series cuts of after lag 1 and the PACF tails off after that lag. In such scenario, the order of the tentative model becomes ARIMA(0,1,1).

Step 4: After identifying a tentative model, the next step is to estimate the model parameters (this can easily be achieved by using standard statistical software such as R, SAS, and Python among others allow maximum likelihood estimation of Box-Jenkins models although these packages may not give the same regression coefficients, but they will be similar) and then fit obtained model.

Step 5: Before the model can be used for forecasting, one must first check the adequacy of the fitted model by performing a residual analysis of the model. There should be no significant autocorrelation between the residuals.

One returns to step 4 and selects a better model if the model does not adequately suit the data. One selects the model with the fewest parameters out of those that represent the data equally well (this can be achieved with the statistical software packages named above). Special cases of ARIMA models some of which has been mentioned earlier are shown in Table 1 below.

Table 1. Special cases of ARIMA (p,d,q) models

Process	Order of model
Autoregression	ARIMA(p,0,0)
Moving average	ARIMA(0,0,q)
Random walk	ARIMA(0,1,0)*
Random walk with drift	ARIMA(0,1,0)*
White noise	ARIMA(0,0,0)*

*Has no coefficients or constants

2.8 LSTM neural network model

Artificial neural networks or neural networks are artificial mimicries of the biological network of neurons in animal brains. Recurrent neural networks are a class of neural networks where the connections between the nodes are cyclic which allows outputs from the nodes within

the network to affect later inputs that are fed into the same nodes. Put simply, this class of neural networks can store features of previous inputs within its internal states in the form of activations (short-term memories) which are later used to update the current and subsequent state of inputs (Elman, 1990; Jordan. 1986; Chen & Soo, 1996) This feature makes RNNs potentially useful and have had some success solving certain problems (Mozer, 1994; Karpathy, Johnson and Fei-Fei, 2015, Li, Li, Cook, Zhu, and Gao, 2018). However, short-term memory becomes a drawback in solving tasks with long sequential inputs, particularly when the time series cannot be divided into a smaller subsequence. RNNs can take a discouraging amount of computing time or might not even be able to learn sequential inputs with long time lags (Hochreiter and Schmidhuber, 1997).

In response to this drawback, Hochreiter and Schmidhuber (1997) proposed the long short-term memory (LSTM) algorithm which is an evolution of RNNs has longer short-term memory for processing sequential data with long-term dependencies. However, the "New" algorithm was unable to forget redundant states of previous values which can become a problem, especially when learning a long continuous stream of sequential inputs. This phenomenon can be compared to a computing system that runs into an indefinite loop. The system will execute the line of commands repeatedly until the system runs out of memory and crash. In the same vein, since the algorithm is unable to forget previous states of inputs, the internal states of the standard LSTM cell will continue to grow out of bound which will lead to the eventual breakdown of the network. (Gers, Schmidhuber, & Cummins 2000) proposed the "forget gate" function within the cell architecture of the standard LSTM network. This function enables the network to reset itself (forget some of the things learned) at appropriate set times without loss of vital information. As a result, there are

variants of LSTM cells, such as LSTM with and without forget gates as well as LSTM with a peephole connection (Gers and Schmidhuber, 2000). In most literature, the phrase "LSTM cell" typically refers to an LSTM that has a forget gate (Yu, Si, Hu & Zhang, 2019). It is worth noting that all LSTM cells are characterized by “gate” functions.

2.8.1 LSTM cell architecture

The standard or initial LSTM cell as mentioned earlier has no forget gate function. Thus, the cell is characterized by only two gate functions namely the input gate and output gate as shown in Figure 2. The input gate determines what raw data should be added to the cell state at a time, and the output gate determines what data should be the output based on the cell state. Every time the cell states are updated, this process is repeated.

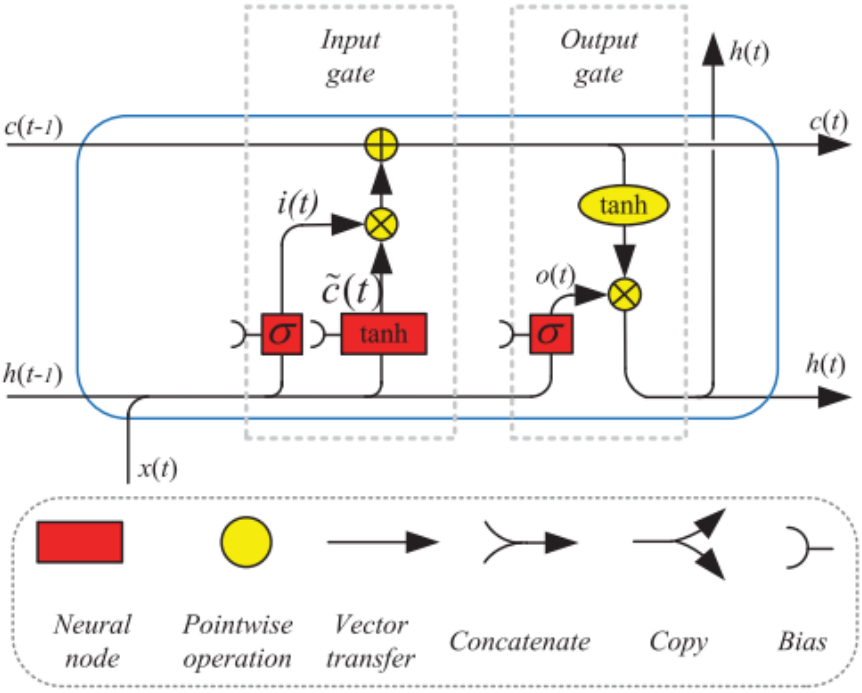

Figure 2: Architectural illustration of the standard LSTM memory cell from (Yu et al., 2019)

The connections within the standard LSTM cell shown in Figure above, each connection is mathematically expressed as:

$$\begin{aligned} i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i), \\ \tilde{c}_t &= \tanh(W_{\tilde{c}h}h_{t-1} + W_{\tilde{c}x}x_t + b_{\tilde{\tau}}), \\ c_t &= c_{t-1} + i_t * \tilde{c}_t, \\ o_t &= \sigma\left(W_{\text{oh}}\, h_{t-1} + W_{ox}x_t + b_o\right), \\ h_t &= o_t * \tanh(c_t), \end{aligned} \tag{21}$$

In equations (21), x_t, and $\mathrm{h_t}$ represent the input, and recurrent information at time t respectively. Whereas b_i denotes the bias, c_t represents the cell state. $W_i, W_{\tilde{c}}$, and W_o are the weights, and the operator ' $*$ ' represents the pointwise multiplication of two vectors. The modified

LSTM is characterized by three gates namely, the input gate, output gate and the forget gate respectively. With the same functions, the input and output gates control the inflow and outflow of data within the LSTM memory cell.

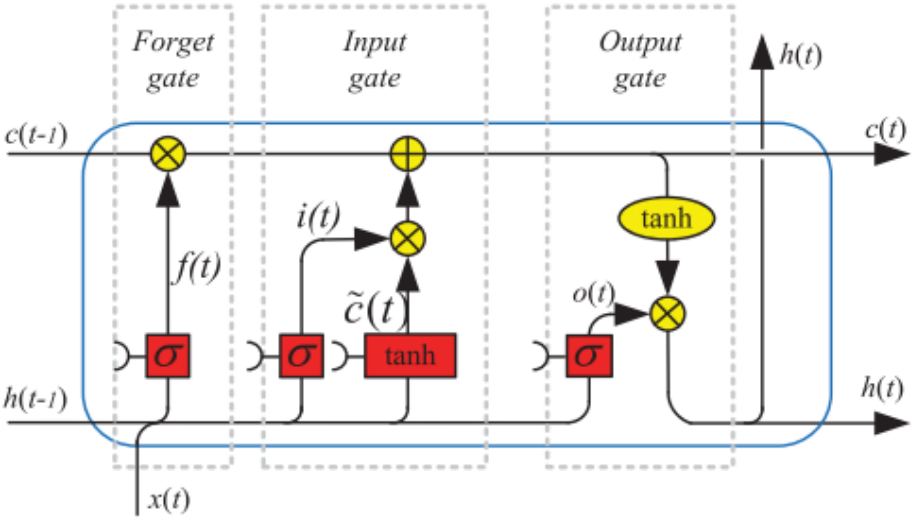

Figure 3. Architectural illustration of LSTM memory cell with forget gate function from (Yu et al., 2019)

The forget gate denoted f_t in equations (22) determines which data to be erased from the cell state. This can be mathematically expressed as,

$$
\begin{aligned}
f_t &= \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + b_f\right) \\
i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \\
\tilde{c}_t &= \tanh(W_{\tau h}h_{t-1} + W_{\tau x}x_t + b_{\tilde{\tau}}) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\
o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \\
h_t &= o_t * \tanh(c_t)
\end{aligned}
\tag{22}
$$

The forget gate is a binary function. Whenever the value of the function is 1, the information within the cell at the time is maintained, whereas a function value of 0 implies deletion of all the information at time t (Yu et al., 2019).

2.9 Mechanisms of missing data

The mechanism of missingness can be represented as a probability function, which depicts the relation between the measured variables and the likelihood of a missing data. The notations used in this section may differ slightly from the one used by Little and Rubin (2019).Suppose $X = (x_1, x_2, \ldots, x_n)^T$ is a vector random variable of the complete rectangular dataset which contains both missing and observed ($X_{missing}$ and $X_{obsrved}$ respectively), with a density function of $f(\varphi)$. The goal is to make inferences about the parameter φ which is unknown by using the given vector. Let $M = (m_1, m_2, \ldots, m_n)^T$ be the missing data indicator variable with entries that are either zero or one depending on whether the accompanying elements within the variable X are missing or not. (That is, $m_i = 1$ whenever the value x_i is missing and $m_i = 0$ whenever the value is observed). The pattern of missingness within the dataset is determined by the missing data indicator as explained above.

In Rubin (1976) each observation is considered as a two-valued vector. The value of the observation is indicated as either $X_{observed}$ or $X_{missing}$ and the corresponding M value (missing value indication code). By treating missing data as some variable, it is enough to suggest that the values assumed by the missing data indicator is governed by an underlying probability distribution. In this case, one should not expect **M**'s distribution to be unrelated to X, although it is difficult to know the exact probability distribution of the missing data indicator. The nature of association between the indicator M and the dataset, on the other hand, is what separates the types of missing data mechanisms, which are described by the conditional distribution of M given a complete data X. Such that, $f(M|X,\varphi)$ where φ, is the vector of unknown parameters which represents the likelihood of missing data (Rubin, 1976; Schafer 1997).

2.9.1 Missing completely at random (MCAR)

When the likelihood of a missing data remains unchanged in all circumstances, the data is considered to be completely missing at random (MCAR). In this case, the probability of a missing observation is unaffected by the values of other variables as well as the value of the observation itself. Suppose X and Y are the columns of a complete bivariate dataset. Then the probability of data in X being missing is not dependent on either Y or X itself. This effectively suggests that the reasons for missing data is not dependent on either the missing or observed values in that variable or any other variables within the dataset. For instance, blood pressure measurements of a patient may be missing because of breakdown of the automatic sphygmomanometer. An unselected unit in random sampling is also another example of MCAR; because every unit in the population could

have been sampled not because of their demographic features. Thus, those units that were not chosen were not selected by pure chance (van Buuren, 2018).

$$P\left(M\middle|X_{observed}, X_{missing}, \varphi\right) = P(M|\varphi) \qquad for\ all\ X, \varphi \tag{23}$$

2.9.2 Missing at random (MAR)

In MAR, the recurring difference between the missing observations and the observed can be explained by the available information. Put differently, the missing value within variable X depends on other variable(s) within the dataset, such that the pattern in which the data becomes missing is traceable. In the context of probability, this can be thought of as the probability of the data X being missing is dependent on the available values of Y and not on the missing observations. This assumption infers that the distribution of missing data is not entirely random, but it can be accounted for by other observed variables. Unlike the MCAR, this assumption is less restrictive.

$$P\left(M\middle|X_{observed}, X_{missing}, \varphi\right) = P(M|X_{observed}, \varphi) \tag{24}$$

2.9.3Missing not at random (MNAR)

In this assumption, the missing observations are dependent on the missing values themselves, such that the available data cannot be used to estimate the missing value(s). Thus, in MNAR the data are missing for reasons linked to the values themselves and cannot be accounted for by the observed values. In other terms, the likelihood of missing data on a variable X is not related to the observed values of X and as such cannot be used to estimate the missing ones.

$$P\left(M\middle|X_{observed}, X_{missing}, \varphi\right) = P(M|X, \varphi) = P\left(M|X_{observed}, X_{missing}, \varphi\right) \tag{25}$$

(Little & Rubin 2002, 2019; Hamzah et al., 2020). For instance, people with normal blood pressure are likely to monitor their blood pressure less often than those with known record of high blood pressure. Thus, in such scenarios, the blood pressure readings would be missing not at random (MNAR) because the reason for missing is dependent on itself rather than the time or the other observed readings within the data.

2.9.4 Univariate equivalent of mechanisms of missingness

Univariate time series data x_i, where $i = 1,2,3, \dots, N$, like any other univariate sample are the simplest of data structures. Thus, in univariate data with missingness, x_i and the missing value indicator m_i are both scalar variables. Then the density function described earlier becomes:

$$p(X = x, M = m|\theta, \varphi) = \prod_{i=1}^{n} f_X(x_i|\theta) \prod_{i=1}^{n} f_{M|X}(m_i|x_i, \varphi) \quad (26)$$

Where $f_Y(x_i|\theta)$ represents the density function of x_i indexed by the unknown vector of parameters θ, and $f_{M|Y}(m_i|x_i, \varphi)$ denotes the density of a Bernouli distribution for the binary missing data indicator m_i, with the probability that an observation x_i is missing as $P(m_i = 1|x_i, \varphi)$. If the mechanism underlying the missing data is independent of X, that is $P(m_i = 1 \mid x_i, \varphi) = \varphi$, which is a constant that does not depend on x_i, then the missingness mechanism is MCAR (or equivalently MAR). Alternatively, if the mechanism responsible for the missing data depends on y_i, then the mechanism is MNAR because it depends on values of y_i, some of which are missing (Little & Rubin, 2019).

2.9.5. Ignorable mechanism

With the exact meaning of m_i and φ as missing data indicator and the unknown parameter which describes the probability of missing data. Suppose $x_{ij} \in \Omega_{ij}$, with Ω_{ij} being its sample space, $m_{ij} = 1$ if x_{ij} is missing and $m_{ij} = 0$ if y_{ij} is observed. When $m_{ij} = 1$, $x_{ij} = \text{NA}$, indicating that x_{ij} can take any value in Ω_{ij}. Then by using the "selection model" factorization (Little and Rubin 2002), the joint distribution of X and M can be model as

$$p(X = x, M = m \mid \theta, \psi) = f_x(x \mid \theta) f_{M|X}(m \mid x, \psi) \tag{27}$$

Where θ denote the parameter governing the data model, and ψ is the parameter vector governing the model for the mechanism of missingness. Then the observed value m of M effects a partition of $x = (x_{(0)}, x_{(1)})$, where $x_{(0)} = [x_{ij}: m_{ij} = 0]$ is the observed part of x and $x_{(1)} = [x_{ij}: m_{ij} = 1]$ is the missing part of x.The full likelihood based on the observed values $(x_{(0)}, m)$ and the assumed model (9) is defined to be

$$L_{\text{full}}(\theta, \psi \mid x_{(0)}, m) = \int f_X(x_{(0)}, x_{(1)} \mid \theta) f_{M|X}(m \mid x_{(0)}, x_{(1)}, \psi) \mathrm{d}x_{(1)} \tag{28}$$

Considered as a function of the parameters(θ, φ). The likelihood of θ ignoring the missingness mechanism is defined to be

$$L_{\text{ign}}(\theta \mid x_{(0)}) = \int f_X(x_{(0)}, x_{(1)} \mid \theta) \mathrm{d}x_{(1)} \tag{29}$$

Since (11) does not involve the model for M, the phrase "ignorable likelihood" is sometimes used for it. Modeling the joint distribution of M and X is often challenging and as such many approaches

to missing data do not model M, and (implicitly or explicitly) base inference about θ on the ignorable likelihood in (11). It is thus important to consider under what conditions inferences about θ can be based on this simpler likelihood (ignorable). Thus, (Little and Rubin, 2019) surmises that the missingness mechanism of the data is ignorable if the following two conditions are met:

1. The missing data are MAR at $(\tilde{m}, x_{(0)})$,
2. The parameters θ and ψ are distinct.

CHAPTER III

METHODS

3.1 Dataset description

We obtained a de-identified univariate time series data of seventy equally spaced observations from a 48-hour ambulatory blood pressure data from the Maracaibo Aging Study. The Maracaibo Aging Study is a prospective, population-based cohort study of individuals ≥55 years of age residing in Maracaibo, Santa Lucia County, Zulia, Venezuela. (Maestre et al., 2002). Detailed methodology of the study is described elsewhere (Maestre et al.,2002). Validated (Gropelli et al., 1992) oscillometric 90207 Spacelabs monitors (Snoqualmie, WA) were programmed to obtain blood pressure readings at 15-minute intervals from 6 AM until 11 PM and at 30-minute intervals from 11 PM until 6 AM. Specifically, the diastolic blood pressure readings were used for this study, the first 56(80%) observations of the data were used as the training data, and the remaining 14(20%) were used as test data for evaluating the prediction performance of the models. A reliable estimate of the autocorrelation function of time series data can be obtained with a minimum of at least 50 equally spaced data points (Box et al., 2015). Therefore the 56 observations used in the training dataset were sufficient to be used for the Box-Jenkins methodology.

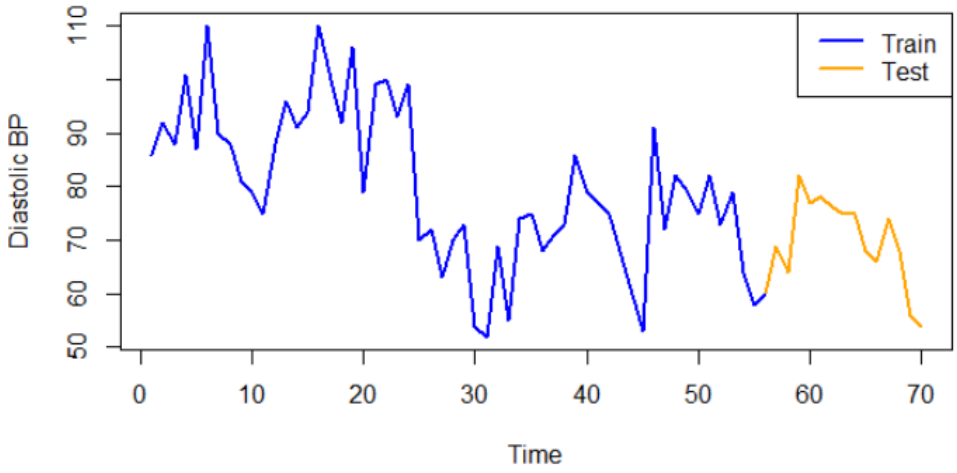

Figure 4: Time series plot of the diastolic blood pressure data depicting the train and test sections. The section colored blue represents the train and orange for the test dataset.

3.1.1 Simulated missing data

The literature showed that in one column, or univariate data, the MAR and MCAR mechanism of missingness are similar in several ways; first, both assume that the missingness of the data is unrelated to missing data themselves. Secondly, the missingness can be treated as a random process, meaning that the probability of missingness is the same or remains constant for all the data points (Little and Rubin, 2019). This further implies that, the missingness cannot be tied to a single cause and in such cases, the reasons behind the missing data can be ignored. For these reasons, we generated MCAR missing data using a Binomial distribution. First, we made four copies of the original training dataset. Subsequently, missing data at different rates (10%, 15%, 25% and 35%) were generated within each of the four copies.

Each of these incomplete datasets were then imputed using the unconditional mean, last observation carried forward (LOCF), linear and stine interpolation (Stineman, 1980), interpolation with Kalman smoothing on ARIMA, interpolation with Kalman smoothing on structural time series technique, k-NN interpolation, simple moving average (SMA), exponentially weighted moving average (EWMA) and cubic spline interpolation techniques. The imputed datasets from the ten imputation algorithms for each rate of missingness, and the original dataset makes up a total of (4*10) +1=41 different datasets. The steps are summarized in the figure below.

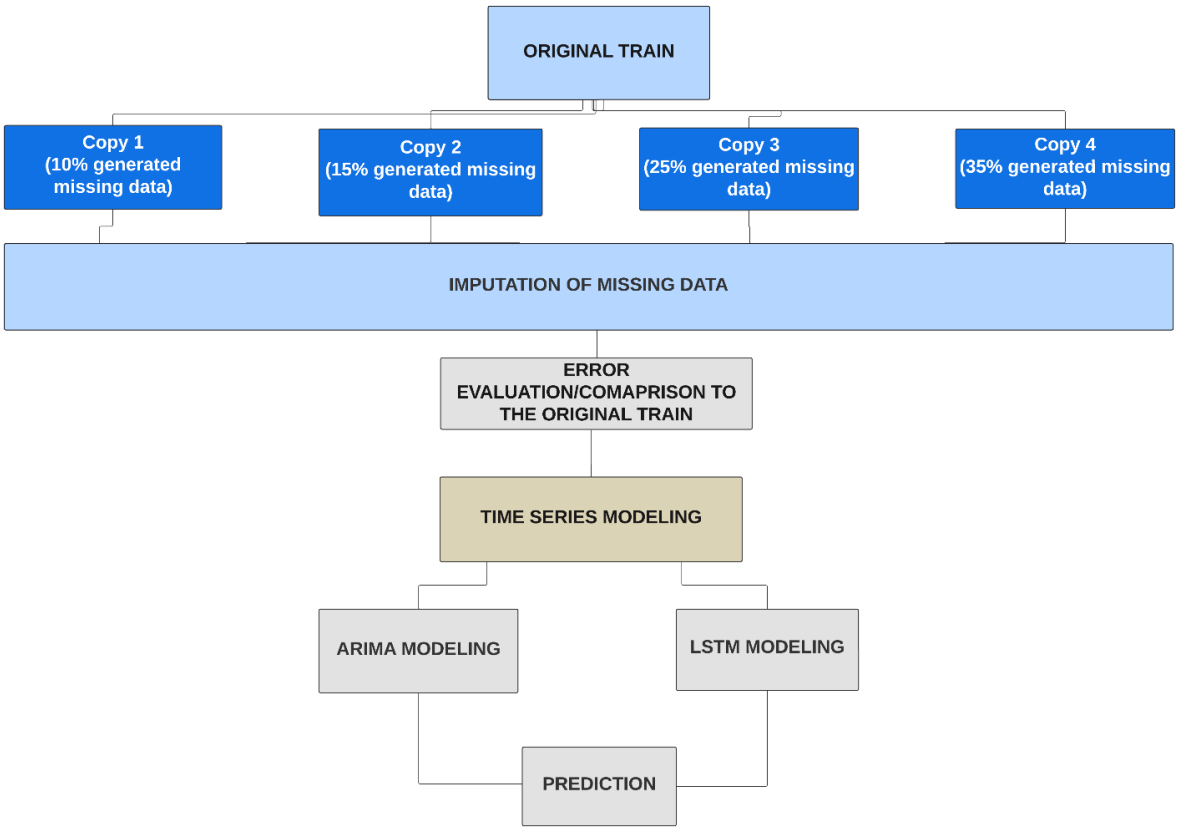

Figure 5: Flowchart of the methodology used in the study

3.2 Imputation of missing data

Imputation techniques for univariate time series cannot solely depend on covariates because univariate time series data is one-column data, although time is given implicitly (Moritz et al, 2015; Moritz & Bartz-Beielstein, 2017). Thus, most of the high-performance techniques such as multivariate or multiple imputations by chain equations (MICE) (Rubin, 2004; van Buuren, Boshuizen & Knook, 1999; van Buuren & Groothuis-Oudshoorn, 2011), Expectation maximization (Dempster, Laird and Rubin, 1977) and Nearest neighbor (Vacek and Ashikaga, 1980; Batista & Monard, 2002) techniques cannot be used directly on univariate time series data. Moritz et al (2015) categorize univariate time series imputation techniques into; univariate imputation (replacing missing values with measures of central tendency such as mean, median, etc.), univariate time series imputation (techniques that can harness the properties of the series such as trend or seasonal cycles to impute missing ones) and multivariate imputation techniques. For the imputation of the missing values, we used the imputeTS (Moritz and Bartz-Beielstein, 2017) package in R which provides a collection of algorithms and tools for univariate time series imputation. The imputation algorithms include: 'Mean', 'LOCF', 'Interpolation', 'Moving Average', 'Seasonal Decomposition', 'Kalman Smoothing on Structural Time Series models', 'Kalman Smoothing on ARIMA models'. Also, the KNN interpolation (See Kulesh, Holscneider and Kurennaya, 2008; Lepot, Aubin and Clemens, 2017) was performed using the DMwR2 package published by (Torgo, 2016). All the imputation methods were done using these packages in R. After the imputation in each dataset, the distribution and autocorrelations within the imputed datasets were compared to the original training using boxplots and the autocorrelation plots.

3.3 Evaluation metrics

We evaluated the performance of the imputation techniques by using the root mean squared error (RMSE) and the mean absolute percentage error (MAPE) metrics. The RMSE was used to compare the difference between the imputed datasets and the actual values, considering the magnitude and direction of the errors. The smaller the RMSE value, the better the imputation algorithm. Thus, the imputed dataset from a given imputation technique with a smaller RMSE indicates better performance. A larger RMSE implies larger prediction errors, which suggests lower confidence in the algorithm's performance in predicting or estimating the missing value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{imp\ (i)} - y_{obs(i)}\right)^2}$$

Similarly, the MAPE was used to compute the average absolute percent difference between the imputed and the original training dataset. In these two metrics, the smaller the value of the error metric, the better the performance of the imputation algorithm.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_{obs(i)} - y_{imp(i)}}{y_{obs(i)}}\right| \times 100\%$$

These two metrics, RMSE and MAPE were also used to evaluate the prediction performance of the models as well.

3.4 Forecasting algorithms

3.4.1 ARIMA model

The ARIMA model has three parameters, $ARIMA(p, d, q)$ where p is the order of the autoregressive component of the model, d represents the number of differencing required to make the data stationary, and q is the order of the moving average component of the model. Due to the number of datasets (41 in total), obtaining the ARIMA models using the traditional way as outlined in the previous section for each of the datasets will take a lot of time. Thus, we used the automatic time series forecasting algorithm proposed by (Hyndman and Khandakar, 2008). The algorithm uses an automated algorithm to select the best ARIMA model for a given time series data. In the R statistical software, the automated time series model is obtained using the **auto.arima()** function. The function performs a grid search over a range of possible ARIMA models and then selects the best model based on a specified criterion. By default, the function uses the Akaike Information Criterion (AIC) to select the best model, but other criteria such as the Bayesian Information Criterion (BIC) or the Hannan-Quinn Information Criterion (HQIC) can also be used.

The algorithm used by **auto.arima()** can be summarized as follows:

1. The function starts by fitting a simple ARIMA model to the data and calculating the AIC.
2. It then tries to improve the model by testing different combinations of AR, MA, and differencing parameters to find the model with the lowest AIC.

3. The function repeats this process for a range of possible models and selects the model with the lowest AIC as the best model.
4. Finally, the function returns the best model along with diagnostic information such as residuals and fitted values.

In addition to the time efficiency of the algorithm in selecting the best ARIMA model, **auto.arima()** also includes functionality for dealing with seasonality and non-stationary data. It can automatically detect and incorporate seasonal components into the model, as well as perform differencing to make the data stationary if necessary. Before any obtained model was used for forecasting, we first perform residual analysis to ensure that the assumption of independent and identically distributed errors was satisfied. This was done by plotting the ACF of the residuals to check for discernible patterns, and also a significant correlation between them.

3.4.2 LSTM model

The LSTM is a Recurrent Neural network (RNN) that can learn and memorize a long sequence of inputs. The algorithm employed performs a multi-step univariate time series forecast (Brownlee, 2018). Keras library with Theano are requisites for the execution of the algorithm. For the reproducibility of results, we used different random seed numbers for each run. The first step in executing the algorithm is to prepare the time series data into a format that is LSTM modeling-friendly. This involves converting the univariate data into three-dimensional input data. For the algorithm to learn a one-step prediction, we split the sequence into several input/output patterns, or samples, where three-time steps served as the input and a one-time step as the output. For

instance a univariate time series (10, 20, 30, 40, 50, and 60) becomes an array of x (three time steps) with a one output time step y.

x	y
10, 20, 30	40
20, 30, 40	50
30, 40, 50	60

The three-dimensional array of the input data consists of the; samples, time steps, and features. Sample denotes the number of observations in the training set. The time steps are the number of steps in each observation and the features are the number of features in each observation. Unlike the ARIMA model where the parameters or coefficients of the model were estimated based on the stochastic behavior of the time series, the LSTM uses hyper-parameters. Parameters are estimated from the training data (using known procedures like maximum likelihood estimation etc.,) but the hyper-parameters are "parameters" that are not estimated directly from the data based on analytical formulae but are rather determined by the analyst (Brownlee, 2018; Kuhn and Johnson, 2013). Usually, trial-and-error, random searching, or using examples that already worked in the past are all common ways to find the model hyperparameters of the LSTM model. We selected a range of parameters; the batch size of (1, 2,3,4,5,6,7), hidden layers of (1,2, and 3), number of neurons within a layer (1,10,25, 32, 50, and 64), and Epochs of (100, 150, 300, 400, 500).

The hyperparameters were then fine-tuned to improve the learning ability of the model using the algorithm outlined explained in (Brownlee, 2018). The experimental results from the number of Epochs showed that 400 Epochs was the best (it had the smallest RMSE values on the training data). 2 LSTM layers were stacked on top of each other. This makes enhances the models ability to learn the temporal dependencies within the train data. The “Adam” optimization algorithm was used in the learning process. After training the LSTM model, the predict () function was used to generate predictions for the testing data. Time series plots of the predicted and observed data were created to give a visual impression of the model’s prediction performance using the matplotlib library. Google Colaboratory or, Google Colab was used to run the LSTM algorithm. Google Colab is a Jupyter interface that enables running Python algorithms in the browser without any complex configuration.

3.5 Forecasting Techniques

The “Rolling Forecasting Origin" technique (Hyndman & Athanasopoulos, 2021), was used for the ARIMA and LSTM algorithms. Variations of this forecasting technique include;

1. One-step forecasts without model re-estimation; in this technique, a fixed historical window of data is used to fit the model and then make a forecast. The window is then moved by a specified amount and the process is repeated.
2. Multi-step forecast without re-estimation of the prediction model. Unlike the one-step mentioned above, this makes multiple-step forecasts without updating the model.
3. Multi-step forecasts with re-estimation. In this variant, the time series model is re-estimated after each prediction.

The Multi-step forecast with re-estimation variant of rolling forecasting was used in this study. This technique is also known in the literature as the Walk forward validation. Walk-forward validation typically involves making forecasts for a fixed period (e.g., one day) and updating the model with the latest data before making the next forecast. Specifically, after fitting a model on training data, the model is used to predict (usually the first value in) the test data. The corresponding actual value in the test data is then added to the training set, and the model is rebuilt using the new dataset (initial training plus the added data from the test set). The updated model is then used to predict the next value and the procedure to give a one-step prediction for the entire test dataset. This method of forecasting accounts for the time dependency in the data. It also allows the model to be updated and refined as new data becomes available.

CHAPTER IV

RESULTS

4.1 Imputation of simulated missing data

The time plot of the incomplete data at each rate of simulated missingness and the missing data map of the incomplete data at each rate are shown in Figure 5.

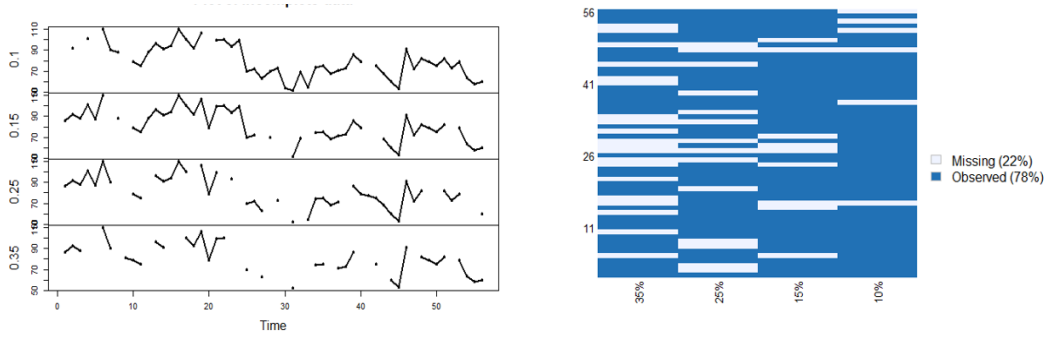

Figure 6.(a) Time series plot of the datasets with generated missing values (left) and (b) Map of the four different rates (10%, 15%, 25%, and 35%) of missingness.

4.1.1 Imputation performance in 10% missingness

The comparative time series plot shows the original train and the imputed data using the ten imputation methods (Figure 6). The descriptive statistics of the datasets, both imputed and original train data, are shown in the boxplots (Figure 7). The unconditional mean (also known as the mean imputation) method outperformed the other missing data replacement techniques. The KNN interpolation technique comes in second to this strategy, with spline interpolation coming in last place as least effective based on the RMSE and SMA based on MAPE values (Table 2). The

probability density plot of the mean imputed dataset at 10% is similar in shape to the original train data (Figure 8).

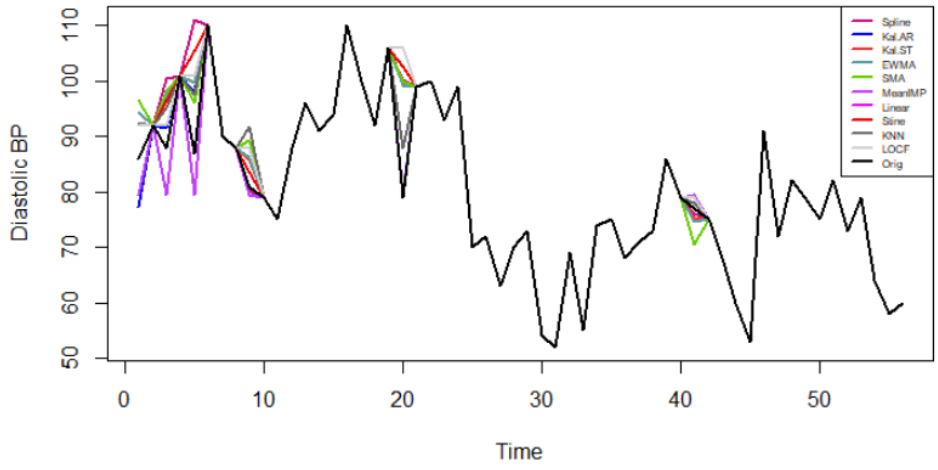

Figure 7. Comparative time series plot of training data and imputed data at 10% missing data rate

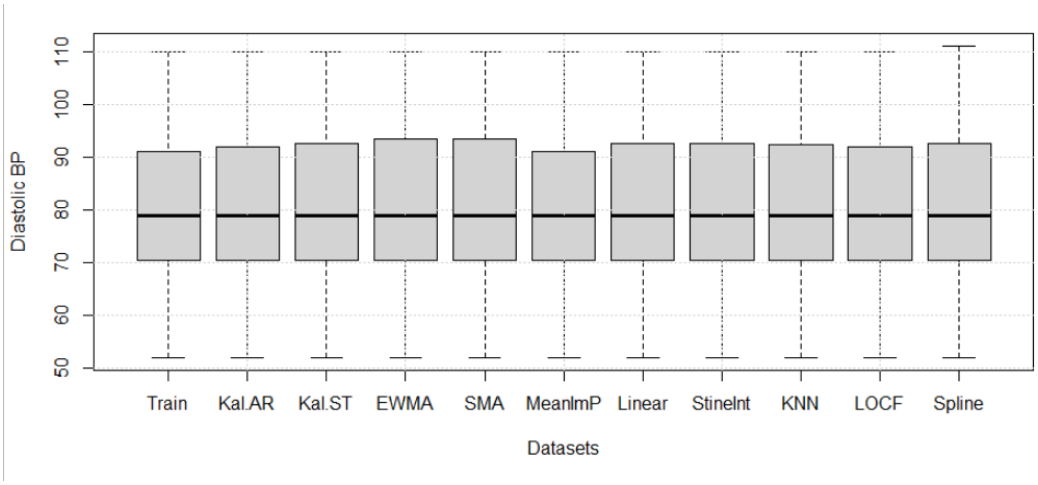

Figure 8. Distribution of training data and imputed data at 10 % missing data rate

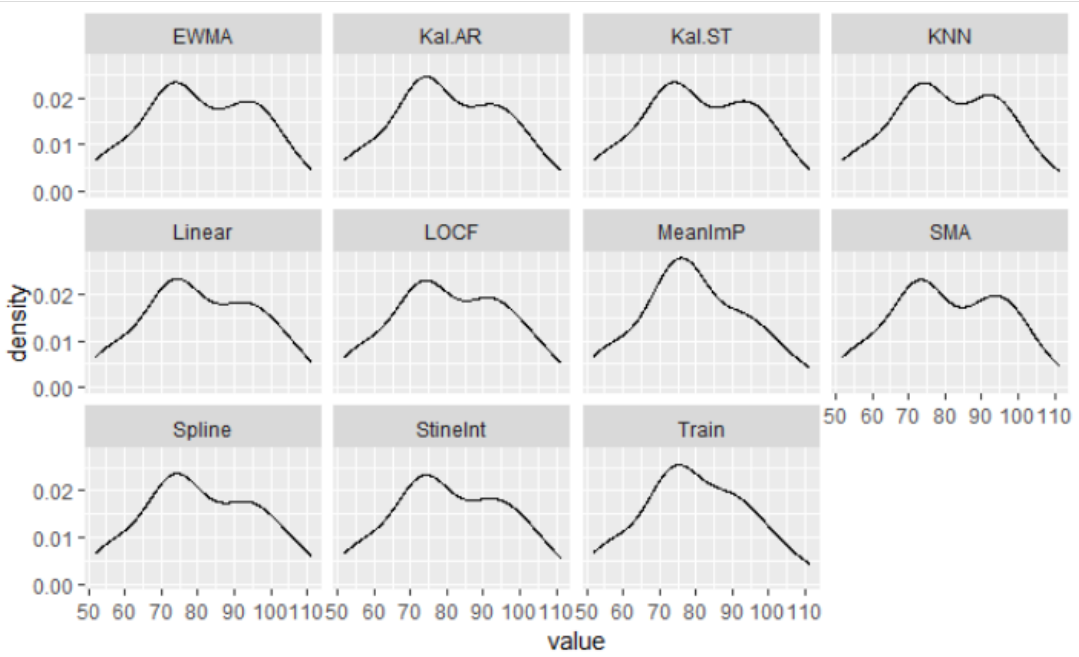

Figure 9: Density plot of training data and imputed data at 10% missing data rate

Table 2. Imputation performance of various techniques at 10% level of generated missing data

Technique	RMSE	MAPE
Kalman AR	3.447427	0.9845257
Kalman ST	3.602193	1.014541
EWMA	3.684186	1.083673
SMA	3.890005	1.254096

Mean Imputation	**1.777848**	**0.6033183**
Linear interpolation	4.244745	1.049757
Stine Interpolation	4.244745	1.049757
KNN	**2.685169**	**0.8665411**
LOCF	4.288689	1.083729
Spline	4.674523	1.143995

4.1.2 Prediction performance in 10% imputed datasets

The summary of the order and coefficient of the ARIMA models in 10% imputed and original training datasets (see Table 3). The missing data replacement methods increased the autocorrelation of the imputed data, except for the mean imputation technique using the original train data as the reference (Figure 9). The prediction performance of the ARIMA in the imputed datasets showed that the model in the spline imputed dataset was the best, and the mean imputed data was the worse. ARIMA models in the imputed datasets performed better than the model in the original train except for the mean imputed data (Table 4). The time plot of the ARIMA prediction on each dataset (Figure 10-12). The LSTM prediction performance on the original train performed better than the ARIMA. The prediction in the spline imputed was the best based on RMSE (Table 5). The prediction performance of the LSTM is in Figure 13, Figure 14, and Figure 15.

Table 3. ARIMA models obtained on 10% level of imputed data

Data	Model	C(SE)	p-value
Train	ARIMA(1,1,0)	-0.4844(0.1161)	0.000000
Kal.AR	ARIMA(1,1,0)	-0.3841(0.1249)	0.002109
Kal.ST	ARIMA(0,1,1)	-0.3934(0.1201)	0.001053
EWMA	ARIMA(0,1,1)	-0.3982(0.1195)	0.0008594
SMA	ARIMA(0,1,1)	-0.4248(0.1164)	0.0002615
MeanImp	ARIMA(1,1,0)	-0.5419(0.1120)	0.000000
Linear	ARIMA(0,1,1)	-0.3646(0.1233)	0.003101
Stine	ARIMA(0,1,1)	-0.3646(0.1233)	0.003101
KNN	ARIMA(0,1,1)	-0.4430(0.1201)	0.0002242
LOCF	ARIMA(0,1,1)	-0.3846(0.1219)	0.001604
Spline	ARIMA(0,1,1)	-0.3521(0.1245)	0.004674

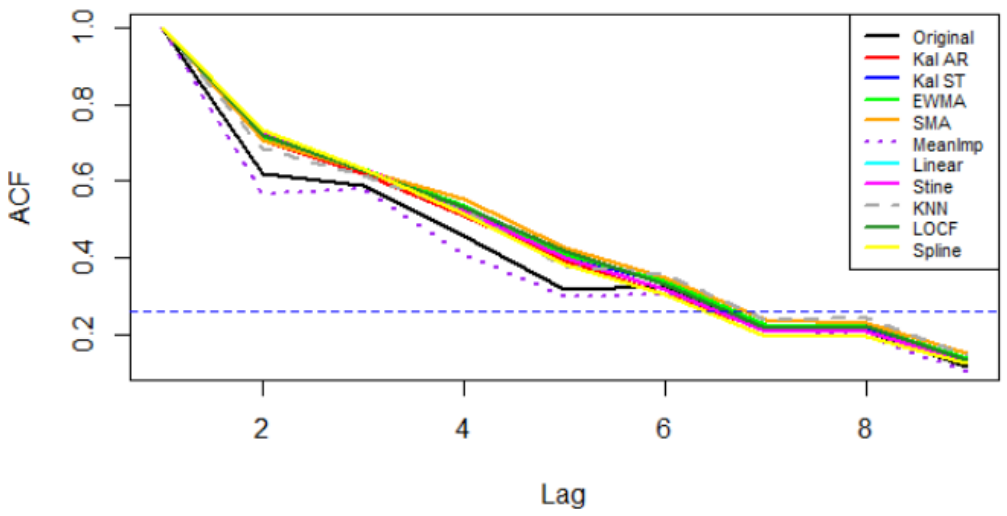

Figure 10. ACF of training data and imputed data at 10% missing data rate

Table 4. Prediction performance of the ARIMA model in original training and imputed data at 10% missing data rate

	RMSE		**MAPE**	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	10.7415	7.4601	10.9258	8.4710
Kal AR	10.27982	7.3106	10.33441	8.1710
Kal ST	10.02822	7.2612	10.12594	8.1460
EWMA	10.02574	7.2651	10.15202	8.1590
SMA	10.09056	7.2886	10.25016	8.230

Mean	11.02351	7.5771	11.25016	8.640
Linear	10.08173	7.2400	10.20420	8.0710
Stine	10.25759	7.2400	10.20420	8.0710
KNN	10.25759	7.30778	10.44693	8.2750
LOCF	10.12764	7.2543	10.24057	8.1240
Spline	10.22252	**7.23189**	10.28856	**8.038**

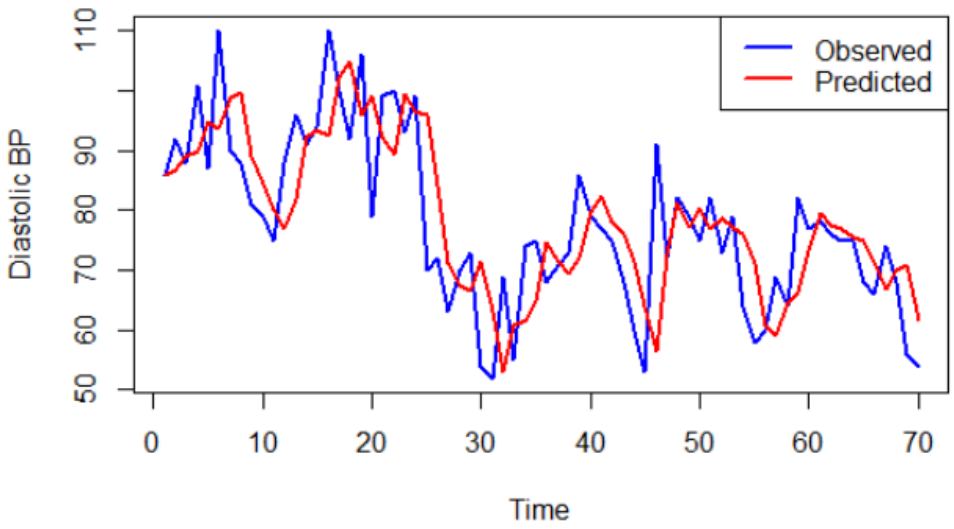

Figure 11: Prediction performance of the ARIMA model in original training data

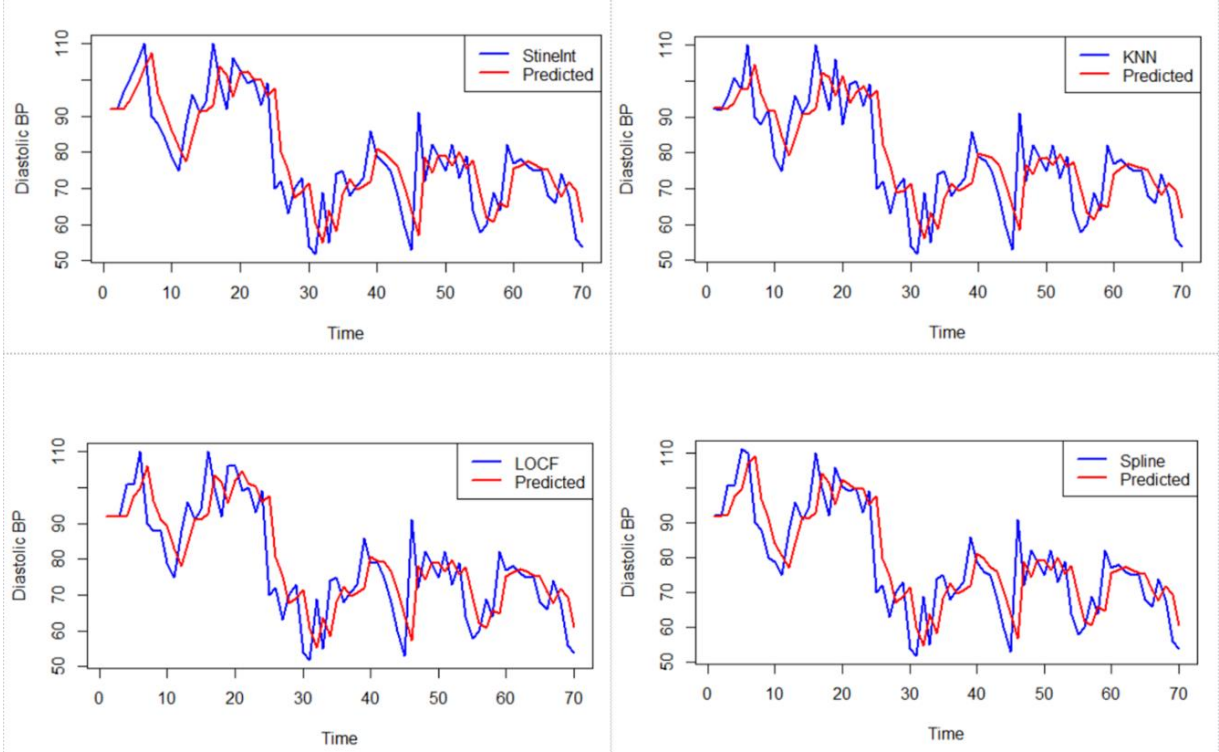

Figure 12: Prediction performance of ARIMA model in stine(top left), KNN (top right), LOCF (bottom left), and spline(bottom right) interpolation imputed dataset at 10% missing data rate.

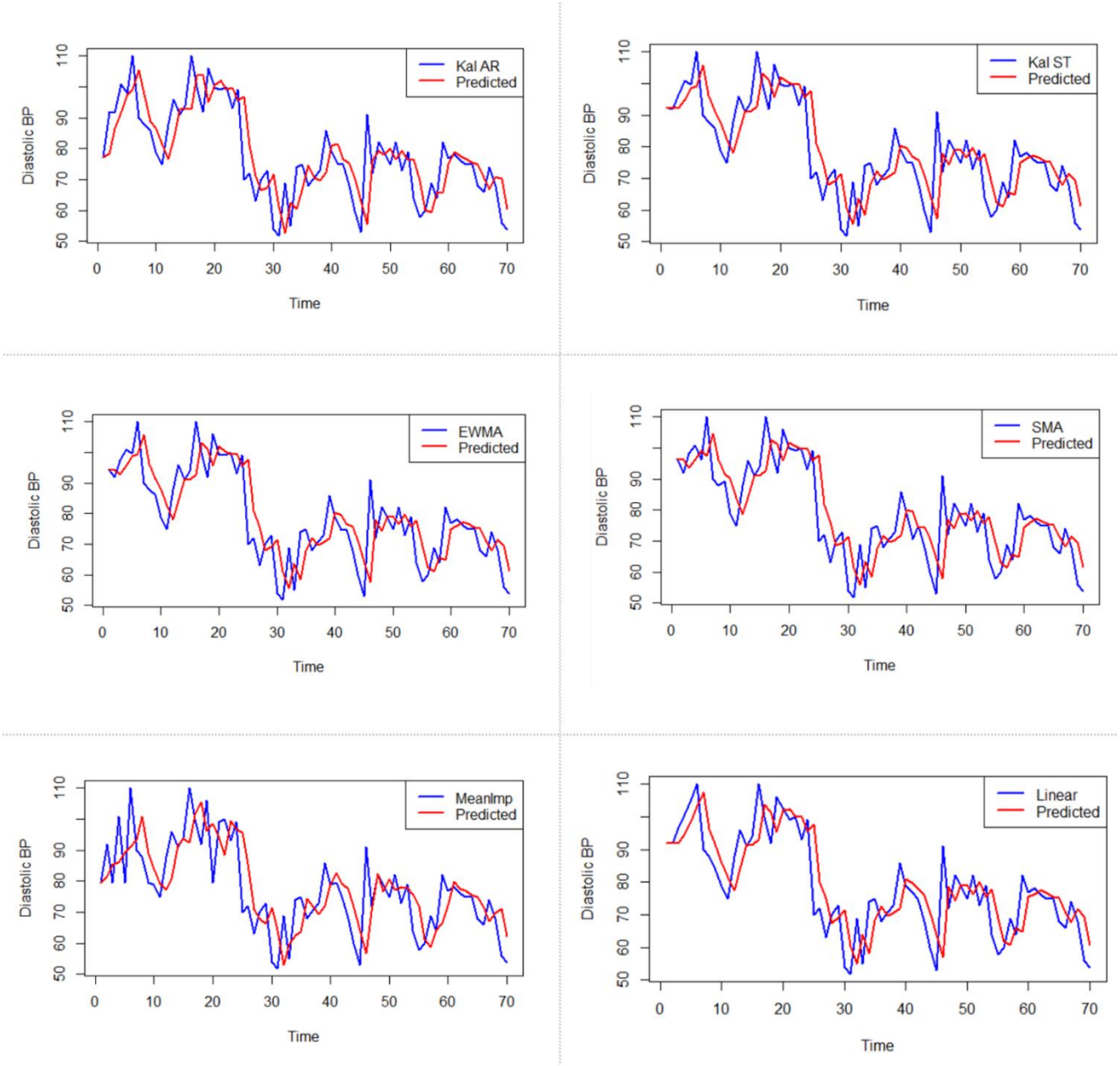

Figure 13: Prediction performance of ARIMA model in Kal AR (top left), Kal ST (top right), EWMA (middle left), SMA (middle right), MeanImp (bottom left), and Linear interpolation (bottom right) imputed dataset at 10% missing data rate.

Table 5. Prediction performance of LSTM on 10% imputed data

	RMSE		MAPE	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	9.1624	6.9976	9.2653	7.8742
Kal AR	9.6553	10.79131	9.572096	11.28484
Kal ST	8.7733	7.1748	7.84416	**6.83658**
EWMA	8.712	10.3995	8.03265	10.263955
SMA	8.9541	9.2407	8.5262	10.24068
Mean	10.6666	8.08152	11.29377	9.20343
Linear	9.2472	7.656	9.10248	8.8898
Stine	9.0847	7.9207	8.46599	7.584497
KNN	10.0431	7.5987	10.75127	8.43368
LOCF	9.2511	7.0612	9.21232	7.1992
Spline	9.0898	**6.7717**	8.758241	7.245654

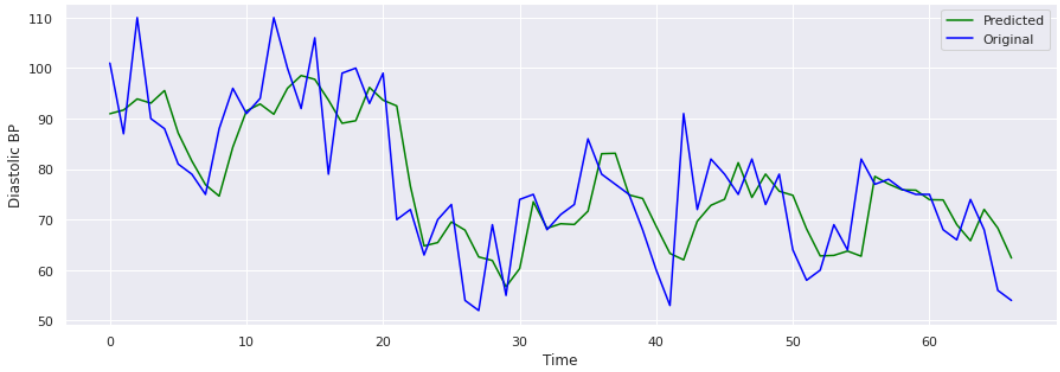

Figure 14: Prediction performance of the LSTM model in the original training data

Figure 15: Prediction performance of LSTM in Kal AR (top left), Kal ST (top right), EWMA (middle left), SMA (middle right), mean (bottom left), and linear interpolation (bottom right) imputed dataset at 10% missing data rate.

Figure 16: Prediction performance of LSTM model in stine(top left), KNN (top right), LOCF (bottom left), and spline(bottom right) interpolation imputed dataset at 10% missing data rate.

4.2.1 Imputation performance in 15% missingness

The comparative time series plot shows the original train and the imputed data using the ten imputation methods (Figure 16). The descriptive statistics of the datasets, both imputed and original train data, are shown in the boxplots (Figure 17). The unconditional mean (also known as the mean imputation) method outperformed the other missing data replacement techniques. The KNN interpolation technique comes in second to this strategy, with spline interpolation coming in last place as least effective based on the RMSE and SMA based on MAPE values (Table 2). The probability density plot of the imputed datasets at 15% is in Figure 18.

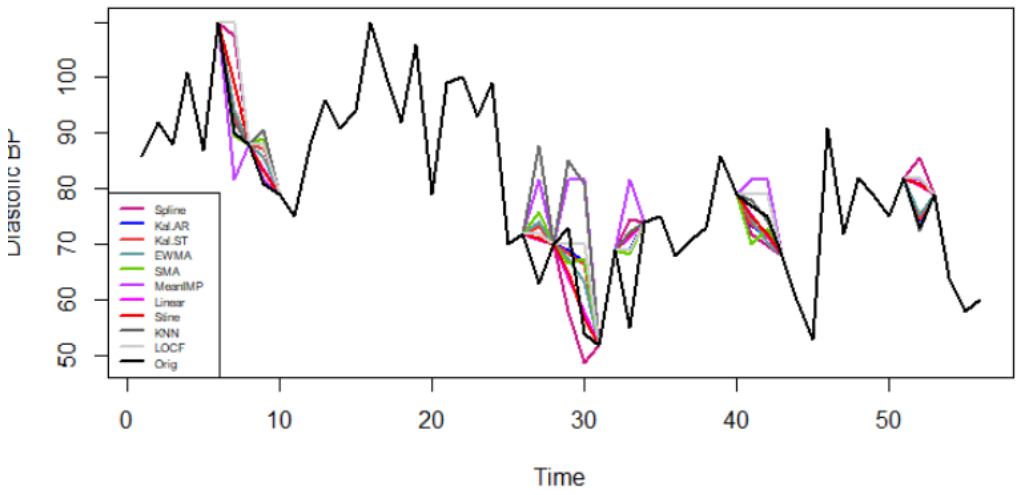

Figure 17.Comparative time series plot of training data and imputed data at 15% missing data rate

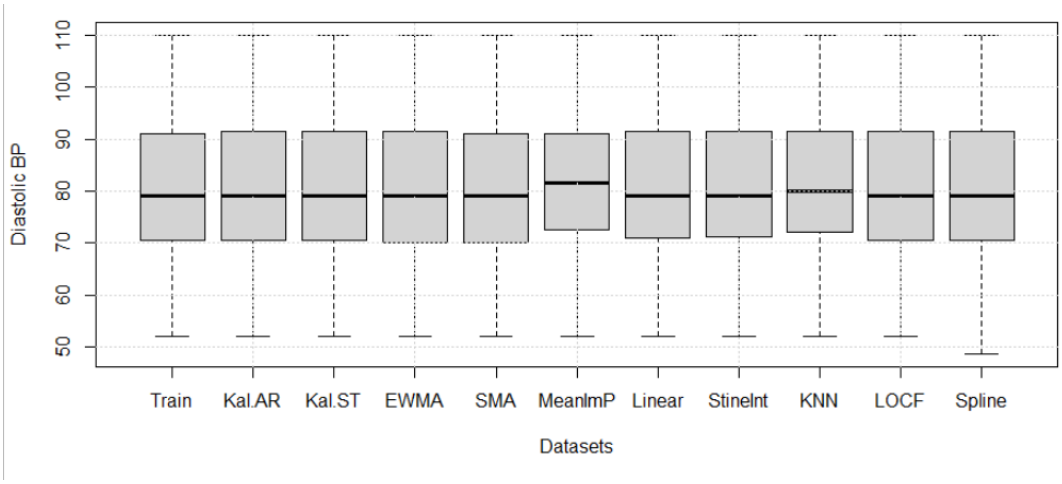

Figure 18: Distribution of training data and imputed data at 15% missing data rate

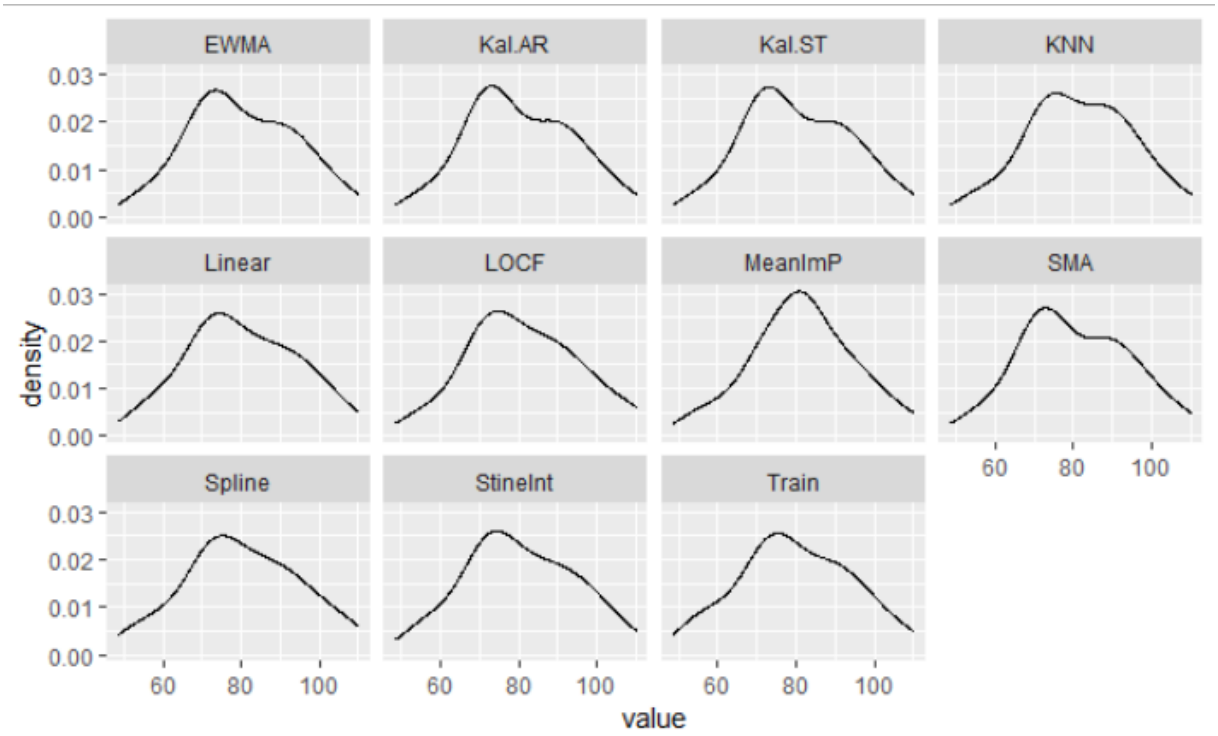

Figure 19. Density plot of training data and imputed data at 15% missing data rate

Table 6. Imputation performance of various techniques at 15% level of generated missing data

Technique	RMSE	MAPE
Kalman AR	3.188437	1.448203
Kalman ST	3.126672	1.451397
EWMA	**3.011556**	1.446414
SMA	3.487081	1.589706
Mean Imputation	6.120874	2.411064
Linear interpolation	3.247633	1.4923

Stine Interpolation	3.224728	**1.429948**
KNN	5.801836	2.065375
LOCF	4.41588	1.868565
Spline	4.858127	2.24265

4.2.2 Prediction performance in 15% imputed datasets

The summary, order, and coefficient of the ARIMA models in 15% imputed and original training datasets (see Table 7). The missing data replacement methods increased the autocorrelation of the imputed data, except for the mean imputation technique using the original train data as the reference. The KNN-imputed dataset showed a reduced autocorrelation in the first four lags (Figure 19). The prediction performance of the ARIMA in the imputed datasets showed that the model in the spline imputed dataset was the best, and the mean imputed data was the worse. ARIMA models in the imputed datasets performed better than the model in the original train except for the mean imputed data (Table 8). The time plot of the ARIMA prediction on each dataset (Figure 20 and Figure 21). The prediction in the KNN imputed dataset was the best based on both RMSE and MAPE (Table 9). The prediction performance of the LSTM is in Figure 22.

Table 7 ARIMA model obtained in 15% imputed data

Data	Model	C(SE)	Pr(>|z|)
Train	ARIMA(1,1,0)	-0.4844(0.1161)	0.0000
Kal.AR	ARIMA(0,1,1)	-0.5115(0.1243)	0.00004
Kal.ST	ARIMA(0,1,1)	-0.5054(0.1245)	0.000049
EWMA	ARIMA(0,1,1)	-0.4861(0.1256)	0.00011
SMA	ARIMA(0,1,1)	-0.5332(0.1202)	0.00000
MeanImp	ARIMA(0,1,1)	-0.6423(0.1346)	0.00000
Linear	ARIMA(0,1,1)	-0.4360(0.1304)	0.000831
Stine	ARIMA(0,1,1)	-0.4351(0.1316)	0.000942
KNN	ARIMA(0,1,1)	-0.6814(0.1233)	0.000000
LOCF	ARIMA(0,1,1)	-0.4878(0.1402)	0.000506
Spline	ARIMA(0,1,1)	-0.3863(0.1426)	0.006749

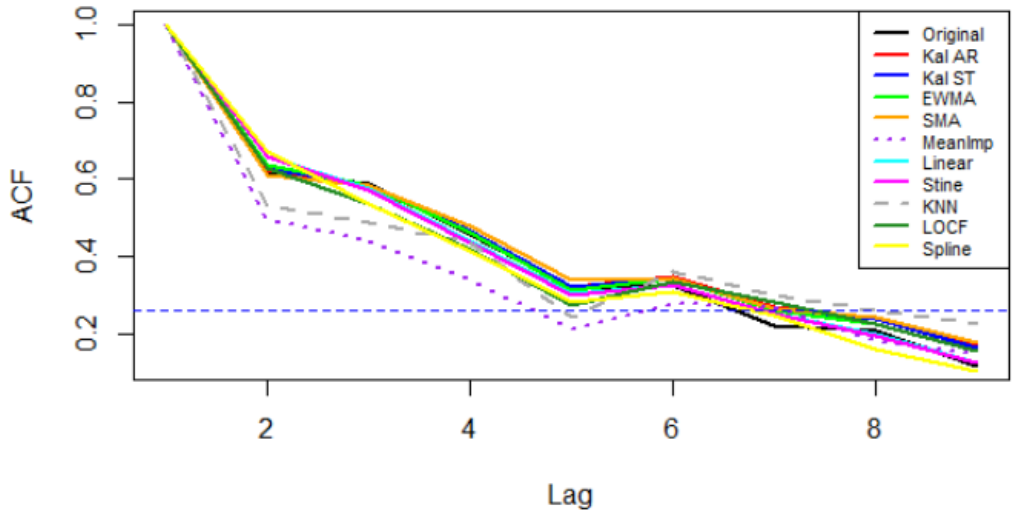

Figure 20. ACF of training data and imputed data at 15% missing data rate

Table 8. ARIMA performance in 15% imputed data

	RMSE		**MAPE**	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	10.7415	7.4601	10.9258	8.4710
Kal AR	10.21487	7.39043	10.24301	8.440
Kal ST	10.21277	7.38141	10.24478	8.4270
EWMA	10.22771	7.41249	10.28395	8.3860
SMA	10.2745	7.42583	10.3443	8.4840
Mean	10.98089	7.56997	10.88417	8.7800
Linear	10.37185	7.29011	10.35169	8.2580

Stine	10.39919	7.28909	10.3907	8.256
KNN	10.80289	7.64289	10.66717	8.8570
LOCF	10.6814	7.3442	10.44967	8.3870
Spline	11.00173	**7.2443**	11.06771	**8.1310**

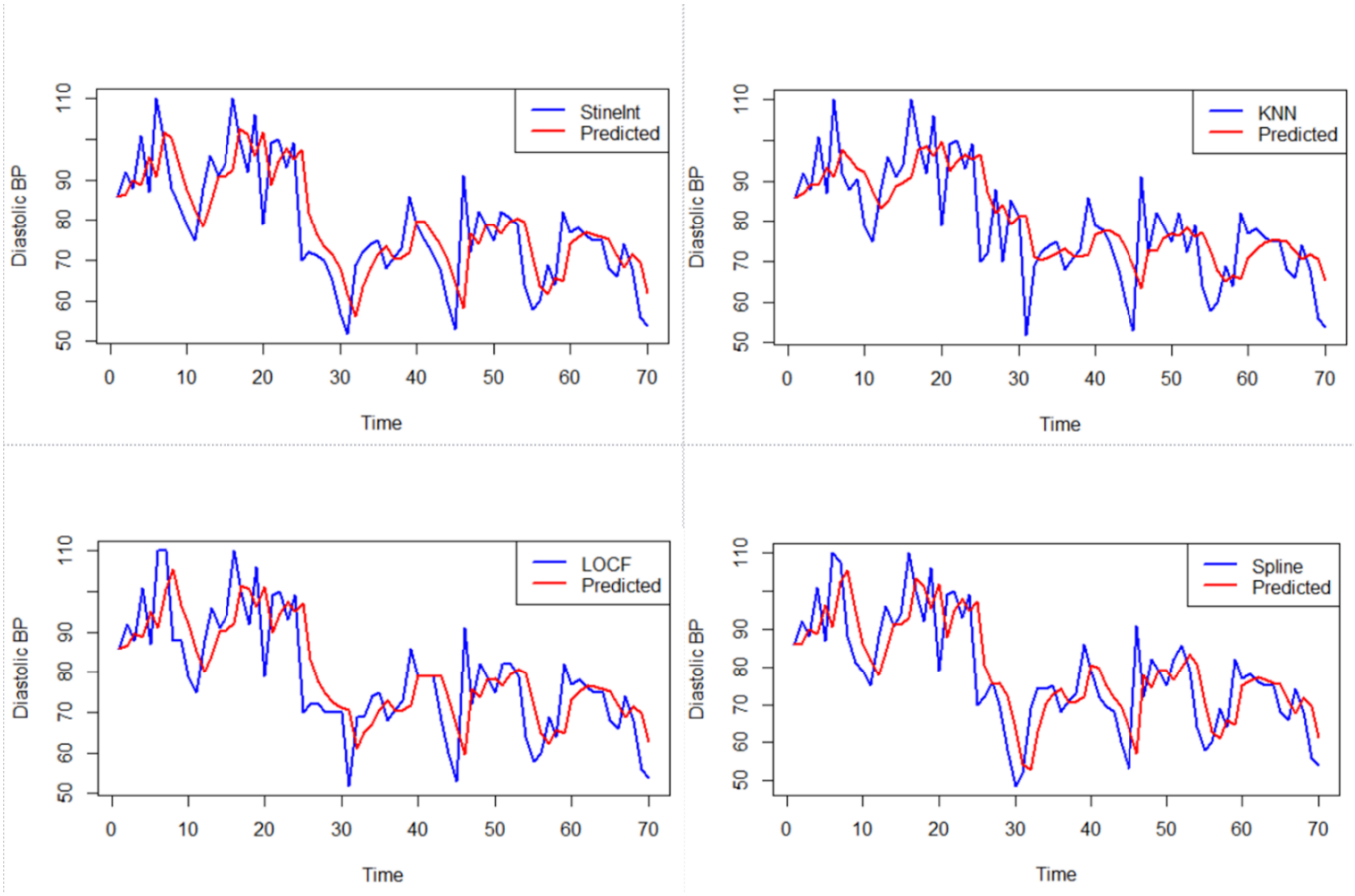

Figure 21: Prediction performance of ARIMA model in stine(top left), KNN (top right), LOCF (bottom left), and spline(bottom right) interpolation imputed dataset at 15% missing data rate.

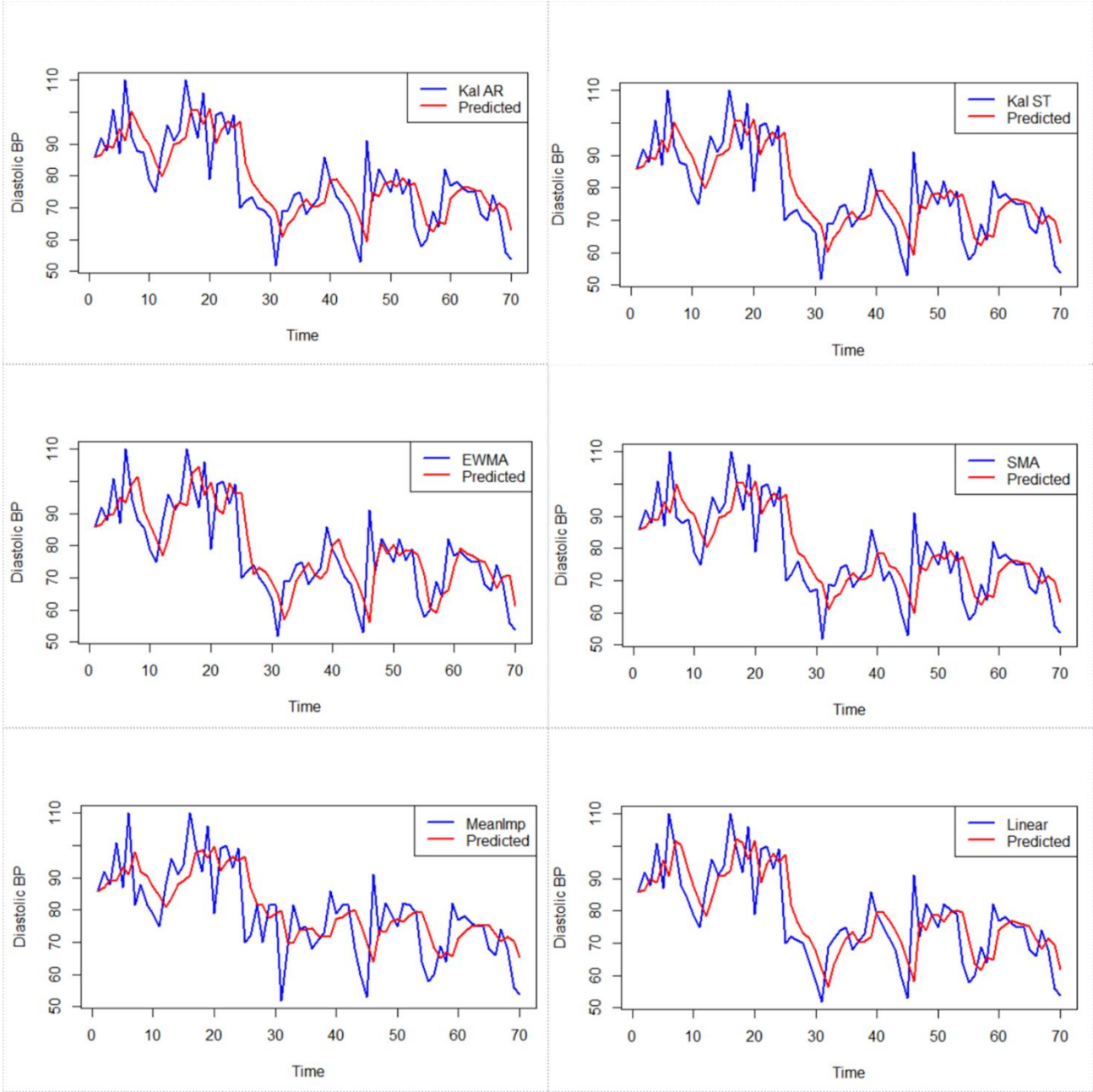

Figure 22: Prediction performance of ARIMA model in Kal AR (top left), Kal ST (top right), EWMA (middle left), SMA (middle right), mean (bottom left), and linear interpolation (bottom right) imputed dataset at 15% missing data rate.

Table 9. LSTM performance on 15% imputed data

	RMSE		MAPE	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	9.1624	6.9976	9.2653	7.8742
Kal AR	9.4448	7.158127	9.21625	8.332801
Kal ST	9.8081	7.9714	10.0917	10.00353
EWMA	9.539	7.1862	9.37526	8.444839
SMA	9.6296	8.6818	9.242071	10.84434
Mean	9.7847	7.6766	9.85172	8.84394
Linear	9.1172	9.0516	7.75999	10.09751
Stine	9.003	9.9418	8.017376	10.0028
KNN	10.1829	**6.4936**	10.28377	**7.65678**
LOCF	10.2038	7.4011	9.88481	8.74803
Spline	9.3754	9.3217	8.70404	10.8177

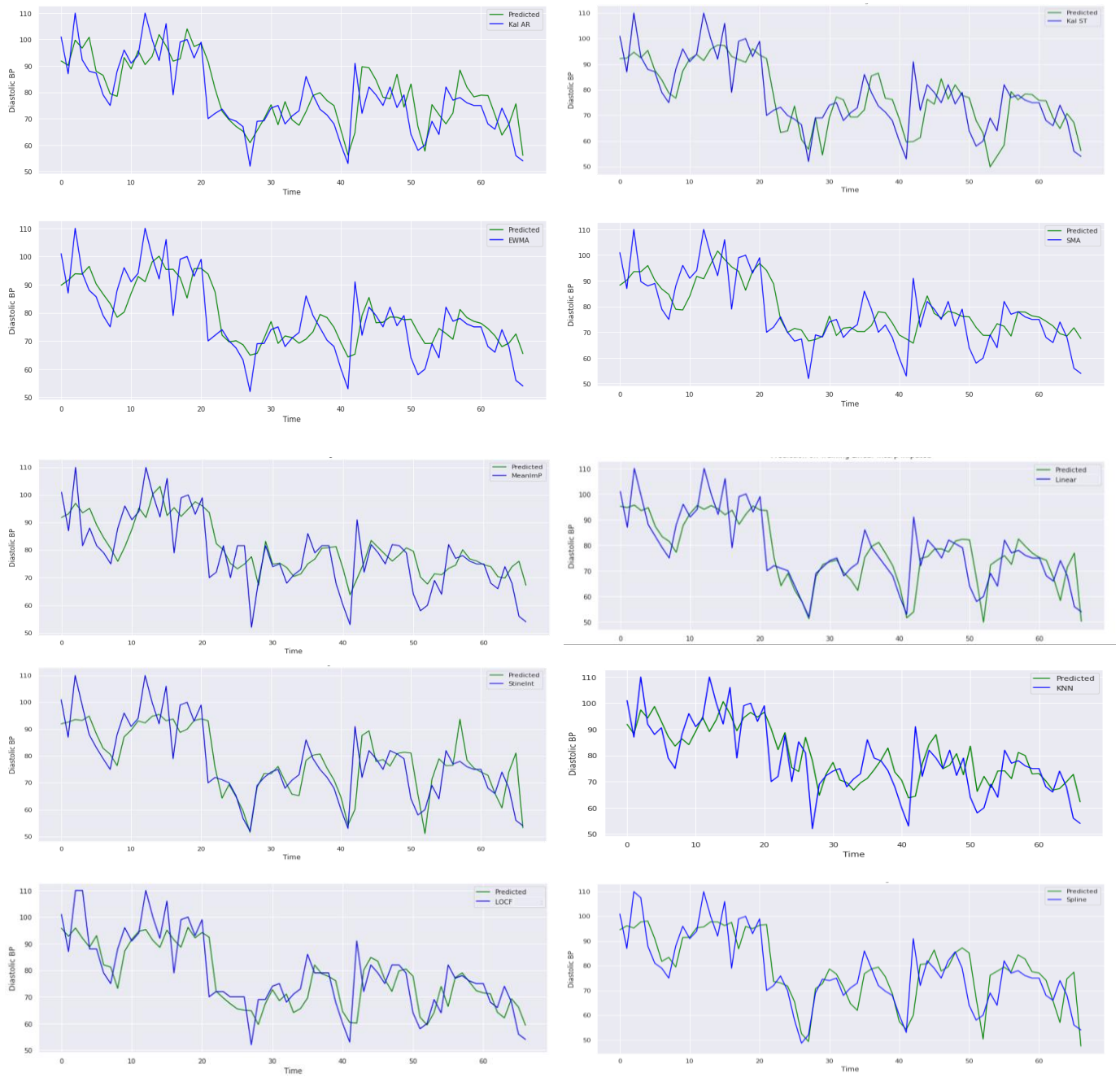

Figure 23 Prediction performance of the LSTM in the 15% imputed datasets; Kal AR(first on first row),Kal ST(second on first row), EWMA(first on second row), SMA(second on second row),mean (first on third row), linear(second on third row), stine(first on fourth row), KNN(second on fourth row), LOCF(first on fifth row),spline(second on fifth row).

4.3.1 Imputation performance in 25% missingness

The comparative time series plot shows the original train and the imputed data using the ten imputation methods (Figure 23). The descriptive statistics of the datasets, both imputed and original train data, are shown in the boxplots. Outliers were detected in the mean imputed dataset using the 1.5IQR rule (Figure 24). The probability density plot of the mean imputed data is distorted compared to the other datasets (Figure 25). The imputation performance at a 25% rate of missingness showed that Kalman ST was the best missing data replacement technique in RMSE and MAPE. EWMA technique came in second place, where spline was the worst data replacement technique (Table 10).

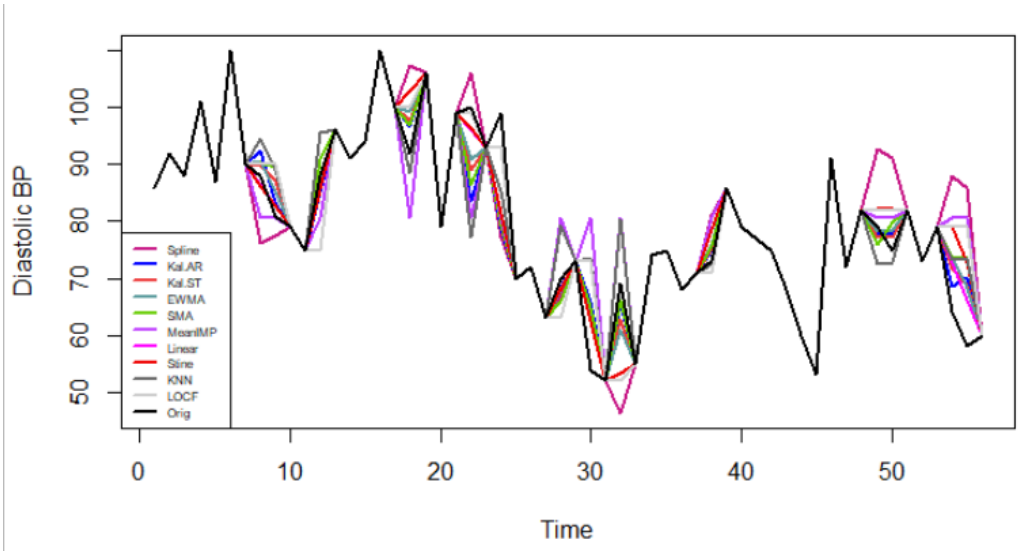

Figure 24: Comparative time series plot of training data and imputed data at 25% missing data rate

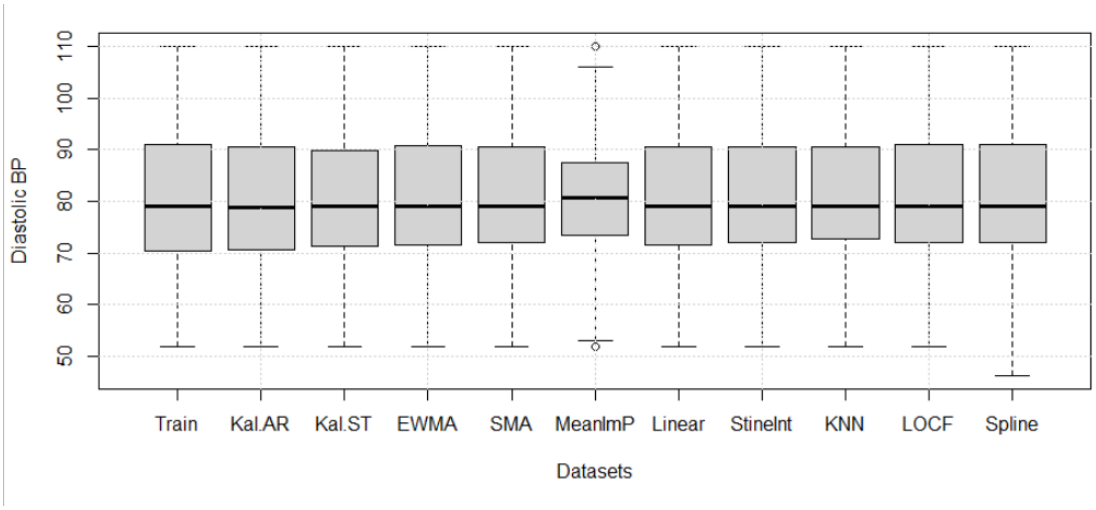

Figure 25: Distribution of training data and imputed data at 25% missing data rate

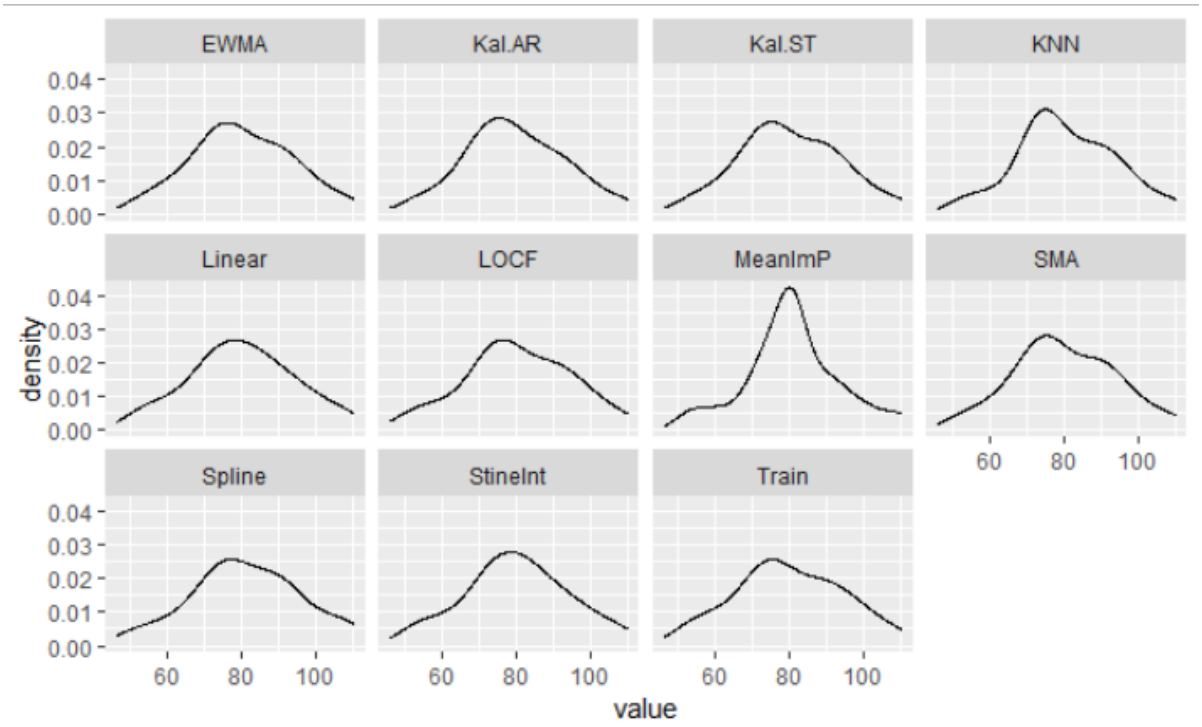

Figure 26: Density plot of training data and imputed data at 25% missing data rate

Table 10. Performance of imputation technique at 25% level of missing data

Technique	RMSE	MAPE
Kalman AR	4.40064	2.122796
Kalman ST	**3.946298**	**2.02317**
EWMA	3.999861	2.112269
SMA	4.611497	2.415885
Mean Imputation	7.056542	3.704329
Linear interpolation	4.234976	2.364167
Stine Interpolation	4.830909	2.638987
KNN	5.748694	3.073106
LOCF	5.641049	3.132674
Spline	7.819639	4.361373

4.3.2 Prediction performance in 25% imputed data

The summary, order, and coefficient of the ARIMA models in 25% imputed and original training datasets (see Table 11). The autocorrelation within the mean imputed dataset was reduced significantly. Spline also showed a lowered autocorrelation decaying exponentially. The KNN imputed data showed a reduced autocorrelation in its first four lags (Figure 26). The prediction performance of ARIMA in LOCF-imputed data was the best in terms of RMSE, with the model in

the original train data being the worst (Table 12). The time plot of the ARIMA prediction in the 25% imputed dataset (Figure 27 and Figure 28). The LSTM prediction in the Kal ST imputed dataset was the best based on RMSE and MAPE, with EWMA being the second best (Table 13). The prediction performance of the LSTM is in Figure 29.

Table 11. ARIMA obtained on 25% imputed data

Dataset	Model	C(SE)	Pr(>|z|)
Train	ARIMA(1,1,0)	-0.4844 (0.1161)	0.0000
Kal.AR	ARIMA(1,1,0)	-0.5560 (0.1108)	0.0000
Kal.ST	ARIMA(1,1,0)	-0.4993 (0.1157)	0.00001
EWMA	ARIMA(1,1,0)	-0.4720 (0.1178)	0.00006
SMA	ARIMA(1,1,0)	-0.5307 (0.1140)	0.000003
MeanImp	ARIMA(1,1,0)	-0.6351 (0.1044)	0.00000
Linear	ARIMA(0,1,1)	-0.3946 (0.1255)	0.001959
Stine	ARIMA(0,1,1)	-0.4046 (0.1287)	0.00166
KNN	ARIMA(1,1,0)	-0.6376(0.1023)	0.00000
LOCF	ARIMA(0,1,1)	-0.4666 (0.1323)	0.00042
Spline	ARIMA(1,1,1)	0.6217 |-0.9275(0.1637 |0.0948)	0.00015| 0.0000

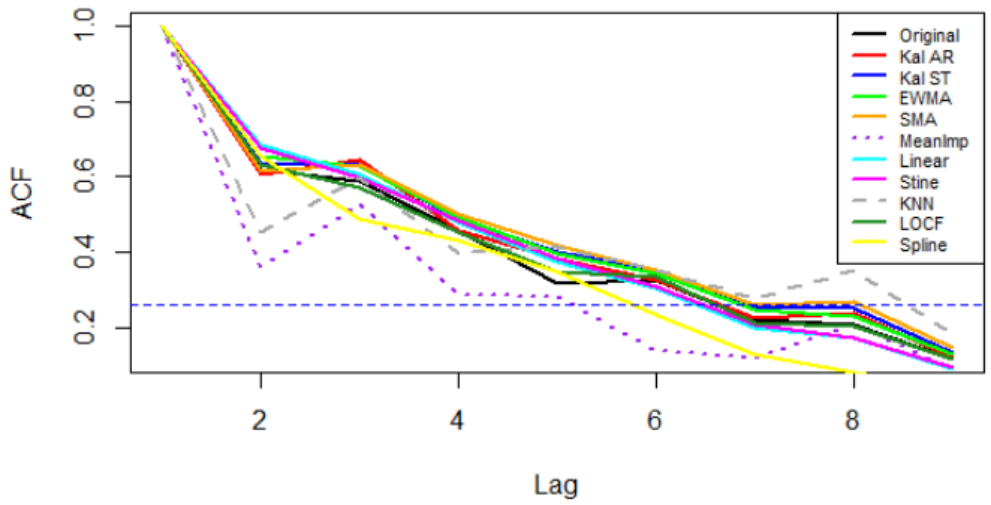

Figure 27. ACF of training data and imputed data at 25% missing data rate

Table 12 ARIMA performance on 25% imputed data

	RMSE		**MAPE**	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	10.7415	7.4601	10.9258	8.4710
Kal AR	9.673936	7.1999	9.470673	8.005
Kal ST	9.77325	7.11293	9.425376	7.967
EWMA	9.81022	7.09267	9.4803	7.952
SMA	9.852183	7.08436	9.535415	**7.768**
Mean	10.86387	7.38017	10.48997	8.4130
Linear	10.08832	7.05886	9.933298	7.8620
Stine	10.15777	6.91784	9.941798	7.8080

KNN	10.51695	7.3272	10.82218	8.080
LOCF	10.78772	**6.91464**	10.45732	7.777
Spline	11.19237	6.99171	11.65583	8.3630

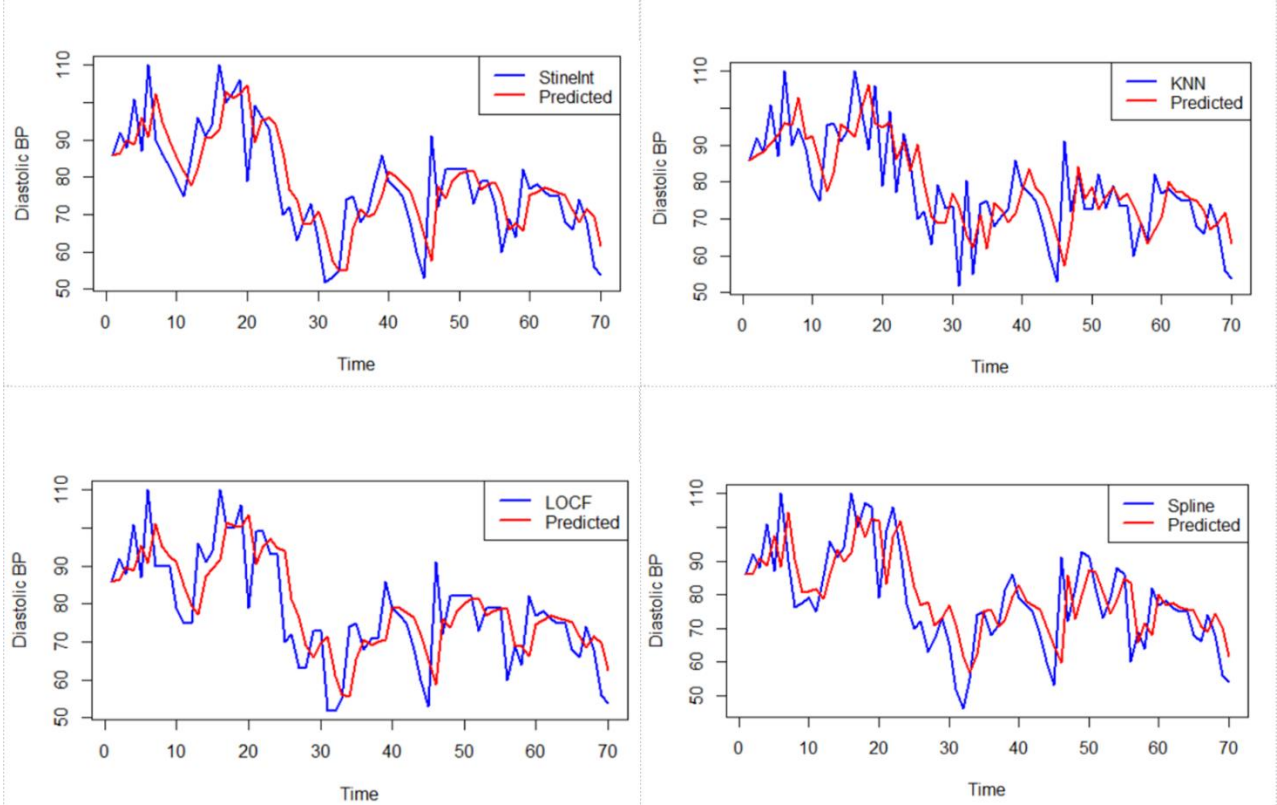

Figure 28: Prediction performance of ARIMA model in stine(top left), KNN (top right), LOCF (bottom left), and spline(bottom right) interpolation imputed dataset at 25% missing data rate.

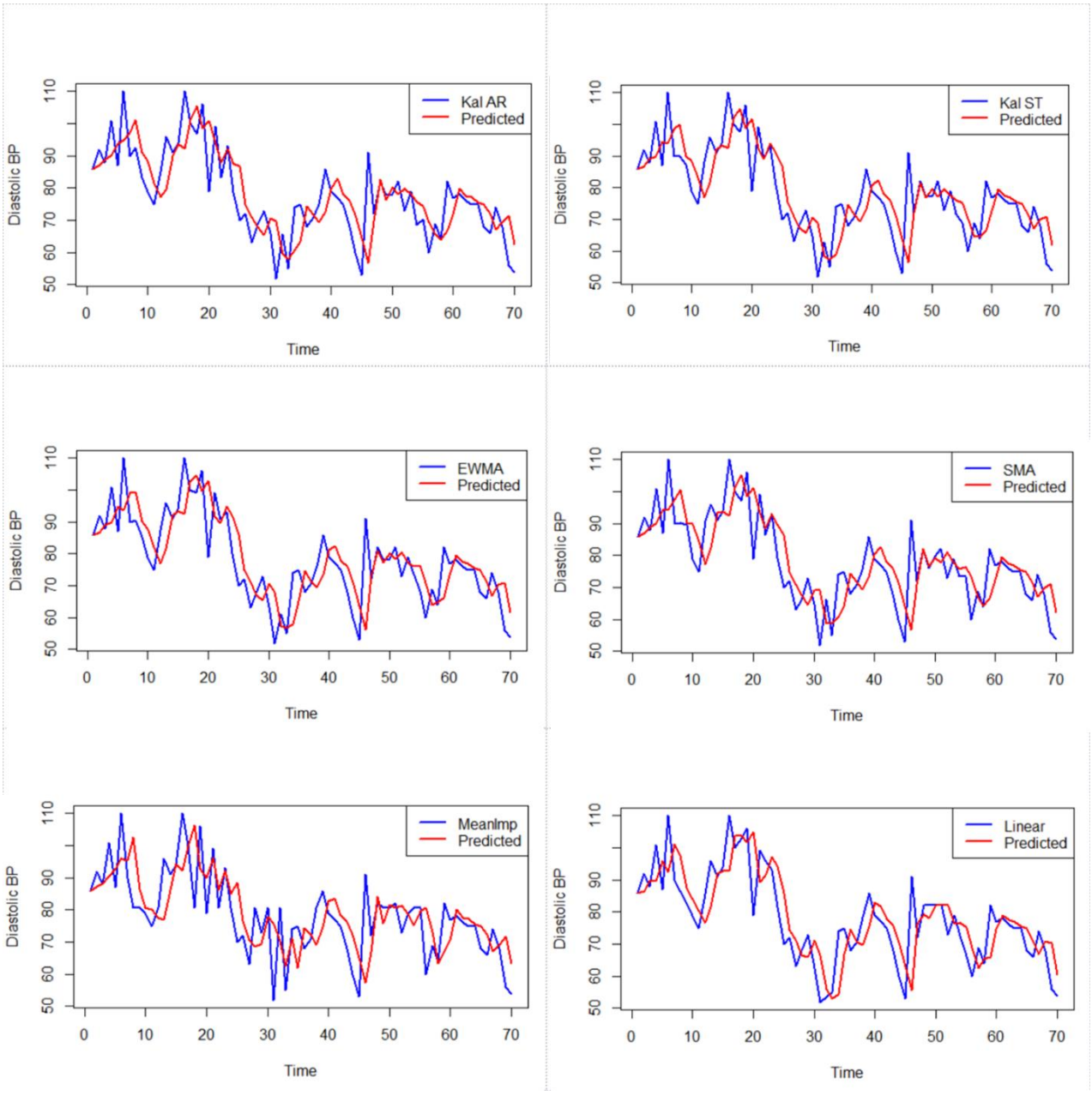

Figure 29: Prediction performance of ARIMA model in Kal AR (top left), Kal ST (top right), EWMA (middle left), SMA (middle right), MeanImp (bottom left), and Linear interpolation (bottom right) imputed dataset at 25% missing data rate.

Table 13. LSTM performance on 25% imputed data

	RMSE		MAPE	
	Train	**Test**	**Train**	**Test**
Original	9.1624	6.9976	9.2653	7.8742
Kal AR	9.1067	7.0614	7.77871	7.778712
Kal ST	9.1049	**6.4184**	8.703598	7.0720944
EWMA	9.2886	6.6478	8.7758337	7.4387604
SMA	9.6926	7.5763	9.67356	8.383058
Mean	10.2237	8.442509	10.64077	10.15677
Linear	9.3534	8.3362	8.7256617	7.904302
Stine	9.6953	8.131101	9.7006114	8.6371184
KNN	9.5688	7.31604	9.600092	8.1603804
LOCF	10.0855	7.0506	9.2044165	7.4510109
Spline	11.158	8.7628	10.05401	8.0435749

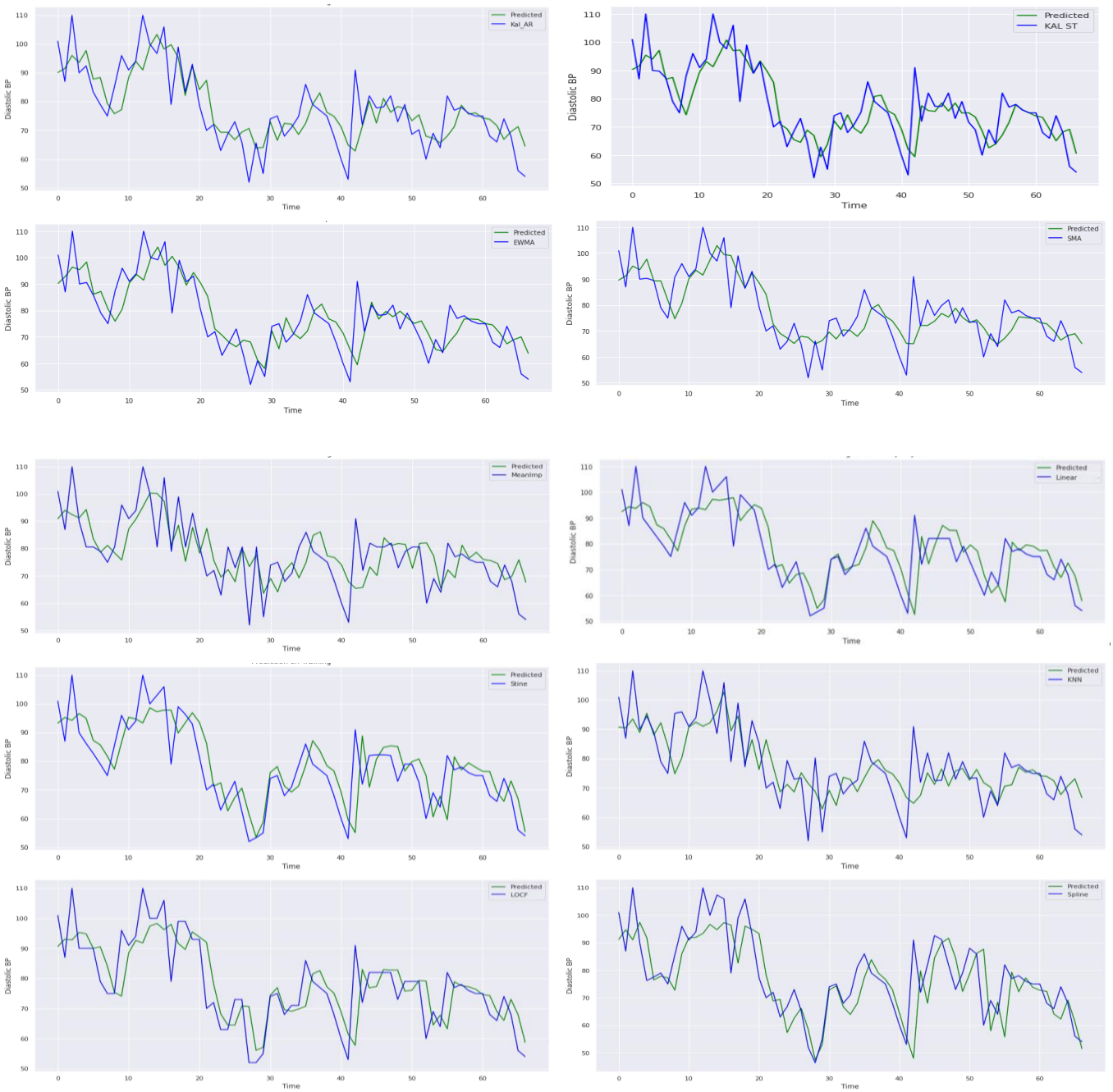

Figure 30: Prediction performance of the LSTM in the 25% imputed datasets; Kal AR(first on first row),Kal ST(second on first row), EWMA(first on second row), SMA(second on second row),mean (first on third row), linear(second on third row), stine(first on fourth row), KNN(second on fourth row), LOCF(first on fifth row),spline(second on fifth row).

4.4.1 Imputation performance in 35% imputed datasets

The comparative time series plot shows the original train and the imputed data using the ten imputation methods (Figure 30). The descriptive statistics of the datasets, both imputed and original train data, are shown in the boxplots. Outliers were also detected in the mean imputed dataset using the 1.5IQR rule (Figure 31). The probability density plot of the mean imputed data is significantly distorted compared to the other datasets (Figure 32). The imputation performance at a 35% rate of missingness showed that Kalman ST was the best missing data replacement technique in RMSE, and Kalman AR was the best in terms of MAPE, with mean imputation being the worst (Table 14)

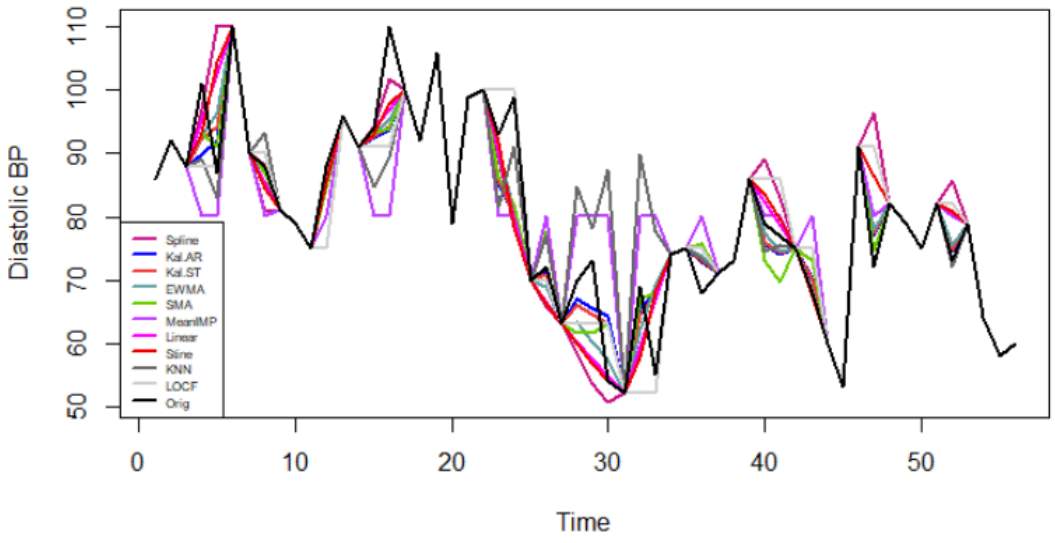

Figure 31: Comparative time series plot of training data and imputed data at 35% missing data rate

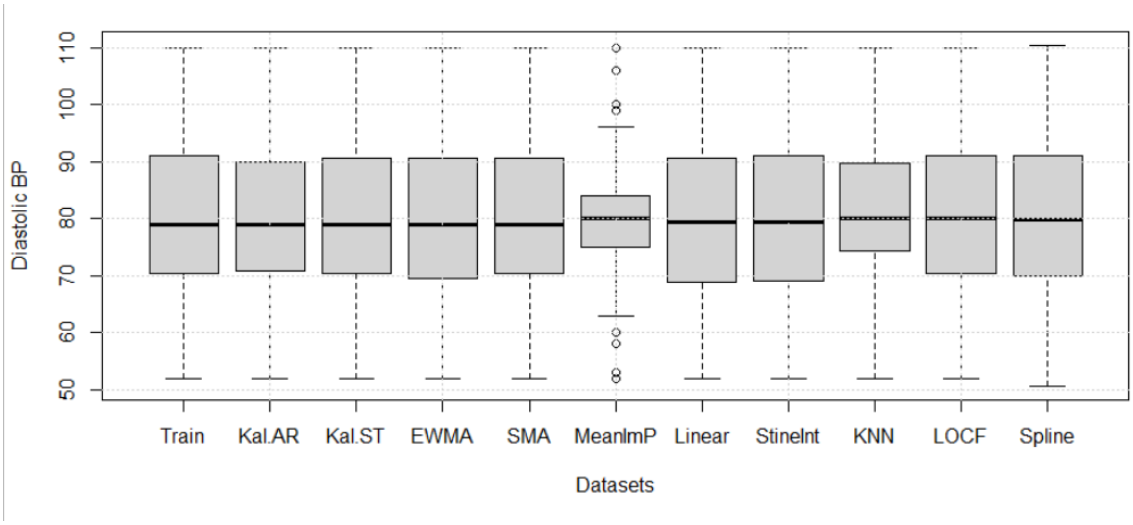

Figure 32: Distribution of training data and imputed data at 35% missing data rate

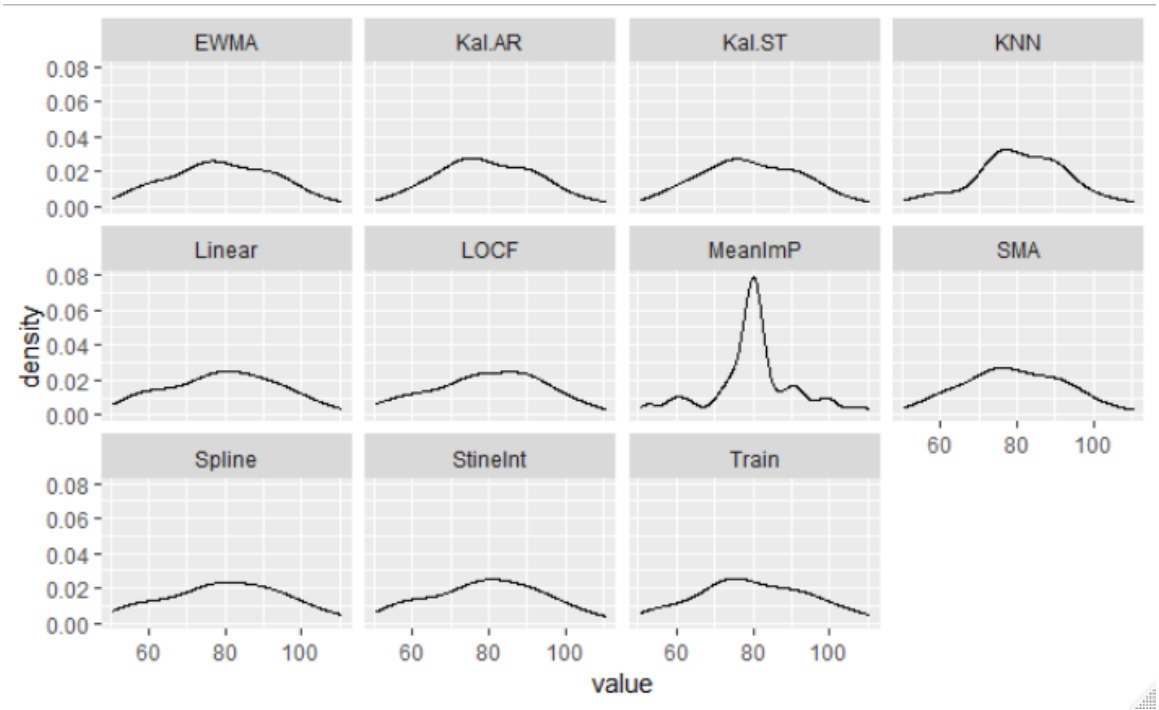

Figure 33: Density plot of training data and imputed data at 35% missing data rate

Table 14. Performance of imputation techniques at 35% of missing data

Technique	RMSE	MAPE
Kalman AR	4.747496	**2.761497**
Kalman ST	**4.680732**	2.80615
EWMA	4.760597	2.885924
SMA	4.899272	3.21842
Mean Imputation	8.756517	5.583606
Linear interpolation	5.558737	3.463321
Stine Interpolation	5.951343	3.726194
KNN	7.704995	4.067641
LOCF	5.923621	3.917054
Spline	7.161102	4.490717

4.4.2 Prediction performance in 35% imputed data

The summary, order, and coefficient of the ARIMA models in 35% imputed and original training datasets. Datasets imputed with linear, stine, LOCF, and stine resulted in a random walk process with inestimable coefficients (see Table 15). At this level of imputation, the mean, KNN, LOCF, spline, linear, and stine showed a lowered autocorrelation (Figure 33). The prediction performance of ARIMA in the EWMA-imputed dataset was the best in terms of RMSE and MAPE, followed by Kal ST and Kal AR, with the model in the KNN dataset being the worst(Table 16).

The ARIMA prediction in the 35% imputed dataset (Figure 34 and Figure 35). The LSTM on the original train dataset outperformed all the imputed datasets in the 35% rate missingness (Table 17). The prediction performance of the LSTM in each dataset is shown in Figure 36.

Table 15. ARIMA model obtained on 35% imputed data

Dataset	Model	C(SE)	Pr(>|z|)
Train	ARIMA(1,1,0)	-0.4844(0.1161)	-
Kal.AR	ARIMA(0,1,1)	-0.4311(0.1276)	0.000727
Kal.ST	ARIMA(0,1,1)	-0.3999(0.1280)	0.001783
EWMA	ARIMA(1,1,0)	-0.3195(0.1267)	0.01165
SMA	ARIMA(0,1,1)	-0.4605(0.1168)	0.000008
MeanImp	ARIMA(0,1,1)	-0.7815(0.1281)	0.000000
Linear	ARIMA(0,1,0)	-	
Stine	ARIMA(0,1,0)	-	
KNN	ARIMA(0,1,1)	-0.7859(0.0942)	0.00000
LOCF	ARIMA(0,1,0)	-	-
Spline	ARIMA(0,1,0)	-	-

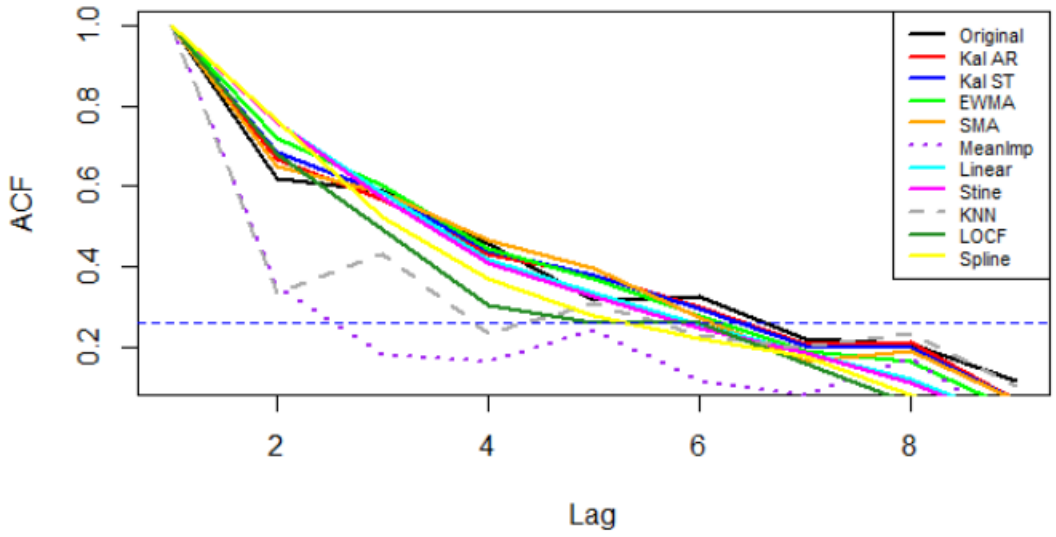

Figure 34. ACF of training data and imputed data at 35% missing data rate

Table 16. ARIMA performance on 35% imputed data

	RMSE		MAPE	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	10.7415	7.4601	10.9258	8.4710
Kal AR	9.3882	7.2973	9.1919	8.2540
Kal ST	9.3405	7.2680	9.1846	8.1730
EWMA	9.3067	**7.2531**	8.8726	**7.9890**
SMA	9.5863	7.3294	9.7408	8.3220
Mean	10.6788	7.8376	10.9086	9.2560
Linear	9.5096	7.3776	8.5189	8.0460

Stine	9.5723	7.3776	8.51685	8.0460
KNN	10.5650	7.8615	10.5417	9.3760
LOCF	11.1692	7.3776	8.5630	8.0460
Spline	10.0214	7.3776	9.2268	8.0460

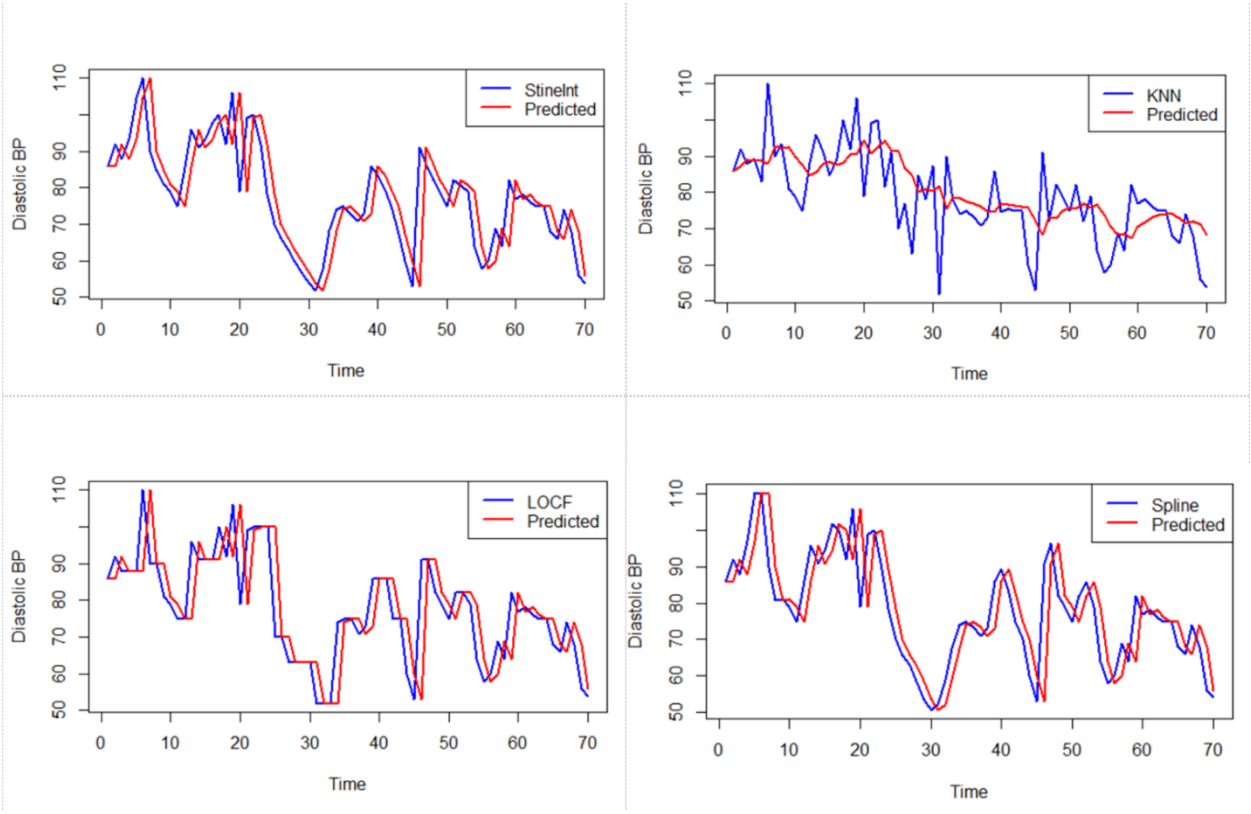

Figure 35: Prediction performance of ARIMA model in stine(top left), KNN (top right), LOCF (bottom left), and spline(bottom right) interpolation imputed dataset at 35% missing data rate.

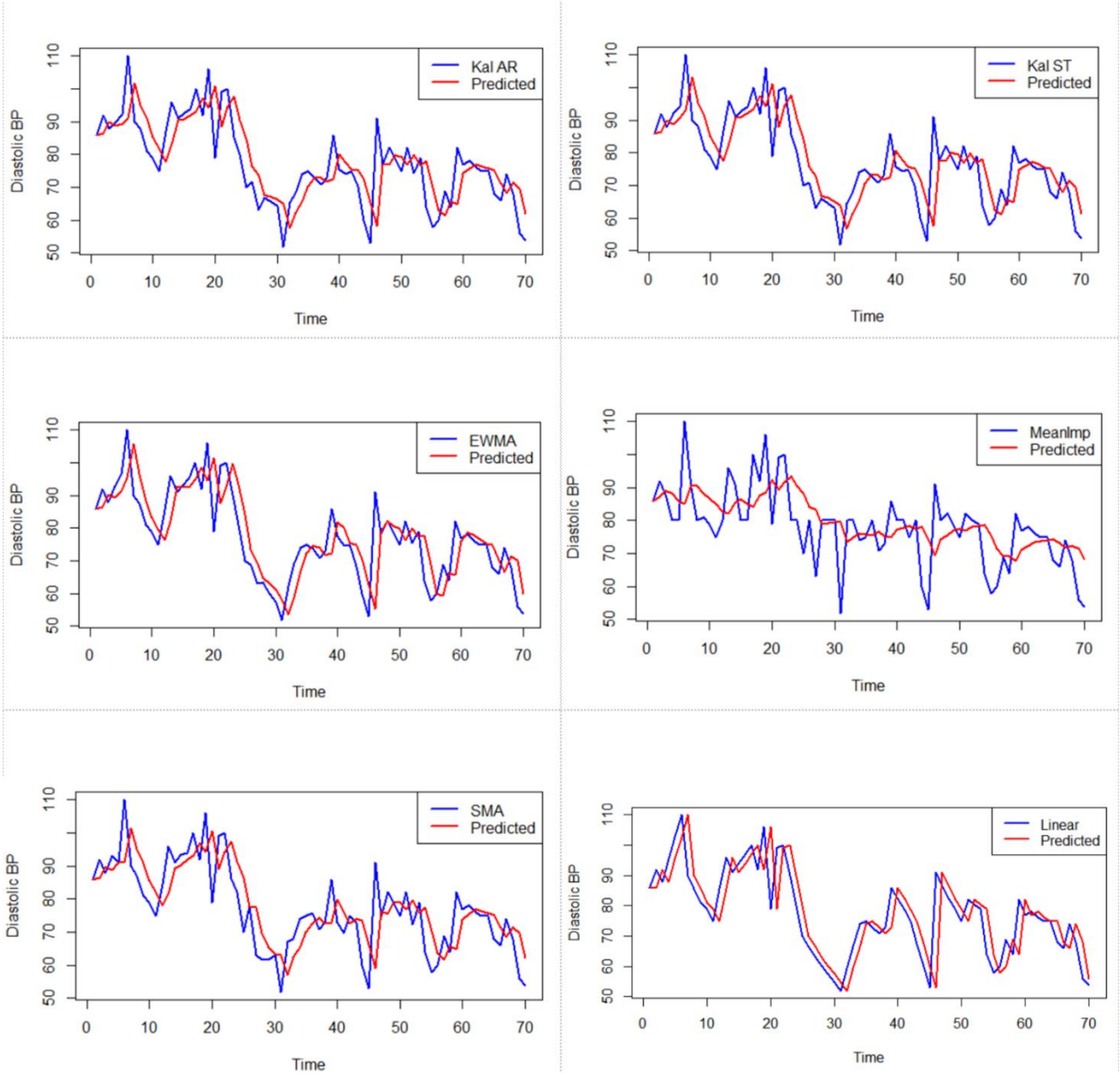

Figure 36: Prediction performance of ARIMA model in Kal AR (top left), Kal ST (top right), EWMA (middle left), SMA (middle right), mean (bottom left), and linear interpolation (bottom right) imputed dataset at 35% missing data rate.

Table 17.LSTM performance on 35% imputed data

	RMSE		MAPE	
Dataset	**Train**	**Test**	**Train**	**Test**
Original	9.1624	**6.9976**	9.2653	**7.8741**
Kal AR	7.9381	7.2527	7.7158	8.09551
Kal ST	8.1695	7.5205	7.7095	9.1685
EWMA	8.9142	7.6226	7.6129	8.3102
SMA	7.6871	9.8699	8.1015	11.5875
Mean	10.4242	9.1676	10.0183	10.9721
Linear	8.4688	8.4115	7.0661	8.7439
Stine	9.7287	8.5534	7.8319	8.7218
KNN	10.3023	7.9139	9.9549	9.0317
LOCF	10.1281	9.2982	8.2839	9.0239
Spline	9.7466	8.3552	8.3684	9.7541

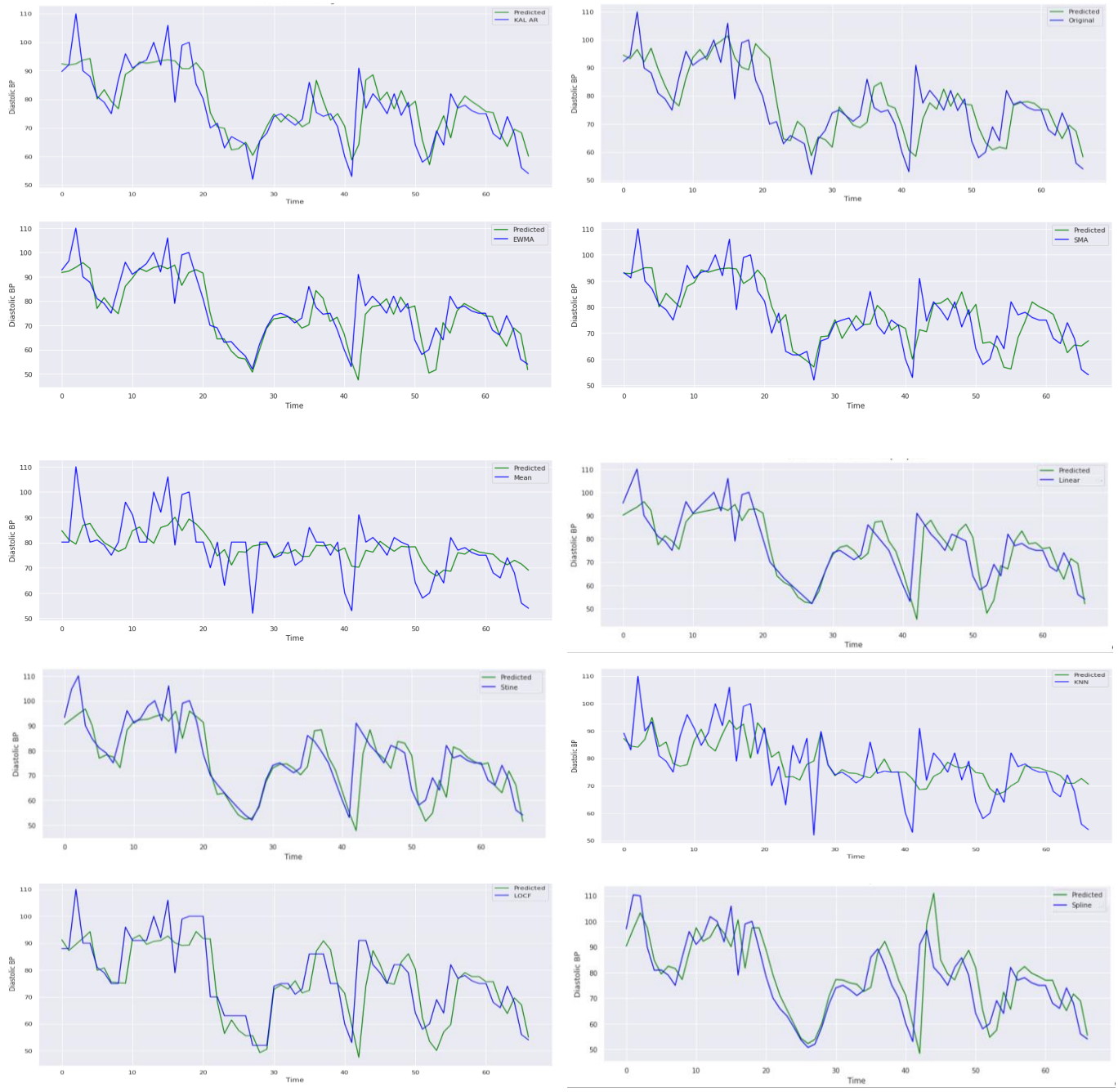

Figure 37: Prediction performance of the LSTM in the 25% imputed datasets; Kal AR(first on first row), Kal ST(second on first row), EWMA(first on second row), SMA(second on second row), mean (first on third row), linear(second on third row), stine(first on fourth row), KNN(second on fourth row), LOCF(first on fifth row),spline(second on fifth row).

CHAPTER V

DISCUSSION

This study investigated the effects of various imputation techniques on univariate time series forecasting using ARIMA and LSTM models. We compared the imputation performance of mean imputation, LOCF, EWMA, SMA, Kal ST, Kal AR, KNN interpolation, linear, cubic spline, and stine interpolation at four different rates (10%, 15%, 25%, and 35%) of missingness under MCAR mechanism. Consideration was given to the effectiveness of these imputation approaches, the modifications made to the data, and the effects of these modifications on time series models.

Our study showed that the mean imputation, followed by the KNN interpolation was the best technique with the smallest MAPE and RMSE at a 10% rate of missing data. This indicated that the mean imputation technique is effective for replacing missing values in time series data when the proportion of missingness is smaller. This finding is consistent with the observations made by (Norazian, Shukri, Azam & AlBakri, 2008; Zakaria & Noor, 2018; Wijesekara & Liyanage 2020). However, the mean imputation is not a reliable method of replacing missing data. Recall, mean imputation was the best imputation technique at a 10% rate of missingness, but as the rate of missingness increased to 15%, it became the worst technique. At a 15% rate of missingness, the exponentially weighted moving average (EWMA) technique outperformed the other imputation techniques in terms of RMSE, and stine interpolation was the best method of imputation based on MAPE. This disparity is not uncommon in literature because each evaluation

metric has a unique sensitivity to extreme values and should be used in conjunction with other metrics while taking the context of the particular problem and the data properties into account. (Chai and Draxler, 2014). Although the EWMA and Kalman smoothing on structural time series techniques were similar in their performance, the Kalman smoothing on structural time series model performed better based on RMSE and MAPE at a 25% rate of missingness. At a 35% rate of missingness, the Kalman smoothing on the ARIMA model (Kalman AR) was the best imputation technique based on MAPE, whereas Kalman ST was the best technique based on RMSE. The results obtained from the imputations also show that aside from the mean imputation, LOCF, KNN interpolation, linear interpolation, stine, and cubic spline interpolation are all gap dependent because they work relatively well for a small rate of missing values, and vice versa when the gap increase. A similar observation was reported by (Junninen et al., 2004) for simple interpolation techniques. However, interpolation with EWMA, SMA, Kal ST, and Kal AR yielded a consistent performance across the four missingness scenarios.

It is worth mentioning that each of the imputation techniques considered in our study either increased or decreased the autocorrelation (i.e., how the data from the past and future are connected) within the imputed dataset. The observations made from the analysis of the ACF plots showed that the mean imputation method, when used to replace missing values in any scenario of missingness reduced the autocovariance and the autocorrelation within the imputed data. In computing the variance and autocovariance, the imputed values cancel out, while increasing the degrees of freedom (sample size). This effect lowers the autocovariance and autocorrelation at

each lag after the first lag. The shape of the probability distribution of the imputed dataset using mean imputation also became significantly distorted as the rate of missingness increased. Other observations including underestimation of errors, disruption of the relationships between variables, and biased estimates have been associated with the mean imputation technique in univariate and multivariate data particularly when the data are not MCAR (Enders, 2010; Tan et al., 2013; Kang, 2013; Dong et al., 2014; van Buuren, 2018). On the other hand, imputed data based on Kal AR, Kal ST, EWMA, SMA, LOCF, KNN, stine, linear, and cubic spline interpolation increased the autocorrelation of the data. These changes in autocorrelation were also observed to impact the prediction performance of the ARIMA models. The higher the autocorrelation, the better the prediction performance (using the original train data as the reference). This pattern of ACF changes and prediction performance were also consistent to some extent in the LSTM models. This pattern was observed in datasets at all the scenarios of missingness except at 25% for cubic spline, and 35% rate for LOCF, KNN, cubic spline, linear, and stine interpolation imputed datasets also led to a reduced autocorrelation and yet, the ARIMA model obtained on them showed slightly better prediction on the test. In the LSTM models, none of the models obtained on the imputed dataset performed better than the original train data. Although many studies report caution against the use of LOCF for imputing missing data due to a lack of theoretical validity (Lachin, 2016; Kenward and Molenberghs, 2009, Moritz et al., 2015) and spurious results in longitudinal analysis (Lachin, 2016), our results showed that the ARIMA model obtained on the LOCF imputed data at 25% rate of missingness had the best prediction performance in terms of RMSE and second in terms of MAPE. The results across the different scenarios indicated that models trained on LOCF imputed

time series data can produce reliable time series predictions but the inadequate literature on time series forecasting with ARIMA and LSTM on imputed time series data, makes it difficult to corroborate and ascertain whether this outcome is a random result or not.

The majority of statistical literature on time series imputation focused on comparing the imputation performance of existing techniques (Walter et al., 2013; Moritz et al., 2015; Wijesekara & Liyanage, 2020). Our results have shown that in any scenario of missingness, the best imputation technique among the methods used does not guarantee that the time series model obtained on the corresponding data will yield better predictions than the others. As mentioned earlier, the mean imputation was the best with cubic spline being the worst imputation technique at a 10% rate of missingness. However, the model obtained on the spline imputed data yielded the best prediction performance (with the smallest RMSE and MAPE for ARIMA and only in RMSE for the LSTM model). Similarly, the EWMA and Kalman ST Imputation techniques performed better at a 15% rate of missingness but the ARIMA model obtained on the spline dataset and the LSTM obtained on the KNN dataset had the best prediction performance. These results are variable across the scenarios of missingness, methods for imputation, and model performance in terms of ARIMA and LSTM. However, there is a pattern that Kalman ST, Kalman AR, and EWMA methods for imputation perform relatively well for both ARIMA and LSTM forecasting.

In this study, the LSTM trained on the original dataset performed better than the ARIMA on the test data. This result is corroborated by (Boulmaiz, Guemoui, and Boutaghane, 2020) who showed that the LSTM algorithm is robust and can give reliable performance both on small and

large training data, but the performance of the LSTM significantly improve by increasing the size of the training data. While there is no rule of thumb regarding the minimum required data size for LSTM, the algorithm performs better on large sequential data with complex long-term dependencies (Hochreiter and Schmidhuber, 1997; Gers et al., 2000; Boulmaiz et al., 2020). In literature, many studies have reported improved performance of the LSTM than the ARIMA in financial time series data (Siami-Namini et al., 2018) and healthcare-expenditure time series data (Kaushik et al., 2020). These prediction improvements are usually accompanied by a more complex computational structure and a higher data processing time. On the other hand, the ARIMA model on several occasions in the imputed datasets outperformed the LSTM algorithm. It is not uncommon in literature for ARIMA models to outperform recurrent neural network architectures like LSTM (Yamak, Yujian, and Gadosey, 2019; Choy, Hoo & Khor, 2021; Kobiela, Krefta, Król & Weichbroth, 2022; Zhang, Song, Chen, Wang & Li, 2022). Several factors, including the changes in the autocorrelation, hyperparameters selected, and the overall size of the train data that contribute to this outcome.

In the literature (Azari, 2019; Choy, Hoo & Khor, 2021) ARIMA models perform well in short-term predictions and the results of this study further suggest that for time series forecasting on imputed data, ARIMA models may be preferable to LSTM because they are comparably easy to implement. Also, in terms of computational convergence, ARIMA models typically converge faster than LSTM models, as they are simpler and have fewer parameters to estimate. However, the convergence of LSTM models depends on several factors, such as the number of layers, the

Epoch size, the size of the hidden layers, and the length of the input sequences. Overall, this choice between ARIMA and LSTM models for time series forecasting depends on the specific characteristics of the data and the forecasting problem at hand. ARIMA models may be more appropriate for short-term forecasting of stationary data, while LSTM models may be better suited for modeling complex and non-linear relationships in long-term forecasting tasks (Gers et al., 2000).

Our study has some limitations, the first being the sample size. Time series with at least 50 observations are required to obtain a reliable estimate of the autocorrelation function in the ARIMA methodology (Box et al., 2015). This sample size may not be ideal for an LSTM algorithm which by design is for handling long-term dependencies within sequential data (Hochreiter and Schmidhuber, 1997; Yu et al., 2019). In our study, missing data were generated under the MCAR mechanism. The assumption of MAR implies that the missing data are related to an observed variable. However, in the univariate time series data, there are no other variables other than the time variable (which is implicit). For this reason, it is practically unlikely for the blood pressure readings of any individual to be missing at 15% or 35% simply because of the monitoring time. In addition, the study focused on MCAR because, in practice, it is empirically impossible to distinguish data that are MAR or MCAR or both in univariate data. Since these assumptions assume a random variable with a probability distribution, several factors or mechanisms unrelated to the missing values can make the data incomplete. The factors underlying the missingness in such circumstances can be ignored (Little and Rubin, 2019). For these reasons, existing literature

assumes that the missing data are either MAR or MCAR (Moritz et al., 2015; Twumasi-Ankrah et al., 2019; Wijesekara & Liyanage, 2020). The effects of these imputation methods on non-ignorable mechanisms (MNAR) would be a great addition to existing literature. However, the small sample size, nature, and distribution of the data limited us from choosing a threshold value for simulating data that are MNAR for the various scenarios studied.

5.1 Conclusion

Our study showed that imputation techniques used on univariate time series either increased or lowered the autocorrelation within the data. These changes in the time series data impacted the prediction performance of the time series forecasting algorithms. Simple interpolation methods like spline, linear, and stine imputation methods are recommended over mean imputation when the gap of missing data is small because they can be effective, and the models obtained on imputed data can give reasonably better predictions than mean imputed data. Overall, Kalman smoothing on Structural time series, Exponentially weighted moving average, and Kalman smoothing on ARIMA techniques not only performed well at every rate of missingness, but the models obtained on the imputed data can also give consistent predictions. Specifically, ARIMA is recommended over LSTM prediction on imputed datasets because it is simple to execute and tends to perform better on imputed data with higher autocorrelation, even though imputed datasets with higher autocorrelation to some extent were also observed to yield better predictions in the LSTM architecture.

REFERENCES

Albano, G., Rocca, M. L., & Perna, C. (2017, February). On the Imputation of Missing Values Univariate $$ PM_ {10} $$ Time Series. In International Conference on Computer Aided Systems Theory (pp. 12-19). Springer, Cham.

Angelini, L., Maestri, R., Marinazzo, D., Nitti, L., Pellicoro, M., Pinna, G. D. ... & Tupputi, S. A. (2007). Multiscale analysis of short term heart beat interval, arterial blood pressure, and instantaneous lung volume time series. Artificial intelligence in medicine, 41(3), 237-250.

Azari, A. (2019). Bitcoin price prediction: An ARIMA approach. arXiv preprint arXiv:1904.05315.

Batista, G. E., & Monard, M. C. (2002). A study of K-nearest neighbour as an imputation method. His, 87(251-260), 48.

Bokde, N., Beck, M. W., Álvarez, F. M., & Kulat, K. (2018). A novel imputation methodology for time series based on pattern sequence forecasting. Pattern recognition letters, 116, 88-96.

Boulmaiz, T., Guermoui, M., & Boutaghane, H. (2020). Impact of training data size on the LSTM performances for rainfall–runoff modeling. *Modeling Earth Systems and Environment*, *6*, 2153-2164.

Box, G. E, & Jenkins, G.M.(1970). Time Series Analysis: Forecasting and Control. San Francisco, Holdan-Day.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

Brownlee, J. (2018). Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.

Caillault, É. P., Lefebvre, A., & Bigand, A. (2020). Dynamic time warping-based imputation for univariate time series data. Pattern Recognition Letters, 139, 139-147.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). Geoscientific model development discussions, 7(1), 1525-1534.

Chaudhry, A., Li, W., Basri, A., & Patenaude, F. (2019). A method for improving imputation and prediction accuracy of highly seasonal univariate data with large periods of missingness. Wireless Communications and Mobile Computing, 2019.

Chen, T. B., & Soo, V. W. (1996, June). A comparative study of recurrent neural network architectures on learning temporal sequences. In Proceedings of International Conference on Neural Networks (ICNN'96) (Vol. 4, pp. 1945-1950). IEEE.

Choy, Y. T., Hoo, M. H., & Khor, K. C. (2021, September). Price Prediction Using Time-Series Algorithms for Stocks Listed on Bursa Malaysia. In 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS) (pp. 1-5). IEEE.

Chuah, M. C., & Fu, F. (2007, August). ECG anomaly detection via time series analysis. In International Symposium on Parallel and Distributed Processing and Applications (pp. 123-135). Springer, Berlin, Heidelberg.

Cowpertwait, P. S., & Metcalfe, A. V. (2009). Introductory time series with R. Springer.

Cox, D. R. (1972). Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 187-202.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-22.

Diggle, P., & Kenward, M. G. (1994). Informative drop-out in longitudinal data analysis. Journal of the Royal Statistical Society: Series C (Applied Statistics), 43(1), 49-73.

Diggle, P., Liang, K. Y., & Zeger, S. L. (1994). Longitudinal data analysis. New York: Oxford University Press, 5, 13.

Dong, C., Shao, C., Richards, S. H., & Han, L. D. (2014). Flow rate and time mean speed predictions for the urban freeway network using state space models. Transportation Research Part C: Emerging Technologies, 43, 20-32.

Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179-211.

Enders, C. K. (2010). Applied missing data analysis. Guilford press.

Gers, F. A., & Schmidhuber, J. (2000, July). Recurrent nets that time and count. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (Vol. 3, pp. 189-194). IEEE.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. Annual review of psychology, 60(1), 549-576.

Granger, C. W. J., & Newbold, P. (2014). Forecasting economic time series. Academic press.

Groppelli, A., Omboni, S., Parati, G., & Mancia, G. (1992). Evaluation of noninvasive blood pressure monitoring devices Spacelabs 90202 and 90207 versus resting and ambulatory 24-hour intra-arterial blood pressure. Hypertension, 20(2), 227-232.

Hamzah, F. B., Mohd Hamzah, F., Mohd Razali, S. F., Jaafar, O., & Abdul Jamil, N. (2020). Imputation methods for recovering streamflow observation: A methodological review. Cogent Environmental Science, 6(1), 1745133.

Harel, O., & Boyko, J. (2013). Missing data: should we care?. American journal of public health, 103(2), 200-201.

Harel, O., Pellowski, J., & Kalichman, S. (2012). Are we missing the importance of missing values in HIV prevention randomized clinical trials? Review and recommendations. AIDS and Behavior, 16(6), 1382-1393.

Helfenstein, U. (1996). Box-Jenkins modelling in medical research. Statistical Methods in Medical Research, 5(1), 3-22.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. Journal of statistical software, 27, 1-22.

Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on 1/26/2023.

Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In Eighth Annual Conference of the Cognitive Science Society, 1986 (pp. 513-546).

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., & Kolehmainen, M. (2004). Methods for imputation of missing values in air quality data sets. Atmospheric Environment, 38(18), 2895-2907.

Kalekar, P. S. (2004). Time series forecasting using holt-winters exponential smoothing. Kanwal Rekhi school of information Technology, 4329008(13), 1-13.

Kang, H. (2013). The prevention and handling of the missing data. Korean journal of anesthesiology, 64(5), 402-406.

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.

Kaushik, S., Choudhury, A., Sheron, P. K., Dasgupta, N., Natarajan, S., Pickett, L. A., & Dutt, V. (2020). AI in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. Frontiers in big data, 3, 4.

Kenward, M. G., & Molenberghs, G. (2009). Last observation carried forward: a crystal ball?. Journal of biopharmaceutical statistics, 19(5), 872-888.

Kobiela, D., Krefta, D., Król, W., & Weichbroth, P. (2022). ARIMA vs LSTM on NASDAQ stock exchange data. Procedia Computer Science, 207, 3836-3845.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling (Vol. 26, p. 13). New York: Springer.

Kulesh, M., Holschneider, M., & Kurennaya, K. (2008). Adaptive metrics in the nearest neighbours method. *Physica D: Nonlinear Phenomena*, *237*(3), 283-291.

Kuligowski, A.T, Gharibvand, L (2020). Dealing with missing data in Epidemiological and Clinical Research. SAS Global Forum 2020, 5072-2020. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://support.sas.com/resources/papers/proceedings20/5072-2020.pdf#:~:text=Listwise%20deletion%20Complete%20case%20analysis%20%28likewise%20deletion%29%20is,unbiased%20estimates%20of%20means%2C%20variances%20and%20regression%20weights

Lachin, J. M. (2016). Fallacies of last observation carried forward analyses. Clinical trials, 13(2), 161-168.

Laird, N. M. (1988). Missing data in longitudinal studies. Statistics in medicine, 7(1-2), 305-315.

Lepot, M., Aubin, J. B., & Clemens, F. H. (2017). Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. *Water*, *9*(10), 796.

Li, L., Su, X., Zhang, Y., Lin, Y., & Li, Z. (2015). Trend modeling for traffic time series analysis: An integrated study. IEEE Transactions on Intelligent Transportation Systems, 16(6), 3430-3439.

Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018). Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5457-5466).

Little, R. J. (1992). Regression with missing X's: a review. Journal of the American statistical association,87(420), 1227-1237.

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Li-wei, H. L., Adams, R. P., Mayaud, L., Moody, G. B., Malhotra, A., Mark, R. G., & Nemati, S. (2014). A physiological time series dynamics-based approach to patient monitoring and outcome prediction. IEEE journal of biomedical and health informatics, 19(3), 1068-1076.

Li-wei, H. L., Nemati, S., Adams, R. P., Moody, G., Malhotra, A., & Mark, R. G. (2013, July). Tracking progression of patient state of health in critical care using inferred shared dynamics in physiological time series. In 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 7072-7075). IEEE.

Longford, N. T. (2007). Studying human populations: An advanced course in statistics. Springer Science & Business Media.

Mack, C., Su, Z., & Westreich, D. (2018). Managing missing data in patient registries: addendum to registries for evaluating patient outcomes: a user's guide.

Maestre, G. E., Pino-Ramírez, G., Molero, A. E., Silva, E. R., Zambrano, R., Falque, L., ... & Sulbarán, T. A. (2002). The Maracaibo Aging Study: population and methodological issues. Neuroepidemiology, 21(4), 194-201.

Moritz, S., & Bartz-Beielstein, T. (2017). imputeTS: time series missing value imputation in R. R J., 9(1), 207.

Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., & Stork, J. (2015). Comparison of different methods for univariate time series imputation in R. arXiv preprint arXiv: 1510.03924.

Mozer, M. C. (1994). Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. Connection Science, 6(2-3), 247-280.

Noor, N. M., Al Bakri Abdullah, M. M., Yahaya, A. S., & Ramli, N. A. (2015). Comparison of linear interpolation method and mean method to replace the missing values in environmental data set. In Materials Science Forum (Vol. 803, pp. 278-281). Trans Tech Publications Ltd.

Norazian, M. N., Shukri, Y. A., Azam, R. N., & Al Bakri, A. M. M. (2008). Estimation of missing values in air pollution data using single imputation techniques. ScienceAsia, 34(3), 341-345.

Penfold, R. B., & Zhang, F. (2013). Use of interrupted time series analysis in evaluating health care quality improvements. Academic pediatrics, 13(6), S38-S44.

Rao, R. K., & Yeragani, V. K. (2001). Decreased chaos and increased nonlinearity of heart rate time series in patients with panic disorder. Autonomic Neuroscience, 88(1-2), 99-108.

Roderick J.A.Little, Donald B.Rubin (2002). Statistical Analysis with Missing Data, second edition. Accessed; https://onlinelibrary.wiley.com/doi/book/10.1002/9781119013563# Retrieved January 10, 2022.

Rubin, D. B. (1976). Inference and missing data. Biometrika, 63(3), 581-592.

Rubin, D. B. (2004). Multiple imputation for nonresponse in surveys (Vol. 81). John Wiley & Sons.

Schafer, J. L. (1997). Analysis of incomplete multivariate data. CRC press.

Shumway, R. H., & Stoffer, D. S. (2019). Time series: a data analysis approach using R. Chapman and Hall/CRC.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, December). A comparison of ARIMA and LSTM in forecasting time series. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1394-1401). IEEE.

Stineman, R. W. (1980). A consistently well-behaved method of interpolation. Creative Computing, 6(7), 54-57.

Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y. J., & Li, F. (2013). A tensor-based method for missing traffic data completion. Transportation Research Part C: Emerging Technologies, 28, 15-27.

Torgo, L. (2016) Data Mining using R: learning with case studies, second edition, Chapman & Hall/CRC (ISBN-13: 978-1482234893).

Twumasi-Ankrah, S., Odoi, B., Adoma Pels, W., & Gyamfi, E. H. (2019). Efficiency of imputation techniques in univariate time series.

Vacek, P. M., & Ashikaga, T. (1980). An examination of the nearest neighbor rule for imputing missing values. Proceeedings of the Statistical Computing Section, American Statistical Association.

Van Buuren, S. (2018). Flexible imputation of missing data. CRC press.

Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. Journal of statistical software, 45, 1-67.

Van Buuren, S., Boshuizen, H. C., & Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. Statistics in medicine, 18(6), 681-694.

Velicer, W. F., & Colby, S. M. (2005). A comparison of missing-data procedures for ARIMA time-series analysis. Educational and Psychological Measurement, 65(4), 596-615.

Walter, Y. O., Kihoro, J. M., Athiany, K. H. O., & Kibunja, H. W. (2013). Imputation of incomplete non-stationary seasonal time series data. Math. Theory Model, 3, 142-154.

Wei, W. W. (2006). Time series analysis. In The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2.

Westreich, D. (2012). Berkson's bias, selection bias, and missing data. Epidemiology (Cambridge, Mass.), 23(1), 159.

Wiens, J., Horvitz, E., & Guttag, J. (2012). Patient risk stratification for hospital-associated c. diff as a time-series classification task. Advances in Neural Information Processing Systems, 25.

Wijesekara, W. M. L. K. N., & Liyanage, L. (2020). Comparison of imputation methods for missing values in air pollution data: Case study on Sydney air quality index. In Advances in Information and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 2 (pp. 257-269). Springer International Publishing.

Yamak, P. T., Yujian, L., & Gadosey, P. K. (2019, December). A comparison between arima, lstm, and gru for time series forecasting. In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (pp. 49-55).

Yang, Y. (2012). Modelling nonlinear vector economic time series. Department of Economics and Business, Business and Social Sciences, Aarhus University, Aarhus, Denmark, 28, 29-30.

Yeragani, V. K. (1995). Heart rate and blood pressure variability: implications for psychiatric research. Neuropsychobiology, 32(4), 182-191.

Yeragani, V. K., Collins, H. L., Rao, K. R., Rodenbaugh, D. W., & DiCarlo, S. E. (2003). Decreased chaos after exercise in cardiac output time series of rats: a preliminary report. Nonlinear analysis: real world applications, 4(2), 307-316.

Yozgatligil, C., Aslan, S., Iyigun, C., & Batmaz, I. (2013). Comparison of missing value imputation methods in time series: the case of Turkish meteorological data. Theoretical and applied climatology, 112(1), 143-167.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270.

Zakaria, N. A., & Noor, N. M. (2018). Imputation methods for filling missing data in urban air pollution data formalaysia. Urbanism. Arhitectura. Constructii, 9(2), 159.

Zeger, S. L., Irizarry, R., & Peng, R. D. (2006). On time series analysis of public health and biomedical data. Annu. Rev. Public Health, 27, 57-79.

Zhang, R., Song, H., Chen, Q., Wang, Y., Wang, S., & Li, Y. (2022). Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China. Plos one, 17(1), e0262009.

BIOGRAPHICAL SKETCH

Nicholas Niako attended Osei Kyeretwie Senior High School in the Ashanti Region of Ghana, where he completed his West African Senior School Certificate Examination (WASSCE) certificate in 2013. Subsequently, he had his bachelor's degree in Statistics from the Kwame Nkrumah University of Science and Technology and graduated with first-class honors in 2018. After his bachelor's, he worked as a research assistant for two years at the Kwame Nkrumah School of Medicine and Dentistry from 2019 to 2021. Later in the fall of 2021, he was admitted into the Master of Science in Mathematics with a concentration in Statistics program. That same year, he was nominated and awarded the Dean of College of Science prestigious scholarship award at the University of Texas Rio Grande Valley. While at the University of Texas Rio Grande Valley, he worked as a Graduate teaching assistant for the first year and as a graduate research assistant for his second year. His research interests are in the application of statistical and machine-learning methods to improve public health outcomes. Nicholas earned his master of science in Mathematics from the University Of Texas Rio Grande Valley in May 2023.

To contact Nicholas, feel free to message him at: nicholasniako@gmail.com.