YAP YI XIAN

B.Sc. (Hons) Statistical Computing and Operations Research

2024

FORECASTING DATA WITH LONG MULTI-SEASONAL PERIODS IN THE ARIMA MODEL USING DISCRETE FOURIER TRANSFORM REGRESSORS

YAP YI XIAN

BACHELOR OF SCIENCE (HONS) STATISTICAL COMPUTING AND OPERATIONS RESEARCH

FACULTY OF SCIENCE

UNIVERSITY TUNKU ABDUL RAHMAN

OCTOBER 2024

FORECASTING DATA WITH LONG MULTI-SEASONAL PERIODS IN THE ARIMA MODEL USING DISCRETE FOURIER TRANSFORM REGRESSORS

By

YAP YI XIAN

A project report submitted to the

Department of Physical and Mathematical Science

Faculty of Science

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements for the degree of

Bachelor of Science (Hons) Statistical Computing and Operations Research

October 2024

ABSTRACT

FORECASTING DATA WITH LONG MULTI-SEASONAL PERIODS IN THE ARIMA MODEL USING DISCRETE FOURIER TRANSFORM REGRESSORS

YAP YI XIAN

Time series data with multiple seasonalities often appear in data observed at high frequency. For instance, daily observed data may exhibit multiple seasonal patterns due to the combination of weekly, monthly, or annual periodicities. Traditional forecasting methods, such as the Autoregressive Integrated Moving Average (ARIMA) model, face significant challenges when dealing with long, multiple seasonal cycles. Specifically, the ARIMA model fitting function may suffer from memory insufficiency when handling long seasonal periods and is generally designed to handle univariate time series with a single seasonal pattern. To address these challenges, this study proposed a novel forecasting approach by integrating Multiple Seasonal Trend decomposition using Loess (MSTL), Discrete Fourier Transform (DFT), and ARIMA. Firstly, the MSTL algorithm was employed to decompose the time series into their constituent components. For the seasonal components, the properties of the Discrete Fourier Transform were utilized to serve as regressors in the ARIMA framework. The non-seasonal components, including the trend and remainder, were fitted using the ARIMA model. The proposed MSTL-DFT-ARIMA approach was then compared with the TBATS model, a known benchmark for handling multiple seasonalities.

From the results, MSTL-DFT-ARIMA approach outperforms TBATS in both forecast accuracy and computational efficiency. Hence, the integration of MSTL, DFT, and ARIMA provides a promising alternative for managing time series data with long multi-seasonal periods.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my project supervisor, Dr. Lem Kong Hoong, for providing valuable insights and support throughout this project. Dr Lem's extensive knowledge and clear guidance were instrumental in the completion of my research. I am incredibly thankful for his patience and the generous manner in which he shared his expertise with me along the way.

Besides, I would also like to extend my heartfelt thanks to my family and friends. Their belief in me, coupled with the mental support they provided, were crucial and meant a lot to me, especially during the challenging times of my studies. Their encouragement kept me motivated and focused on achieving my goals.

This project would not have been possible without the support and guidance of these individuals. Thank you all for your invaluable contributions.

DECLARATION

I hereby declare that the project report is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

YAP YI XIAN

APPROVAL SHEET

This project report entitled "**FORECASTING DATA WITH LONG MULTI-SEASONAL PERIODS IN THE ARIMA MODEL USING DISCRETE FOURIER TRANSFORM REGRESSORS**" was prepared by YAP YI XIAN and submitted as partial fulfilment of the requirements for the degree of Bachelor of Science (Hons) Statistical Computing and Operations Research at Universiti Tunku Abdul Rahman.

Approved by:

_____

(Dr. Lem Kong Hoong)

Supervisor

Department of Physical and Mathematical Science

Faculty of Science

Universiti Tunku Abdul Rahman

Date: 3 Sept 2024

FACULTY OF SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 3 September 2024

PERMISSION SHEET

It is hereby certified that **YAP YI XIAN** (ID No: **20ADB05640**) has completed this final year project entitled "**FORECASTING DATA WITH LONG MULTI-SEASONAL PERIODS IN THE ARIMA MODEL USING DISCRETE FOURIER TRANSFORM REGRESSORS**" under the supervision of Dr. Lem Kong Hoong (Supervisor) from the Department of Physical and Mathematical Science, Faculty of Science.

I hereby give permission to the University to upload the softcopy of my final year project in pdf format into the UTAR Institutional Repository, which may be made accessible to the UTAR community and public.

Yours truly,

_____

(YAP YI XIAN)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACD	Absolute Coverage Difference
ACF	Autocorrelation Function
ADMM	Alternative Direction Multiplier Method
ADMM	Alternative Direction Multiplier Method
AIC	Akaike Information Criterion
AICc	corrected Akaike Information Criterion
ANN	Artificial Neural Network
ARIMA	Auto-Regressive Integrated Moving Average
BATS	Exponential Smoothing State Space model with Box-Cox transformation, ARMA errors, Trend and Seasonal Components
BIC	Bayesian Information Criterion
DARIMA	Distributed Auto-Regressive Integrated Moving Average
DARIMA_SA	Simply Averaging the estimated parameters for split subseries when implementing Distributed Auto-Regressive Integrated Moving Average
DAX	Deutscher Aktien Index
DFT	Discrete Fourier Transform
DGP	Data-Generating Process
DHR	Dynamic Harmonic Regression
DIMS	Discrete-Interval Moving Seasonalities
DS	Double Seasonal

DSARIMAX	Double Seasonal Autoregressive Integrated Moving Average with Exogenous Factors
ETS	Exponential Smoothing
FFT	Fast Fourier Transform
GBR	Gradient Boosting Regressor
HW	Holt-Winters
IDFT	Inverse Discrete Fourier Transform
LOESS	Locally Weighted Scatterplot Smoothing
LSTM	Long Short-term Memory
MA	Moving Average Process
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron
MS	Multiple Seasonal
MSE	Mean Square Error
MSTL	Multiple Seasonal Trend decomposition using Loess
PACF	Partial Autocorrelation Function
RF	Random Forest
RMSE	Root Mean Square Error
RobustSTL	Robust Seasonal-Trend Decomposition Algorithm
RW	Random Walk
SARIMA	Seasonal Autoregressive Integrated Moving Average
SEATS	Seasonal Extraction in ARIMA Time Series

STL	Seasonal Trend decomposition using Loess
TBATS	Trigonometric Exponential Smoothing State Space model with Box-Cox transformation, ARMA errors, Trend and Seasonal Components

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Time series refers to a set of ordered data points collected over a time interval (Umer et al., 2022). Time series analysis aims to develop a mathematical model that can forecast future values based on past observations. Seasonality is said to be present in a time series if there is a repeating pattern over a fixed period. Moreover, time series data can be observed over various lengths, ranging from short intervals like hours or days to much longer periods like years. When data are observed over longer time intervals, they often exhibit complex seasonalities, where multiple overlapping seasonal patterns emerge. For instance, daily data might display both weekly and annual patterns, whereas hourly data can be even more complex, exhibiting daily, weekly, and annual seasonalities (Hyndman and Athanasopoulos, 2021). These complex seasonal patterns are increasingly common in real-world scenarios. For example, daily minimum temperatures were recorded over centuries and stock indices were recorded every minute for several months (Wang et al., 2023).

The Autoregressive Integrated Moving Average (ARIMA) model is a traditional time series analysis tool developed by Box and Jenkins (1976). The model assumes that future values in a time series can be predicted from its past values. ARIMA's strength lies in its ability to handle various types of data models such as non-stationary and seasonal patterns (Rizkya et al., 2019). Furthermore, it is a linear model that is good at handling linear relationships between variables.

However, it struggles with capturing more complex and nonlinear patterns such as sudden shocks (Ridwan, Sadik and Afendi, 2023). The requirement for applying ARIMA is the time series needs to be stationary, meaning that it has to hold a constant variance, covariance, and mean over time (Schaffer, Dobbins and Pearson, 2021). While ARIMA performs well in short to medium-term forecasting, its accuracy tends to decrease over longer forecast horizons. This is because the model's reliance on past data and its assumptions about linearity and stationarity become less reliable, leading to less accurate predictions in the long term (Liu, 2024).

On top of that, Multiple Seasonal-Trend decomposition using LOESS (MSTL) represents a multi-seasonal decomposition approach introduced by Bandara, Hyndman and Bergmeir (2022). This method is an extension of the classical Seasonal Trend decomposition using LOESS (STL) algorithm which uses Locally Estimated Scatterplot Smoothing (LOESS) to extract seasonal components from time series data (Manani, 2022). LOESS can be regarded as a non-parametric method that fits multiple local regressions to subsets of data points. Hence, it allows STL to capture complex and nonlinear more effectively than traditional linear methods by reducing the influence of noise and outliers (Jacoby, 2000). In context, the MSTL algorithm works by iteratively applying the STL procedure to detect and separate seasonal and non-seasonal components (Bandara, Hyndman and Bergmeir, 2022). The MSTL simplifies the time series analysis by breaking down time series into trend, seasonality, and residual. This breakdown aids in understanding the underlying patterns and making predictions. However, MSTL does not automatically account for

trading day or calendar variations, and there is limited information on specific scenarios where the model might underperform (Arneric, 2021).

In Fourier analysis, a time series is often regarded as a mix of signals with different frequencies. The Fourier transform is a mathematical technique devised by a French mathematician from the 18th century, namely Jean-Baptiste Joseph Fourier (IEEE Pulse, 2016). It breaks down a time series into its individual frequency components and identifies the most important frequencies in the data (Keil et al., 2022). Some frequency components may appear as random, high-frequency signals, which often represent noise. These components can be filtered out by setting an appropriate threshold. Furthermore, the Fourier transform shifts the time series into the frequency domain. While the time domain representation only provides direct information about the values of the signal over time, the frequency domain representation offers a different perspective that can often be more insightful for certain types of analysis (Parsons, Boonman and Obrist, 2000). However, its limitation is that it cannot pinpoint exactly when a particular frequency occurs, which can be a drawback in analyzing signals with time-varying frequencies (Sakhuja, 2024).

On the other hand, the Trigonometric, Box-Cox, ARMA, Trend, Seasonal (TBATS) model, developed by De Livera, Hyndman, and Snyder (2011), is an advanced approach designed for handling complex seasonal time series data. It is capable of handling seasonal patterns that are non-integer, non-nested, and of long periods, without imposing any constraints on the type of seasonality. Thus, it is possible for long-term forecasts (Nadeem, 2021). In addition, as a state-

space model, TBATS can handle a larger parameter space to yield accurate forecasts (Munim, 2022). However, this flexibility comes at the cost of slower computation. In addition, TBATS requires evaluating numerous model candidates, which can be time-consuming, especially for multiple parallel time series (Hyndman and Athanasopoulos, 2021).

1.2 Problem Statement

Forecasting time series data with long seasonal periods poses significant challenges. One major issue is that ARIMA model fitting functions in software often struggle with memory insufficiency when dealing with long seasonal periods. This difficulty arises because the seasonal differencing process requires comparing current data with observations from many time points in the past. Such comparisons are computationally intensive and can lead to running out of memory, especially when the seasonal period exceeds 200 units (Wang et al., 2023). Additionally, estimating model coefficients via maximum likelihood estimation adds further complexity, as it involves solving numerous linear systems, which further strains computational resources (Hyndman and Athanasopoulos, 2021).

Moreover, most of the traditional methods are generally unable to account for multiple seasonal patterns (Hyndman and Athanasopoulos, 2021). For example, the traditional ARIMA models are designed to handle a single type of seasonality. Furthermore, Seasonal Autoregressive Integrated Moving Average (SARIMA) extends ARIMA to include seasonality but it can only handle one

seasonal pattern at a time (Williams, Sperl and Chung, 2023). On the other hand, even though Exponential Smoothing (ETS) is a popular time series approach for handling single seasonality, it performs poorly when applied to time series containing multiple seasonal patterns (Naim, Mahara and Idrisi, 2018).

To tackle these challenges, this study proposed a novel algorithm to improve time series forecasting with long seasonal periods and multiple seasonal patterns. It will be discussed in greater detail in Chapter 3.

1.3 Research Objectives

The objectives of this study are:

1. To develop a model capable of forecasting long multi-seasonal time series data by integrating the MSTL, ARIMA, and Discrete Fourier Transform.
2. To compare the forecasting performance of the proposed method with TBATS in terms of computational efficiency and forecasting accuracy.

1.4 Significance of the Study

Today's technology has made data collection easy and frequent, hence, long and multi-seasonal data can be commonly encountered in real life. Therefore, there is a need for a fast and effective model to handle these complexities. This study is significant because it explores the development of a model specifically designed to manage both long and multi-seasonal data simultaneously. Moreover, this model is expected to address the limitations of ARIMA in handling such data, while also improving the prediction accuracy and reducing computational time. The insights gained from this study could also pave the way for future research, by offering a preliminary idea for developing an alternative to multi-seasonal data analysis.

1.5 Outline

The remainder of this thesis is structured as follows:

Chapter 2 reviews the literature relevant to this study. Chapter 3 elaborates on the research methods employed. Chapter 4 presents and discusses the results. Finally, Chapter 5 concludes the study and recommends potential extensions for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Past Approaches to Multi-seasonal Time Series

There is a large body of literature that accommodates multiple seasonal patterns. Among the early studies, Harvey and Koopman (1993) developed an unobserved components method to model a time series with two seasonal periods, daily and weekly. Specifically, they used time-varying periodic splines within a state-space framework to model the seasonal components. Furthermore, Taylor (2003) extended the simple Holt-Winters model to capture seasonalities by incorporating multiple seasonal components into the linear framework of the model. This double seasonal exponential smoothing method allowed one seasonal cycle to be nested within another, but it assumed that the intra-day cycle was identical for every day of the week.

Next, the Multiple Seasonal (MS) process model was introduced by Gould et al. (2008) to model multiple seasonalities. It was built upon the exponential smoothing techniques. The MS model was capable of handling additive and multiplicative seasonal effects. It also accounted for public holidays and missing data in the time series. However, they pointed out that the MS model must identify the recurring patterns across different cycles for multiple seasonal data, thus, this approach might be time-consuming. Despite this limitation, the model outperformed the traditional approaches such as Taylor's Double Seasonal (DS) and Holt-Winters (HW) models, in forecasting utility demand and vehicle flows across various prediction horizons. However, most of these techniques were

prone to issues with optimization and overparameterization, as well as unable to capture complex seasonalities in time series data (De Livera, Hyndman and Snyder, 2011).

To tackle the challenges in modelling complex seasonal patterns, De Livera, Hyndman and Snyder (2011) developed the TBATS model, an extension of the earlier BATS framework proposed by De Livera (2010). The BATS model was designed to manage multiple seasonalities in time series data. However, it struggled with capturing complex and high-frequency seasonal patterns. Therefore, De Livera et al. (2011) proposed the TBATS model, which integrated trigonometric functions into the BATS framework, thus, creating a more parsimonious and flexible version of the innovation state-space modelling framework. This framework could model multiple seasonal periods, calendar effects, or even non-integer seasonality. Hence, it can accommodate time series with complex seasonal patterns.

A study conducted by Vieira, Sousa, and Dória-Nóbrega (2023) also found that the TBATS model was strong in handling non-integer frequencies, such as the 365.25-day annual seasonality. However, despite its strengths in managing complex seasonalities, TBATS demonstrated lower predictive power than SARIMA and ETS in scenarios consisting of holiday effects. This might be due to the inability of TBATS to include external variables for holidays or other significant events that can influence a time series. As a result, the model might miss important variations that could improve the accuracy of the forecast,

particularly in cases where holidays or other external factors hold a significant role in the behaviour of the time series (Hyndman, 2017).

To address this gap, Taylor and Letham (2018) introduced the Prophet model to decompose data into multiple components, then, fit each one separately, and combine them to generate forecasts. Prophet handled multiple non-integer seasonal periods through the use of the Fourier series. Additionally, Prophet included a component specifically designed to model holidays or special events, which added a layer of precision that many other models lack. However, the Prophet approach struggled with multidimensional data and did not account for scale information. On top of that, it tends to function more as a curve-fitting model instead of fully capturing the temporal dependency structure within the data (Sousa, Tom and Moreira, 2022).

Apart from that, Lakshmanan and Das (2017) introduced a two-stage framework to model time series with multiple levels of seasonalitiies. The first stage focused on fitting a regression model to capture lower frequency seasonalities, such as daily and weekly data. It used dummy variables to represent days of the week and incorporated weather-related covariates to account for annual patterns. This regression model helped to remove the major seasonal components from the data. If the regression model's residuals still contain significant patterns, methods like ARIMA or TBATS were applied to further refine the fit. In the second stage, the framework addressed high-frequency components, such as hourly or minute-level patterns. This involved using the residuals from the first stage and applying classical decomposition

methods or polynomial functions to estimate the high-frequency seasonality. The study revealed that when dealing with time series that have multiple levels of seasonalities, the two-stage method yielded superior computational efficiency and accuracy compared to TBATS. Additionally, this approach could avoid overfitting and inconsistency issues.

Another effective approach to handle multi-seasonal time series is through time series decomposition. The STL method developed by Cleveland et al. (1990) emerged as the most commonly used decomposition method. However, Hyndman and Athanasopoulos (2021) pointed out that STL was originally designed to handle only a single seasonality and did not account for calendar effects or special events. In reference to Moon, Lee and Song (2022), only a few traditional statistical models could decompose time series with multiple seasonalities, such as *x*11, and SEATS. Nonetheless, they revealed that these methods were limited in their ability to capture only quarterly and monthly periodicities. To overcome these limitations, Bandara, Hyndman and Bergmeir (2022) proposed a Multiple Seasonal Trend decomposition using Loess (MSTL) approach, which iteratively applied the STL to decompose time series with more than one seasonality. This method first separated the time series into individual components such as trends, residuals, and seasonalities. Then, each of these components was modelled independently. Lastly, the time series was fitted into the forecasting model that was capable of handling time series with multiple seasonalities. Hence, the complexity of the model was reduced compared to forecasting the entire time series in its entirety (Bandara, Hyndman and Bergmeir, 2022; Moon, Lee and Song, 2022). Next, Trull, García-Díaz and

Peiró-Signes (2022) also enhanced the STL methods by integrating discrete-interval moving seasonalities (DIMS) to account for special events.

Moreover, MSTL proved its capability in analyzing high-frequency data, as shown in the study by Arneric (2021), where it was applied to trading volume data for the DAX stock index. The 15-minute trading volume observations over five years were decomposed into monthly, hourly, and daily seasonal components. This decomposition offered deeper insights into trading behaviours. For instance, the monthly seasonality highlighted higher trading in January and downturns in May and August, daily seasonality highlighted the trading activity increased towards the end of the week, whereas hourly seasonality was most dominant and consistent, in which the trading volumes peaked at the beginning and end of the trading day. Notably, the decomposition captured over 50% of the variations in trading volume through these multi-seasonal patterns.

MSTL has found widespread application across various domains as it enabled each component to be modelled and forecast independently. For instance, in the study by Nan, Zhu, and Ma (2023), MSTL was a critical element in predicting wireless traffic in cellular networks. By breaking down cellular traffic data into daily and weekly seasonalities, trends, and residuals, MSTL enabled the separate modelling of each component. The seasonal components were refined using a global model that employed clustering techniques and a distance-assisted attention mechanism to effectively capture both common and unique

spatiotemporal patterns across different cells. This approach resulted in superior performance in terms of forecasting accuracy and model efficiency.

Similarly, Krechiem and Khadir (2023) utilized the MSTL method to analyze electricity consumption in Algeria by decomposing the data into daily and weekly seasonalities, trends, and residual components. This decomposition allowed them to understand the underlying structure of electricity demand. The residual component that captured high-frequency fluctuations, was modelled using Artificial Neural Network (ANN), specifically the Multilayer Perceptron (MLP) and Long Short-term Memory (LSTM) types. The modelled components were then combined to produce a final load forecast. The results of their study demonstrated that MSTL, in conjunction with ANN models, provided more accurate short-term load forecasting compared to classical predictive approaches. This was evidenced by lower RMSE and MAPE values.

However, most of the seasonal-trend decomposition algorithms struggled with a high computational burden and demand a huge amount of data when multiple seasonalities and long-time series exist. To tackle this issue, Yang et al. (2021) explored a novel model that simplified the process. They proposed a multi-scale seasonal-trend decomposition method that first down-sampled the time series to a lower resolution and then transformed it into a time series with a single seasonal component. Hence, the existing decomposition algorithms could be applied more directly to estimate the trend and seasonal components. To further boost efficiency, the model incorporated an optimization technique using the Alternative Direction Multiplier Method (ADMM), which helped recover the

high-resolution components effectively. When this method was tested on synthetic data, it showed much lower MSE, and shorter computation times compared to traditional methods like STL and RobustSTL.

2.2 Related Studies on Research Topic

The use of Fourier analysis in time series forecasting has proven effective in overcoming the limitations of other methods in capturing seasonality (Lye et al., 2009). Therefore, Fourier analysis has been extensively applied in various disciplines, including physics, economics, engineering, and seismology.

In the context of electricity demand forecasting, McLoughlin, Duffy, and Conlon (2013) demonstrated how Fourier analysis could replicate the temporal characteristics of demand profiles at the level of individual dwellings. They found that Fourier transforms managed to replicate the temporal characteristics of demand profiles using only half the number of variables. However, the findings suggested that while Fourier transforms were valuable in scenarios where demand was more evenly distributed across the day, they might struggle with scenarios that involved peak demand.

On top of that, Kang et al. (2023) advanced the application of Fourier analysis by integrating it with a Transformer architecture to enhance electric load forecasting. They approached electric load forecasting as a time series problem to extract periodic features from data. The methodology involved preprocessing the dataset by extracting time-based features, applying FFT to transform the

data to the frequency domain, and then using a Transformer model to process this frequency domain data and forecast load. Their model was tested against LSTM, GRU, ARIMA, and ResNet using data from 2,000 households over three years. The results demonstrated that the FFT-Transformer outperformed the benchmark models in forecasting accuracy through lower MSE and MAE.

Besides, Fourier analysis is also useful in predicting consumer behaviour. FFT was applied by Lewis, Herbert, and Bell (2003) to predict call arrivals at a 24-hour inbound telephone call center. Their approach first transformed the time series data into the frequency domain to identify dominant periodic components. Then, the phase shift, amplitude, and frequency were estimated using a nonlinear fitting procedure in the frequency domain. These components were then summed to reconstruct the time series and predict future values. Furthermore, they used a controlled dataset with a linear trend and three sine waves of different frequencies to validate the model. The result showed that the model's predictions matched the actual call data, showing good alignment of the daily and weekly cycles with the call center's workload patterns.

In addition to these applications, Singh and McAtackney (2002) discussed a pattern modelling and recognition system that utilized Fourier transform for noise filtering. They noted that this technique enhanced the accuracy of their forecasting system. Similarly, Rao et al. (2006) focused on refining forecasts by using Fourier transform techniques. They shaped, filtered, and smoothed the forecasts by first applying FFT to transform the data into the frequency domain.

This process allowed them to eliminate higher harmonics that cause sudden fluctuation in the forecasted load.

On the other hand, Hyndman and Athanasopoulos (2021) emphasized that the Dynamic Harmonic Regression (DHR) model with Fourier series terms often outperformed other approaches when dealing with long seasonal periods. Therefore, Permata, Prastyo and Wibawati (2022) explored forecasting methods for short-term electricity load, with a focus on long seasonal periods and calendar variations. This study compared the DHR model and its hybrid version with Double Seasonal ARIMA (DSARIMAX). The DHR model effectively captured long seasonal patterns by applying the Fourier series, which allowed it to address daily, weekly, and yearly harmonics in the data. However, they noticed that the DHR model alone performed reasonably well in capturing long seasonal patterns, but it was less effective in handling calendar variations when compared to the hybrid model.

Moreover, Rausch, Albrecht, and Baier (2021) extended the DHR model specifically designed to handle long seasonal periods by incorporating additional predictor variables other than Fourier terms. The model utilized Fourier series terms to capture periodic components of the time series and ARIMA error terms to address short-term dynamics. They compared the DHR model's performance against traditional time series models like TBATS, ETS, and ARIMA, as well as machine learning models such as Gradient Boosting Regressor (GBR) and Random Forest (RF), using two datasets of 174 weeks of call and e-mail arrival data. The results showed that the DHR model, achieved

superior performance against all traditional time series models and machine learning approaches across all lead times, particularly in capturing long seasonal periods.

2.3 Research Gap

Although many studies have addressed either multi-seasonal patterns or long seasonal periods independently, there are relatively few approaches that tackle both complexities concurrently. Furthermore, previous studies have explored the use of the Fourier Series in handling long and multiple seasonal patterns, but the application of DFT as an exogenous variable within the ARIMA framework still remains underexplored. Since Kang et al. (2023) emphasized that the Fourier transform can handle various types of data efficiently, hence, this study aims to bridge this gap by employing the DFT to handle the time series with long and multi-seasonal periods.

CHAPTER 3

METHODOLOGY

3.1 Dataset Overview

In this study, the dataset by Taylor (2003) was used for the development of the forecasting model. The dataset was a collection of electricity demand data taken in England and Wales, from 5th of June 2000 (Monday) to 27th of August (Sunday) of the same year. It comprised 4032 half-hourly records of electricity consumption in megawatts for 84 days. However, the data was truncated to 1600 data points to simplify the analysis. The data demonstrated multiple seasonal effects, particularly showing a daily cycle with 48 half-hour periods and a weekly cycle with 336 half-hour periods.

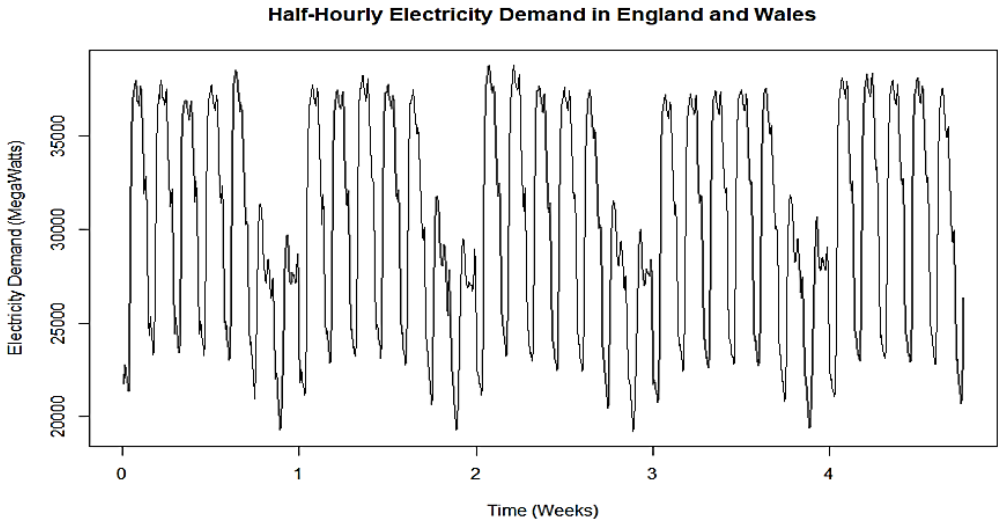

Figure 3.1: The time series graph of the electricity demand in England and Wales data (Truncated data with 1600 data points)

The time plot of the dataset is shown in Figure 3.1. Multiple seasonality patterns were observed. The large pattern, which repeated after approximately seven up-and-down movements, represented the weekly cycle. On the other hand, the small pattern, that repeated within the weekly cycle represented the daily cycle. Furthermore, the amplitude of these cycles remained relatively stable throughout the observed period.

3.2 Data Decomposition Using MSTL Method

The MSTL algorithm decomposes the time series data using an additive approach. While traditional STL decompositions extract only a single seasonal component, MSTL iteratively applies the STL method to extract multiple seasonal elements within the time series.

The STL decomposition performs smoothing on the time series using Loess in two recursive loops. In the inner loop, the seasonal component is calculated through detrending and seasonal smoothing with Loess. The trend component is then calculated through deseasonalized, and trend smoothing with Loess. The remainder is determined by subtracting the seasonal and trend components from the time series. Meanwhile, the outer loop minimizes the impact of outliers on the trend and seasonal components (Rehman, Shahrizal and Noorasiah, 2023).

The decomposition of a time series Z_t using MSTL can be expressed as:

$$Z_t = T_t + \sum_{i=1}^{N} S_{i,t} + R_t$$

where T_t is the trend component, $S_{i,t}$ are the seasonal components, R_t is the remainder component, t is the time index, which is the specific point in the time series being evaluated and N is the number of seasonal components (Bandara, Hyndman and Bergmeir, 2022).

The MSTL decomposition was performed using the 'mstl' function from the built-in 'forecast' package in R (Rdocumentation.org, n.d.). The MSTL effectively separated the time series into its constituent parts, which were the multiple seasonal components ($S_{i,t}$), trend component (T_t), and the remainder component (R_t). MSTL not only provided a clear view of the underlying trends and seasonal patterns but also allowed for the separate modelling and forecasting of each decomposed component. Once the modelling and forecasting were completed, these components were summed up to recreate a data structure identical to the original time series.

3.3 ARIMA Model Fitting

The ARIMA model is a prominent statistical approach that is extensively employed for time series analysis and forecasting. ARIMA modelling is a regression process where the current value, Z_t is predicted using both past values of Z and past forecast errors (ϵ_t). It combines autoregressive (AR) terms, differencing to make the data stationary (I for Integrated), and moving average (MA) terms to model the underlying data patterns (Bakar and Rosbi, 2017). Specifically, the AR component captures the influence of previous time series

values on the current value while the MA component captures the influence of past prediction errors on the current value.

The $ARIMA(p, d, q)$ is expressed using the equation:

$$Z_t' = c + \emptyset_1 Z_{t-1}' + \ldots + \emptyset_p Z'_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Furthermore, the equation can also be expressed in backshift notation:

$$\emptyset(B)(1-B)^d Z_t = c + \theta(B)\varepsilon_t$$

where the $\emptyset_i$ is the parameter of AR, θ_i is the parameter of MA, $\emptyset(B) = (1 - \emptyset_1 B - \cdots - \emptyset_p B^p)$ is a polynomial of degree p in B, $\theta(B) = (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$ $(1-B)^d$ is a polynomial of degree q in B, Z_t is the time series, Z_t' is the differenced series, d is the order of differencing to achieve stationary, c is the constant term and B is the backshift operator, $B^i Z_t = Z_{t-i}$, hence, $Z'_t = (1-B)Z_t$ (Schaffer, Dobbins and Pearson, 2021).

In this study, the automatic algorithm developed by Hyndman and Khandakar (2008) was applied to identify the best-fitting ARIMA models. The automated ARIMA algorithm is advantageous as it can automatically return the best ARIMA model based on either Akaike Information Criterion (AIC), Corrected Akaike Information Criterion (AICc), or Bayesian Information Criterion (BIC). In the default setting, the model with the lowest AICc will be returned. It improves the overall model-fitting process by choosing the optimal model that achieves both the goodness of fit and complexity. Not only that, it also accurately identifies the appropriate lag values for the autoregressive and moving average components, which traditionally require manual inspection of

the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots (Box and Jenkins, 1976).

AIC is used to measure the performance of statistical models by considering both fit and complexity. It penalizes models by the number of parameters. It is defined as:

$$AIC = 2s - 2Log(L)$$

where L refers to the maximized value of the likelihood function for the model and s refers to the number of estimated parameters in the model (Chakrabarti and Ghosh, 2011).

In contrast, BIC also assesses model quality but imposes a stronger penalty for the number of parameters compared to AIC. It is derived from Bayesian probability and defined as:

$$BIC = Log(n)s - 2Log(L)$$

where L is the maximized value of the likelihood function for the model. n is the number of observations and s is the number of estimated parameters in the model (Shi and Tsai, 2002).

Besides, AICc is an adjusted version of AIC to improve model selection for small sample sizes. This correction applies a stronger penalty when the number of parameters is high compared with the sample size, thus rectifying the bias that occurs in AIC due to the small, limited, and insufficient sample size. When n increases, the correction term diminishes and AICc will converge to AIC.

Therefore, AICc is a more reliable criterion for small samples and gives similar results to AIC for larger datasets.

It is defined as:

$$AICc = AIC + \frac{2s(s+1)}{n-s-1}$$

where s is the number of estimated parameters in the model and n is the number of observations (Brewer, Butler and Cooksley, 2016).

In this study, the best-fitting ARIMA model was selected based on minimizing the AICc criterion in the model identification process. AICc was chosen because it could balance the model fit and complexity, ensuring high parsimony and optimal forecast performance (Hyndman and Athanasopoulos, 2021). The ARIMA fitting was conducted using the 'auto.arima' function from the built-in 'forecast' package in R (Rdocumentation.org, n.d.).

3.4 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) maps a time-domain signal into a frequency domain and vice versa using the Inverse Discrete Fourier Transform (IDFT) (Kong, Siauw and Bayen, 2020). The transformation is visualized in Figure 3.2 below. Importantly, no information is lost when moving between the frequency domain to the time domain. Therefore, crucial information such as harmonics, amplitude, and phase are preserved in the transformation (Parsons, Boonman and Obrist, 2000).

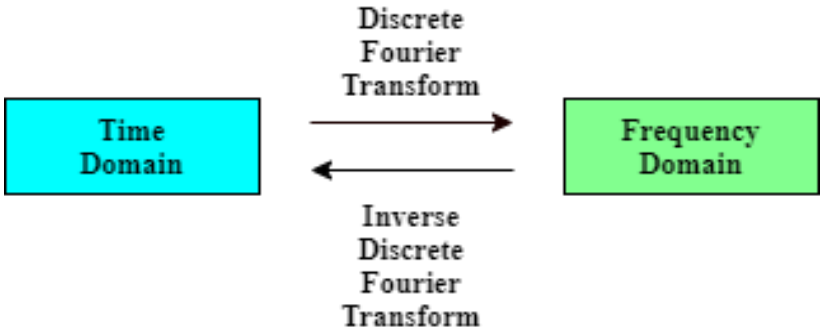

Figure 3.2: The transformation between the time domain and frequency domain using DFT and IDFT

The DFT can be mathematically expressed as:

$$y_k = \frac{1}{N}\sum_{t=1}^{N} z_t e^{\frac{-2\pi i(k-1)(t-1)}{N}}, k = 1,2,3,\ldots\ldots,N \quad (1)$$

The IDFT can be mathematically expressed as:

$$f_t = \sum_{k=1}^{N} y_k e^{\frac{2\pi i(k-1)(t-1)}{N}}, t = 1,2,\ldots,N \quad (2)$$

where z_t, f_t are the time-domain data, y_k is the frequency-domain coefficient, k is the index of the frequency component, N is the number of data points, and i is the imaginary unit (Jain and Singh, 2011).

In this study, the DFT was applied to the seasonal components of the time series data. The underlying seasonal patterns were represented as combinations of sinusoids with different frequencies, amplitudes, and phases. By analyzing

those frequency components, important seasonal patterns were identified (Dhuriya, 2021).

The DFT was executed using the 'fft' function from R's built-in 'stats' package (Rdocumentation.org, n.d.). The 'fft' function computed the DFT in a fast manner using the Fast Fourier Transform (FFT) algorithm. It optimized the computation process, reducing both time and memory usage when compared to the standard DFT calculation methods.

The output of the FFT was a set of complex numbers, each representing a different frequency component of the seasonal data. These complex numbers encoded both amplitude and phase information of the frequency components that made up the seasonality.

By applying the IDFT to these frequency components, the original data points could be accurately reconstructed. However, if the IDFT was applied manually using formula (2) in R, rather than relying on a built-in function, it allowed the time variable t to be extended to the desired length. By extending t beyond the original data length, the periodicity characteristics of the IDFT caused the transformation to create a repeating pattern. These repeating patterns corresponded to the seasonal patterns in the data and could be used as the future value, serving as the exogenous regressor in the Arima model. The result of this process was illustrated in Figure 3.4.2 using an example with the data points $z = (2,1,4,5,5,4)$ and extending to a forecast horizon of 16. As can be seen in

Figure 3.3, those repeating patterns preserved the original pattern of the data points.

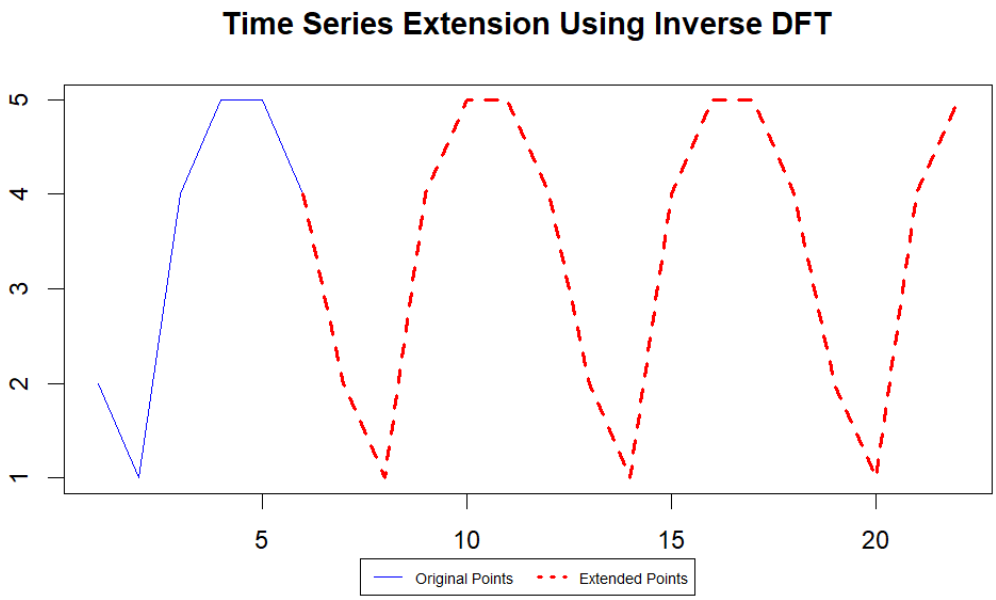

Figure 3.3: The graph of original and extended data points using IDFT

Notably, the exogenous variable must have values for every single time point in the dataset (Andres, 2023). Therefore, the t must be extended so that the number of data points matches the forecast horizon before model forecasting. This ensures that the ARIMA model could use both the past values of the time series and the corresponding values of the exogenous variable to generate reliable forecasts.

3.5 The MSTL-DFT-ARIMA Model

To facilitate model training and evaluation, the data was first divided into training and testing sets. Then, the MSTL-DFT-ARIMA algorithm was

performed on the training set. Figure 3.4 presents a block diagram that delineates the entire process of how the algorithm works.

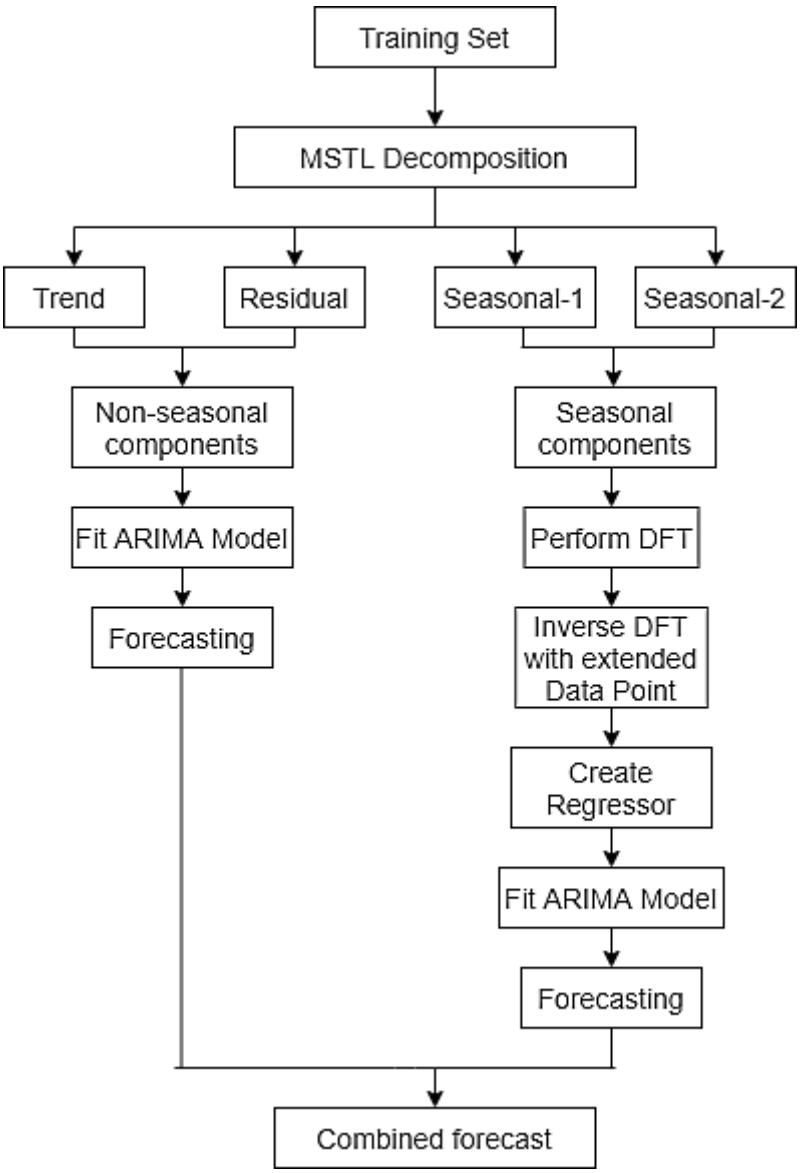

Figure 3.4: The block diagram of the MSTL-DFT-ARIMA algorithm

The time series data was first decomposed using MSTL, isolating the residual, trend, and seasonal components. Then, the residual and trend components were summed up and modelled using an ARIMA model to capture the non-seasonal

patterns and generate forecasts. On the other hand, the seasonal components were combined and converted into the frequency domain using discrete Fourier transform (DFT). In the frequency domain, the seasonal component was decomposed into a series of sinusoids to provide a more detailed and interpretable representation of the underlying periodic patterns. Furthermore, the process was reversed using the inverse DFT. During the inverse DFT, the data point was extended to match the size of the forecast horizon.

For model fitting, only the inverse-transformed data segment corresponding to the length of the training data was used as an exogenous regressor in the ARIMA model. Next, the fitted ARIMA model, where the inverse-transformed data with a length matching the forecast horizon served as the regressor was used for forecasting. This ensured that the seasonal patterns were accurately captured across the forecast period, with each forecasted point having a corresponding regressor value that reflected the extended seasonal patterns. Lastly, the two sets of forecasted values, from the non-seasonal components and seasonal components, were summed to get the total forecast.

An exogenous variable is an external input to the model that is not predicted by the model itself but is used to enhance the forecasting accuracy (Howell, 2023). In this study, the exogenous variable was derived from the data after applying the inverse DFT. It represented the periodic patterns identified in the seasonal components. By including this exogenous feature in the ARIMA model, the model not only relies on past values but also incorporates the specific seasonal

cycles found in the data. This combination allows the ARIMA model to make better use of the repeating patterns in the data, resulting in more accurate predictions of seasonal components.

Therefore, the equation of the ARIMA model with the inverse discrete Fourier transform regressor will be the combination of the ARIMA model equation and the regression model equation. The model can be expressed as:

$$Z_t' = \beta_1 X_t' + \eta_t'$$

where Z_t' is the differenced series, X_t' is the differenced exogenous variable (derived from the inverse DFT), and β_1 is the coefficient of the regressor. The term η_t' captures the residual component that is further modelled using the ARIMA process:

$$\eta_t' = \emptyset_1 \eta_{t-1}' + \ldots + \emptyset_p \eta'_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

where $\emptyset_i$ are the autoregressive parameters, θ_j are the moving average parameters, and ε_t is the error term (Hyndman and Athanasopoulos, 2021).

3.6 Forecasting (TBATS)

The methodology behind TBATS is structured to automate the modelling process and generate accurate forecasts by decomposing overall seasonal patterns into several individual components, each with distinct frequencies.

The TBATS model can be expressed as:

$$\text{TBATS}(q, p, \omega, \varphi, \{m_1, k_1\}, \{m_2, k_2\}, \ldots, \{m_T, k_T\})$$

where each component has a specific function:

ω = the Box-Cox transformation parameter, which stabilizes the time series variance.

φ = damping parameter, regulating the impact of past values on future ones.

q = the orders of the moving average (MA) components

p= the orders of the autoregressive (AR) components

$\{m_1, k_1\}, \{m_2, k_2\}, \ldots, \{m_T, k_T\}$ = the seasonal periods and the number of harmonics used to model each seasonal element (Yu et al., 2021).

The modelling process starts with the Box-Cox transformation to deal with nonlinearity in the data. This transformation makes the time series more linear and homoscedastic so that the model fits better. This transformation is defined as:

$$z_t^{(\omega)} = \begin{cases} \dfrac{z_t^{\omega} - 1}{\omega} \; if \; \omega \neq 0 \\ \log(z_t) \, if \; \omega = 0, \end{cases}$$

where z_t is the original time series, and ω is the Box-Cox transformation parameter and $z_t^{(\omega)}$is the observations that have undergone transformation using the Box-Cox method with the parameter ω.

Next, the TBATS model accommodates multiple and complex seasonalities by incorporating trigonometric terms. In this case, each seasonal component is represented using a Fourier series, which includes a sum of sine and cosine terms.

The modelling equation is defined as:

$$z_t = \sum_{j=1}^{J} \alpha_j \cos\left(\frac{2\pi jt}{m}\right) + \sum_{j=1}^{J} \beta_j \sin\left(\frac{2\pi jt}{m}\right)$$

where m is the seasonal period, α_j and β_j are coefficients estimated from the data and the number of harmonics, J determines the smoothness of the seasonal component, with more harmonics capturing more detailed seasonal patterns.

Furthermore, the state space representation of the TBATS model is then constructed. It is composed of seasonal components, a trend component, and an ARMA error component. The parameters of the TBATS model are estimated using a MLE approach. More specifically, it is done by iteratively updating the state vector and associated variances by the use of the Kalman filter (Hyndman and Athanasopoulos, 2021).

The automation of the TBATS model involves heuristic searching for the best parameters, supplemented by model selection criteria like AIC. This ensures that the model is both parsimonious and capable of capturing the features of the time series. The state space structure propagates the state vector forward in time using the estimated parameters to do forecasting. This process automatically accounts for the trend, seasonal patterns, and autocorrelations in the residuals to obtain reliable forecasts (De Livera, Hyndman and Snyder, 2011).

In this study, the TBATS model was applied to the same dataset using the ‘tbats’ function from the built-in ‘forecast’ package in R (Rdocumentation.org, n.d.). The forecast horizon for both the TBATS model and the proposed model was

set identically. This alignment allowed for a direct comparison of both model performances over the same time period, which will be further discussed in the subsequent section 3.7.

3.7 Model Evaluation

To compare the proposed MSTL-DFT-ARIMA model with the TBATS model, both models were applied to the same dataset with an identical forecast horizon of 1,000 data points. The size of the training and testing sets was kept the same for both models, and their forecast accuracy and computing time were measured to evaluate their effectiveness.

3.7.1 Quantitative Forecast Accuracy

The out-of-sample predictive performances of the MSTL-DFT-ARIMA and TBATS models were evaluated using the testing set. Three forecast accuracy metrics, namely Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) were calculated from the differences between the observed values in the testing set and the corresponding forecasts. The detailed information on these three error measures is presented in Table 3.1 below.

Table 3.1: Summary of Accuracy Metrics

Measure	Formula	Purpose
MAE	$\frac{1}{N}\sum_{i=1}^{N}\lvert Z_t - \widehat{Z}_t\rvert$	Quantifies the average magnitude of errors without taking the direction into account, providing a straightforward measure of forecast accuracy.
MAPE	$\frac{1}{N}\sum_{t=1}^{N}\left\lvert\frac{Z_t - \hat{Z}_t}{Z_t}\right\rvert$	Expresses forecast accuracy in percentage form, making it easier to compare performance across different models.
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(Z_t - \widehat{Z}_t)^2}$	It highlights significant deviations in forecasts.

In context, N refers to the total number of data points in the testing set, $\widehat{Z}_t$ is the predicted value from the forecasted data and Z_t is the actual value from the test data (Hyndman and Koehler, 2006).

3.7.2 Computational Efficiency

The computational efficiency was assessed using the elapsed time required for model fitting and forecasting for both the TBATS and MSTL-DFT-ARIMA models.

3.8 Forecast Performance Comparison – Time Series Cross-Validation

The cross-validation method that used in classification with independent observations is not directly applicable to time series forecasting. This is because time series data must preserve their temporal order, where the observations are ordered chronologically and depend on preceding values (Assaad and Fayek, 2021).

For time series forecasting, the time series cross-validation or rolling window approach is analogous to the traditional cross-validation methods. In this approach, the model is trained using either a fixed window or an expanding window. In the fixed window approach, the size of the training set remains constant while the window slides forward in time (refer to Figure 3.5 A). This method uses a fixed amount of past data to predict future values, allowing for a consistent comparison across iterations. On the other hand, the expanding window approach starts with an initial training size and gradually incorporates more recent data as time progresses (refer to Figure 3.5 B). This enables the model to learn from an increasing amount of information (Hewamalage, Ackermann and Bergmeir, 2022).

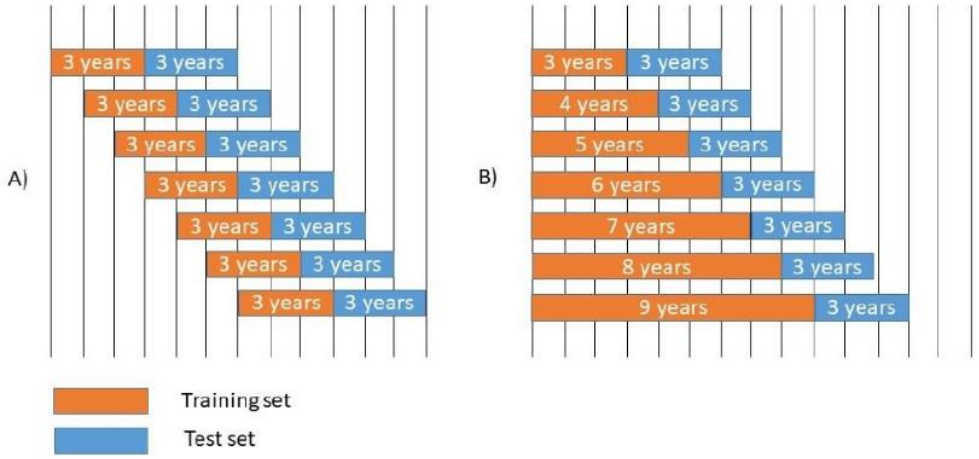

Figure 3.5: (A) Cross-validation based on Fixed Window Approach (B) Cross-validation based on Expanding Window Approach (Shojaei and Flood, 2018)

In this study, a rolling origin cross-validation method was applied to compare the performance of both the proposed model (MSTL-DFT-ARIMA) and the TBATS model in handling the long, multi-seasonal data. The original dataset of 4032 points was truncated to 1600 points to shorten the validation time. The cross-validation procedure began with an initial training set of 1400 data points, and the next 200 points were designated as the testing set. This train-test split preserved the time order of the data.

Contrary to the typical expanding window approach, this study started with the longest training window and gradually shortened it in each iteration. The training set decreased by 20 data points while the testing set shifted backward in time in each subsequent iteration. Each iteration was analogous to one-fold in traditional cross-validation. The process involved five repetitions, resulting in five distinct iterations. Notably, a different model was created and evaluated

for its forecast performance for every iteration. The idea is illustrated in Figure 3.6 below.

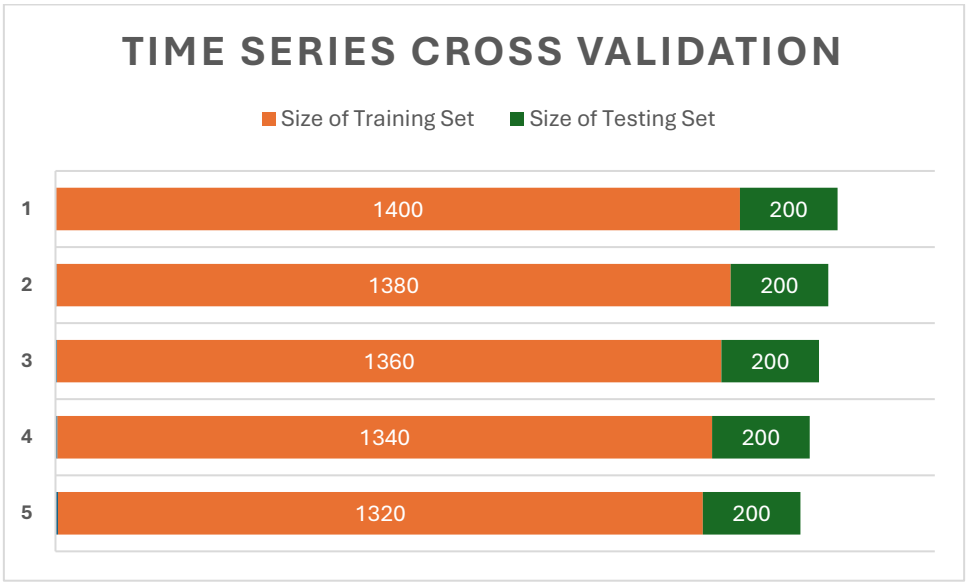

Figure 3.6: The bar chart of the rolling origin cross-validation procedure with a step-back loop

Furthermore, each model's accuracy and computational efficiency were evaluated in each iteration. If the proposed model consistently showed a lower MAE, MAPE, RMSE, and computing time than the TBATS model, it indicated superior performance on the given data. Conversely, if the TBATS model showed lower error metrics and computing time, it was regarded as the better-performing model.

The study recognized that using a single dataset might lead to biased or misleading comparisons between forecast models. Therefore, the rolling origin cross-validation procedure with a step-back loop was employed. This approach evaluated the models' performance across different historical periods with

different amounts of training data. Hence, the risk of bias from randomly choosing a single training and testing set was reduced. Consequently, the model performance assessments were more reliable and not merely a result of chance (Hyndman and Athanasopoulos, 2021).

CHAPTER 4

RESULTS AND DISCUSSION

This section discussed and compared the forecasting results of the MSTL-DFT-ARIMA and TBATS models. Five-fold cross-validation has been performed for both methods to ensure a more reliable evaluation.

4.1 Data Decomposition Using MSTL Method

The training set was first decomposed using the MSTL method, which separates the series into its trend, multiple seasonal components, and remainder. The decomposition is represented by the following equation:

$$Z_t = T_t + S_{48,t} + S_{336,t} + R_t$$

where Z_t is the original time series, T_t is the trend component, $S_{48,t}$ and $S_{336,t}$ are the seasonal components with periods of 48 and 336, respectively, and R_t is the remainder component.

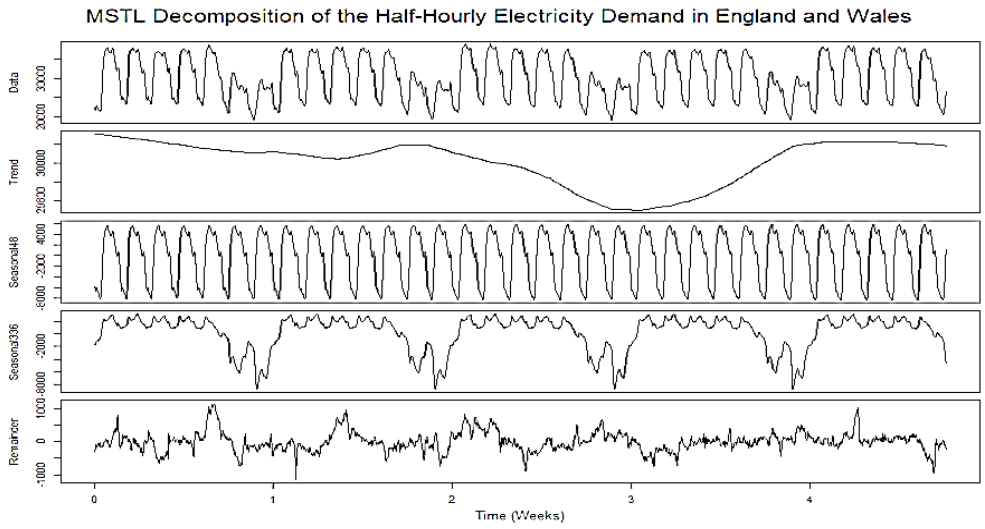

Figure 4.1: The components of the 1380 training data points under MSTL decomposition

Figure 4.1 shows the individual plots of these decomposed components. Compared to the original time series, the decomposed components exhibit simpler patterns. This reduction in complexity leads to a more nuanced understanding of the data. It also simplifies the modelling of each component, thus improving the overall model fitting.

Based on Figure 4.1, the original time series data (Z_t) appears to move up and down regularly, which suggests that there are underlying seasonal trends. Next, the trend component (T_t) shows a clear downward slope followed by an upward trend. Moreover, the seasonal48 components ($S_{48,t}$) shows a clear repeating pattern every day, corresponding to 48-time units. Moreover, the seasonal336 components ($S_{336,t}$) shows a clear repeating pattern every week, corresponding to 336-time units. Lastly, the remainder component (R_t) fluctuates irregularly without any obvious pattern.

4.2 ARIMA Model Fitting

The ARIMA model was fitted to the training set through the automatic algorithm developed by Hyndman and Khandakar (2008). The model orders selected for the five iterations are presented in Table 4.1.

Table 4.1: Optimal model selection for each iteration

Iteration	Non-seasonal Model	Seasonal Model
1	$ARIMA(0,1,4)$	$ARIMA(3,0,1)$
2	$ARIMA(0,1,4)$	$ARIMA(3,0,1)$
3	$ARIMA(0,1,4)$	$ARIMA(3,0,1)$
4	$ARIMA(1,1,3)$	$ARIMA(3,0,1)$
5	$ARIMA(1,1,3)$	$ARIMA(5,0,0)$

The optimal models identified in each iteration are detailed below, along with their equivalent representations using the backshift operator.

<u>Iteration 1</u>

Non-Seasonal Model (ARIMA(0,1,4)):

$$Z_t' = \epsilon_t - 0.0627\epsilon_{t-1} + 0.0298\epsilon_{t-2} - 0.0658\epsilon_{t-3} - 0.0505\epsilon_{t-4}$$

$$(1 - B)Z_t = (1 - 0.0627B + 0.0298B^2 - 0.0658B^3 - 0.0505B^4)\epsilon_t$$

Seasonal Model (ARIMA(3,0,1)):

$$Z_t = 0.5067Z_{t-1} + 0.6715Z_{t-2} - 0.3845Z_{t-3} + \epsilon_t - 0.9350\epsilon_{t-1} + 0.9975X_t$$

$$(1 - 0.5067B - 0.6715B^2 + 0.3845B^3)Z_t = (1 - 0.9350B)\epsilon_t + 0.9975X_t$$

Iteration 2

Non-Seasonal Model (ARIMA(0,1,4)):

$$Z_t' = \epsilon_t - 0.0590\epsilon_{t-1} + 0.0292\epsilon_{t-2} - 0.0657\epsilon_{t-3} - 0.0487\epsilon_{t-4}$$

$$(1 - B)Z_t = (1 - 0.0590B + 0.0292B^2 - 0.0657B^3 - 0.0487B^4)\epsilon_t$$

Seasonal Model (ARIMA(3,0,1)):

$$Z_t = 0.5125Z_{t-1} + 0.6541Z_{t-2} - 0.3890Z_{t-3} + \epsilon_t - 0.9096\epsilon_{t-1} + 0.9985X_t$$

$$(1 - 0.5125B - 0.6541B^2 + 0.3890B^3)Z_t = (1 - 09096B)\epsilon_t + 0.9985X_t$$

Iteration 3

Non-Seasonal Model (ARIMA(0,1,4)):

$$Z_t' = \epsilon_t - 0.0591\epsilon_{t-1} + 0.0325\epsilon_{t-2} - 0.0627\epsilon_{t-3} - 0.0482\epsilon_{t-4}$$

$$(1 - B)Z_t = (1 - 0.0591B + 0.0325B^2 - 0.0627B^3 - 0.0482B^4)\epsilon_t$$

Seasonal Model (ARIMA(3,0,1)):

$$Z_t = 0.5284Z_{t-1} + 0.6173Z_{t-2} - 0.3759Z_{t-3} + \epsilon_t - 0.8868\epsilon_{t-1} + 0.9986X_t$$

$$(1 - 0.5284B - 0.6173B^2 + 0.3759B^3)Z_t = (1 - 08868B)\epsilon_t + 0.9986X_t$$

Iteration 4

Non-Seasonal Model (ARIMA(1,1,3)):

$$Z_t' = 0.8713Z_{t-1}' + \epsilon_t - 0.9414\epsilon_{t-1} + 0.0864\epsilon_{t-2} - 0.0881\epsilon_{t-3}$$

$$(1 - 0.8713B)(1 - B)Z_t = (1 - 0.9414B + 0.0864B^2 - 0.0881B^3)\epsilon_t$$

Seasonal Model (ARIMA(3,0,1)):

$$Z_t = 0.9994X_t + 0.5207Z_{t-1} + 0.6111Z_{t-2} - 0.3801Z_{t-3} - 08999\epsilon_{t-1} + \epsilon_t$$

$$(1 - 0.5207B - 0.6111B^2 + 0.3801B^3)Z_t = (1 - 08999B)\epsilon_t + 0.9994X_t$$

Iteration 5

Non-Seasonal Model (ARIMA(1,1,3)):

$$Z_t' = 0.8570Z_{t-1}' + \epsilon_t + 0.9266\epsilon_{t-1} - 0.0803\epsilon_{t-2} + 0.0885\epsilon_{t-3}$$

$$(1 - 0.8570B)(1 - B)Z_t = (1 + 0.9266B - 0.0803B^2 + 0.0885B^3)\epsilon_t$$

Seasonal Model (ARIMA(5,0,0)):

$$Z_t = 1.4706Z_{t-1} - 0.7143Z_{t-2} + 0.1569Z_{t-3} - 0.1809Z_{t-3} + 0.1694Z_{t-3} + 1.0013X_t$$

$$(1 - 1.4706B + 0.7143B^2 - 0.1569B^3 + 0.1809B^4 - 0.1694B^5)Z_t = 1.0013X_t$$

where Z_t is the time series, Z_t' is the differenced series, B is the backshift operator, ϵ_t is the error term and X_t refers to the exogenous regressor (IDFT) included in the ARIMA model.

These models ensure that both non-seasonal and seasonal components are accurately captured. Thus, providing robust forecasts for the time series data.

4.3 Forecasting Accuracy Comparison and Visualization: TBATS vs MSTL-DFT-ARIMA

4.3.1 Visualization

The forecasts from the MSTL-DFT-ARIMA and TBATS models are plotted together with the corresponding testing set and a segment of the training set on the same figure for a visual accuracy comparison. Notably, the forecasted data extends beyond the testing data to assess the models' ability to capture the long (i.e. the weekly) seasonal pattern besides the short (i.e. daily) seasonal pattern. Figure 4.2 presents the accuracy comparison for the initial iteration, while Figures 4.3, 4.4, 4.5, and 4.6 display the plots for the remaining four iterations.

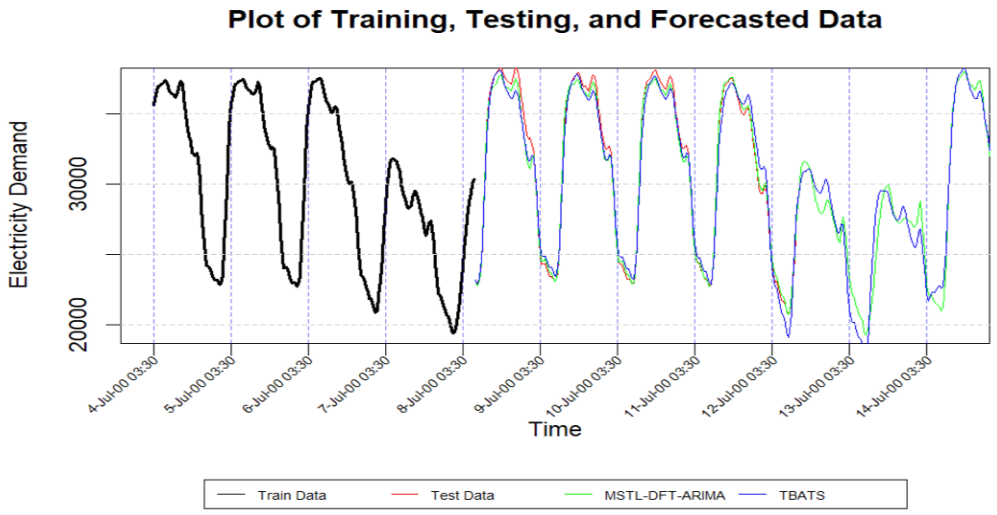

Figure 4.2: The plot of truncated training data (1400 data points), testing data & forecasted data using MSTL-DFT-ARIMA and TBATS for iteration 1

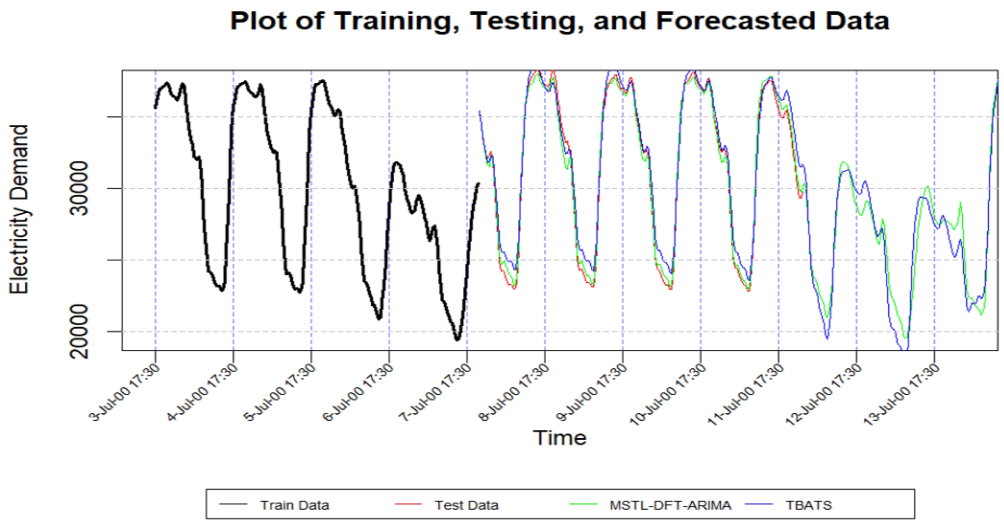

Figure 4.3: The plot of truncated training Data (1380 data points), testing data & forecasted data using MSTL-DFT-ARIMA and TBATS for iteration 2

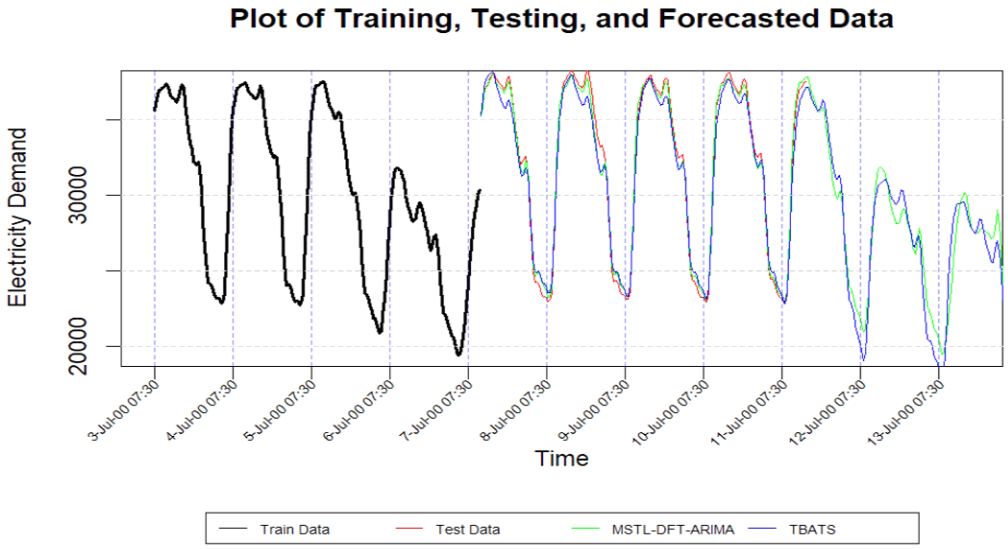

Figure 4.4: The plot of truncated training Data (1360 data points), testing data & forecasted data using MSTL-DFT-ARIMA and TBATS for iteration 3

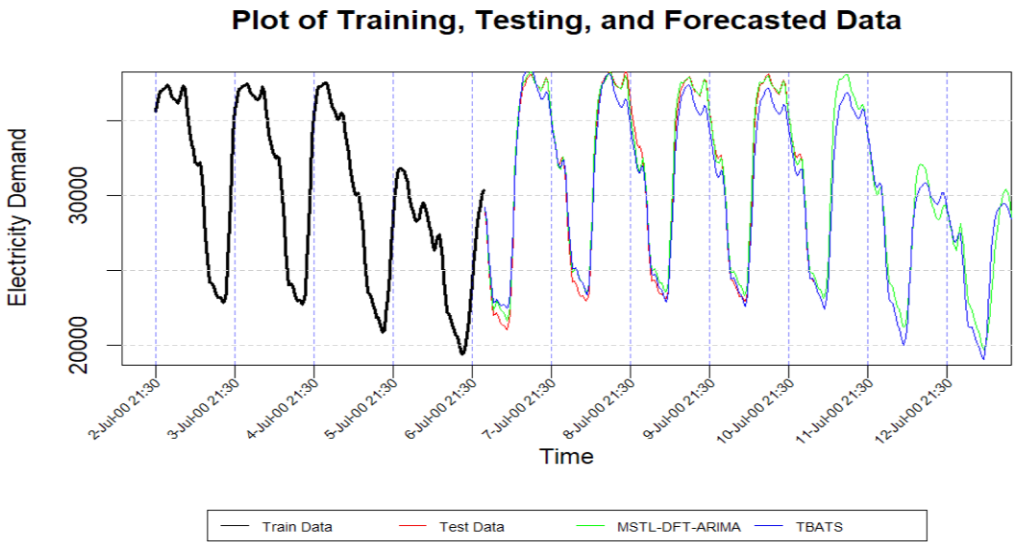

Figure 4.5: The plot of truncated training Data (1340 data points), testing data & forecasted data using MSTL-DFT-ARIMA and TBATS for iteration 4

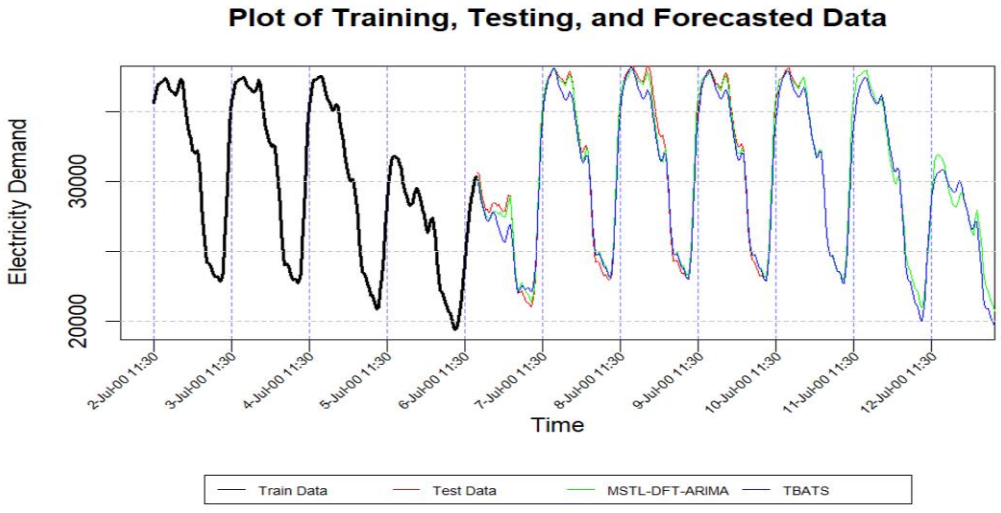

Figure 4.6: The plot of truncated training Data (1320 data points), testing data & forecasted data using MSTL-DFT-ARIMA and TBATS for iteration 5

In all the plots, the forecasted data for TBATS (blue line) and MSTL-DFT-ARIMA (green line) closely match the patterns of the test data (red line), indicating that both MST-FFT-ARIMA and TBATS models effectively capture the seasonalities and trends of the testing data. However, with a more zoomed-in observation of the forecasted lines, it is observed that the MSTL-DFT-ARIMA forecast line adheres more tightly to the fluctuations in the test data, whereas TBATS shows slight deviations from the actual test data.

4.3.2 Quantitative Forecast Performance

The forecast accuracies of both the MSTL-DFT-ARIMA and TBATS models for each iteration are summarized in Table 4.2 in terms of RMSE, MAPE, and MAE.

Table 4.2: Forecast error comparison for MSTL-DFT-ARIMA and TBATS for each iteration

Metrics	Iteration	MSTL-DFT-ARIMA	TBATS
RMSE	1	514.91	792.51
	2	487.37	948.19
	3	464.94	774.55
	4	510.07	936.24
	5	471.29	863.24
MAE	1	373.54	653.67
	2	400	763.58
	3	368.74	661.16

	4	397.2	775.78
	5	373.32	691
MAPE (%)	1	1.14	2.14
	2	1.3	2.7
	3	1.19	2.06
	4	1.43	2.46
	5	1.29	2.25

Across all five iterations, the MSTL-DFT-ARIMA model demonstrates good forecasting capabilities, as evidenced by its lower RMSE, MAE, and MAPE values.

4.3.3 Computational Efficiency

The total computational times for the model fitting and forecasting processes are presented in Table 4.3.

Table 4.3: Computational time for MSTL-DFT-ARIMA and TBATS for each iteration

Iteration	Proposed Model (seconds)	TBATS Model (seconds)
1	2.59	116.60
2	2.40	77.86
3	2.60	77.50
4	0.90	100.57
5	0.85	92.98

The MSTL-DFT-ARIMA model took significantly less time to fit the model and forecast for each of the five iterations, with computation durations ranging from roughly 0.85 to 2.60 seconds. On the other hand, the computing times of the TBATS model range from about 77.50 seconds up to 116.60 seconds. Thus, the proposed model outranks the TBATS model in handling long and multi-seasonal data in terms of computational efficiency.

Based on the above results, the MSTL-DFT-ARIMA model shows a superior fit when comparing the forecasted data to the test data. It also demonstrates both high speed and accuracy. These superior performances are primarily due to the ability of the proposed model to partition the data into manageable subseries, such as trend, seasonal, and residual components using MSTL. This decomposition reduces the data complexity and permits a deeper understanding of underlying patterns and trends. Additionally, the model further enhances its ability to capture long and multi-seasonal variations by fully utilizing the periodicity of inverse discrete Fourier transform to create the exogenous regressor, which is then included in the ARIMA model. This approach allows the model to more accurately capture and forecast seasonal dynamics. Moreover, the efficiency of the FFT algorithm also speeds up the computation. Hence, the decomposition of the time series and the integration of these regressors into the ARIMA model improves the overall forecasting performance.

While the TBATS model is effective in capturing complex seasonality, it is computationally intensive. Furthermore, the parameter estimation in TBATS is slower because the prediction length often requires extensive computation.

Hence, a longer time is needed to fit the model, and do forecasting (Hyndman and Athanasopoulos, 2021).

CHAPTER 5

CONCLUSIONS

5.1 Summary of Research

In today's world, where scientific research is greatly valued and digital technology is rapidly evolving, data collection has become easier and more frequent. Hence, multiple seasonal data will be encountered commonly. That said, a fast and good model is in demand. Therefore, this research has introduced a forecasting method using MSTL, ARIMA, and DFT, mainly to handle long and multi-seasonal data. The remarkable features of the proposed model are the use of DFT and the extended inverse DFT (IDFT) to serve as an exogenous regressor in the ARIMA framework. Since only one dataset was used in this study, TSCV was adopted to ensure a more valuable evaluation. Furthermore, the forecast performances of MSTL-DFT-ARIMA were compared with TBATS in terms of RMSE, MAPE, and MAE. The result shows that the MSTL-DFT-ARIMA outperformed TBATS in both prediction accuracy and computational time across all 5 iterations of TSCV.

5.2 Limitations and Recommendations

One of the limitations of the proposed MSTL-DFT-ARIMA model is it considers that the data is available every day, without accounting for gaps like weekends, holidays, or festivals. Furthermore, the model was tested using only one dataset throughout the study, which did not exhibit an increasing trend or variance. This limitation restricts the generalizability of the model, as it may not perform as well on datasets with different characteristics, such as those showing an upward trend or increasing variance over time. Another limitation is that the model was compared only with the TBATS model. This limited comparison may not fully highlight the strengths and weaknesses of the MSTL-DFT-ARIMA model. On top of that, the study did not perform residual diagnostic tests such as the Ljung-box test to assess the adequacy of the model. Those diagnostic tests ensure the residuals are uncorrelated and follow a white noise process. Thus, the absence of such tests may cause issues related to model assumptions and fit unaddressed, affecting the forecasting results.

To address these limitations, future research should explore the effects of weekends and holidays on the model performance. Next, future studies should experiment with a broader range of datasets with varying characteristics, rather than relying on a single dataset. For instance, further research may also try to apply the model to non-seasonal data or data with changing periodicity over time to see whether the model works well or if modifications are needed. Moreover, further study should evaluate the MSTL-DFT-ARIMA model by benchmarking it against other forecasting methods than TBATS, such as Prophet, to better understand its strengths and limitations.

REFERENCES

Andres, D. (2023). *Exogenous variables in ARIMA models - ML Pills*. [online] ML Pills - Machine Learning Pills. Available at: https://mlpills.dev/time-series/exogenous-variables-in-arima-models/ [Accessed 27 Aug. 2024].

Arneric, J. (2021). Multiple STL decomposition in discovering a multi-seasonality of intraday trading volume. *Croatian Operational Research Review*, 12(1), pp.61–74. https://doi.org/10.17535/crorr.2021.0006

Assaad, R.H. and Fayek, S. (2021). Predicting the Price of Crude Oil and its Fluctuations Using Computational Econometrics: Deep Learning, LSTM, and Convolutional Neural Networks. *Econometric Research in Finance*, 6(2), pp.119–137. https://doi.org/10.2478/erfin-2021-0006

Bakar, N.A. and Rosbi, S. (2017). Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction. *International Journal of Advanced Engineering Research and Science*, 4(11), pp.130–137. https://doi.org/10.22161/ijaers.4.11.20

Bandara, K., Hyndman, R. and Bergmeir, C. (2022). MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns. *International Journal of Operational Research*, 1(1), p.1. https://doi.org/10.1504/ijor.2022.10048281

Box, G.E.P. and Jenkins, G.M. (1976) Time Series Analysis: Forecasting and Control. Revised Edition, Holden Day, San Francisco.

Brewer, M.J., Butler, A. and Cooksley, S.L. (2016). The relative performance of AIC, AICC and BIC in the presence of unobserved heterogeneity. *Methods in Ecology and Evolution*, 7(6), pp.679–692. https://doi.org/10.1111/2041-210x.12541

Chakrabarti, A. and Ghosh, J.K. (2011). AIC, BIC and Recent Advances in Model Selection. *Philosophy of Statistics*, 7, pp.583–605. https://doi.org/10.1016/b978-0-444-51862-0.50018-6

Cleveland, R.B., Cleveland, W.S., McRae, J.E. and Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, 6(1), pp.3–73.

De Livera, A. (2010). Automatic forecasting with a modified exponential smoothing state space framework. *Clayton: Department of Econometrics & Business Statistics, Monash University*

De Livera, A.M., Hyndman, R.J. and Snyder, R.D. (2011). Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing. *Journal of the American Statistical Association*, 106(496), pp.1513–1527. https://doi.org/10.1198/jasa.2011.tm09771

Dhuriya, A. (2021). *Why Fourier Transform is so important?* [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/why-fourier-transform-is-so-important-cb7841733bb8 [Accessed 27 Aug. 2024].

Gould, P.G., Koehler, A.B., Ord, J.K., Snyder, R.D., Hyndman, R.J. and Vahid-Araghi, F. (2008). Forecasting time series with multiple seasonal

patterns. *European Journal of Operational Research*, 191(1), pp.207–222. https://doi.org/10.1016/j.ejor.2007.08.024

Harvey, A. and Koopman, S.J. (1993). Forecasting Hourly Electricity Demand Using Time-Varying Splines. *Journal of the American Statistical Association*, 88(424), pp.1228–1236. https://doi.org/10.1080/01621459.1993.10476402

Hewamalage, H., Ackermann, K. and Bergmeir, C. (2022). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2), pp.788–832. https://doi.org/10.1007/s10618-022-00894-5

Howell, E. (2023). *Take Your Forecasting to the Next Level with Harmonic Regression*. [online] Medium. https://towardsdatascience.com/take-your-forecasting-to-the-next-level-with-harmonic-regression-5a8515f63295 [Accessed 27 Aug. 2024]

Hyndman, R.J. (2017). *Forecasting with daily data | Rob J Hyndman*. [online] robjhyndman.com. Available at: https://robjhyndman.com/hyndsight/dailydata/

Hyndman, R.J. and Athanasopoulos, G. (2021). *Forecasting : Principles and Practice*. 3rd ed. [online] Heathmont, Vic.: Otexts. https://otexts.com/fpp3/

Hyndman, R.J. and Khandakar, Y. (2008). Automatic Time Series Forecasting: the Forecast Package for R. *Journal of Statistical Software*, 27(3). https://doi.org/10.18637/jss.v027.i03

Hyndman, R.J. and Koehler, A.B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), pp.679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

IEEE Pulse. (2016). *Highlights in the History of the Fourier Transform – EMBS*. [online] Available at: https://www.embs.org/pulse/articles/highlights-in-the-history-of-the-fourier-transform/

Jacoby, W.G. (2000). Loess: a nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, 19(4), pp.577–613. https://doi.org/10.1016/s0261-3794(99)00028-1

Jain, S.K. and Singh, S.N. (2011). Harmonics estimation in emerging power system: Key issues and challenges. *Electric Power Systems Research*, 81(9), pp.1754–1766. https://doi.org/10.1016/j.epsr.2011.05.004

Kang, L., Guo, Y., Liao, S., Xu, Y., Wang, T. and Cheng, D. (2023). Electric Load Forecasting with Fast Fourier Transform-Optimized Transformers. *3rd International Conference on Electronic Information Engineering and Computer Communication (EIECC)*. https://doi.org/10.1109/eiecc60864.2023.10456747

Keil, A., Bernat, E.M., Cohen, M.X., Ding, M., Fabiani, M., Gratton, G., Kappenman, E.S., Maris, E., Mathewson, K.E., Ward, R.T. and Weisz, N. (2022). Recommendations and publication guidelines for studies using frequency domain and time-frequency domain analyses of neural time series. *Psychophysiology*, 59(5). https://doi.org/10.1111/psyp.14052

Kong, Q., Siauw, T. and Bayen, A. (2020). *Python Programming And Numerical Methods : a guide for engineers and scientists.* 1st ed. [online] S.L.: Elsevier Academic Press. Available at: https://pythonnumericalmethods.berkeley.edu/notebooks/Index.html [Accessed 27 Aug. 2024].

Krechiem, A. and Khadir, M.T. (2023). Algerian Electricity Consumption Forecasting with Artificial Neural Networks Using a Multiple Seasonal-Trend Decomposition Using LOESS. *International Conference on Decision Aid Sciences and Applications (DASA)*, pp.586–591. https://doi.org/10.1109/dasa59624.2023.10286694

Lakshmanan, A. and Das, S. (2017). Two-stage models for forecasting time series with multiple seasonality.

Lewis, B.G., Herbert, R.D. and Bell, R.D. (2003). The Application of Fourier Analysis to Forecasting the Inbound Call Time Series of a Call Centre. *In Proceedings of the International Congress on Modeling and Simulation*, [online] pp.1281–1286. Available at: https://mssanz.org.au/MODSIM03/Volume_03/B10/06_Lewis.pdf [Accessed 23 Aug. 2024].

Liu, J. (2024). Navigating the Financial Landscape: The Power and Limitations of the ARIMA Model. *Highlights in Science, Engineering and Technology*, [online] 88, pp.747–752. https://doi.org/10.54097/9zf6kd91

Lye, K.W., Yuan, X.M. and Cai, T.X. (2009). A spectrum comparison method for demand forecasting. *SIMTech technical reports*, 10(1), pp.32–35.

Manani, K. (2022). *Multi-Seasonal Time Series Decomposition using MSTL in Python*. [online] Medium. Available at: https://towardsdatascience.com/multi-seasonal-time-series-decomposition-using-mstl-in-python-136630e67530 [Accessed 27 Aug. 2024].

McLoughlin, F., Duffy, A. and Conlon, M. (2013). Evaluation of time series techniques to characterise domestic electricity demand. *Energy*, 50, pp.120–130. https://doi.org/10.1016/j.energy.2012.11.048

Moon, H., Lee, H. and Song, B. (2022). Mixed pooling of seasonality for time series forecasting: An application to pallet transport data. *Expert Systems with Applications*, 201, p.117195. https://doi.org/10.1016/j.eswa.2022.117195

Munim, Z.H. (2022). State-space TBATS model for container freight rate forecasting with improved accuracy. *Maritime Transport Research*, 3, p.100057. https://doi.org/10.1016/j.martra.2022.100057

Nadeem (2021). *Time Series Forecasting using TBATS Model*. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/time-series-forecasting-using-tbats-model-ce8c429442a9

Naim, I., Mahara, T. and Idrisi, A.R. (2018). Effective Short-Term Forecasting for Daily Time Series with Complex Seasonal Patterns. *Procedia Computer Science*, 132, pp.1832–1841. https://doi.org/10.1016/j.procs.2018.05.136

Nan, H., Zhu, X. and Ma, J. (2023). MSTL-GLTP: A Global-Local Decomposition and Prediction Framework for Wireless Traffic. *IEEE Internet of Things Journal*, 10(6), pp.5024–5034. https://doi.org/10.1109/jiot.2022.3221743

Parsons, S., Boonman, A.M. and Obrist, M.K. (2000). Advantages and disadvantages of techniques for transforming and analyzing chiropteran echolocation calls. *Journal of Mammalogy*, 81(4), pp.927–938. https://doi.org/10.1644/1545-1542(2000)081%3C0927:aadotf%3E2.0.co;2

Permata, R.P., Prastyo, D.D. and Wibawati (2022). Hybrid dynamic harmonic regression with calendar variation for Turkey short-term electricity load forecasting. *Procedia Computer Science*, 197, pp.25–33. https://doi.org/10.1016/j.procs.2021.12.114

Rao, M.S.S., Soman, S.A., Menezes, B.L., Chawande, N.P., Dipti, P. and Ghanshyam, T. (2006). An expert system approach to short-term load forecasting for Reliance Energy Limited, Mumbai. *2006 IEEE Power India Conference*. https://doi.org/10.1109/poweri.2006.1632604

Rausch, T.M., Albrecht, T. and Baier, D. (2021). Beyond the beaten paths of forecasting call center arrivals: on the use of dynamic harmonic regression with predictor variables. *Journal of Business Economics*, 92(4), pp.675–706. https://doi.org/10.1007/s11573-021-01075-4

Rdocumentation.org. (n.d.). *auto.arima function - RDocumentation*. [online] Available at: https://www.rdocumentation.org/packages/forecast/versions/8.21/topics/auto.arima [Accessed 27 Aug. 2024].

Rdocumentation.org. (n.d.). *fft function - RDocumentation*. [online] Available at: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/fft [Accessed 27 Aug. 2024].

Rdocumentation.org. (n.d.). *mstl function - RDocumentation*. [online] Available at: https://www.rdocumentation.org/packages/forecast/versions/8.21/topics/mstl [Accessed 27 Aug. 2024].

Rdocumentation.org. (n.d.). *tbats function - RDocumentation*. [online] Available at: https://www.rdocumentation.org/packages/forecast/versions/8.23.0/topics/tbats [Accessed 27 Aug. 2024].

Rehman, K.A., Shahrizal, A.R.M. and Noorasiah, M. (2023). Improving Long-Term Wave Forecasting Through Seasonal Adjustment Based On Seasonal Trend Decomposition LOESS And CNN-GRU Network. *Journal of sustainability science and management/Journal of Sustainability Science and Management*, 18(4), pp.119–137. https://doi.org/10.46754/jssm.2023.04.009

Ridwan, M., Sadik, K. and Afendi, F.M. (2023). Comparison of ARIMA and GRU Models for High-Frequency Time Series Forecasting. *Scientific Journal of Informatics*, 10(3), pp.389–400. https://doi.org/10.15294/sji.v10i3.45965

Rizkya, I., Syahputri, K., Sari, R.M., Siregar, I. and Utaminingrum, J. (2019). Autoregressive Integrated Moving Average (ARIMA) Model of Forecast Demand in Distribution Centre. *IOP Conference Series: Materials Science and Engineering*, 598, p.012071. https://doi.org/10.1088/1757-899x/598/1/012071

Sakhuja, S. (2024). *Limitations of Fourier Transform and the Role of Wavelet Transform*. [online] Medium. Available at: https://sakhujasaiyam.medium.com/disadvantages-of-fourier-transform-and-the-role-of-wavelet-transform-a4218d8cc6de [Accessed 20 Aug. 2024].

Schaffer, A.L., Dobbins, T.A. and Pearson, S.-A. (2021). Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. *BMC Medical Research Methodology*, 21(1). https://doi.org/10.1186/s12874-021-01235-8

Shi, P. and Tsai, C.-L. (2002). Regression model selection-a residual likelihood approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2), pp.237–252. https://doi.org/10.1111/1467-9868.00335

Shojaei, A. and Flood, I. (2018). Univariate Modeling of the Timings and Costs of Unknown Future Project Streams: A Case Study. *International Journal on Advances in Systems and Measurements*, [online] 11(1&2), pp.36–46. https://www.iariajournals.org/systems_and_measurements/

Singh, S. and McAtackney, P. (2002). Dynamic time-series forecasting using local approximation. *IEEE Press*, pp.392–399. https://doi.org/10.1109/tai.1998.744870

Sousa, M., Tom, A.M. and Moreira, J. (2022). Forecasting hourly retail customer flow on intermittent time series with multiple seasonality. *Data Science and Management*, pp.137–148. https://doi.org/10.1016/j.dsm.2022.07.002

Taylor, J.W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8), pp.799–805. https://doi.org/10.1057/palgrave.jors.2601589

Taylor, S.J. and Letham, B. (2018). Forecasting at Scale. *The American Statistician*, [online] 72(1), pp.37–45. https://doi.org/10.1080/00031305.2017.1380080

Trull, O., García-Díaz, J.C. and Peiró-Signes, A. (2022). Multiple seasonal STL decomposition with discrete-interval moving seasonalities. *Applied Mathematics and Computation*, 433, p.127398. https://doi.org/10.1016/j.amc.2022.127398

Umer, M.A., Junejo, K.N., Jilani, M.T. and Mathur, A.P. (2022). Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection*, p.100516. https://doi.org/10.1016/j.ijcip.2022.100516

Vieira, A., Sousa, I. and Dória-Nóbrega, S. (2023). Forecasting daily admissions to an emergency department considering single and multiple seasonal patterns. *Healthcare Analytics*, 3, p.100146. https://doi.org/10.1016/j.health.2023.100146

Wang, X., Kang, Y., Hyndman, R.J. and Li, F. (2023). Distributed ARIMA models for ultra-long time series. *International Journal of Forecasting*, 39(3), pp.1163–1184. https://doi.org/10.1016/j.ijforecast.2022.05.001

Williams, A.J., Sperl, R.E. and Chung, S. (2023). Anomaly Detection in Multi-Seasonal Time Series Data. *IEEE Access*, 11, pp.106456–106464. https://doi.org/10.1109/access.2023.3317791

Yang, L., Wen, Q., Yang, B. and Sun, L. (2021). A Robust and Efficient Multi-Scale Seasonal-Trend Decomposition. *IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP) |. https://doi.org/10.1109/icassp39728.2021.9413939

Yu, C., Xu, C., Li, Y., Yao, S., Bai, Y., Li, J., Wang, L., Wu, W. and Wang, Y. (2021). Time Series Analysis and Forecasting of the Hand-Foot-Mouth Disease Morbidity in China Using an Advanced Exponential Smoothing State Space TBATS Model. *Infection and Drug Resistance*, Volume 14, pp.2809–2821. https://doi.org/10.2147/idr.s304652

APPENDICES

Appendix A

Acceptance Letter of International Conference Proceedings

ICONMAA 2024

International Conference on Mathematical Analysis and Its Applications 2024
ITB Ganesha Campus (Hybrid), 31 July - 2 August 2024
Website: https://fmipa.itb.ac.id/ICONMAA2024
Email: iconmaa2024@gmail.com

Date: 18 July 2024

Letter of Acceptance for Abstract

Dear Authors: Kong Hoong Lem, Yi Xian Yap

We are pleased to inform you that your abstract (ABS-97, Oral Presentation), entitled:

"Multiple seasonal time series forecast using decomposition, ARIMA model and discrete Fourier transform: a preliminary study."

has been reviewed and accepted to be presented at ICONMAA 2024 conference to be held on 31 July - 2 August 2024 in Bandung, Indonesia.

Please submit your full paper and make the payment for registration fee before the deadlines, visit our website for more information.

Thank You.

Best regards,

Yudi Soeharyadi, Ph.D.
ICONMAA 2024 Chairperson

Konfrenzi.com - Conference Management System

Appendix B

Accepted Abstract

:: Abstract ::

<< back

Multiple seasonal time series forecast using decomposition, ARIMA model and discrete Fourier transform: a preliminary study.

Kong Hoong Lem, Yi Xian Yap

Universiti Tunku Abdul Rahman

Abstract

Multiple seasonalities often appear in time series observed at high frequency. For example, an hourly observed data may exhibit multiple seasonal patterns due to combination of daily, weekly, monthly or even yearly periodicity. In this study, we first decomposed the data into components using MSTL. For the seasonal components, we leveraged the properties of discrete Fourier transform to serve as a regressor, whereas the trend and the remainder components underwent an ARIMA model. Experiments were done on two datasets and compared with the TBATS approach. The proposed method yielded superior forecast performance.

Keywords: multiple seasonal time series, discrete fourier transform, ARIMA, MSTL

Topic: Others

Plain Format | Corresponding Author (Kong Hoong Lem)

Share Link https://confbeam.net/2024/iconmaa/kfz/abstract/fmTqdgpCD Copy Text

Share your abstract link to your social media or profile page

Appendix C

R Codes for the MSTL-DFT-ARIMA model

```r
```{r}
# Load necessary libraries
library(forecast)
library(Metrics) # to compute errors
library(fpp2)
library(timetk)
library(scales)

# Load the dataset
dataset = taylor
setFreq = frequency(taylor)

# Initialize variables
loops     = 5           # number of stepback-loops (runs)
stepback  = 20
dlist.k   = list()      # the individual k-th data list
dlist     = list()      # the overall data list
errmtx    = {}          # create a blank matrix
timelist1 = list()
timelist2 = list()

# Loop through the steps
for (k in 0:(loops-1)) {
  sample_size = 1600 - k * stepback # the chop amount from original dataset
  test_size = 200                   # testing set length
  hn = 1000                         # true forecast horizon beyond test data
```

```r
  # Chop dataset
  x0 = head(dataset, sample_size)             # original data
  x1 = head(x0, sample_size - test_size)      # training set
  x2 = tail(x0, test_size)                    # testing set

  # Check lengths
  lenx1 = sample_size - test_size             # length of training set
  lenx2 = test_size                           # length of testing set
  t1 = Sys.time()

  # MSTL the training set and extract components
  mstlset = mstl(x1)

  for (i in 1:ncol(mstlset)) {
    assign(colnames(mstlset)[i], mstlset[, i])
  }

  x9 = Trend + Remainder

  if (ncol(mstlset) == 5) {
    seasonalSum = mstlset[, 3] + mstlset[, 4]
  } else if (ncol(mstlset) == 6) {
    seasonalSum = mstlset[, 3] + mstlset[, 4] + mstlset[, 5]
  } else if (ncol(mstlset) == 7) {
    seasonalSum = mstlset[, 3] + mstlset[, 4] + mstlset[, 5] + mstlset[, 7]
  }
```
```

```
  # Create xregressor for combined seasonal 48 + 336
  lendftdata = setFreq * 3     # =1008, multiple of 336, length of dft-data
  dftdatas = head(seasonalSum, lendftdata)
  fftsl = fft(dftdatas) / lendftdata

  # invfft-extend for the length = sample_size
  # (before train-test length of the ts)
  xexts = c()
  inner.k = 1:lendftdata # inner index for inv-dft, is 1 to length of dft data
  for (j in 1:(sample_size + hn)) {  # extend beyond training size
    xexts[j] = sum(fftsl * exp(2i * pi * (inner.k - 1) * (j - 1) / lendftdata))
  }
  xexts.re = Re(xexts) # extract the real-part (since it is complex)

  # Model-fit, forecast combined-seasonal 48+336 and extract fcast-data
  mdls = auto.arima(seasonalSum, seasonal = FALSE, xreg = xexts.re[1:lenx1])
  summary(mdls)

  fcs = forecast(mdls, xreg = xexts.re[(lenx1 + 1):(lenx1 + hn)], h = hn)
  fcsdat = fcs$mean

  # Model-fit, forecast combined remaining and extract fcast-data
  md19 = auto.arima(x9, seasonal = FALSE)
  summary(md19)
  summary(md19,trace=TRUE)
  fc9 = forecast(md19, h = hn)
  fc9dat = fc9$mean
```

```
   # Combine forecast data
   fcdat = fc9dat + fcsdat
   dlist.k$dat = x0
   dlist.k$dat.train = x1
   dlist.k$dat.test = x2
   dlist.k$dat.fc = fcdat
   dlist = c(dlist, list(dlist.k)) # make the whole dlist.k as a list also

   t = difftime(Sys.time(), t1, unit = "secs")

   # Create accuracy matrix
   # using only the first part of the forecast within test data
   rmse1 = sqrt(mean((x2 - head(fcdat, lenx2))^2))
   mae1  = mean(abs(x2 - head(fcdat, lenx2)))
   mape1 = mean(abs((x2 - head(fcdat, lenx2)) / x2)) * 100

   errmtx = cbind(errmtx, c(rmse1, mae1, mape1))

   # Print loop indicator
   cat(k+1,', ',t,' secs.\n')
   timelist1[[k+1]] = as.numeric(t)
}

rownames(errmtx) = c('rmse', 'mae', 'mape')
colnames(errmtx) = paste0('fft-arima', 1:loops)

cat('\n\n')
errmtx
```

Appendix D

R Codes for the TBATS model

```r
```{r}
loops = 5 # amount of stepback loop
stepback = 20
bats.dmtx = {}    # empty matrix to store tbats-fcast
bats.errmtx = {}  # create a blank matrix

for (k in 0: (loops-1)){
  # chop dataset, set train test & length
  sample_size = 1600 - k * stepback       # chop amount from original dataset
  test_size = 200                         # training set length
  hn = 1000
  x0 = head(dataset, sample_size)           # original data
  x1 = head(x0, sample_size - test_size)      # training set
  x2 = tail(x0, test_size)                # testing set
  lenx1 = sample_size - test_size             # length of training set
  lenx2 = test_size                       # length of testing set

  # tbats model-fit & forecast
  t1 = Sys.time()
  tb1 = tbats(x1)
  fctb = forecast(tb1, h = hn)
  bats.dmtx = rbind(bats.dmtx,fctb$mean)
  t = difftime(Sys.time(), t1, unit = "secs")
  # "auto","secs","mins","hours","days","weeks"
  cat(k+1,', ',t,' secs.\n')
  timelist2[[k+1]] = as.numeric(t)
```

```r
   #accuracy and errors
   acc.auto = forecast::accuracy(fctb,x0)
   acc.tbats = acc.auto[2,c('RMSE','MAE','MAPE')]
   bats.errmtx = cbind(bats.errmtx, acc.tbats)
}

rownames(bats.errmtx) = c('rmse','mae','mape')
colnames(bats.errmtx) = paste0('tbats-', 1:loops)

cat('\n')
bats.errmtx
```
```

Appendix E

R Codes for Plotting the Training Set, Testing Set, and Forecasted Results

```
Control k to indicate loop count
Compiled Results for above date_time display chunk
#loop1 k = 1 2000-07-04 03:30:00
#loop2 k = 2 2000-07-03 17:30:00
#loop3 k = 3 2000-07-03 07:30:00
#loop4 k = 4 2000-07-02 21:30:00
#loop5 k = 5 2000-07-02 11:30:00
```

```
```{r}
k = 5

dat.train = dlist[[k]]$dat.train
dat.test  = dlist[[k]]$dat.test
dat.fc    = dlist[[k]]$dat.fc
bats.fc   = bats.dmtx[k,]
```
```

```
```{r}
timeh = seq(as.POSIXct('2000-07-02 11:30:00'), length = hn, by = '30 min') |> format('%e-%b-%y %H:%M')
idx = seq(1, by = 48, length = 11) # 2*24=48=daily
timehidx = timeh[idx]

# Adjust the bottom margin to create space for the legend
par(mar = c(7, 5, 4, 2) + 0.1)

l = length(c(tail(as.numeric(dat.train), 200), as.numeric(dat.test)))

plot(
  #c(tail(as.numeric(x1), 200), as.numeric(dat.test)),
  tail(as.numeric(x1), 200),
  type = 'l',
  xaxt = 'n',
  xlim = c(1, 500),
  lwd = 2,
  main = paste0("Plot of Training, Test, and Forecasts"),
  xlab = "Time",
  ylab = "Electricity Demand",
  col = c('black')
)
```

```
# Custom x-axis
axis(1, at = idx, labels = FALSE)
text(
  x = idx,
  y = par("usr")[3] - 1000, # Adjust this value as needed to position the labels correctly
  labels = timehidx,
  adj = 1,
  srt = 45,
  xpd = NA,
  cex = 0.6
)

start_pos = 201

lines(seq(start_pos, by = 1, length.out = length(dat.test)), as.numeric(dat.test), col = 'red')
lines(seq(start_pos, by = 1, length.out = length(dat.fc))  , as.numeric(dat.fc)  , col = 'green')
lines(seq(start_pos, by = 1, length.out = length(dat.fc))  , as.numeric(bats.fc) , col = 'blue')

# Add grid and background tweaks
abline(v = idx, lty = 2, col = alpha('blue1', 0.5))

grid(
  nx = NA,
  ny = NULL,
  lty = 2,
  col = gray(0.8),
  lwd = 1
)
```
```

```
legend(
  "bottom",
  inset = c(0, -0.6),
  legend = c("Train Data", "Test Data", "MSTL-DFT-ARIMA", "TBATS"),
  col = c("black", "red", "green", "blue"),
  xpd = TRUE,
  horiz = TRUE,
  lty = 1,
  cex = 0.6
)
```

Appendix F

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1

FACULTY OF SCIENCE

Full Name(s) of Candidate(s)	YAP YI XIAN
ID Number(s)	20ADB05640
Programme / Course	STATISTICAL COMPUTING AND OPERATIONS RESEARCH
Title of Final Year Project	FORECASTING DATA WITH LONG MULTI-SEASONAL PERIODS IN THE ARIMA MODEL USING DISCRETE FOURIER TRANSFORM REGRESSORS

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: **14** **%** **Similarity by source** Internet Sources: **11** % Publications: **9** % Student Papers: **3** %	
Number of individual sources listed of more than 3% similarity: **-**	

Parameters of originality required and limits approved by UTAR are as follows:

(i) Overall similarity index is 20% and below , and

(ii) Matching of individual sources listed must be less than 3% each , and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Lem Kong Hoong

Date: 3 Sept 2024

Signature of Co-Supervisor

Name: _____

Date: _____