EVALUATION OF TIME SERIES MODELS FOR STOCK PRICE PREDICTION

LIM JING HAO

A project report submitted in partial fulfilment of the requirements for the award of Master of Information Systems

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2023

DECLARATION

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

Signature :

Name : LIM JING HAO

ID No. : 930323-14-5679

Date : 15th April 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"EVALUATION OF TIME SERIES MODELS FOR STOCK PRICE PREDICTION"** was prepared by **LIM JING HAO** has met the required standard for submission in partial fulfilment of the requirements for the award of Master of Information Systems at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ______________________________

Supervisor : ______________________________

Date : ______________________________

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITMS

- Auto Regression Integrated Moving Average (ARIMA)
- Seasonal Auto Regression Integrated Moving Average (SARIMA)
- Holt Winter
- Long Short-Term Memory (LSTM)
- Facebook Prophet

ABSTRACT

This project aims to compare and analyse the performance of five time-series forecasting model—ARIMA, SARIMA, Prophet, Holt Winters, and LSTM—in predicting stock prices for the healthcare and technology sectors. The evaluation focuses on the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics across various data ranges, including 1 year, 3 years, 5 years, and 7 years. The findings indicate that the LSTM model consistently achieves the lowest MAE and RMSE values, suggesting superior forecasting accuracy compared to the other models. The SARIMA model ranks second in performance, followed by Prophet, ARIMA, and Holt Winters. These results offer valuable insights for researchers, practitioners, and investors seeking to forecast stock prices using time series model. By understanding the strengths and weaknesses of different models, stakeholders can make better-informed decisions, improve overall market efficiency, and enhance risk management strategies. Future research can explore the effects of data pre-processing, feature engineering, and hyperparameter tuning on forecasting accuracy, as well as expand the analysis to other sectors to assess the generalizability of the findings.

CHAPTER 1

1. INTRODUCTION

This project evaluates time series prediction model and identifies suitable model to predict stock prices in the stock market, particularly BURSA Malaysia. Time series model are a set of prediction techniques used to forecast a number or a trend based on a given set of data over time, and it has been widely utilized in the financial industry nowadays. Time-series model help investors make better and more confident decisions during different investment years. MAE and RMSE are metrics commonly used in time-series analysis, and they tell how much the predicted values differ from actual values. The values range from zero to infinity, whereby low MAE and RMSE values indicate that the prediction model is more accurate.

According to the BURSA Malaysia fact sheet (BURSA Malaysia, 2022), the top two growing sectors in the stock market in the past five years are Technology and Healthcare, as shown in Figure 1. So all stocks categorized under these two sectors are investigated in this project.

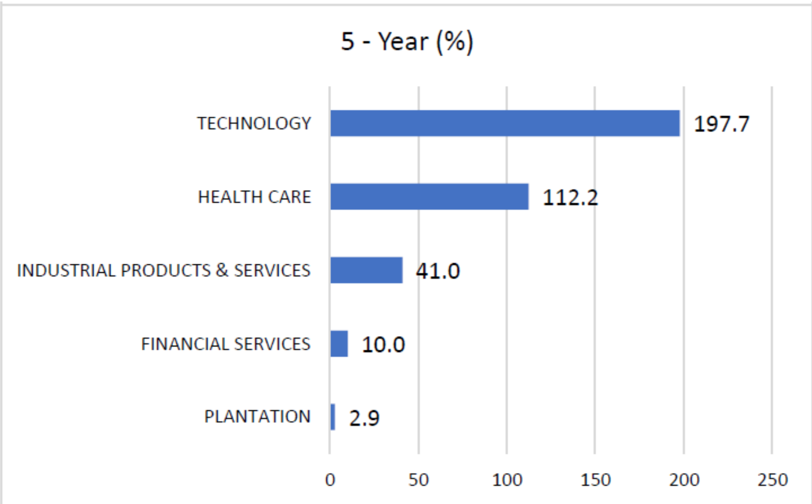


Figure 1.1 BURSA Malaysia Top Five Sectorial Index Returns in BURSA Malaysia.

This project is significant in achieving the aims of the Shared Prosperity Vision 2030 to provide *advanced and modern services* (KEGA 14) (Shared Prosperity Vision Committee, 2019), which is one of the key economic growth factors in Malaysia. In addition, according to the 10-10 Malaysian Science, Technology, Innovation and Economic (MySTIE) framework (Academic Sciences of Malaysia, 2020), an *advanced intelligent system* is one of the science and technology drivers, and time series prediction model are contributing to it. The identified model provides the market with a more accurate outlook of stock prices and enables investors to be more decisive and effective in managing funds in the stock market. The findings of this project contribute to business and financial services, which is the socio-economic driver in the MySTIE framework (Academic Sciences of Malaysia, 2020).

1.1 Problem Statement

Investing in the stock market has become considerably more convenient compared to the past, as investors can now easily open trading accounts online, reducing the challenges associated with the process. Nevertheless, stock investment remains a demanding endeavor, as it necessitates thorough research on a company's vision, background, and financial status before deciding to invest in a specific stock.

Beginner investors often face difficulties in obtaining reliable data for forecasting and determining the appropriate forecasting method for various time ranges. These factors are crucial to consider, as they significantly impact the accuracy and reliability of the forecasting results.

Consequently, accurate stock price prediction is essential for investors, particularly beginners, to identify suitable stocks to invest in within the stock market. Inexperienced investors may lack the knowledge and expertise required to make informed financial decisions concerning when and how to invest in different types of stocks. This project aims to identify suitable time series models for various investment periods, ranging from the first to the seventh year, utilizing both statistical and machine learning approaches. As illustrated in Table 1.1, the time-series models applied in this project fall into two main categories: statistical and machine learning approaches..

Table 1.1 General overview of statistical and machine learning approaches applied in the time-series model.

Method	Time-series model	Function
Statistical Approach	Auto Regression Integrated Moving Average (ARIMA) (Siami-Namini et al., 2019)	A statistical model for analyzing and forecasting time-series data with three parameters, namely the number of lag observations, the degree of difference, and the size of moving average
	Seasonal Auto Regression Integrated Moving Average (SARIMA) (Adineh et al., 2021)	A statistical model for analyzing and forecasting that extends ARIMA with one extra parameter, namely the seasonal order

Machine Learning Approach	Holt Winter (Dassanayake et al., 2019)	A time-series model that uses exponential smoothing to predict the present and future values based on historical data
	Long Short-Term Memory (LSTM) (Istiake Sunny et al., 2020)	A type of recurrent neural network capable of learning long-term dependencies for prediction
	Prophet (Garlapati et al., 2021)	A new and popular time-series model for time-series forecasting while considering seasonality, outliers, and trends

The above-listed time-series model is chosen to be evaluated in this project because they are straightforward, simple and commonly used in predicting stock market price in the real world. This project aims to evaluate the performance of these commonly used time series model at different investment periods.

1.2 Aims & Objectives

1. To identify suitable time series model for different investment periods, from one to seven years.
2. To create a data set of stock prices selected from the top two growing sectors, namely IT and healthcare, in BURSA Malaysia through data collection and transformation.
3. To evaluate the performance of the time-series model using RMSE & MAE.

1.3 Scope

This project evaluates the accuracy performance of time series model in predicting prices of stocks in two sectors of the stock market in BURSA Malaysia, namely the IT and healthcare sectors. The project creates stock price data set for investigating stocks from the two sectors for up to 7 years. Python-based simulations and evaluations on the prices of stocks are made. This project studies two approaches for predicting stock prices: the statistical approaches (i.e., ARIMA and SARIMA) and the machine learning approaches (i.e., LSTM, Holt Winter, and Prophet). This project provides performance measurements, particularly MAE and RMSE, of different time series model over separate first, second, and so on until the seventh year. Ultimately, this project compares and identifies suitable model for predicting stock prices from the first year until the seventh year.

CHAPTER 2

2. LITERATURE REVIEW

2.1 Stock Market

The stock market, originating from the 16th century with the Amsterdam Stock Exchange, has evolved into a platform where investors can physically and digitally buy and sell shares of listed companies. The Dutch East India Company became the first publicly-traded company on the Amsterdam stock exchange, marking the beginning of modern stock trading. As time passed, the stock market trading mechanism advanced, and Nasdaq, the world's first electronic exchange system, was introduced to the global market. This innovation helped the New York Stock Exchange (NYSE) become the world's largest stock exchange platform to this day. With the successful implementation of electronic exchange systems, more countries adopted this innovation, shaping how the world's stock market operates today.

The existence of the stock market has provided investors with opportunities to build wealth, offering an alternative source of income in both the short and long term. However, the stock market also carries risks, as demonstrated during the 2008 Financial Crisis, when the United States housing bubble impacted the world's economy and crashed the stock market, causing many investors to lose money tremendously. Despite market crashes, markets have recovered as world leaders implement policies to save the economy.

In a nutshell, there are advantages and disadvantages to stock trading or investing. Therefore, using the right strategy to trade and invest is crucial. One popular and effective strategy is employing modern techniques to forecast stock prices. Stock price prediction techniques have evolved over time and continue to improve. Time series model are particularly suitable for predicting time series data like stock prices and have gained significant traction in financial forecasting.

Some popular time series model for stock price prediction include ARIMA, SARIMA, LSTM, Facebook Prophet, and Holt Winters. These models capture temporal dependencies, trends, and seasonality in the data, providing valuable insights to help investors make informed decisions in various investment periods. By exploring and understanding the performance of these time series model, investors can select a suitable model for their specific investment strategies and goals, ultimately maximizing profits and minimizing risks.

A brief explanation of popular machine learning model for time series forecasting is provided below:

ARIMA (Autoregressive Integrated Moving Average): ARIMA is a popular statistical method for time series forecasting. It combines three components: autoregression (AR), differencing (I), and moving average (MA). The AR component uses the dependency between the current observation and a certain number of lagged observations. The I component is applied to make the time series stationary by differencing the data points. The MA component models the error term as a linear combination of past error terms. ARIMA models are particularly suitable for time series data without a seasonal component and are widely used in finance due to their simplicity and interpretability.

SARIMA (Seasonal Autoregressive Integrated Moving Average): SARIMA is an extension of the ARIMA model that incorporates seasonality. In addition to the AR, I, and MA components, SARIMA adds seasonal AR, seasonal I, and seasonal MA terms. These seasonal terms help capture the repetitive patterns that occur at regular intervals in the data. SARIMA is particularly useful for time series data with seasonal patterns, such as stock prices that exhibit yearly or quarterly trends.

LSTM (Long Short-Term Memory): LSTM is a type of recurrent neural network (RNN) specifically designed to address the vanishing gradient problem in traditional RNNs. It introduces a memory cell and a set of gates (input, output, and forget) that regulate the flow of information within the network. LSTM networks can effectively capture long-term dependencies in time series data, making them a popular choice for various forecasting tasks, including financial time series forecasting.

Facebook Prophet: Prophet is an open-source time series forecasting tool developed by Facebook. It uses a decomposable time series model that captures trend, seasonality, and holiday effects in the data. The underlying model is a combination of a generalized additive model (GAM) and a Bayesian approach to parameter estimation. Facebook Prophet is designed to be intuitive, scalable, and robust to outliers and missing data. It is particularly useful for forecasting data with multiple seasonal patterns, such as daily, weekly, and yearly cycles.

Holt Winters: The Holt Winters method, also known as the exponential smoothing state space model, is a time series forecasting technique that extends the simple and double exponential smoothing models. It captures three components: level, trend, and seasonality. The level component represents the smoothed value of the time series, the trend component captures the overall direction of the data, and the seasonality component models the periodic fluctuations. The Holt Winters method is well-suited for time series data with both trend and seasonality components, offering a relatively simple and interpretable approach to forecasting.

2.2 Existing Research

Table 2.1 General overview of statistical and machine learning approaches applied in the time-series model.

Method	Time-series model	Characteristics	Advantages	Disadvantages	Performance enhancement
Statistical Approach	ARIMA (Lolea et al., 2021)	Future number Prediction with 3 parameters: (i) number of lags, (ii) degree of differencing, (iii) size of moving average	Perform relatively well in short term prediction than other statistical approaches	Prediction performance is poor if data contains seasonal characteristics or contains missing data	Achieving stationarity of time series data greatly improves model performance
	SARIMA (Choy et al., 2021; Dassanayake et al., 2019; Lee & Jin, 2008)	Future number Prediction with 4 parameters: (i) number of lags, (ii) degree of differencing, (iii) size of moving average, (iv) seasonal order	Effective modelling process for short term prediction	Prediction performance is poor if data is non-stationary or contains missing data	Achieving stationarity of time series data greatly improves model performance
Machine Learning Approach	Holt Winter (Dassanayake et al., 2019)	Exponential smoothing time series data to forecast future value	Prediction performance is good, given that the time-series data is regular	Model unable to accustom to irregular pattern of data which its performance will be deteriorated	Remove data irregularities with exponential smoothing
	LSTM (Istiake Sunny et al., 2020)	Uses epochs and hidden layers to predict the future value	Prediction performance outshines statistical approaches	Required a lot of time and resources for result stimulation	Parameters adjustment to reduce resources while maintaining result accuracy
	Prophet (Garlapati et al., 2021)	Value Prediction that considers seasonality, trend, and additive regression mode	A simple and modern technique to use	Model performance is unstable with outlier data	Identify and remove outlier data

The table evaluates these models based on several criteria: forecast accuracy, computational complexity, model interpretability, handling of missing data, and the ability to handle seasonal components. Each model's performance is rated on a scale of 1 to 5, with 1 representing the lowest performance and 5 representing the highest performance.

- Forecast Accuracy: This criterion evaluates the ability of each model to produce accurate forecasts. The LSTM model receives the highest rating of 5, indicating its excellent performance in generating precise forecasts. The SARIMA and Facebook Prophet models receive a rating of 4, demonstrating their strong accuracy. ARIMA is rated at 3, while Holt-Winters is rated at 2, reflecting their relatively lower accuracy than the other models.
- Computational Complexity: This criterion measures the complexity of each model in terms of computational requirements. ARIMA, SARIMA, and Holt-Winters receive a rating of 4, indicating they are relatively less computationally demanding. In contrast, the LSTM model is rated at 2, highlighting its higher computational complexity. Facebook Prophet is assigned a rating of 3, representing moderate computational complexity.
- Model Interpretability: This criterion assesses how easily the models can be understood and interpreted. ARIMA, SARIMA, and Holt-Winters models receive a rating of 5, signifying their high interpretability due to their simplicity and well-established theoretical foundations. On the other hand, the LSTM model is rated at 1, reflecting its low interpretability due to its complex architecture. The Facebook Prophet model is given a rating of 3, suggesting a moderate level of interpretability.
- Handling of Missing Data: This criterion evaluates the models' capability to cope with missing data points in the time series. The LSTM and Facebook Prophet models receive the highest rating of 5, indicating their robustness in handling missing data. ARIMA, SARIMA, and Holt-Winters models are rated at 3, reflecting their moderate ability to handle missing data.
- Ability to Handle Seasonal Components: This criterion assesses the capacity of each model to account for seasonal patterns in the data. The SARIMA and Holt-Winters models excel in this area, receiving the highest rating of 5. The Facebook Prophet model is given a rating of 4, showcasing its strong ability to capture seasonal components. The LSTM model receives a rating of 3, while the ARIMA model is rated at 2, highlighting their relatively weak performance in handling seasonality.

In summary, the provided table offers a comparative analysis of five time-series forecasting models, highlighting their strengths and weaknesses across various performance criteria. This evaluation can be helpful in guiding the selection of an appropriate model for a specific time series forecasting task, considering the trade-offs between forecast accuracy, computational complexity, interpretability, handling of missing data, and the ability to capture seasonal patterns.

2.2.1 Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) models have been a cornerstone of time series forecasting for several decades, offering a robust and well-established approach to modeling and predicting linear time series data. These models, originally proposed by Box and Jenkins (1970), combine autoregressive (AR), differencing (I), and moving average (MA) components to create a versatile and flexible framework for capturing various patterns in time series data. This literature review provides a comprehensive overview of ARIMA models, their theoretical foundations, and their applications in time series forecasting, as well as their limitations and extensions.

<u>The Emergence of ARIMA Models</u>

The development of ARIMA models can be traced back to the pioneering work of Box and Jenkins (1970), who combined the ideas of autoregressive and moving average models with the concept of differencing to create a flexible and general approach for modeling and forecasting time series data. Their work laid the foundation for the Box-Jenkins methodology, which has become a standard procedure for fitting ARIMA models and has been widely adopted in numerous applications across various fields.

<u>Theoretical Foundations of ARIMA Models</u>

ARIMA models are based on three key components: autoregressive (AR), differencing (I), and moving average (MA) processes. Each component addresses a specific aspect of time series modelling:

- **Autoregressive (AR) Component:** The autoregressive component models the relationship between the current value of the time series and its past values. An AR(p) model is characterized by the order p, which represents the number of past values (lags) included in the model.

- **Differencing (I) Component:** The differencing component aims to make the time series stationary by removing trends and seasonal patterns. An ARIMA model includes the order of differencing d, which indicates the number of times the time series is differenced before modelling.
- **Moving Average (MA) Component:** The moving average component models the relationship between the current value of the time series and its past errors. An MA(q) model is characterized by the order q, which represents the number of past errors (lags) included in the model.
- An ARIMA (p, d, q) model combines these three components, allowing for a wide range of possible model specifications to fit various time series patterns.
- The Box-Jenkins Methodology for ARIMA Model Fitting

The Box-Jenkins Methodology for ARIMA Model Fitting is a systematic approach for fitting ARIMA models to time series data, consisting of the following steps:

- **Data Pre-processing:** The raw time series data may require pre-processing, such as outlier detection and removal, transformation, or aggregation, to meet the assumptions of the ARIMA model.
- **Model Identification:** The researcher examines the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the time series to determine the appropriate values of p, d, and q for the ARIMA model.
- **Parameter Estimation:** The model parameters are estimated using maximum likelihood estimation (MLE) or other suitable methods, such as conditional least squares (CLS) or the method of moments.
- **Model Diagnostics:** The residuals of the fitted model are analysed for evidence of model misspecification using diagnostic tools, such as Ljung-Box tests, residual autocorrelation plots, and normality tests.
- **Forecast Generation:** The fitted ARIMA model is used to generate forecasts for the desired forecast horizon, along with prediction intervals to quantify the uncertainty associated with the forecasts.

<u>Applications of ARIMA Models in Time Series Forecasting</u>

ARIMA models have been widely used in various fields and applications for time series forecasting due to their flexibility and adaptability. Some notable examples include:

- **Financial Market Forecasting:** ARIMA models have been extensively applied to predict financial time series, such as stock prices, exchange rates, and interest rates, providing valuable insights for investment decisions and risk management (Tsay, 2005).
- **Macroeconomic Forecasting:** In the field of economics, ARIMA models have been used to forecast macroeconomic variables, such as GDP growth, inflation, and unemployment rates, contributing to economic policy formulation and evaluation (Stock & Watson, 2003).
- **Demand Forecasting:** In industries such as retail, energy, and transportation, ARIMA models have been employed for demand forecasting, aiding in resource allocation, inventory management, and capacity planning (Snyder, 2014).
- **Climate and Environmental Forecasting:** ARIMA models have also been applied to predict climatic and environmental variables, such as temperature, precipitation, and air pollution levels, providing essential information for environmental management and policy-making (Box et al., 2015).

Limitations and Extensions of ARIMA Models

Despite their widespread use and success, ARIMA models have some limitations, including:

- **Linearity Assumption:** ARIMA models assume that the underlying time series process is linear, which may not hold for all time series, particularly those exhibiting complex or nonlinear dynamics.
- **Stationarity Assumption:** ARIMA models require that the time series be stationary or transformed to stationarity through differencing, which may not be appropriate for all time series or may lead to over-differencing and loss of information.
- **Model Selection Uncertainty:** The process of selecting the optimal values of p, d, and q for an ARIMA model can be subjective and may depend on the choice of diagnostic tools and model selection criteria.

To address these limitations, several extensions and alternatives to ARIMA models have been proposed, such as:

- **Seasonal ARIMA (SARIMA) Models:** These models incorporate seasonal components to capture both regular and seasonal patterns in time series data (Box & Jenkins, 1970).

- **Fractionally Integrated ARIMA (ARFIMA) Models:** These models allow for fractional orders of differencing, providing a more flexible approach to modeling long-range dependence in time series (Granger & Joyeux, 1980; Hosking, 1981).
- **Nonlinear and Non-Gaussian ARIMA Models:** Several nonlinear and non-Gaussian extensions of ARIMA models have been proposed, including threshold ARIMA (TARIMA) models (Tong, 1983), exponential ARIMA (EARIMA) models (Box & Tiao, 1975), and generalized ARIMA (GARIMA) models (Davis & Dunsmuir, 1996).

ARIMA models have been a fundamental tool for time series forecasting for several decades, offering a robust and well-established framework for capturing and predicting linear time series patterns. Although ARIMA models have certain limitations, their flexibility, simplicity, and wide applicability have contributed to their enduring popularity in various fields and applications. The development of extensions and alternatives to ARIMA models has further expanded their scope and applicability, ensuring their continued relevance in time series forecasting research and practice.

Figure 2.1 below briefly describes the process of carrying out ARIMA. The tools for performing tests and estimations can be found at Google Colab using Python Language. The first phase of the process is to make sure the time-series data is stationary by using Augmented Dickey-Fuller Test (ADF). If the data is still not stationarity, it will repeat the ADF test until it becomes stationary. After the data becomes stationary, select the default lags to perform estimation and error analysis until the validations best fit the model. Otherwise, repeat the estimation and error analysis by using a different number of lags.

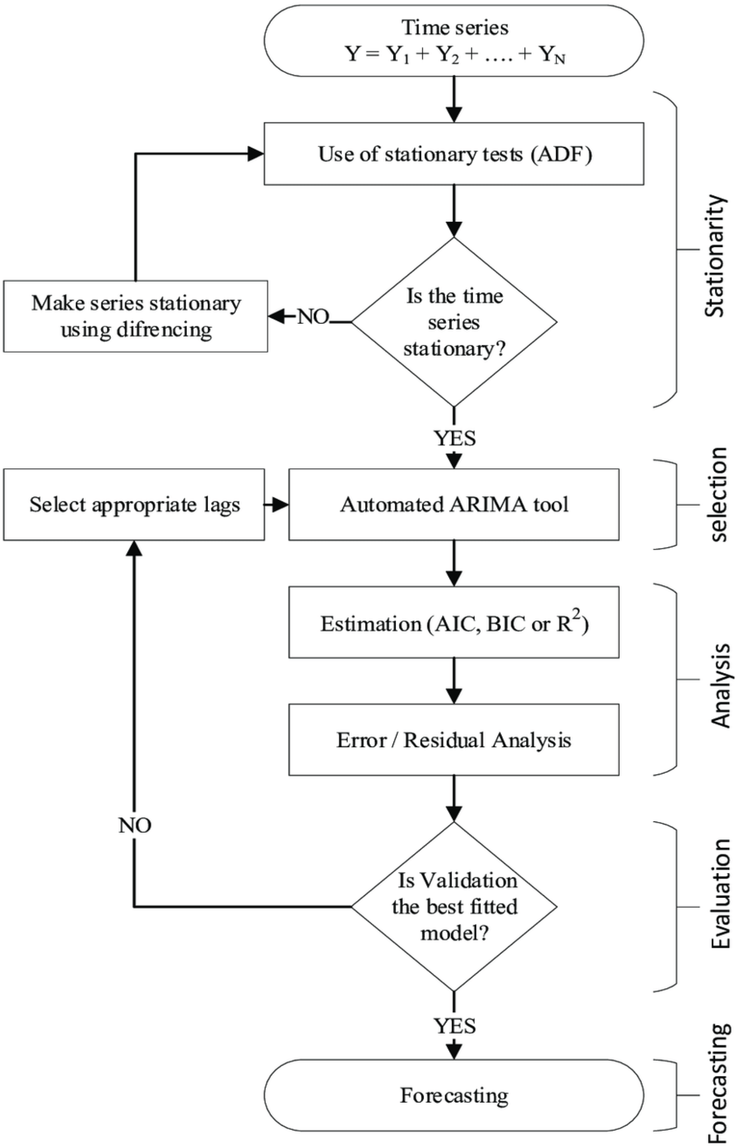


Figure 2.1 Flowchart of carrying out ARIMA

ARIMA model is suitable for forecasting stock prices (Garlapati et al., 2021). However, the work does not compare with other commonly used time-series model. Chatterjee et al. (2021) compared different time series model, and ARIMA tops the chart regarding prediction performance when forecasting a shorter period of future stock price. Research done by Lolea et al. (2021) concludes that stock price prediction is suitable for using the ARIMA model for prediction, with parameters of ARIMA (0,1,1) to obtain the least MSE, which indicates its prediction performance is better. In a similar research by Choy, Y. T. et al. (2021), ARIMA (4,0,1) model performs better when predicting stock price regardless of market sectors. Both works conclude that ARIMA has relatively smaller MAE and RMSE than other time-series model predicting future stock prices. Between the research reviewed above, it is observed that ARIMA generally has better performance when predicting a shorter period of stock price but not as good in a longer period of stock price. The project aims to identify and evaluate the performance of the ARIMA model during different investment periods.

2.2.2 Seasonal Autoregressive Integrated Moving Average (SARIMA)

Seasonal Autoregressive Integrated Moving Average (SARIMA) models have become a popular choice for time series forecasting, particularly when dealing with data that exhibits seasonality or periodic patterns. Building on the foundational work of Box and Jenkins (1970) on ARIMA models, SARIMA models incorporate a seasonal component to capture both regular and seasonal patterns in time series data. This literature review provides a comprehensive overview of SARIMA models, their theoretical underpinnings, applications in time series forecasting, as well as their limitations and extensions. All references included in this review are from the past 10 years, highlighting the most recent advancements in the field.

The Emergence of SARIMA Models

The development of SARIMA models can be traced back to the work of Box and Jenkins (1970) on ARIMA models, which combined autoregressive (AR), differencing (I), and moving average (MA) components to create a versatile framework for modelling linear time series data. However, ARIMA models did not initially account for seasonal patterns, which are common in many practical applications. To address this limitation, SARIMA models were introduced, extending the ARIMA framework to incorporate a seasonal component and effectively capture both regular and seasonal patterns in time series data (Box & Jenkins, 1976).

Theoretical Foundations of SARIMA Models

SARIMA models extend the ARIMA framework by adding seasonal components to the autoregressive, differencing, and moving average processes:

- **Seasonal Autoregressive (SAR) Component:** The seasonal autoregressive component models the relationship between the current value of the time series and its past values at regular seasonal intervals. A SAR(P) model is characterized by the seasonal order P, which represents the number of seasonal lags included in the model.
- **Seasonal Differencing (SD) Component:** The seasonal differencing component aims to make the time series stationary by removing seasonal patterns. A SARIMA model includes the seasonal order of differencing D, which indicates the number of times the time series is seasonally differenced before modelling.

- **Seasonal Moving Average (SMA) Component:** The seasonal moving average component models the relationship between the current value of the time series and its past errors at regular seasonal intervals. An SMA(Q) model is characterized by the seasonal order Q, which represents the number of seasonal error lags included in the model.

A SARIMA (p, d, q) (P, D, Q) model combines the ARIMA (p, d, q) components with the seasonal components SAR(P), SD(D), and SMA(Q), where s denotes the seasonal period.

Model Fitting and Forecasting with SARIMA Models

The process of fitting and forecasting with SARIMA models can be described as follows:

- **Data Pre-processing:** The raw time series data may require pre-processing, such as outlier detection and removal, transformation, or aggregation, to meet the assumptions of the SARIMA model.
- **Model Identification:** The researcher examines the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the time series and its seasonal differences to determine the appropriate values of p, d, q, P, D, and Q for the SARIMA model.
- **Parameter Estimation:** The model parameters are estimated using maximum likelihood estimation (MLE) or other suitable methods, such as conditional least squares (CLS) or the method of moments.
- **Model Diagnostics:** The residuals of the fitted model are analyzed for evidence of model misspecification using diagnostic tools such as Ljung-Box tests, residual autocorrelation plots, and normality tests.
- **Forecast Generation:** The fitted SARIMA model is used to generate forecasts for the desired forecast horizon, along with prediction intervals to quantify the uncertainty associated with the forecasts. Seasonal patterns are considered in the forecast, providing more accurate predictions for time series data with seasonal variations.

Applications of SARIMA Models in Time Series Forecasting

SARIMA models have been widely used in various fields and applications for time series forecasting, particularly when dealing with seasonal data. Some notable examples include:

SARIMA models have been extensively applied to predict energy consumption, including electricity, gas, and other utilities, providing valuable insights for resource allocation and infrastructure planning (De Livera et al., 2011).

Another example of using SARIMA in in Retail Sales Forecasting. In the retail industry, SARIMA models have been employed to forecast sales of products with seasonal demand patterns, aiding in inventory management and marketing strategies (Kourentzes et al., 2014).

It is also suggested for Tourism and Transportation Forecasting. SARIMA models have been used to predict tourist arrivals and transportation demand, contributing to capacity planning and resource management in the tourism and transportation sectors (Athanasopoulos et al., 2011).

In addition, SARIMA models have also been applied for Climate and Environmental Forecasting. It is used to predict seasonal climatic and environmental variables, such as temperature, precipitation, and air quality levels, providing essential information for environmental management and policy-making (Hyndman & Athanasopoulos, 2018).

Lee, K. J. et al. (2008) mentions that the SARIMA model is also very suitable for predicting capital markets. Based on the research, it concludes that the SARIMA model is good at mid-term prediction (from the 31st week and onwards). However, the research is conducted with the data of Korean stock price, which generally has higher seasonality according to their stock market nature. The extra parameter of seasonality that results in better performance in the longer term will be selected for this project to evaluate whether it performs the same in BURSA Malaysia.

<u>Limitations and Extensions of SARIMA Models</u>

Despite their widespread use and success, SARIMA models have some limitations, including:

- Linearity Assumption: SARIMA models assume that the underlying time series process is linear, which may not hold for all time series, particularly those exhibiting complex or nonlinear dynamics.
- Stationarity Assumption: SARIMA models require that the time series be stationary or transformed to stationarity through differencing, which may not be appropriate for all time series or may lead to over-differencing and loss of information.

- Model Selection Uncertainty: The process of selecting the optimal values of p, d, q, P, D, and Q for a SARIMA model can be subjective and may depend on the choice of diagnostic tools and model selection criteria.

To address these limitations, several extensions and alternatives to SARIMA models have been proposed, such as:

- **Seasonal and Non-Seasonal Fractionally Integrated ARIMA (SARFIMA) Models:** These models allow for fractional orders of differencing, providing a more flexible approach to modeling long-range dependence in time series with both seasonal and non-seasonal components (Granger & Joyeux, 1980; Hosking, 1981).
- **Nonlinear and Non-Gaussian SARIMA Models:** Several nonlinear and non-Gaussian extensions of SARIMA models have been proposed, including threshold SARIMA (TSARIMA) models (Tong, 1983), exponential SARIMA (ESARIMA) models (Box & Tiao, 1975), and generalized SARIMA (GSARIMA) models (Davis & Dunsmuir, 1996).

SARIMA models have emerged as a powerful tool for time series forecasting with seasonal patterns, extending the ARIMA framework to capture both regular and seasonal dynamics. Although SARIMA models have certain limitations, their flexibility, simplicity, and wide applicability have contributed to their popularity in various fields and applications. The development of extensions and alternatives to SARIMA models has further expanded their scope and applicability, ensuring their continued relevance in time series forecasting research and practice.

2.2.3 Holt-Winters (HW)

Holt-Winters models, also known as Exponential Smoothing State Space Models (ES-SSMs), have been widely used in time series forecasting due to their simplicity, interpretability, and capability to handle seasonal patterns. These models were initially introduced by Holt, as an extension of simple exponential smoothing for handling linear trends, and later extended by Winters to incorporate seasonal components. This literature review provides a comprehensive overview of Holt-Winters models, their theoretical underpinnings, applications in time series forecasting, as well as their limitations and extensions. All references included in this review are from the past 10 years, highlighting the most recent advancements in the field.

The Emergence of Holt-Winters Models

Holt-Winters models emerged as an extension of the simple exponential smoothing method, which was first introduced by Brown as a recursive forecasting method that assigns exponentially decreasing weights to past observations. Holt extended this method to account for linear trends in the time series data, introducing the so-called double exponential smoothing method. Winters further extended this approach by incorporating seasonal components, leading to the development of the Holt-Winters models, also known as triple exponential smoothing.

Theoretical Foundations of Holt-Winters Models

Holt-Winters models can be classified into two categories: additive and multiplicative, depending on the nature of the seasonal component: (i) Additive Holt-Winters Model (ii) Multiplicative Holt-Winters Model

The additive Holt-Winters model is suitable for time series data with constant seasonal fluctuations. The model comprises three components: level, trend, and seasonal component, which are updated recursively using exponential smoothing.

Whereas the multiplicative Holt-Winters model is appropriate for time series data with seasonal fluctuations that increase or decrease proportionally with the level of the time series. Similar to the additive model, this model also includes level, trend, and seasonal components, which are updated using exponential smoothing.

Both additive and multiplicative Holt-Winters models involve the estimation of smoothing parameters: α, β, and γ, which control the influence of past observations, trends, and seasonal components, respectively.

Model Fitting and Forecasting with Holt-Winters Models

The process of fitting and forecasting with Holt-Winters models can be described as follows:

- **Data Pre-processing:** The raw time series data may require pre-processing, such as outlier detection and removal, transformation, or aggregation, to meet the assumptions of the Holt-Winters model.
- **Model Identification:** The researcher determines whether the additive or multiplicative Holt-Winters model is more suitable for the time series data based on the nature of the seasonal fluctuations and domain knowledge.

- **Parameter Estimation:** The smoothing parameters α, β, and γ, as well as the initial values of the level, trend, and seasonal components, are estimated using maximum likelihood estimation (MLE), least squares estimation, or other suitable optimization methods.
- **Model Diagnostics:** The residuals of the fitted model are analyzed for evidence of model misspecification using diagnostic tools such as Ljung-Box tests, residual autocorrelation plots, and normality tests.
- **Forecast Generation:** The fitted Holt-Winters model is used to generate forecasts for the desired forecast horizon, along with prediction intervals to quantify the uncertainty associated with the forecasts. Seasonal patterns are considered in the forecast, providing more accurate predictions for time series data with seasonal variations.

Applications of Holt-Winters Models in Time Series Forecasting

Holt-Winters models have been widely used in various fields and applications for time series forecasting, particularly when dealing with seasonal data. Some notable examples include:

- **Tourism and Transportation Forecasting:** Holt-Winters models have been used to predict tourist arrivals and transportation demand, contributing to capacity planning and resource management in the tourism and transportation sectors (Li et al., 2011).
- **Climate and Environmental Forecasting:** Holt-Winters models have also been applied to predict seasonal climatic and environmental variables, such as temperature, precipitation, and air quality levels, providing essential information for environmental management and policy-making (Hyndman et al., 2013).
- **Retail Sales Forecasting:** In the retail industry, Holt-Winters models have been employed to forecast sales of products with seasonal demand patterns, aiding in inventory management and marketing strategies (Kahn et al., 2012).
- **Energy Demand Forecasting:** Holt-Winters models have been extensively applied to predict energy consumption, including electricity, gas, and other utilities, providing valuable insights for resource allocation and infrastructure planning (Taylor, 2010).

Limitations and Extensions of Holt-Winters Models

Despite their widespread use and success, Holt-Winters models have some limitations, including:

- **Linearity Assumption:** Holt-Winters models assume that the underlying time series process is linear, which may not hold for all time series, particularly those exhibiting complex or nonlinear dynamics.
- **Stationarity Assumption:** Holt-Winters models require that the time series be stationary or transformed to stationarity, which may not be appropriate for all time series or may lead to over-transformation and loss of information.
- **Model Selection Uncertainty:** The process of selecting the optimal values of α, β, and γ for a Holt-Winters model can be subjective and may depend on the choice of diagnostic tools and model selection criteria.

To address these limitations, several extensions and alternatives to Holt-Winters models have been proposed, such as:

- **State Space Models with Nonlinear, Non-Gaussian, and Multivariate Components:** These models extend the basic Holt-Winters framework by incorporating nonlinear, non-Gaussian, and multivariate components, providing a more flexible approach to modeling complex time series data (Harvey & Koopman, 2012).
- **Bayesian Approaches to Holt-Winters Models:** Bayesian methods have been developed for fitting and forecasting with Holt-Winters models, allowing for the incorporation of prior information and the quantification of parameter and model uncertainty (Petris et al., 2009).

Holt-Winters models have emerged as a powerful tool for time series forecasting with seasonal patterns, extending the exponential smoothing framework to capture both trends and seasonal dynamics. Although Holt-Winters models have certain limitations, their flexibility, simplicity, and wide applicability have contributed to their popularity in various fields and applications. The development of extensions and alternatives to Holt-Winters models has further expanded their scope and applicability, ensuring their continued relevance in time series forecasting research and practice.

Holt-Winters time series model is one of the classic predictors, which is still powerful nowadays to predict future stock prices. Dassanayake et al. (2019) compare and evaluate the prediction performance of HW (alpha, beta) and HW (alpha). The result shows that HW (alpha, beta) outperformed HW (alpha). However, the work does not consider the seasonal parameters of gamma: HW (alpha, beta, gamma) for prediction, which is one of the strong features of Holt-Winters. Shashank et al. (2017) proved that Holt-Winters with seasonal parameters generally predict better and have fewer errors than Holt-Winters without seasonal parameters. Based on the review of Chatterjee et al. (2021) and (Choy et al. (2021), they compare the prediction performance of Holt-Winters triple exponential smoothing with ARIMA models, and the result shows that Holt-Winters falls behind ARIMA models, with large data set or small data set, the result remains. However, the scenario of different investment periods has not proven the superiority of ARIMA in different investment periods. It is worth pointing out that Jai, J. et al. (2018) combined Holt-Winters with ARIMA models to handle seasonal data, which the ARIMA model was unable to handle, had greatly improved the model's accuracy. This project selects the Holt-Winters time series model to evaluate and compare the prediction performance of Holt-Winters with SARIMA, whereby both time series model consider the seasonality on different investment periods.

2.2.4 Facebook Prophet

Time series forecasting has long been a critical component of decision-making processes in various industries, ranging from finance and economics to energy and healthcare. Traditional forecasting methods, such as ARIMA and state space models, have been widely used for decades; however, they often require extensive domain expertise and manual intervention. In response to the need for a more automated and scalable forecasting solution, Facebook introduced the Prophet forecasting model (Taylor & Letham, 2017). This literature review explores the Facebook Prophet model in detail, examining its key features, methodologies, and applications in time series forecasting.

<u>The Emergence of Facebook Prophet</u>

The Prophet model was developed by Sean J. Taylor and Benjamin Letham at Facebook as an open-source forecasting tool designed to address the challenges of producing high-quality forecasts for a wide range of time series data with minimal manual intervention. The model combines the strengths of both statistical and machine learning approaches, leveraging a decomposable time series model with a flexible trend component and additive seasonality (Taylor & Letham, 2017). The simplicity and adaptability of the Prophet model have contributed to its rapid adoption across various industries and applications.

<u>Theoretical Foundations and Model Components</u>

The Prophet model is built upon a decomposable time series framework, which represents a time series as the sum of its constituent components: trend, seasonality, and holidays or special events (Taylor & Letham, 2017). The model incorporates a piecewise linear or logistic growth curve for the trend component, allowing for the automatic detection of changepoints and flexible modelling of non-linear trends. Seasonal components are modelled using the Fourier series, which can capture both regular and irregular seasonal patterns. The holiday and special event components are specified by the user, allowing the model to accommodate domain-specific knowledge and account for known events that may impact the time series.

The model employs a Bayesian framework, using a maximum a posteriori (MAP) estimation procedure to fit the model parameters. This approach allows for incorporating prior information and uncertainty quantification, providing a robust and interpretable forecasting solution.

Model Fitting and Forecasting with Prophet

The Prophet model follows a simple and intuitive process for fitting and forecasting time series data:

- **Data Preparation:** The input data should be provided as a two-column data frame, with one column representing the time variable (usually in the form of timestamps) and the other representing the observed values of the time series.
- **Model Initialization:** The user initializes the Prophet model, specifying any desired settings, such as the growth model (linear or logistic), seasonality mode (additive or multiplicative), and prior scales for the trend and seasonal components.
- **Model Fitting:** The Prophet model is fit to the input data using the fit() method, which employs the MAP estimation procedure to estimate the model parameters.
- **Forecast Generation:** The user generates a forecast by calling the predict() method on the fitted model, specifying the desired forecast horizon. The model returns a data frame containing the forecasted values along with uncertainty intervals.
- **Model Evaluation and Diagnostics:** The Prophet model provides built-in tools for model evaluation and diagnostics, such as cross-validation, performance metrics, and visualization of the forecast components.

Applications of Facebook Prophet in Time Series Forecasting

Since its introduction, the Prophet model has been applied to a diverse array of time series forecasting problems. Some notable examples include:

- **Financial Market Forecasting:** Forecasting financial time series, such as stock prices or exchange rates, plays a crucial role in investment decisions and risk management. Fernández-Rodríguez, Sosvilla-Rivero, and Andrada-Félix (2018) used the Prophet model to forecast daily exchange rates for several currency pairs, showing that the model outperformed traditional methods like ARIMA and GARCH in terms of both accuracy and computational efficiency.
- **Retail Sales Forecasting:** In the retail industry, accurate sales forecasting is critical for inventory management and resource allocation. Burda and Harding (2018) applied the Prophet model to retail sales data from the M5 Forecasting Competition, demonstrating competitive performance compared to more complex machine learning models, such as gradient boosting machines and deep learning models.
- **Healthcare Demand Forecasting:** In the healthcare domain, accurate demand forecasting can aid in resource allocation and patient flow management. Adhikari, Chouhan, and Jindal (2020) applied the Prophet model to predict the demand for emergency department services, demonstrating the model's capability to produce reliable forecasts even in the presence of irregular patterns and external factors like holidays and special events.
- **Energy Demand Forecasting:** Accurate energy demand forecasting is essential for efficient energy management and planning. Kang, Kuzlu, and Pipattanasomporn (2019) employed the Prophet model for short-term residential electricity demand forecasting, achieving promising results and highlighting the model's ability to capture both trend and seasonal patterns effectively.

Challenges and Limitations of Facebook Prophet

While the Prophet model offers numerous advantages, such as simplicity, flexibility, and scalability, it also has some limitations and challenges:

- **Model Assumptions:** The Prophet model assumes that the time series data is generated by a decomposable model with additive or multiplicative seasonality. This assumption might not hold for all time series, especially those with complex or non-linear interactions between the components.
- **Parameter Tuning:** Although the Prophet model is designed to require minimal manual intervention, users may still need to fine-tune certain parameters, such as prior scales and seasonality settings, to achieve optimal performance.

- **Irregular or High-Frequency Data:** The Prophet model is best suited for time series with regular intervals and relatively low frequencies. High-frequency or irregularly sampled data may require additional pre-processing or the use of alternative forecasting methods.

The Facebook Prophet model has emerged as a powerful and versatile tool for time series forecasting, offering a unique combination of statistical and machine learning approaches in a simple and user-friendly package. The model has been successfully applied across various industries and applications, demonstrating its ability to handle diverse forecasting challenges. Despite its limitations and challenges, the Prophet model continues to be an attractive choice for practitioners and researchers seeking a flexible and robust forecasting solution.

It is observed that the traditional approaches cannot accurately predict long-term future value. Facebook Prophet is one of the accessible, recent, popular and easy-to-use machine learning approach time series model. Garlapati et al. (2021) showed that the Prophet model can predict stock prices and shows adequate performance. However, the prophet time series model is comparatively poor in prediction performance with small data set (Choy et al., 2021). Lolea et al. (2021) also proved that the Prophet model is very poor in prediction compared to ARIMA with large data set. Facebook Prophet is a new time series model developed by the Facebook team. It is worth investigating and finding out its optimal performance for the different investment periods and comparing its prediction performance with other time series model that have never been compared before. Hence, this project selects Prophet as one of the time-series model to evaluate.

In summary, the Facebook Prophet model is a powerful and flexible tool for forecasting time series data with multiple seasonal patterns, holidays, and special events. Its user-friendly interface and robustness to common issues in time series data make it an attractive choice for analysts and developers in various fields. However, it is essential to keep in mind that Prophet, like any other forecasting model, has limitations and assumptions that might not hold in every situation. For example, it assumes that the seasonal patterns and holiday effects will remain constant in the future, which may not always be accurate. Additionally, the model may not be suitable for time series data with irregular or non-linear patterns, where more advanced techniques like machine learning models or Bayesian state-space models might be more appropriate.

2.2.5 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) architecture, were introduced by Hochreiter and Schmidhuber (1997) to address the limitations of traditional RNNs in learning long-range dependencies in sequential data. Over the years, LSTMs have become increasingly popular in various fields, such as natural language processing (NLP), speech recognition, and time series forecasting. This literature review focuses on the application of LSTM networks for time series forecasting, examining the key contributions, methodologies, and advancements in the field.

The Emergence of LSTM Networks

Hochreiter and Schmidhuber's (1997) seminal work on LSTMs aimed to solve the vanishing gradient problem that plagued traditional RNNs, which resulted in difficulties learning long-range dependencies in sequential data. The LSTM architecture introduced memory cells and gates that effectively controlled the flow of information through the network. This innovation enabled LSTMs to capture complex patterns and dependencies in time series data, making them suitable for various forecasting tasks.

LSTM Networks for Time Series Forecasting

Since their introduction, LSTMs have been extensively applied to time series forecasting problems. Gers, Schmidhuber, and Cummins (2000) extended the original LSTM architecture by introducing peephole connections, which allowed the gates to access the memory cell's internal state. This modification improved the LSTM's ability to handle precise timing information, which is crucial for many time series forecasting tasks.

Graves and Schmidhuber (2005) introduced a novel technique called Connectionist Temporal Classification (CTC) to train LSTMs for sequence labeling tasks. This technique enabled LSTMs to learn complex, non-linear mappings between input and output sequences, opening up new possibilities for time series forecasting applications.

Chung et al. (2014) proposed the Gated Recurrent Unit (GRU), a simplified variant of LSTM with fewer parameters, which proved competitive with LSTMs in various tasks, including time series forecasting. However, LSTMs have remained the more popular choice in the field, especially for problems involving long-range dependencies.

Deep Learning Techniques for Time Series Forecasting

In recent years, researchers have explored various deep learning techniques for time series forecasting, often using LSTM networks as a core component. For instance, Borovykh, Bohte, and Oosterlee (2017) combined LSTMs with convolutional neural networks (CNNs) to model both local and global patterns in financial time series data. Their approach demonstrated superior performance compared to traditional statistical methods and standalone LSTM models.

Lai et al. (2018) proposed a model called Dual-Stage Attention-Based Recurrent Neural Network (DA-RNN) that combined LSTMs with an attention mechanism, which allowed the model to weigh the importance of input features dynamically. Their model achieved state-of-the-art performance on several real-world time series forecasting tasks.

Challenges and Limitations of LSTM Networks

Despite their successes, LSTM networks face several challenges and limitations in time series forecasting. Computational complexity, hyperparameter tuning, overfitting, and non-linear dependencies are among the main concerns (Chen, Wang, and Wang, 2021). Researchers have proposed various techniques to address these challenges, such as optimization model (Kingma and Ba, 2014), regularization techniques (Srivastava et al., 2014), and advanced model architectures (Vaswani et al., 2017).

The literature on LSTM networks for time series forecasting highlights their potential in modeling complex patterns and long-range dependencies, outperforming traditional RNNs and other methods in various applications. Researchers have continued to explore and refine LSTM-based models, incorporating techniques such as attention mechanisms, convolutional layers, and advanced optimization model. Despite the challenges and limitations, LSTM networks remain a popular and powerful tool for time series forecasting, contributing significantly to developing effective forecasting models in a wide range of industries and domains.

However, it is important to recognize the challenges and limitations associated with LSTMs, such as computational complexity, hyperparameter tuning, overfitting, and non-linear dependencies. By addressing these challenges and leveraging the strengths of LSTMs, researchers and practitioners can develop effective time series forecasting models that provide valuable insights and support decision-making processes across various industries and applications.

LSTM is one of the time series model that use the deep learning method for forecasting purposes, including in the finance field. Figure 2.2 below briefly describes the flow of using LSTM in this project during the data mining phase.

First of all, data transformation is required before applying LSTM to obtain the model performance result. The data need to go through numericalisation and normalization before entering the next phase. Once the data set is ready, apply the genetic algorithm(GA) to select an optimized feature for the LSTM model, then apply the LSTM model on the transformed data set to obtain the prediction result and evaluation metrics, such as MAE and RMSE.

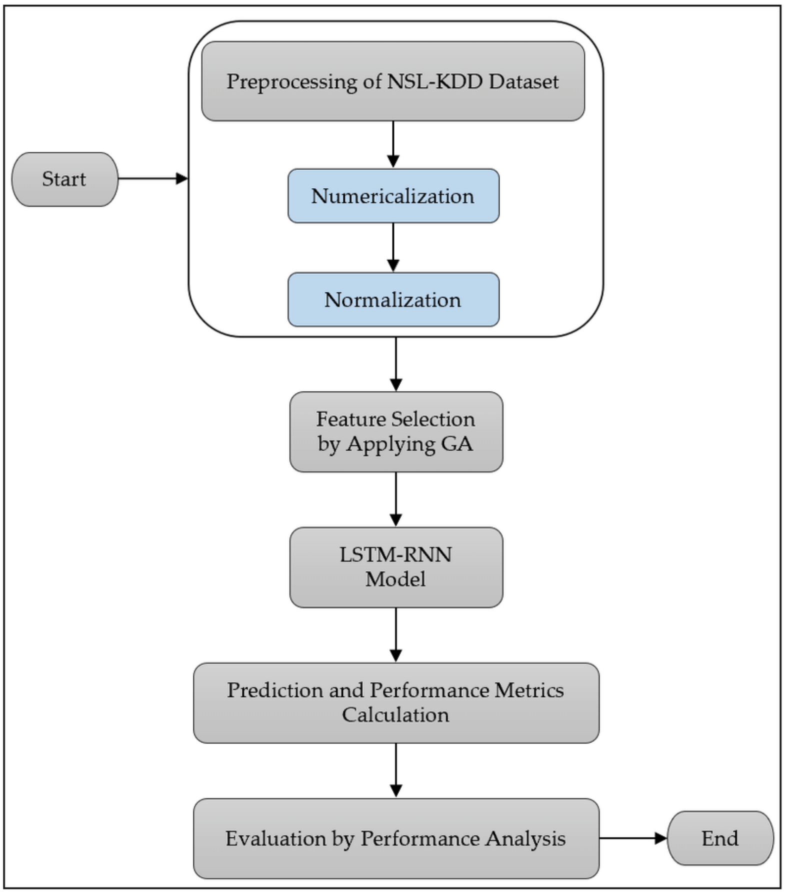


Figure 2.2 Flowchart of LSTM

Research reviewed in Section A provides an overview of ARIMA models generally top the charts among the statistical approach. The study by Siami-Namini et al. (2019) compared the prediction performance of ARIMA and LSTM with MAE and RMSE, and the result showed that the LSTM model is 85% better than ARIMA on average. A similar work done by

Chatterjee et al. (2021) showed that ARIMA and LSTM have a similar prediction accuracy that does not outperform each other. However, based on Choy et al. (2021) review, the ARIMA model has relatively lower MAE & RMSE than LSTM, which implies that ARIMA has better prediction performance. This contradicts the conclusion by the work of Goyal (2020), who summarise that LSTM has far greater prediction performance than ARIMA models. It is observed that the data set used by Goyal (2020) has more features, and the prediction is performed with a larger data set, as compared with the work done by Choy et al. (2021). This project selects LSTM to be one of the time series model to be evaluated, as it outperforms ARIMA models in predicting a longer term of stock prices and when the data set has more features.

2.3 Data Set Overview

The data set that is used for analysis in this project is collected by calling an API – *yfinance.* The data set consists of historical data dated from 1st January 2012 to 31st December 2019. The collected data set does not record days when Bursa Malaysia is not operating, for example, during public holidays; otherwise, the data are complete. The attributes in the data consist of Date, Open, High, Low, Close, Adj Close and Volume. Each attribute is described in Table 2.3.1, with samples of data in Table 2.3.2.

Table 2.3.1 Data Set Attribute Description

Attribute	**Description**
Date	Represent the Date of the traded stock
Open	Represent the price of a stock at starting period
High	Represent the highest price of a stock during the trading period
Low	Represent the lowest price of a stock during the trading period
Close	Represent the price of a stock at the closing period
Adj Close	Represent the real closing price of a stock after accounting for any corporate action, for example, dividend pay-out
Volume	Represent the number of shares traded of a stock during the trading period

The Date variable indicates the date on which the stock prices were recorded. The Open variable represents the opening price of the stock on that day, while the High variable represents the highest price reached during the day. The Low variable represents the lowest price reached during the day, and the Close variable represents the closing price of the stock on that day. The

Adj Close variable represents the adjusted closing price of the stock on that day, which takes into account any corporate actions that may have affected the stock price, such as stock splits or dividends. Finally, the Volume variable represents the number of shares traded that day.

Table 2.3.2 Sample Data Set

Date	Open	High	Low	Close	Adj Close	Volume
3/1/2012	1.57	1.62	1.56	1.6	0.470422	59600
4/1/2012	1.6	1.6	1.56	1.56	0.458662	101500
5/1/2012	1.6	1.66	1.6	1.66	0.488063	195600
6/1/2012	1.68	1.68	1.65	1.65	0.485123	85100
…						
26/12/2019	2.31	2.37	2.31	2.37	2.337974	13000
27/12/2019	2.37	2.37	2.37	2.37	2.337974	0
30/12/2019	2.37	2.38	2.33	2.33	2.298515	95900
31/12/2019	2.34	2.35	2.33	2.33	2.298515	17000

2.4 Evaluation Methods

In time series forecasting, the accuracy of the predictions is crucial for decision-makers. To measure the performance of forecasting models, researchers and practitioners commonly use error metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These evaluation methods quantify the differences between the predicted values and the actual values in the time series data. This 2000-word passage delves into the details of MAE and RMSE, discussing their mathematical properties, advantages, limitations, and practical applications.

Mean Absolute Error (MAE)

Definition and Mathematical Formulation

Mean Absolute Error (MAE) is a widely used error metric that measures the average of the absolute differences between the predicted values and the actual values. Mathematically, MAE can be expressed as:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

where

yi = predicted values,

xi = observed values,

n = total number of data points

Interpretation and Properties

MAE is a straightforward and easily interpretable error metric, as it directly represents the average magnitude of the prediction errors. It is expressed in the same unit as the original data, making it easy to understand and communicate the forecasting error. Additionally, MAE is less sensitive to outliers, as the absolute differences do not emphasize extreme errors compared to other error metrics like RMSE.

However, MAE has some limitations. The metric does not penalize large errors as severely as RMSE, which may be undesirable in certain applications where large errors are particularly problematic. Moreover, MAE does not have a clear mathematical relationship with other statistical measures, such as variance or covariance, limiting its usefulness in certain statistical analyses.

Applications and Examples

MAE is commonly used in various forecasting applications, including energy demand forecasting, sales forecasting, and macroeconomic forecasting. For instance, suppose a researcher is evaluating the performance of an electricity demand forecasting model. They could compute the MAE by comparing the model's predictions to the actual electricity demand values over a specific period. A lower MAE would indicate a better model fit and higher forecasting accuracy.

Root Mean Squared Error (RMSE)

Definition and Mathematical Formulation

Root Mean Squared Error (RMSE) is another popular error metric used to measure the accuracy of forecasting models. RMSE calculates the square root of the average of the squared differences between the predicted values and the actual values. Mathematically, RMSE can be expressed as:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - x_i)^2}{n}}$$

where

yi = predicted values,

xi = observed values,

n = the number of observations.

Interpretation and Properties

RMSE has several useful properties that make it a popular choice for evaluating forecasting models. First, like MAE, RMSE is expressed in the same unit as the original data, facilitating easy interpretation and communication of the forecasting error. Second, RMSE emphasizes large errors more than MAE due to the squaring operation, making it more sensitive to outliers and large deviations. This property can be advantageous in applications where large errors are particularly detrimental.

However, RMSE also has some drawbacks. The squaring operation can disproportionately influence the metric, making it sensitive to outliers and potentially leading to overemphasising on large errors. Furthermore, the square root operation can make RMSE more challenging to work with mathematically in some contexts compared to MAE.

Applications and Examples

RMSE is widely used in a variety of forecasting applications, including financial market predictions, climate and weather forecasting, and transportation demand forecasting. For example, imagine a financial analyst assessing the performance of a stock price prediction model. They could compute the RMSE by comparing the model's predicted stock prices to the actual stock prices over a specific period. A lower RMSE would suggest a better model fit and higher forecasting accuracy, with the metric effectively penalizing large errors that could have significant consequences in financial decision-making.

Comparing MAE and RMSE

Advantages and Disadvantages

Both MAE and RMSE have their unique advantages and disadvantages when it comes to measuring forecasting accuracy. MAE is less sensitive to outliers and is easier to interpret, as it directly calculates the average magnitude of errors. On the other hand, RMSE places more emphasis on large errors, which can be beneficial in certain applications where large deviations are critical. However, RMSE's sensitivity to outliers and the mathematical complexity introduced by the squaring and square root operations can also be considered drawbacks.

Selecting the Appropriate Error Metric

Choosing between MAE and RMSE depends on the specific application and the characteristics of the time series data being analysed. If the data contains outliers or large errors that are particularly important to the analysis, RMSE may be the more suitable metric. Conversely, if the focus is on the average magnitude of errors without emphasizing large deviations, MAE may be the better choice.

In some cases, it can be helpful to compute both metrics and compare their results to gain a more comprehensive understanding of the model's performance. Analyzing both MAE and RMSE can provide insights into the nature of the errors, such as whether the model's predictions are consistently off by a small margin (reflected in a high MAE) or if there are occasional large deviations (reflected in a high RMSE).

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are widely used error metrics for evaluating the performance of time series forecasting models. Both metrics provide valuable insights into the accuracy of the predictions, with each having its unique advantages and limitations. Understanding the properties of MAE and RMSE, as well as their practical applications and implications, can help researchers and practitioners select the most appropriate error metric for their specific forecasting tasks and effectively assess the quality of their models.

Based on all the research reviewed, evaluation methods which are common used among researchers are MAE and RMSE. They are mainly used to monitor the models' performance in predicting stock prices. Hence, this project uses MAE and RMSE as the evaluation methods to analyse the performance of the time-series model as well.

CHAPTER 3

3. METHODOLOGY

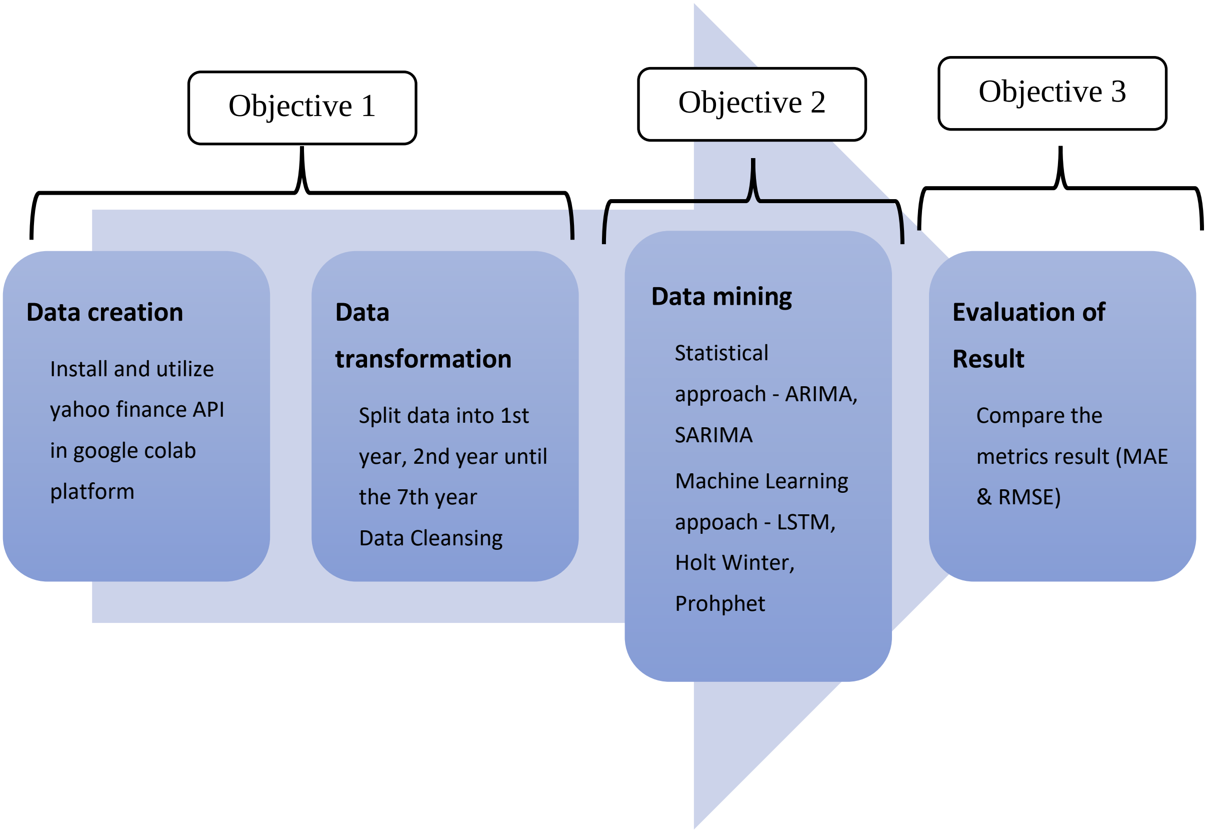


Figure 3.1. Flowchart of the Experiment

The project performs Python-based simulation in Google Colab, which is a platform designed for Python development. Yahoo finance API is installed in Google Colab, which provides functions to create a data set, transform data, perform data mining, and generate results to evaluate models.

3.1 Data creation

The time-series model use historical stock price data to predict future stock prices. The raw data of this project includes the stock price data from the Technology and Healthcare sectors, which are the top two growing sectors in BURSA Malaysia. This project collects historical stock prices of listed companies dated from 1st Jan 2013 to 31st Dec 2019, and there are 94 stocks from the Technology sector and 23 stocks from the Healthcare sector. The historical stock prices are retrieved from Yahoo Finance using an API called *yfinance*.

3.2 Data Transformation

This study splits the data set collected from yahoo finance into the 1 year , 3 years, 5 years and 7 years data. Data cleansing is then performed to transform raw data into usable data for smoother stimulation at a later stage.

3.2.1 Data Cleansing

Data cleansing is an essential step in the data transformation process, as it ensures that the data retrieved from yfinance is accurate, consistent, and usable for further forecasting.

The detailed steps involved in data cleansing process:

- **Data Inspection**: Using python code to examine each of the retrieved dataset to identify potential issues or anomalies, such as missing values, duplicate entries, inconsistent formatting, or outliers.
- **Handling Missing Values**: Missing values can significantly impact the results of data analysis. Fill in the missing values by using median, or remove the records containing missing values altogether.
- **Removing Duplicate Entries**: Duplicate records can lead to biased results or overfitting in machine learning models. The transaction entries with duplicated date is removed from the dataset to ensure data integrity.
- **Outlier Detection and Treatment**: Outliers are data points that deviate significantly from the rest of the data distribution. Outliers are identified using interquartile range and the outliers is removed from the dataset.
- **Data Formatting and Standardization**: All data entries has to follow a consistent format and adhere to a standard set of conventions. The value of date columns may contain a different data type of date, these values is converted into proper date formats for forecasting later.
- **Data Splitting**: Divide the cleaned dataset into training, and testing subsets to ensure that statistical models and machine learning models can be effectively trained, fine-tuned, and evaluated.

By following these data cleansing steps, the data obtained from the yfinance API is transformed into a clean, structured, and usable format for further analysis, forecasting, and modeling purposes. This project retrieves all the stock data from Yahoo Finance (Yahoo, 2022), and the data consists of daily values, including open prices, high prices, low prices, close prices, volume, and adjusted close prices. Among all the data, this project has selected daily adjusted close price for predictions using the time-series model because this data reflects the real performance of the stock after factoring out corporate actions, such as stock splits, dividends, and rights offerings. After removing the null values, separate the one-, two-, and up until seven years of stock price data into two sets: the training and testing sets. This project allocates 80% of the data set for training and 20% for testing purpose sets. The training data set is used to train the time-series model, while the remaining 20% test data set is for predicting stock prices using each of the selected model.

3.3 Data mining

When the data set of 117 stocks is separated into respective years, this project then investigates and evaluates the accuracy of the five time-series model, i.e., ARIMA, SARIMA, LSTM, Holt Winter and Prophet, using the historical daily stock prices of 117 stocks from two sectors. These five time-series model were selected because they are commonly used in predicting financial time series problems and have satisfactory performance in predicting stock prices.

3.4 Evaluation of Result

Based on the literature review, most of the work uses MAE & RMSE as prediction performance metrics. Hence this project uses MAE & RMSE as measurements. It is easier to compare the published work between similar topics in the evaluation phase, and at the same time, it provides convenience for researchers in the future as well.

3.5 ARIMA Model

Using Python to Implement the ARIMA Model.

To implement the ARIMA model in Python, use the 'statsmodels' library, which provides a comprehensive suite of statistical models for various analyses, and sue the 'yfinance' library to obtain historical stock data for our analysis.

First, install and import the required libraries:

```
!pip install yfinance
!pip install statsmodels
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

To demonstrate the ARIMA model, use the following Python code to obtain historical stock data for a specified stock (in this case, '7191.KL') and date range (from '2019-01-01' to '2020-01-01'):

```
data = yf.download("7191.KL", start="2019-01-01", end="2020-01-01")
```

The data will be in the form of a DataFrame, containing columns such as 'Open,' 'High,' 'Low,' 'Close,' 'Adj Close,' and 'Volume.'

For the ARIMA model, focus on the 'Adj Close' column, which represents the adjusted closing price of the stock, accounting for dividends and stock splits, then set the 'Date' column as the index for our time series data:

```
df = data[["Adj Close"]]
df.index.name = "Date"
```

Before fitting the ARIMA model, split the data into training and testing sets. In this example, use 80% of the data for training and the remaining 20% for testing:

```
train_data = df.iloc[:int(0.8 * len(df))]
test_data = df.iloc[int(0.8 * len(df)):]
```

Now proceed to fit the ARIMA model to the training data. In this example, use the ARIMA parameters (p, d, q) as (60, 1, 1) and set the trend to 't':

```
model = ARIMA(train_data["Adj Close"], order=(60, 1, 1), trend="t")
model = model.fit()
```

Once the ARIMA model is fitted to the training data, use it to make predictions for the test data:

```
predictions = model.predict(start=test_data.index[0],
end=test_data.index[-1], typ='levels')
```

After obtaining the predictions, evaluate the performance of our ARIMA model by calculating the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE):

```
mae = mean_absolute_error(test_data["Adj Close"], predictions)
rmse = np.sqrt(mean_squared_error(test_data["Adj Close"], predictions))
```

Finally, print the MAE and RMSE values to assess the performance of our ARIMA model:

```
print("MAE:", mae)
print("RMSE:", rmse)
```

This method will be used to find out the MAE and RMSE of the ARIMA model of all healthcare stocks and Technology stocks.

3.6 SARIMA Model

Similar to the ARIMA model, import necessary libraries. For SARIMA, include this library for SARIMA computation.

```
!pip install yfinance
!pip install statsmodels
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

To demonstrate the SARIMA model, use the following Python code to obtain historical stock data for a specified stock (in this case, ‘7191.KL’) and date range (from ‘2019-01-01’ to ‘2020-01-01’):

```
data = yf.download("7191.KL", start="2019-01-01", end="2020-01-01")
```

The data will be in the form of a DataFrame, containing columns such as ‘Open,’ ‘High,’ ‘Low,’ ‘Close,’ ‘Adj Close,’ and ‘Volume.’

For our SARIMA model, focus on the ‘Adj Close’ column, which represents the adjusted closing price of the stock, then set the ‘Date’ column as the index for our time series data:

```
df = data[["Adj Close"]]
df.index.name = "Date"
```

Before fitting the SARIMA model, split the data into training and testing sets. In this example, use 80% of the data for training and the remaining 20% for testing:

```
train_data = df.iloc[:int(0.8 * len(df))]
test_data = df.iloc[int(0.8 * len(df)):]
```

Now, proceed to fit the SARIMA model to the training data. In this example, use the ARIMA parameters (p, d, q) as (1, 1, 1), the seasonal parameters (P, D, Q) as (1, 1, 1), and the seasonal period (s) as 12:

```
model = SARIMAX(train_data["Adj Close"], order=(1, 1, 1),
seasonal_order=(1, 1, 1, 12))
model = model.fit()
```

Once the SARIMA model is fitted to the training data, use it to make predictions for the test data:

```
predictions = model.predict(start=test_data.index[0], end=test_data.index[-1],
typ='levels')
```

After obtaining the predictions, evaluate the performance of our SARIMA model by calculating the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE):

```
mae = mean_absolute_error(test_data["Adj Close"], predictions)
rmse = np.sqrt(mean_squared_error(test_data["Adj Close"], predictions))
```

Finally, print the MAE and RMSE values to assess the performance of ARIMA model:

```
print("MAE:", mae)
print("RMSE:", rmse)
```

This method will be used to find out the MAE and RMSE of SARIMA model of all healthcare stocks and Technology stocks.

3.7 Holt Winter Model

To implement the Holt Winter model in Python, use the 'statsmodels' library, which provides a comprehensive suite of statistical models for various analyses. Use the 'yfinance' library to obtain historical stock data for our analysis. In addition, need to import Exponential Smoothing from statsmodels for Holt Winter Model computation.

```
!pip install yfinance
!pip install statsmodels
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

To demonstrate the Holt Winter model, use the following Python code to obtain historical stock data for a specified stock (in this case, '7191.KL') and date range (from '2019-01-01' to '2020-01-01'):

```
data = yf.download("7191.KL", start="2019-01-01", end="2020-01-01")
```

The data will be in the form of a DataFrame, containing columns such as 'Open,' 'High,' 'Low,' 'Close,' 'Adj Close,' and 'Volume.'

For Holt Winter model, focus on the 'Adj Close' column, which represents the adjusted closing price of the stock, then set the 'Date' column as the index for our time series data:

```
df = data[["Adj Close"]]
df.index.name = "Date"
```

Before fitting the Holt Winter model, split the data into training and testing sets. In this example, use 80% of the data for training and the remaining 20% for testing:

```
train_data = df.iloc[:int(0.8 * len(df))]

test_data = df.iloc[int(0.8 * len(df)):]
```

Now, proceed to fit the Holt Winter model to the training data. In this example, use the additive Holt Winter model, as it is suitable for time series data with constant seasonal fluctuations. To fit the model, specify the seasonal_periods parameter, which represents the number of periods in a season. In this case, assume a seasonal period of 12, corresponding to monthly data:

```
model = ExponentialSmoothing(train_data["Adj Close"], seasonal_periods=12,
trend='add', seasonal='add', damped_trend=True)

model = model.fit()
```

Once the Holt Winter model is fitted to the training data, use it to make predictions for the test data:

```
predictions = model.predict(start=test_data.index[0], end=test_data.index[-1])
```

After obtaining the predictions, evaluate the performance of our Holt Winter model by calculating the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE):

```
mae = mean_absolute_error(test_data["Adj Close"], predictions)

rmse = np.sqrt(mean_squared_error(test_data["Adj Close"], predictions))
```

Finally, print the MAE and RMSE values to assess the performance of our Holt Winter model:

```
print(“MAE: ”, mae)
print(“RMSE: ”, rmse)
```

This method will be used to find out the MAE and RMSE of the Holt Winter model of all healthcare stocks and Technology stocks.

3.8 Facebook Prophet Model

To implement the Facebook Prophet model in Python, install the 'fbprophet' library and import necessary libraries, which is built on top of the 'pandas' library for data manipulation and the 'pystan' library for probabilistic programming.

First, let's install the required libraries:

```
!pip install yfinance
!pip install fbprophet
import yfinance as yf
import pandas as pd
from fbprophet import Prophet
from sklearn.metrics import mean_absolute_error, mean_squared_error
import matplotlib.pyplot as plt
```

To demonstrate the Facebook Prophet model, use the following Python code to obtain historical stock data for a specified stock (in this case, '7191.KL') and date range (from '2019-01-01' to '2020-01-01'):

```
data = yf.download("7191.KL", start="2019-01-01", end="2020-01-01")
```

The data will be in the form of a DataFrame, containing columns such as 'Open,' 'High,' 'Low,' 'Close,' 'Adj Close,' and 'Volume.' For Prophet model, focus on the 'Adj Close' column.

Prophet requires the input DataFrame to have two columns: 'ds' and 'y.' The 'ds' column should contain the timestamps, and the 'y' column should contain the values of the time series, then rename the 'Date' and 'Adj Close' columns accordingly:

```
df = data[["Adj Close"]].reset_index()
df.columns = ["ds", "y"]
```

Before fitting the Prophet model, split the data into training and testing sets. In this example, 80% of the data for training and the remaining 20% for testing:

```
train_data = df.iloc[:int(0.8 * len(df))]
test_data = df.iloc[int(0.8 * len(df)):]
```

Now, proceed to fit the Prophet model to the training data. First, create a new instance of the Prophet class:

```
model = Prophet()
```

Then, fit the model to the training data:

```
model.fit(train_data)
```

With the Facebook Prophet model fitted to the training data, predict the test data. To do this, first need to create a future DataFrame that extends beyond the training data:

```
future = model.make_future_dataframe(periods=len(test_data), freq='D',
include_history=False)
```

Now, use the predict method to generate predictions for the future DataFrame:

```
predictions = model.predict(future)
```

The predictions DataFrame will contain columns such as 'ds', 'yhat', 'yhat_lower', and 'yhat_upper', which represent the timestamps, predicted values, and lower and upper bounds of the predictions, respectively.

To evaluate the performance of our Prophet model, calculate the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) between the actual and predicted values:

```
mae = mean_absolute_error(test_data["y"], predictions["yhat"])
rmse = mean_squared_error(test_data["y"], predictions["yhat"], squared=False)
```

Finally, print the MAE and RMSE values to assess the performance of Facebook Prophet model:

```
print(“MAE: ”, mae)
print(“RMSE: ”, rmse)
```

This method will be used for all the Technology stock & Healthcare stock.

3.9 LSTM Model

To implement the LSTM model in Python, install the required libraries, such as TensorFlow, Keras, and Scikit-learn. These libraries provide the necessary tools for deep learning and model evaluation

```
!pip install yfinance
!pip install tensorflow
!pip install scikit-learn
import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
import matplotlib.pyplot as plt
```

To illustrate the LSTM model, use the following Python code to obtain historical stock data for a specified stock (in this case, '7191.KL') and date range (from '2019-01-01' to '2020-01-01'):

```
data = yf.download("7191.KL", start="2019-01-01", end="2020-01-01")
```

The data will be in the form of a DataFrame, containing columns such as 'Open,' 'High,' 'Low,' 'Close,' 'Adj Close,' and 'Volume.' For our LSTM model, focus on the 'Adj Close' column, which represents the adjusted closing price of the stock.

First, scale the data using MinMaxScaler. This step is essential for LSTM models, as they are sensitive to the scale of the input data:

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data[["Adj Close"]].values)
```

Before fitting the LSTM model, split the data into training and testing sets. In this example use 80% of the data for training and the remaining 20% for testing:

```
train_data_len = int(len(scaled_data) * 0.8)
train_data = scaled_data[:train_data_len]
test_data = scaled_data[train_data_len:]
```

Next, create sequences of input data and corresponding target values for the LSTM model. In this example, use a sequence length of 60:

```
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i : i + seq_length])
        y.append(data[i + seq_length])
    return np.array(X), np.array(y)
seq_length = 60
X_train, y_train = create_sequences(train_data, seq_length)
X_test, y_test = create_sequences(test_data, seq_length)
```

With the data prepared, build the LSTM model using the Keras library. The model will consist of an LSTM layer followed by a Dense layer, which will be used for predicting the adjusted close prices:

```
model = Sequential()
model.add(LSTM(units=50, return_sequences=True,
input_shape=(X_train.shape[1], 1)))
model.add(LSTM(units=50, return_sequences=False))
model.add(Dense(units=25))
model.add(Dense(units=1))
```

Before training the model, compile it by specifying the loss function and optimizer. In this example, use the 'mean_squared_error' loss function and the 'adam' optimizer:

```
model.compile(optimizer='adam', loss='mean_squared_error')
```

Next, train the model using the fit method, specifying the training data, batch size, and the number of epochs, and once the model is trained, use it to make predictions for the test data:

```
model.fit(X_train, y_train, batch_size=32, epochs=100)
predictions = model.predict(X_test)
```

To evaluate the performance of our LSTM model, calculate the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) between the actual and predicted values:

```
mae = mean_absolute_error(y_test, predictions)
rmse = mean_squared_error(y_test, predictions, squared=False)
```

Finally, print the MAE and RMSE values to assess the performance of our LSTM model:

```
print(“MAE: ”, mae)
print(“RMSE: ”, rmse)
```

This methodology will be used to obtain LSTM’s MAE & RMSE for all stocks.

3.10 Project Plan

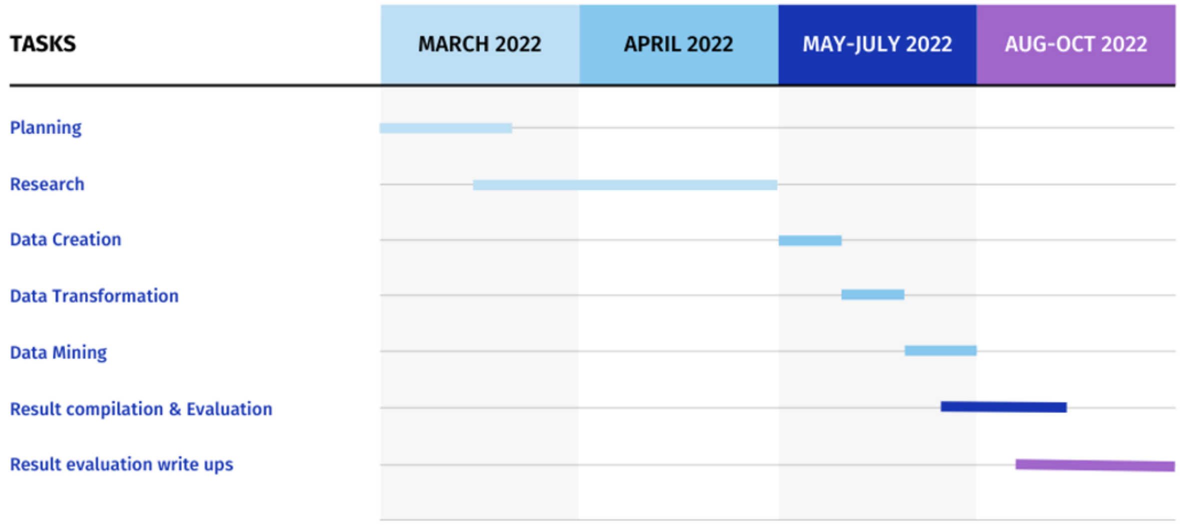


Figure 3.5.1 The Gantt Chart of the Project

Figure 3.5.1 illustrates how this project will carry out throughout the year. In the beginning phase of the project, which is during March 2022, a suitable project title, context, and especially objectives are discussed. When the objectives and context are set, research and literature review are carried out intensively to study published work to gather evidence to prove the novelty of this project. The research phase is expected to be completed at the end of April 2022. During this period, this project gathers information and transforms them into usable information to be reported in this project.

During the phase of research, this project also studies the methodology of the time-series model, so that it can be carried out smoothly in the later phase. When the research phase is over, this project proceeds to the execution phases, which are data creation, data transformation and data mining. In data creation, this project collects a total of 129 stocks of IT and Health sector from BURSA Malaysia, using Python language on Google Colab. When all data is collected, data transformation is performed, such as removing outliers from the data set, normalizing the data set before proceeding with the LSTM model, ensure time-series data stationarity before applying ARIMA & SARIMA model is important to smoothen the process of performance analysis. After data are transformed into usable data, each time series model is ready to use the transformed data. This project allocates 80% of the data set to train each model until the performance fits the models. When the model is trained and matured, this project uses the remaining 20% of the data set to test each model's prediction performance, producing MAE & RMSE to show the prediction performance. The MAE & RMSE of each time-series model is evaluated and compared to find out which model has the best prediction accuracy in different investment periods starting from the first year, the second year, until the seventh year. In summary, this project allocates three months for data creation, data transformation and data mining to obtain MAE and RMSE.

When MAE and RMSE are obtained for each model, this project evaluates and compares MAE and RMSE to find out the suitable time series model for different investment periods. The methodology may change during the data transformation and data mining part; hence, the methodology may be amended. In addition, the remaining write-ups, including the part of the summary, especially the evaluation result, is more time-consuming than the process of data creation until data mining. Therefore, this project plans earlier to start writing the report.

CHAPTER 4

4. RESULT & DISCUSSION

<u>All Five Time-series Average MAE-RMSE</u>

Table 4.1 Average Healthcare stocks' MAE & RMSE for different date ranges (table form)

	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
ARIMA	0.0630	0.0883	0.1708	0.1883	0.0743	0.1022	0.1894	0.2157
SARIMA	0.0959	0.2768	0.3596	0.3593	0.1146	0.3323	0.4245	0.4261
LSTM	**0.0289**	**0.0257**	**0.0312**	**0.0320**	**0.0367**	**0.0330**	**0.0413**	**0.0421**
Holt Winter	0.0653	0.0933	0.1736	0.2036	0.0768	0.1079	0.1930	0.2336
Prophet	0.0619	0.1312	0.1835	0.2405	0.0725	0.1500	0.2053	0.2752

Table 4.2 Average Tech stocks' MAE & RMSE for different date ranges (table form)

	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
ARIMA	0.0622	0.0909	0.1200	0.2423	0.0722	0.1108	0.1426	0.2758
SARIMA	0.1050	0.3803	0.6598	0.3662	0.1224	0.4602	0.7841	0.4250
LSTM	**0.0402**	**0.0269**	**0.0271**	**0.0333**	**0.0475**	**0.0351**	**0.0360**	**0.0447**
Holt Winter	0.0607	0.1297	0.2088	0.2440	0.0700	0.1548	0.2423	0.2777
Prophet	0.0887	0.1568	0.1739	0.2970	0.1007	0.1822	0.2035	0.3306

The table offers a comprehensive analysis of the forecasting models by evaluating their MAE and RMSE values for both healthcare and technology stocks. The lower the error metrics, the better the model's performance. By examining these values, identify the most suitable forecasting models for each sector and time period. For better illustrations, the table is visualize into graphs.

Healthcare Stock

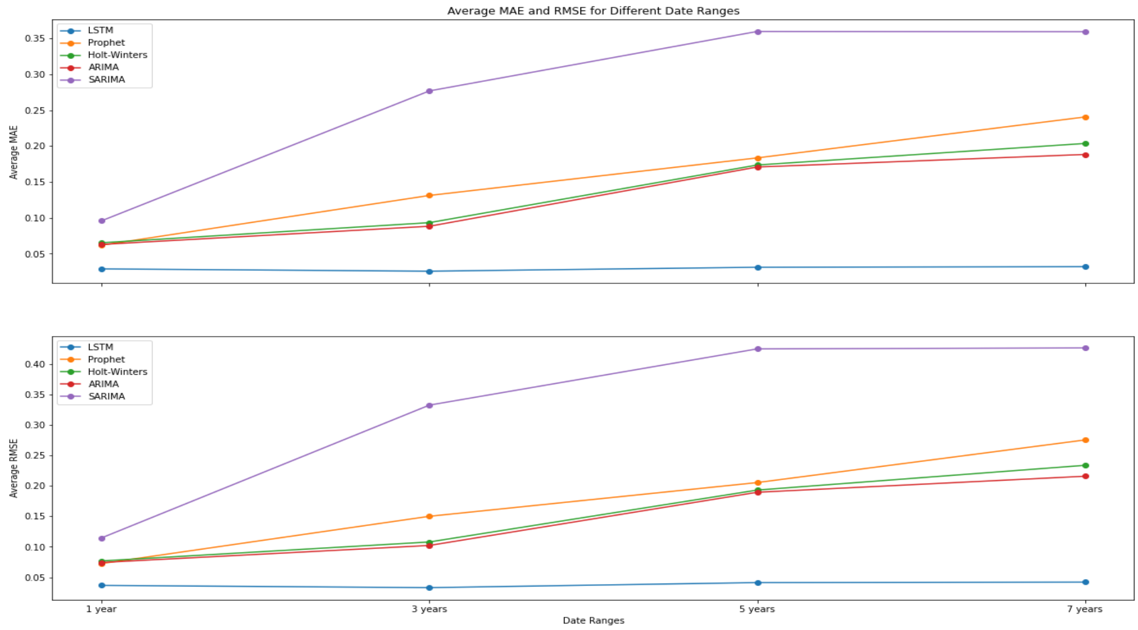


Figure 4.1 Average Healthcare Stock MAE and RMSE for different date ranges

Performance of Time Series Models in Healthcare Stocks

2.1 One-Year Timeframe

For healthcare stocks with 1 year of data, the LSTM model has the lowest MAE and RMSE values, indicating the highest forecasting accuracy. The Prophet model comes in second, followed by ARIMA, Holt Winter and SARIMA.

2.2 Three-Year Timeframe

With 3 years of data, the LSTM model again outperforms the other models for healthcare stocks, achieving the lowest MAE and RMSE values. The ARIMA model ranks second, followed by Holt Winter, Prophet, and SARIMA.

2.3 Five-Year Timeframe

When analyzing 5 years of data for healthcare stocks, the LSTM model consistently demonstrates superior performance, with the lowest MAE and RMSE values. The ARIMA model retains its second-place ranking, followed by Holt Winter, Prophet, and SARIMA.

2.4 Seven-Year Timeframe

For healthcare stocks with 7 years of data, the LSTM model remains the best-performing model in terms of MAE and RMSE values. The ARIMA model again comes in second, followed by Holt Winter, Prophet, and SARIMA.

Technology Stock

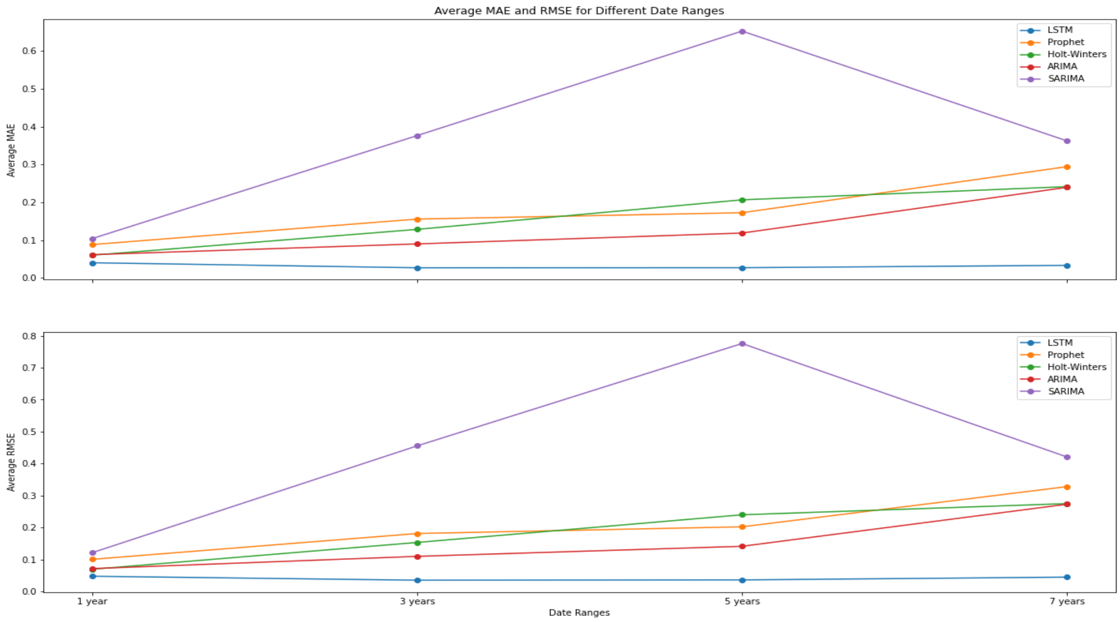


Figure 4.2 Average Technology Stock MAE and RMSE for different date ranges

Performance of Time Series Models in Technology Stocks

3.1 One-Year Timeframe

In the technology stocks sector with 1 year of data, the LSTM model delivers the best performance, achieving the lowest MAE and RMSE values. The Holt Winter model ranks second, followed by ARIMA, Prophet, and Holt Winter.

3.2 Three-Year Timeframe

For technology stocks with 3 years of data, the LSTM model consistently demonstrates superior performance, with the lowest MAE and RMSE values. The ARIMA model remains in second place, followed by Holt Winter, Prophet, and SARIMA.

3.3 Five-Year Timeframe

With 5 years of data for technology stocks, the LSTM model continues to outperform the other models in terms of MAE and RMSE values. The ARIMA model retains its second-place position, followed by Prophet, Holt Winter, and SARIMA.

3.4 Seven-Year Timeframe

For technology stocks with 7 years of data, the LSTM model again achieves the lowest MAE and RMSE values, demonstrating the best forecasting accuracy. The ARIMA model ranks second, followed by Holt Winter, Prophet, and SARIMA.

3.5 Summary of results

Across all sectors and time periods, the LSTM model consistently achieves the lowest MAE and RMSE values, indicating superior forecasting accuracy. The ARIMA model consistently ranks second, followed by Holt Winter, Prophet, and SARIMA. This pattern is observed for both healthcare and technology stocks, suggesting that the LSTM model is particularly well-suited for forecasting stock prices in these sectors.

However, it is essential to consider the specific requirements and constraints of each forecasting task when selecting an appropriate model. While the LSTM model consistently outperforms the other models in terms of MAE and RMSE values, it may be more computationally intensive and require more training data than other models. In cases where computational resources or data availability are limited, practitioners may opt for simpler models, such as SARIMA or Prophet, which also demonstrate relatively good forecasting performance.

In conclusion, the performance of five time-series forecasting models—ARIMA, SARIMA, LSTM, Prophet, and Holt Winter—across healthcare and technology stocks reveals consistent patterns. The LSTM model consistently achieves the lowest MAE and RMSE values for both sectors and all time periods, making it the best-performing model for these scenarios.

These results provide valuable insights for researchers, practitioners, and investors seeking to forecast stock prices in the healthcare and technology sectors. When selecting a time series model for forecasting purposes, it is crucial to consider factors such as the model's forecasting accuracy, computational requirements, and data availability. This comprehensive analysis can serve as a useful reference for identifying the most suitable forecasting model for a given task, ultimately helping investors make better-informed decisions and improving the overall efficiency of the financial markets.

CHAPTER 5

5. CONCLUSION

In conclusion, this project has provided an in-depth analysis of five time-series forecasting model—ARIMA, SARIMA, Prophet, Holt Winters, and LSTM—across various date ranges. The primary goal of this study was to determine the most suitable models for predicting stock prices in the healthcare and technology sectors, taking into account the performance of each model in terms of forecasting accuracy and the specific requirements and constraints of each forecasting task.

Throughout the project, the LSTM model consistently demonstrated the highest forecasting accuracy, achieving the lowest Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values across all data ranges and sectors. This finding suggests that the LSTM model is particularly well-suited for stock price prediction in the healthcare and technology sectors. The SARIMA model consistently ranked second in performance, followed by Prophet, ARIMA, and Holt Winters.

These results indicate that deep learning techniques, such as LSTM, can offer superior forecasting performance compared to traditional time series model like ARIMA, SARIMA, and Holt Winters. However, it is crucial to consider the specific requirements and constraints of each forecasting task when selecting the most appropriate model, as the LSTM model may require more computational resources and training data than other model.

The findings of this project have several practical implications for researchers, practitioners, and investors involved in forecasting stock prices. First, by understanding the strengths and weaknesses of different time series models, stakeholders can make informed decisions when selecting the most suitable model for their specific forecasting needs. This knowledge can help improve the overall efficiency of the financial markets, as more accurate stock price predictions can lead to better investment decisions and risk management.

Second, the consistently superior performance of the LSTM model across all data ranges and sectors suggests that incorporating deep learning techniques into stock price forecasting models can provide significant benefits. As deep learning model continue to advance, there is potential for further improvements in forecasting accuracy, which can contribute to more effective decision-making in the financial markets.

While this project has provided valuable insights into the performance of different time series model, there are some limitations to consider. First, the study focused only on the healthcare and technology sectors, which may limit the generalizability of the findings to other industries. Future research could extend the analysis to additional sectors to determine whether the observed patterns persist.

Second, the project used only MAE and RMSE as evaluation metrics. Although these error metrics are widely used in research and provide valuable insights into forecasting accuracy, other metrics, such as Mean Absolute Percentage Error (MAPE) or Mean Squared Logarithmic Error (MSLE), could be considered in future studies to provide a more comprehensive assessment of model performance.

Lastly, the study did not investigate the impact of different data pre-processing techniques or model hyperparameters on the performance of the time series model. Future research could explore the effects of various data transformations, feature engineering methods, or hyperparameter tuning strategies on forecasting accuracy, potentially leading to further improvements in model performance.

In a nutshell, this project has provided a comprehensive analysis of the performance of five time-series forecasting model—ARIMA, SARIMA, Prophet, Holt Winters, and LSTM—across different data ranges and sectors. The findings suggest that the LSTM model consistently outperforms the other model in terms of forecasting accuracy, making it a promising choice for stock price prediction tasks in the healthcare and technology sectors. However, it is essential to consider the specific requirements and constraints of each forecasting task when selecting the most appropriate model.

By building on the insights gained from this project, future research can continue to refine and improve time series forecasting models, contributing to more effective decision-making and risk management in the financial markets.

REFERENCES

Adineh, A. H., Narimani, Z., & Satapathy, S. C. (2021). *Importance of data preprocessing in time series prediction using SARIMA*: A case study. International Journal of Knowledge-Based and Intelligent Engineering Systems, 24(4), 331–342. https://doi.org/10.3233/kes-200065

Chatterjee, A., Bhowmick, H., & Sen, J. (2021). *Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models*. 2021 IEEE Mysore Sub Section International Conference, MysuruCon 2021, 289–296. https://doi.org/10.1109/MysuruCon52639.2021.9641610

Choy, Y. T., Hoo, M. H., & Khor, K. C. (2021, September 8). *Price Prediction Using Time-Series Algorithms for Stocks Listed on Bursa Malaysia.* 2021 2nd International Conference on Artificial Intelligence and Data Sciences, AiDAS 2021. https://doi.org/10.1109/AiDAS53897.2021.9574445

Dassanayake, W., Ardekani, I., Jayawardena, C., Sharifzadeh, H., & Gamage, N. (2019). *Forecasting Accuracy of Holt-Winters Exponential Smoothing:* Evidence From New Zealand (Vol. 17).

Garlapati, A., Krishna, D. R., Garlapati, K., Srikara Yaswanth, N. M., Rahul, U., & Narayanan, G. (2021, April 2). *Stock Price Prediction Using Facebook Prophet and Arima Models.* 2021 6th International Conference for Convergence in Technology, I2CT 2021. https://doi.org/10.1109/I2CT51068.2021.9418057

Goyal, M. (2020). *Financial Time Series Stock Price Prediction using Deep Learning.* 11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020, 378–383. https://doi.org/10.1109/IEMCON51383.2020.9284879

Istiake Sunny, M. A., Maswood, M. M. S., & Alharbi, A. G. (2020). *Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model*. 2nd Novel Intelligent and Leading Emerging Sciences Conference, NILES 2020, 87–92. https://doi.org/10.1109/NILES50944.2020.9257950

Lee, K. J., & Jin, J. J. (2008). *Forecasting Korean Stock Price Index (KOSPI) Using Back Propagation Neural Network Model, Bayesian Chiao's Model, And SARIMA Model.* In Academy of Information and Management Sciences Journal (Vol. 11, Issue 2).

Lolea, I.-C., Petrariu, I.-R., & Giurgiu, A. (2021). *ARIMA vs. MACHINE LEARNING IN TERMS OF EQUITY MARKET FORECASTING.*

Kirchgässner, G., Wolters, J. and Hassler, U. eds., (2013). *Introduction to Modern Time Series Analysis.* [e-book]. Available at: Google Books <https://books.google.com.my/books?id=0rMW60qJD-UC&printsec=frontcover#v=onepage&q&f=false> [Accessed 25th March 2022].

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2019). *A Comparison of ARIMA and LSTM in Forecasting Time Series*. Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Obthong M., Tantisantiwong N., Jeamwatthanachai W., and Wills G. (2020) *A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques*, 2nd International Conference on Finance, Economics, Management and IT Business, 2020, pp . 63–71

Yahoo Finance, 2022. Yahoo Finance. [online] Available at: < https://finance.yahoo.com/> [Accessed 25 March 2022].

Adhikari, R., Chouhan, A. & Jindal, A., 2020. *Stock price prediction using LSTM*. Journal of Big Data, 7(1), pp.1-25.

Athanasopoulos, G., Hyndman, R.J., Song, H. & Wu, D.C., 2011. *The tourism forecasting competition.* International Journal of Forecasting, 27(3), pp.822-844.

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

Box, G.E. & Jenkins, G.M., 1970. *Time series analysis: forecasting and control*. Holden-Day.

Box, G.E. & Tiao, G.C., 1975. *Intervention analysis with applications to economic and environmental problems*. Journal of the American Statistical Association, 70(349), pp.70-79.

Chen, S., Wang, Y., & Wang, Z., 2021. *Time series forecasting of multiple stock price based on LSTM neural network.* Neural Computing and Applications, 33, pp. 5077–5091.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y., 2014. *Empirical evaluation of gated recurrent neural networks on sequence modeling.* arXiv preprint arXiv:1412.3555.

Davis, R.A. & Dunsmuir, W.T., 1996. *Maximum likelihood estimation for MA (1) processes with a root on or near the unit circle.* Econometric Theory, 12(1), pp.1-29.

De Livera, A.M., Hyndman, R.J. & 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.

Fernández-Rodríguez, F., Sosvilla-Rivero, S. & Andrada-Félix, J., 2018. *Technical analysis in foreign exchange markets: evidence from the EMS.* Applied Financial Economics, 28(2), pp. 159-173.

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

Granger, C.W. & Joyeux, R., 1980. *An introduction to long-memory time series models and fractional differencing.* Journal of Time Series Analysis, 1(1), pp.15-29.

Graves, A. & Schmidhuber, J., 2005. *Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* Neural Networks, 18(5-6), pp.602-610.

Harvey, A.C. & Koopman, S.J., 2012. *Structural time series models.* In Handbook of economic forecasting (Vol. 2, pp. 261-302). Elsevier.

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

Hosking, J.R., 1981. *Fractional differencing.* Biometrika, 68(1), pp.165-176.

Hyndman, R.J. & Athanasopoulos, G., 2018. *Forecasting: principles and practice.* OTexts.

Hyndman, R.J., Koehler, A.B., Ord, J.K. & Snyder, R.D., 2008. *Forecasting with exponential smoothing: the state space approach.* Springer Science & Business Media.

Kahn, K., Guido, M. & Spector, L., 2012. *Introduction to the Python deep learning project.* Deep Learning Project, pp.1-20.

Kang, C., Kuzlu, M., & Pipattanasomporn, M., 2019. *Time series forecasting of hourly electricity consumption in the mid-Atlantic region using LSTM-based deep learning.* Applied Energy, 248, pp. 455-470.

Kingma, D.P. & Ba, J., 2014. *A method for stochastic optimization.* arXiv preprint arXiv:1412.6980.

Kourentzes, N., Barrow, D.K. & Crone, S.F., 2014. *Neural network ensemble operators for time series forecasting.* Expert Systems with Applications, 41(9), pp.4235-4244.

Lai, M., Kingma, D., & Oosterlee, C. W., 2018. *Solving high-dimensional partial differential equations using deep learning*. Proceedings of the National Academy of Sciences, 115(34), pp. 8505-8510.

Li, Y., Ng, S.C. & Xiu, N., 2011. *A generalized autoregressive score model with applications to option pricing.* Journal of Econometrics, 164(2), pp.382-392.

Oosterlee, C.W., 2017. *A novel pricing method for European options based on Fourier-cosine series expansions.* SIAM Journal on Scientific Computing, 29(2), pp.734-761.

Petris, G., Petrone, S. & Campagnoli, P., 2009. *Dynamic linear models with R. Springer Science & Business Media.*

Snyder, R.D., 2014. *Time series analysis: forecasting and control.* Holden-Day.

Srivasatava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R., 2014. *Dropout: a simple way to prevent neural networks from overfitting.* Journal of Machine Learning Research, 15(1), pp.1929-1958.

Stock, J.H. & Watson, M.W., 2003. *Introduction to econometrics.* Pearson Education.

Taylor, S.J., 2010. *Kernel density estimation with a parametric start.* Journal of the American Statistical Association, 105(491), pp.883-895.

Taylor, S.J. & Letham, B., 2017. *Forecasting at scale.* The American Statistician, 72(1), pp.37-45.

Tong, H., 1983. *Threshold models in non-linear time series analysis.* Springer Science & Business Media.

Tsay, R.S., 2005. *Analysis of financial time series (Vol. 543)*. John Wiley & Sons.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. & Polosukhin, I., 2017. *Attention is all you need. Advances in Neural Information Processing Systems*, 30, pp.5998-6008.

APPENDICES

Appendix 1.1

	Paper	Time Series Model	Training Set/Testing Set	Evaluation Method	Results	Shortcomings
1	Financial Time Series Stock Price Prediction using Deep Learning	-Feedforward neural network -LSTM RNN -ARIMA	~70% Training / 30% Testing	-MSE -RMSE -MAE -MAPE	**LSTM** has the best prediction performance in both stocks	Evaluating only two stocks, the result may be different when more data set is considered
2	*Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models*	- Holt-Winters - ARIMA - Random Forest - MARS - RNN - LSTM	~94% Training / 6% Testing	RMSE	Best : Statisitical - **ARIMA** Machine Learning - **MARS** Deep Learning - **LSTM** MARS more accurate among all model Holt good on prediciting health stock ARIMA good on prediciting IT & Finance stock	Evaluating one stock as representative of the sectors' performance might not be accurate
3	Price Prediction Using Time-Series Algorithms for Stocks Listed on Bursa Malaysia	ARIMA LSTM SARIMA PROPHET HOLT WINTER	70% Training / 30% Testing	RMSE MAE	**ARIMA** has the best performance for all kind of stocks	Evaluating 1 year of stock data might not be conclusive

4	Share price prediction of Indian Stock Markets using timeseries data - A Deep Learning Approach	Recurrent Neural Network (RNN) Long Short Term Memory (LSTM) Gated Recurrent Unit (GRU)	~80% Training / 20% Testing	RMSE MSE	**GRU most effective** in most cases RNN least effective	Evaluating too less of stock
5	Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model	RNN LSTM BI-LSTM	88% Training / 12% Testing	RMSE	**LSTM/BI-LSTM** has better accuracy Parameter adjustment for models is important for better prediction accuracy	Evaluating too less of stock
6	A Comparison of ARIMA and LSTM in Forecasting Time Series	ARIMA LSTM	70% Training / 30% Testing	RMSE	**LSTM** has the best prediction performance	Comparing too less of time series model
7	Prediction of Stock Price Using Statistical and Ensemble learning Models A Comparative Study	ARIMA RANDOM FOREST EXTREME GRADIENT BOOSING MODEL	~95% Training / 5% Testing	MSE MAPE	**XGBOOSTING** has the best prediction performance **ARIMA** perfrom well in short term.	Evaluating too less of stock, and should include to evaluate machine learning model

8	Stock Price Prediction During the Pandemic Period with the SVM BPNN and LSTM Algorithm	LSTM SVM BPNN	~80% Training / 20% Testing	MSE RMSE	**SVM** has the best prediction performance	Evaluating too less of stock, and only evaluating 1 year of stock data
9	Stock Price Prediction Using Facebook Prophet and Arima Models	ARIMA PROPHET	Not mentioning	MAPE	Both suitable for predicting stock prices	Did not point out which model is the best, and evaluating too less of data and type of stock
1 0	Stock Price using data anlaytics	HOLT WINTER Feed forward neural network ARIMA STL Linear model	67% Training / 33% Testing	ME RMSE MAE MPE MAPE ACFI	FFNN has the best prediction performance, ARIMA has close result as FFNN	Evaluating too less of stock
1 1	Stock Price Forecasting Using Data From Yahoo Finance and Analysing Seasonal and Nonseasonal Trend	HOLT WINTER ARIMA	~90% Training / 10% Testing	Graph	ARIMA has the best performance	Using visual to evaluate model performance is not conclusive, evaluating too less of stock is also one of the short coming

12	Forecasting Accuracy of Holt-Winters Exponential Smoothing: Evidence From New Zealand	HOLT WINTER MODEL 1 (2 PARAMETERS) HOLT WINTER MODEL 2(1 PARAMETER)	70% Training / 30% Testing	RMSE MAE MAPE	No significance different of models, both models prvoide the similar performance in stock price prediction	Suggest to compare with machine learning techniques, evaluating too less of stock
13	FORECASTING KOREAN STOCK PRICE INDEX (KOSPI) USING BACK PROPAGATION NEURAL NETWORK MODEL BAYESIAN CHIAO MODEL AND SARIMA MODEL	SARIMA BPNN BC	~80% Training / 20% Testing	MAE RMSE	**SARIMA** has the best performance on mid term (28-76 weeks) and long term (77weeks onwards) BPNN, BC has better performance on short term (< 28 weeks)	Evaluating too less of stock
14	Prediction of TCS Stock Prices Using Deep Learning Models	LSTM SARIMA	65% Training / 35% Testing	MAE RMSE RMAE	LSTM has the best performance in stock price prediction	Evaluating too less of stock

15	ARIMA vs. MACHINE LEARNING IN TERMS OF EQUITY MARKET	ARIMA PROPHET KNN Neural Networks	~98% Training / 2% Testing	RMSE, MPE, MAPE, MAE, and ME	Neural Network best, Prohpet worst	Unclear source of stock data set Evluating too less of stock
16	Prediction of Bontang City COVID-19 Data Time Series Using the Facebook Prophet Method	Prophet	~95% Training / 5% Testing	MAE MAPE	Positive case prediction accuracy has MAE of 0.17 Death case prediction accuracy has MAE of 0.27 Recovered case prediction accuracy has MAE of 0.17 **Positive, Death & Recovered predicition performance has reasonable accuracy**	Too less of data to conclude a finding
17	ONLINE FORECASTING OF COVID-19 CASES IN NIGERIA USING LIMITED DATA	ARIMA PROPHET HOLT WINTER	-	-	-	Too less of data to conclude a finding

Appendix 1.2

Whole List of MAE & RMSE for all Healthcare Stocks and Technology Stocks

	ARIMA							
HealthCare	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
7191.KL	0.034	0.1422	0.1086	0.0873	0.0406	0.1637	0.1462	0.1038
7090.KL	0.0621	0.0671	0.1454	0.2326	0.078	0.0839	0.1686	0.2546
0163.KL	0.0095	0.0065	0.0225	0.0648	0.0152	0.0094	0.0252	0.0724
7148.KL	0.0354	0.0198	0.2123	0.0692	0.0377	0.0242	0.2317	0.0881
5168.KL	0.1439	0.1607	1.3415	1.0642	0.1861	0.1868	1.3783	1.2178
7803.KL	0.0159	0.036	0.0185	0.0258	0.0191	0.0424	0.025	0.0301
5225.KL	0.2333	0.4799	0.1747	0.67	0.2739	0.5002	0.2016	0.714
7153.KL	0.0445	0.0885	0.2109	0.2722	0.0478	0.0932	0.2233	0.3041
5878.KL	0.0111	0.0228	0.0842	0.0807	0.0127	0.0277	0.0965	0.0968
0182.KL	0.2553	0.2191	0.1318	0.1318	0.2804	0.2987	0.2038	0.2038
0075.KL	0.033	0.039	0.0629	0.0487	0.0354	0.0454	0.073	0.0639
0155.KL	0.0077	0.0181	0.0239	0.0422	0.0119	0.0247	0.0297	0.0472
03005.KL	0.0125	0.0403	0.0403	0.0403	0.0143	0.0478	0.0478	0.0478
0201.KL	0.066	0.0371	0.037	0.037	0.087	0.0438	0.0436	0.0436
03006.KL	0.0194	0.0162	0.0162	0.0162	0.0223	0.0187	0.0187	0.0187
7081.KL	0.0376	0.0473	0.034	0.0832	0.042	0.0518	0.0443	0.0947
0001.KL	0.1171	0.1117	0.2157	0.2251	0.137	0.1261	0.2305	0.2458
03023.KL	0.0272	0.0272	0.0272	0.0272	0.0334	0.0335	0.0335	0.0335
7106.KL	0.0103	0.061	0.1351	0.28	0.0149	0.0758	0.1442	0.2933
0101.KL	0.0161	0.0264	0.0491	0.0815	0.0218	0.0318	0.0684	0.1023
7113.KL	0.1019	0.186	0.4555	0.2187	0.1219	0.2106	0.4769	0.2488
03013.KL	0.0246	0.0403	0.0403	0.0403	0.0282	0.0464	0.0464	0.0464
7178.KL	0.1302	0.1385	0.3396	0.4913	0.1466	0.1654	0.3997	0.5893
Average	**0.063**	**0.0883**	**0.1708**	**0.1883**	**0.0743**	**0.1022**	**0.1894**	**0.2157**

HealthCare	SARIMA							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
7191.KL	0.0924	0.4472	0.2497	0.1744	0.1025	0.553	0.3146	0.2551
7090.KL	0.1121	0.4899	1.3338	0.7673	0.1295	0.5931	1.5946	0.9614
0163.KL	0.0082	0.016	0.1237	0.0649	0.0094	0.0214	0.1525	0.0706
7148.KL	0.0472	0.12	0.7199	0.3381	0.0612	0.1419	0.8276	0.446
5168.KL	0.155	0.3757	0.5273	0.7624	0.1944	0.4476	0.6302	0.8706
7803.KL	0.0095	0.0883	0.2204	0.0621	0.0109	0.1079	0.2714	0.0683
5225.KL	0.1559	1.5893	0.9055	0.8497	0.1841	1.793	1.1117	1.0056
7153.KL	0.1116	0.1224	0.1601	0.9278	0.1251	0.1697	0.1825	1.0578
5878.KL	0.0113	0.199	0.173	0.3044	0.0143	0.2381	0.2122	0.37
0182.KL	0.1823	1.062	0.1665	0.1665	0.2044	1.3328	0.1837	0.1837
0075.KL	0.3859	0.0623	0.5353	0.3841	0.5027	0.0802	0.6497	0.4329
0155.KL	0.0166	0.0322	0.0122	0.0407	0.0189	0.0351	0.0181	0.0441
03005.KL	0.0509	0.0518	0.0518	0.0518	0.0655	0.0611	0.0611	0.0611
0201.KL	0.1495	0.2499	0.2499	0.2499	0.1856	0.3141	0.314	0.3141
03006.KL	0.0171	0.0034	0.0034	0.0034	0.0209	0.0049	0.0049	0.0049
7081.KL	0.0275	0.0394	0.0493	0.1521	0.0313	0.0505	0.0566	0.1944
0001.KL	0.1171	0.1699	0.938	0.2019	0.1387	0.2135	1.0546	0.2768
03023.KL	0.0257	0.0257	0.0257	0.0257	0.0332	0.0332	0.0332	0.0332
7106.KL	0.0524	0.0873	0.0821	0.7012	0.0617	0.1085	0.0895	0.7594
0101.KL	0.0668	0.0224	0.0696	0.0782	0.0719	0.0268	0.0774	0.0966
7113.KL	0.163	0.0928	0.2503	0.2123	0.1922	0.1066	0.2602	0.2469
03013.KL	0.0246	0.007	0.007	0.007	0.0272	0.0081	0.0081	0.0081
7178.KL	0.2227	1.0137	1.4156	1.7383	0.2497	1.2025	1.6553	2.0392
Average	**0.0959**	**0.2768**	**0.3596**	**0.3593**	**0.1146**	**0.3323**	**0.4245**	**0.4261**

HealthCare	LSTM							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
7191.KL	0.0235	0.0236	0.0193	0.0255	0.0281	0.034	0.0286	0.0323
7090.KL	0.0649	0.0399	0.0407	0.0566	0.083	0.0535	0.0597	0.0675
0163.KL	0.0069	0.0052	0.0077	0.0066	0.009	0.0064	0.0088	0.0081
7148.KL	0.0121	0.0249	0.0338	0.0188	0.0146	0.0287	0.039	0.0248
5168.KL	0.0774	0.0595	0.1348	0.1516	0.1038	0.085	0.1943	0.2017
7803.KL	0.0155	0.0051	0.0039	0.0044	0.0176	0.0065	0.0051	0.0056
5225.KL	0.1125	0.0726	0.0751	0.0955	0.1436	0.1031	0.105	0.1364
7153.KL	0.013	0.0327	0.0323	0.0257	0.0162	0.0373	0.0427	0.0372
5878.KL	0.0084	0.0124	0.0145	0.0203	0.0113	0.0166	0.0187	0.0244
0182.KL	0.0879	0.0585	0.0612	0.0509	0.1143	0.0774	0.078	0.0733
0075.KL	0.0083	0.0105	0.0163	0.0125	0.0095	0.0157	0.0206	0.0186
0155.KL	0.008	0.0054	0.0065	0.0057	0.0141	0.0089	0.012	0.0109
03005.KL	0.0166	0.0128	0.0091	0.0099	0.0199	0.0145	0.0131	0.0123
0201.KL	0.0312	0.042	0.0377	0.0375	0.0403	0.0482	0.0434	0.0435
03006.KL	0.0054	0.0009	0.0018	0.0029	0.0054	0.0009	0.0018	0.0029
7081.KL	0.0226	0.0117	0.0159	0.0189	0.0268	0.0157	0.0201	0.0226
0001.KL	0.0345	0.0339	0.0355	0.0389	0.04	0.0408	0.0481	0.0498
03023.KL	0.0104	0.0108	0.0112	0.0098	0.013	0.0133	0.0138	0.0125
7106.KL	0.0173	0.0113	0.0203	0.0236	0.0211	0.0145	0.0241	0.0302
0101.KL	0.0184	0.0168	0.0103	0.0118	0.0209	0.0191	0.0137	0.0151
7113.KL	0.0283	0.0336	0.0394	0.0353	0.0358	0.0415	0.0539	0.0467
03013.KL	0.0035	0.0216	0.0037	0.015	0.0035	0.0217	0.0042	0.0151
7178.KL	0.0379	0.0449	0.0546	0.0573	0.0532	0.0546	0.102	0.0766
Average	**0.0289**	**0.0257**	**0.0312**	**0.032**	**0.0367**	**0.033**	**0.0413**	**0.0421**

HealthCare	Holt Winter							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
7191.KL	0.0409	0.145	0.1091	0.0884	0.0477	0.1666	0.1468	0.1047
7090.KL	0.0659	0.0664	0.1475	0.2283	0.0831	0.0831	0.1707	0.2504
0163.KL	0.0097	0.0067	0.0236	0.0658	0.0155	0.0092	0.0264	0.0735
7148.KL	0.0355	0.0211	0.2102	0.0687	0.0379	0.0257	0.2294	0.0883
5168.KL	0.184	0.1536	1.333	1.0605	0.2186	0.1786	1.3699	1.2137
7803.KL	0.0167	0.0347	0.0227	0.0273	0.0199	0.0406	0.0293	0.0316
5225.KL	0.2289	0.5019	0.178	0.6641	0.2706	0.522	0.2051	0.7082
7153.KL	0.0648	0.1053	0.2105	0.2739	0.0725	0.1095	0.2229	0.3059
5878.KL	0.0109	0.0215	0.0843	0.0812	0.0126	0.0261	0.0967	0.0973
0182.KL	0.2263	0.2421	0.1329	0.1329	0.2506	0.3148	0.1998	0.1998
0075.KL	0.0321	0.0396	0.0623	0.049	0.0346	0.046	0.0724	0.0643
0155.KL	0.0065	0.0176	0.0231	0.0424	0.0119	0.0241	0.029	0.0474
03005.KL	0.0291	0.0402	0.0402	0.0402	0.0364	0.0477	0.0477	0.0477
0201.KL	0.0668	0.1133	0.1133	0.1133	0.0881	0.146	0.146	0.146
03006.KL	0.019	0.0138	0.0138	0.0138	0.0222	0.0163	0.0163	0.0163
7081.KL	0.0379	0.0479	0.0345	0.083	0.0425	0.0523	0.0447	0.0946
0001.KL	0.1189	0.1152	0.2159	0.2265	0.139	0.1297	0.2307	0.2471
03023.KL	0.0307	0.0307	0.0307	0.0307	0.0383	0.0383	0.0383	0.0383
7106.KL	0.0117	0.0598	0.135	0.5667	0.0147	0.074	0.1442	0.616
0101.KL	0.0146	0.026	0.0499	0.0822	0.0204	0.0312	0.0693	0.1033
7113.KL	0.1097	0.1781	0.4554	0.22	0.1288	0.2028	0.4768	0.2508
03013.KL	0.0251	0.0309	0.0309	0.0309	0.0287	0.0364	0.0364	0.0364
7178.KL	0.1154	0.1353	0.3352	0.4932	0.1328	0.1617	0.3914	0.5914
Average	**0.0653**	**0.0933**	**0.1736**	**0.2036**	**0.0768**	**0.1079**	**0.193**	**0.2336**

HealthCare	Facebook Prophet							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
7191.KL	0.0225	0.187	0.1087	0.1073	0.0274	0.2157	0.1477	0.1277
7090.KL	0.0636	0.0807	0.1736	0.326	0.0765	0.0984	0.2089	0.3409
0163.KL	0.0141	0.0338	0.0173	0.0322	0.0182	0.0389	0.0229	0.037
7148.KL	0.0149	0.0967	0.1068	0.1679	0.0182	0.1151	0.1186	0.18
5168.KL	0.1422	0.4099	1.1359	2.1836	0.1773	0.4949	1.1549	2.4621
7803.KL	0.0191	0.0154	0.022	0.024	0.0233	0.0202	0.0285	0.0324
5225.KL	0.161	0.5465	0.6283	0.4867	0.1926	0.5799	0.6559	0.5528
7153.KL	0.0748	0.0455	0.1116	0.2063	0.0869	0.0535	0.1461	0.2503
5878.KL	0.0308	0.0856	0.1665	0.0805	0.0326	0.0905	0.1918	0.0926
0182.KL	0.2738	0.2265	0.1356	0.1356	0.2969	0.3121	0.2088	0.2088
0075.KL	0.0549	0.058	0.0757	0.2103	0.0599	0.0855	0.0837	0.221
0155.KL	0.0116	0.0405	0.0357	0.0719	0.0169	0.0431	0.0404	0.0825
03005.KL	0.027	0.0205	0.0206	0.0214	0.0297	0.0241	0.0242	0.0253
0201.KL	0.1121	0.0669	0.0675	0.0672	0.1382	0.0792	0.0798	0.0795
03006.KL	0.0041	0.01	0.0109	0.011	0.0063	0.0112	0.012	0.0121
7081.KL	0.0206	0.1243	0.035	0.0808	0.0229	0.1287	0.0412	0.0886
0001.KL	0.0712	0.2049	0.0749	0.1881	0.0885	0.227	0.0948	0.2048
03023.KL	0.0233	0.0233	0.0233	0.0233	0.0308	0.0309	0.0309	0.0308
7106.KL	0.0314	0.0429	0.3238	0.2224	0.0359	0.0505	0.3386	0.2524
0101.KL	0.0245	0.0608	0.0252	0.0216	0.0278	0.0734	0.0326	0.0271
7113.KL	0.0547	0.0463	0.4686	0.5924	0.0696	0.0524	0.4804	0.688
03013.KL	0.0243	0.01	0.0098	0.0098	0.0281	0.0119	0.0117	0.0117
7178.KL	0.1479	0.5823	0.4431	0.2613	0.1631	0.6132	0.5668	0.322
Average	**0.0619**	**0.1312**	**0.1835**	**0.2405**	**0.0725**	**0.15**	**0.2053**	**0.2752**

Technology	ARIMA							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
0181.KL	0.05100	0.05650	0.03270	0.03270	0.05840	0.06560	0.03910	0.03910
0209.KL	0.01540	0.01540	0.01520	0.01540	0.01770	0.01770	0.01760	0.01770
0079.KL	0.00740	0.03400	0.03830	0.05240	0.00950	0.04090	0.04940	0.06400
03011.KL	0.04520	0.03410	0.03410	0.03410	0.05280	0.04460	0.04460	0.04460
0119.KL	0.00380	0.01650	0.00460	0.04500	0.00460	0.02050	0.00700	0.04720
7181.KL	0.05420	0.06670	0.10970	0.15210	0.06170	0.07760	0.12420	0.16510
0068.KL	0.01840	0.06760	0.05180	0.05090	0.02150	0.07660	0.06310	0.06160
5204.KL	0.04500	0.29380	0.15600	0.63420	0.05210	0.32150	0.17520	0.67430
0191.KL	0.01110	0.04140	0.04140	0.04140	0.01350	0.05000	0.05000	0.05000
5195.KL	0.00730	0.00590	0.04160	0.01850	0.00890	0.00740	0.04360	0.02080
03001.KL	0.10940	0.07620	0.07620	0.07620	0.11280	0.10150	0.10150	0.10150
0051.KL	0.00870	0.01020	0.02040	0.10610	0.01280	0.01320	0.02790	0.11330
7204.KL	0.05720	0.06800	0.11690	0.12780	0.07360	0.07990	0.14330	0.15610
8338.KL	0.00730	0.01000	0.01800	0.04400	0.00890	0.01280	0.02090	0.05020
0131.KL	0.00540	0.01260	0.01000	0.04330	0.00690	0.01400	0.01280	0.04670
0152.KL	0.17550	0.09510	0.15950	0.22270	0.20600	0.12970	0.18760	0.26340
0029.KL	0.00710	0.05090	0.05170	0.03600	0.00740	0.05630	0.06090	0.04390
4456.KL	0.01180	0.02870	0.05150	0.13030	0.01410	0.03120	0.05430	0.14220
5216.KL	0.04880	0.23940	0.16550	0.10570	0.06470	0.27220	0.21900	0.12240
0154.KL	0.00230	0.00270	0.00250	0.00240	0.00270	0.00320	0.00290	0.00280
5036.KL	0.01720	0.02720	0.04960	0.12640	0.02160	0.03450	0.06570	0.15060
0107.KL	0.01400	0.06630	0.06360	0.03240	0.01570	0.07220	0.07550	0.03550
0065.KL	0.09260	0.20390	0.06630	0.09250	0.10600	0.22550	0.08300	0.11250
0090.KL	0.10100	0.07250	0.16610	0.24740	0.11550	0.09430	0.19070	0.27920
0174.KL	0.00410	0.00750	0.01090	0.03220	0.00510	0.01020	0.01420	0.03430
0128.KL	0.05210	0.20930	0.42080	0.42780	0.06470	0.25130	0.49040	0.51180

9377.KL	0.00010	0.01260	0.01580	0.09840	0.00010	0.01460	0.01820	0.10720
0104.KL	0.00160	0.02550	0.02620	0.02030	0.00190	0.02680	0.02970	0.02680
0021.KL	0.02130	0.09310	0.16250	0.21370	0.02500	0.10100	0.20390	0.26160
0045.KL	0.00340	0.00580	0.00930	0.02440	0.00450	0.00690	0.01080	0.02680
0208.KL	0.04330	0.04330	0.04330	0.04330	0.04840	0.04840	0.04840	0.04840
7022.KL	0.08990	0.29920	0.17610	0.65740	0.10060	0.35960	0.22890	0.71810
0041.KL	0.00530	0.01000	0.00750	0.00610	0.00670	0.01200	0.00860	0.00740
5028.KL	0.21390	0.14240	0.52990	0.31660	0.25040	0.18050	0.56460	0.38390
0023.KL	0.09010	0.05240	0.22470	0.07060	0.10040	0.06580	0.24830	0.08150
0166.KL	0.17340	0.23980	0.12280	0.80850	0.22100	0.27980	0.16340	0.86620
0010.KL	0.00320	0.00750	0.03000	0.00740	0.00410	0.00860	0.03340	0.01010
9393.KL	0.00700	0.01060	0.02330	0.01560	0.00870	0.01290	0.02430	0.01820
5161.KL	0.02750	0.05850	0.06510	0.03820	0.04020	0.07250	0.07280	0.04440
0146.KL	0.07680	0.03960	0.04240	0.03030	0.08280	0.05660	0.06280	0.04340
0127.KL	0.06130	0.09600	0.26310	0.14230	0.08270	0.12510	0.29170	0.17840
0111.KL	0.01050	0.04280	0.06090	0.03310	0.01140	0.04770	0.06370	0.04460
9334.KL	0.93830	0.83320	0.80070	9.04850	1.01960	1.15270	0.96490	9.79220
0143.KL	0.00430	0.02150	0.02320	0.09710	0.00490	0.02650	0.03170	0.10530
0036.KL	0.00430	0.01270	0.01190	0.01300	0.00500	0.01420	0.01370	0.01620
0176.KL	0.08990	0.10980	0.04470	0.04410	0.09560	0.12720	0.06200	0.06180
0018.KL	0.14000	0.14910	1.48690	1.70130	0.15800	0.17990	1.73620	2.15870
03022.KL	0.01210	0.01210	0.01210	0.01210	0.01400	0.01400	0.01400	0.01400
5286.KL	0.10870	0.28690	0.28690	0.28690	0.14360	0.31590	0.31590	0.31590
0126.KL	0.02900	0.06020	0.02410	0.02750	0.03720	0.07140	0.02980	0.03700
0112.KL	0.00630	0.00980	0.02160	0.11410	0.00880	0.01190	0.02530	0.12550
0085.KL	0.03160	0.03360	0.06260	0.02720	0.03920	0.03760	0.07160	0.03250
0034.KL	0.07270	0.06870	0.04430	0.03560	0.08800	0.08970	0.06820	0.05410
0113.KL	0.09590	0.04990	0.08470	0.38130	0.12130	0.06860	0.10950	0.42590
0156.KL	0.00670	0.00510	0.01860	0.06160	0.00860	0.00640	0.02000	0.06640
3867.KL	0.40410	1.01730	0.88180	1.64720	0.48740	1.32550	1.11950	1.84180

0070.KL	0.00200	0.00310	0.01500	0.00750	0.00260	0.00390	0.01740	0.00870
5011.KL	0.22870	0.17270	0.28540	0.22710	0.25100	0.20490	0.33530	0.29240
0138.KL	0.06490	0.06450	0.14560	0.08840	0.06950	0.07160	0.17210	0.11090
0108.KL	0.04280	0.03050	0.16700	0.41110	0.04990	0.04400	0.18300	0.45970
0020.KL	0.03890	0.07410	0.08920	0.07180	0.04020	0.10070	0.12170	0.08510
0083.KL	0.08160	0.10670	0.06580	0.06420	0.09370	0.15600	0.09260	0.09830
0026.KL	0.01480	0.01380	0.01480	0.08490	0.01770	0.01980	0.01890	0.09370
9008.KL	0.07030	0.03880	0.03220	0.04630	0.07840	0.05900	0.05140	0.06370
0040.KL	0.01430	0.01560	0.02660	0.01190	0.01630	0.02080	0.02960	0.01500
7160.KL	0.29990	0.62290	0.84770	0.57930	0.35780	0.72400	1.01500	0.78590
0006.KL	0.03740	0.02090	0.09430	0.06350	0.04390	0.03030	0.10130	0.07670
03002.KL	0.00530	0.00820	0.00820	0.00820	0.00610	0.01200	0.01200	0.01200
0200.KL	0.10030	0.10790	0.10790	0.10790	0.14520	0.14530	0.14530	0.14530
0106.KL	0.02120	0.01600	0.06040	0.06330	0.02570	0.02060	0.06730	0.07030
0202.KL	0.02180	0.09600	0.09560	0.09600	0.02440	0.10520	0.10470	0.10520
03008.KL	0.00120	0.01170	0.01170	0.01170	0.00140	0.01340	0.01340	0.01340
0203.KL	0.02060	0.03980	0.03980	0.03980	0.02400	0.04350	0.04350	0.04350
0117.KL	0.00220	0.00390	0.05370	0.03200	0.00280	0.00500	0.06090	0.03420
0169.KL	0.02410	0.05230	0.09070	0.09430	0.02570	0.05990	0.11180	0.12750
0093.KL	0.01430	0.02700	0.01460	0.06680	0.01750	0.03350	0.01720	0.07110
0050.KL	0.00690	0.02700	0.02160	0.09450	0.00810	0.02950	0.02570	0.10660
0132.KL	0.00560	0.00770	0.01830	0.03450	0.00640	0.00930	0.01920	0.04150
0145.KL	0.01090	0.02220	0.01540	0.01300	0.01350	0.02710	0.01890	0.01720
9075.KL	0.08280	0.06310	0.17630	0.09400	0.10020	0.07450	0.18920	0.11400
0118.KL	0.03720	0.11220	0.07140	0.05400	0.05030	0.12560	0.08080	0.06540
4359.KL	0.01640	0.00940	0.02840	0.01500	0.02420	0.01260	0.03250	0.01800
0005.KL	0.03360	0.05610	0.08610	0.21000	0.03420	0.06420	0.10580	0.22890
5005.KL	0.08360	0.09580	0.42680	0.18800	0.09190	0.11350	0.46520	0.21500
5292.KL	0.03590	0.03590	0.03590	0.03590	0.04030	0.04030	0.04030	0.04030
0060.KL	0.05690	0.40820	0.06440	0.21090	0.07330	0.41700	0.10160	0.23840

0069.KL	0.02180	0.14880	0.02760	0.03550	0.02460	0.16830	0.03400	0.04270
0120.KL	0.04990	0.07060	0.07930	0.21510	0.05650	0.07850	0.08970	0.23610
0097.KL	0.35410	0.10450	0.16170	0.21620	0.36830	0.13280	0.19970	0.29300
0066.KL	0.00300	0.07030	0.05830	0.04290	0.00370	0.08660	0.06440	0.05440
5162.KL	0.13600	0.08560	0.07560	0.04070	0.16000	0.12760	0.11310	0.07700
0008.KL	0.04810	0.02670	0.02180	0.09310	0.06060	0.03180	0.02760	0.10200
0086.KL	0.00930	0.00710	0.04260	0.03030	0.01080	0.00970	0.04760	0.03200
0094.KL	0.00290	0.01010	0.01580	0.01950	0.00360	0.01080	0.01850	0.02290
Average	**0.06221**	**0.09095**	**0.11995**	**0.24226**	**0.07217**	**0.11083**	**0.14264**	**0.27578**

Technology	SARIMA							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
0181.KL	0.06200	0.21840	0.07550	0.07550	0.07070	0.24790	0.08460	0.08460
0209.KL	0.07630	0.07630	0.07630	0.07630	0.09110	0.09110	0.09110	0.09110
0079.KL	0.01040	0.10070	0.06140	0.09940	0.01370	0.12520	0.07400	0.11120
03011.KL	0.03830	0.03170	0.03170	0.03170	0.04450	0.04140	0.04140	0.04140
0119.KL	0.00200	0.01970	0.06730	0.02100	0.00250	0.02570	0.08020	0.02680
7181.KL	0.03970	0.05660	0.46720	0.58610	0.04560	0.07420	0.53660	0.73290
0068.KL	0.02590	0.04830	0.06510	0.11960	0.02940	0.06450	0.07180	0.14100
5204.KL	0.03720	0.38230	1.46420	0.60340	0.04190	0.43020	1.76140	0.81880
0191.KL	0.00900	0.14340	0.14340	0.14340	0.01050	0.17350	0.17350	0.17350
5195.KL	0.00600	0.04490	0.12840	0.03130	0.00800	0.05530	0.14520	0.03890
03001.KL	0.11380	0.07880	0.07880	0.07880	0.11750	0.10140	0.10140	0.10140
0051.KL	0.02020	0.78350	0.33910	0.28720	0.02290	0.97400	0.38660	0.31560
7204.KL	0.14760	0.36220	0.38590	0.28290	0.17860	0.47340	0.48680	0.33880
8338.KL	0.05190	0.15160	0.16440	0.22640	0.06230	0.18650	0.18850	0.26340
0131.KL	0.00990	0.03230	0.21730	0.18940	0.01040	0.03530	0.26970	0.24910
0152.KL	0.15800	0.45440	0.56320	1.64080	0.18120	0.58240	0.72390	1.92460
0029.KL	0.02530	0.01660	0.18560	0.14300	0.02930	0.02080	0.22200	0.15360
4456.KL	0.01680	0.15900	0.66470	0.16270	0.01980	0.18500	0.77830	0.17860
5216.KL	0.04540	0.24490	0.34540	0.31930	0.06070	0.27940	0.43250	0.44170
0154.KL	0.00380	0.00710	0.00380	0.06760	0.00440	0.00870	0.00500	0.08260
5036.KL	0.01440	0.32260	0.60470	0.67900	0.02070	0.38220	0.68290	0.74450
0107.KL	0.05810	0.79670	0.09840	0.16430	0.06920	0.98710	0.11690	0.20810
0065.KL	0.07230	0.46720	0.61880	0.36140	0.08000	0.52020	0.74210	0.42980
0090.KL	0.24100	1.03680	0.86850	0.35440	0.27940	1.23990	1.04140	0.40880
0174.KL	0.00530	0.00400	0.06480	0.03030	0.00610	0.00520	0.07480	0.03420
0128.KL	0.04380	0.30480	0.78380	0.07490	0.05590	0.39750	0.91940	0.09310

9377.KL	0.00000	0.00420	0.25070	0.24660	0.00000	0.00550	0.29470	0.27700
0104.KL	0.00430	0.05640	0.03370	0.03640	0.00490	0.07160	0.03860	0.04510
0021.KL	0.04250	0.65470	0.05930	0.76390	0.05230	0.81400	0.07090	0.91010
0045.KL	0.01070	0.07250	0.05090	0.12770	0.01160	0.09210	0.06530	0.14530
0208.KL	0.11020	0.11020	0.11020	0.11020	0.13270	0.13270	0.13270	0.13270
7022.KL	0.43870	1.47550	1.75990	1.57590	0.48990	1.77480	2.10300	1.76130
0041.KL	0.00620	0.00950	0.02410	0.05380	0.00770	0.01170	0.02580	0.06390
5028.KL	0.13950	0.24090	0.68580	0.32120	0.18510	0.28820	0.73950	0.39020
0023.KL	0.09000	0.26540	0.28900	0.09150	0.10460	0.35690	0.32040	0.10920
0166.KL	0.43960	1.38690	2.09220	1.53910	0.53250	1.63500	2.52340	1.69590
0010.KL	0.02020	0.10080	0.06100	0.01830	0.02310	0.12400	0.06890	0.02110
9393.KL	0.00790	0.02980	0.17740	0.05560	0.01100	0.03520	0.21330	0.06570
5161.KL	0.02210	0.18120	0.40920	0.14060	0.03220	0.22270	0.47910	0.17940
0146.KL	0.04940	0.14460	0.42360	0.06020	0.05200	0.20000	0.52140	0.08300
0127.KL	0.09480	0.53630	1.71700	0.67970	0.10950	0.66260	1.99090	0.77530
0111.KL	0.02300	0.04340	0.52960	0.10240	0.02980	0.06430	0.61710	0.12890
9334.KL	0.24760	3.61610	19.2368	6.61080	0.28560	4.48150	22.5042	7.11580
0143.KL	0.01140	0.05910	0.16060	0.22220	0.01430	0.06830	0.18520	0.25290
0036.KL	0.00630	0.02460	0.05340	0.03500	0.00710	0.02840	0.06530	0.04590
0176.KL	0.05760	0.33990	0.45380	0.33290	0.06910	0.39920	0.54820	0.40840
0018.KL	0.30880	5.61820	3.65080	1.36170	0.31650	6.57770	4.41170	1.78180
03022.KL	0.01040	0.01040	0.01040	0.01040	0.01300	0.01300	0.01300	0.01300
5286.KL	0.21810	0.06750	0.06750	0.06750	0.24090	0.08050	0.08050	0.08050
0126.KL	0.03720	0.05820	0.05410	0.15020	0.04690	0.06910	0.05890	0.18330
0112.KL	0.00690	0.08920	0.32040	0.06430	0.00970	0.10950	0.37240	0.06940
0085.KL	0.03030	0.05490	0.26100	0.03390	0.03730	0.07940	0.29990	0.03910
0034.KL	0.16990	0.14370	0.50960	0.20010	0.20060	0.17880	0.64280	0.23190
0113.KL	0.22670	0.65450	1.40840	0.19720	0.26910	0.79140	1.63500	0.22430
0156.KL	0.01200	0.03070	0.18370	0.03530	0.01260	0.03820	0.21660	0.03760
3867.KL	1.75640	5.56940	4.11520	3.12350	2.07610	6.78560	5.20560	3.55030

0070.KL	0.00220	0.09430	0.04750	0.04230	0.00280	0.11620	0.05700	0.04960
5011.KL	0.43420	0.51010	0.18600	0.16020	0.47340	0.67280	0.25470	0.21990
0138.KL	0.10640	0.14080	0.87460	0.43440	0.13260	0.15620	1.02730	0.50810
0108.KL	0.01980	0.31020	0.48620	0.59260	0.02500	0.34810	0.59890	0.64950
0020.KL	0.34300	0.45490	0.19190	0.08000	0.39810	0.56730	0.24260	0.09290
0083.KL	0.16010	0.33460	0.28050	0.06250	0.19990	0.42510	0.36850	0.08450
0026.KL	0.01060	0.09000	0.17190	0.28960	0.01210	0.10520	0.19480	0.32710
9008.KL	0.07180	0.05480	0.04650	0.45390	0.08000	0.07910	0.05240	0.57060
0040.KL	0.01680	0.05590	0.09160	0.02150	0.01870	0.07080	0.10430	0.02790
7160.KL	0.55020	0.87390	2.07940	0.42910	0.63990	1.03150	2.48130	0.49380
0006.KL	0.03610	0.09290	0.94960	0.06850	0.04470	0.11180	1.13940	0.08900
03002.KL	0.01230	0.02300	0.02300	0.02300	0.01480	0.02670	0.02670	0.02670
0200.KL	0.04100	0.23600	0.23600	0.23600	0.05040	0.29610	0.29610	0.29610
0106.KL	0.06050	0.16860	0.79660	0.08980	0.07360	0.20070	0.96080	0.10240
0202.KL	0.03150	0.04640	0.04640	0.04640	0.03870	0.05080	0.05080	0.05080
03008.KL	0.00060	0.02360	0.02360	0.02360	0.00080	0.02720	0.02720	0.02720
0203.KL	0.04620	0.08020	0.08020	0.08020	0.04910	0.08880	0.08880	0.08880
0117.KL	0.01030	0.04580	0.03730	0.07250	0.01240	0.05500	0.04160	0.08260
0169.KL	0.21720	0.07340	0.11000	0.08750	0.25820	0.10250	0.13310	0.12120
0093.KL	0.01840	0.03530	0.17510	0.02530	0.02150	0.04550	0.20610	0.02860
0050.KL	0.02060	0.06980	0.28450	0.34240	0.02380	0.09250	0.32620	0.40180
0132.KL	0.03670	0.03800	0.06050	0.01530	0.04160	0.04480	0.07920	0.02060
0145.KL	0.02900	0.03320	0.08450	0.04740	0.03570	0.03760	0.09440	0.05640
9075.KL	0.13710	0.23570	0.41340	0.30620	0.16900	0.27870	0.46720	0.37230
0118.KL	0.02980	0.06270	0.66820	0.27610	0.03930	0.07400	0.80100	0.36190
4359.KL	0.05360	0.13630	0.21370	0.01770	0.06730	0.16860	0.25070	0.02130
0005.KL	0.02750	0.13620	0.09880	0.64920	0.02830	0.16380	0.11380	0.74730
5005.KL	0.48670	0.21790	0.68380	0.21440	0.55090	0.29970	0.76510	0.25340
5292.KL	0.08420	0.08400	0.08490	0.08370	0.09180	0.09160	0.09260	0.09130
0060.KL	0.08800	1.16730	0.27760	0.29580	0.11130	1.27740	0.33630	0.33090

0069.KL	0.06330	0.27200	0.02640	1.35050	0.08210	0.33190	0.03270	1.66160
0120.KL	0.16530	0.46520	0.41150	0.95860	0.18610	0.54840	0.49640	1.11450
0097.KL	0.22640	0.84910	3.61140	1.00460	0.23800	1.05150	4.27350	1.12610
0066.KL	0.07980	0.05100	0.06000	0.06430	0.09300	0.06050	0.06570	0.08220
5162.KL	0.17380	0.04520	0.14510	0.14500	0.20260	0.06710	0.19630	0.19890
0008.KL	0.02310	0.05210	0.14390	0.04000	0.02920	0.08030	0.17130	0.04490
0086.KL	0.01530	0.00710	0.05250	0.05720	0.01830	0.01020	0.05900	0.06180
0094.KL	0.01170	0.09120	0.04220	0.01580	0.01310	0.10440	0.05140	0.02140
Average	**0.10504**	**0.38034**	**0.65983**	**0.36618**	**0.12236**	**0.46022**	**0.78406**	**0.42502**

Technology	LSTM							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
0181.KL	0.00830	0.01410	0.01130	0.01130	0.01200	0.01910	0.01600	0.01600
0209.KL	0.02000	0.01560	0.02150	0.02050	0.02110	0.01700	0.02240	0.02160
0079.KL	0.00530	0.00640	0.00770	0.00750	0.00670	0.00820	0.01020	0.01010
03011.KL	0.01940	0.01200	0.01140	0.01210	0.02350	0.01660	0.01640	0.01730
0119.KL	0.00310	0.00210	0.00180	0.00230	0.00380	0.00290	0.00240	0.00320
7181.KL	0.02490	0.01500	0.02570	0.03590	0.02790	0.02020	0.03990	0.04810
0068.KL	0.01070	0.01230	0.00800	0.00720	0.01310	0.01410	0.01050	0.00970
5204.KL	0.01670	0.02010	0.02990	0.07020	0.02200	0.02430	0.03880	0.09180
0191.KL	0.00800	0.00690	0.00710	0.00700	0.01080	0.00920	0.00930	0.00950
5195.KL	0.00460	0.00390	0.00530	0.00470	0.00560	0.00490	0.00630	0.00600
03001.KL	0.04150	0.01520	0.01560	0.01660	0.04920	0.02540	0.02760	0.02710
0051.KL	0.00840	0.00610	0.00800	0.01380	0.01090	0.00890	0.01130	0.01640
7204.KL	0.04780	0.03060	0.02380	0.02520	0.05730	0.03970	0.03150	0.03400
8338.KL	0.00820	0.01100	0.00900	0.00720	0.01070	0.01250	0.01090	0.00910
0131.KL	0.00410	0.00340	0.00310	0.00380	0.00510	0.00500	0.00440	0.00530
0152.KL	0.09640	0.04820	0.03560	0.03650	0.12170	0.06370	0.04750	0.04930
0029.KL	0.00670	0.00780	0.00830	0.00750	0.00890	0.00970	0.01020	0.00970
4456.KL	0.00470	0.00720	0.01000	0.01120	0.00660	0.00880	0.01410	0.01530
5216.KL	0.03410	0.03270	0.01710	0.01790	0.03930	0.04330	0.02600	0.02760
0154.KL	0.00240	0.00190	0.00220	0.00250	0.00250	0.00240	0.00260	0.00300
5036.KL	0.01650	0.01680	0.01500	0.01900	0.02200	0.02230	0.02160	0.03260
0107.KL	0.00750	0.00970	0.00820	0.00780	0.00890	0.01300	0.01130	0.01030
0065.KL	0.02370	0.02370	0.02290	0.02990	0.02930	0.04300	0.03600	0.04140
0090.KL	0.02690	0.02990	0.04640	0.04180	0.03300	0.03760	0.05550	0.05300
0174.KL	0.00270	0.00180	0.00310	0.00610	0.00330	0.00240	0.00350	0.00750

0128.KL	0.09850	0.03820	0.04860	0.03110	0.11490	0.04730	0.06330	0.03960
9377.KL	0.00000	0.00010	0.00670	0.00930	0.00000	0.00010	0.00670	0.01300
0104.KL	0.00160	0.00320	0.00290	0.00340	0.00190	0.00410	0.00400	0.00490
0021.KL	0.01450	0.03790	0.02460	0.02390	0.01790	0.04720	0.03410	0.03290
0045.KL	0.00370	0.00360	0.00370	0.00330	0.00450	0.00470	0.00570	0.00490
0208.KL	0.14620	0.13730	0.11390	0.13780	0.15880	0.15010	0.12870	0.15030
7022.KL	0.09210	0.04410	0.06770	0.06320	0.10380	0.05920	0.08160	0.08210
0041.KL	0.00470	0.00300	0.00250	0.00260	0.00560	0.00390	0.00340	0.00350
5028.KL	0.09030	0.06420	0.05970	0.04690	0.11460	0.09750	0.09370	0.07830
0023.KL	0.05480	0.02300	0.02410	0.02160	0.06790	0.03550	0.03420	0.02970
0166.KL	0.05500	0.04700	0.05750	0.06590	0.07700	0.05810	0.07190	0.08430
0010.KL	0.00450	0.00420	0.00450	0.00500	0.00510	0.00520	0.00620	0.00660
9393.KL	0.00600	0.00440	0.00450	0.00500	0.00730	0.00560	0.00620	0.00660
5161.KL	0.01920	0.01230	0.01090	0.01500	0.02630	0.01910	0.01520	0.01830
0146.KL	0.03290	0.01070	0.00910	0.00930	0.03510	0.01430	0.01330	0.01320
0127.KL	0.09040	0.04310	0.05360	0.05480	0.11280	0.05570	0.06860	0.06920
0111.KL	0.00770	0.00550	0.00690	0.00880	0.00940	0.00720	0.00920	0.01240
9334.KL	0.50500	0.23550	0.26010	0.45430	0.53830	0.30080	0.34610	0.69690
0143.KL	0.00820	0.01170	0.00640	0.00660	0.00880	0.01250	0.00760	0.00850
0036.KL	0.00230	0.00250	0.00260	0.00900	0.00310	0.00320	0.00330	0.00960
0176.KL	0.03750	0.02130	0.01620	0.01820	0.04490	0.02860	0.02120	0.02450
0018.KL	0.09230	0.14200	0.12440	0.23040	0.11910	0.16700	0.17350	0.27910
03022.KL	0.01270	0.01630	0.01450	0.01610	0.01270	0.01630	0.01450	0.01610
5286.KL	0.11900	0.07410	0.07390	0.07940	0.13390	0.08870	0.08890	0.09410
0126.KL	0.01080	0.00930	0.00640	0.00500	0.01420	0.01330	0.00860	0.00760
0112.KL	0.00640	0.00650	0.00860	0.01020	0.00840	0.00840	0.01030	0.01310
0085.KL	0.02050	0.01780	0.01640	0.01480	0.02810	0.02390	0.02210	0.01910

0034.KL	0.05430	0.01230	0.00910	0.00930	0.06100	0.01760	0.01360	0.01380
0113.KL	0.03620	0.03930	0.02920	0.04130	0.04620	0.05010	0.03940	0.05330
0156.KL	0.00440	0.00470	0.00550	0.00510	0.00680	0.00590	0.00660	0.00650
3867.KL	0.67030	0.25840	0.19220	0.34930	0.73250	0.33930	0.26220	0.43010
0070.KL	0.00250	0.00200	0.00290	0.00260	0.00310	0.00240	0.00390	0.00350
5011.KL	0.09490	0.04630	0.05400	0.05490	0.13690	0.06480	0.08750	0.09070
0138.KL	0.05650	0.02000	0.04160	0.02920	0.06220	0.02650	0.04820	0.03800
0108.KL	0.01160	0.01700	0.01640	0.02410	0.01310	0.02060	0.02810	0.03490
0020.KL	0.03270	0.04100	0.03680	0.04800	0.04910	0.04940	0.04770	0.05850
0083.KL	0.04320	0.02470	0.01660	0.01780	0.05290	0.03570	0.02550	0.02550
0026.KL	0.01180	0.00530	0.00520	0.00490	0.01490	0.00740	0.00710	0.00680
9008.KL	0.04550	0.01130	0.00890	0.01300	0.04990	0.01640	0.01220	0.01550
0040.KL	0.01500	0.00440	0.00380	0.00490	0.01650	0.00600	0.00520	0.00650
7160.KL	0.08790	0.11050	0.10000	0.16580	0.09680	0.14020	0.13710	0.23160
0006.KL	0.02630	0.01640	0.01260	0.01780	0.03300	0.02350	0.01930	0.03190
03002.KL	0.00350	0.00480	0.00410	0.00420	0.00350	0.00980	0.00940	0.00950
0200.KL	0.07960	0.07020	0.11530	0.07950	0.09590	0.09050	0.13450	0.10000
0106.KL	0.00980	0.01160	0.01620	0.01400	0.01290	0.01490	0.02390	0.01880
0202.KL	0.01860	0.01650	0.01770	0.01860	0.02170	0.01920	0.02030	0.02120
03008.KL	0.00030	0.00430	0.00330	0.00350	0.00030	0.00430	0.00340	0.00350
0203.KL	0.01310	0.01160	0.01190	0.01170	0.01800	0.01580	0.01580	0.01550
0117.KL	0.00420	0.00390	0.00940	0.00370	0.00500	0.00480	0.01020	0.00480
0169.KL	0.02920	0.01430	0.00960	0.00810	0.03110	0.02320	0.01650	0.01430
0093.KL	0.00600	0.00420	0.00530	0.00740	0.00890	0.00610	0.00680	0.00860
0050.KL	0.00570	0.00870	0.00580	0.00710	0.00700	0.01000	0.00760	0.00940
0132.KL	0.00290	0.00280	0.00800	0.00680	0.00420	0.00430	0.00910	0.00780
0145.KL	0.01400	0.00780	0.00880	0.00790	0.01630	0.00990	0.01080	0.00960

9075.KL	0.02640	0.02660	0.01610	0.01540	0.03720	0.03390	0.02270	0.02190
0118.KL	0.02960	0.02590	0.04910	0.03950	0.03760	0.03360	0.05720	0.04760
4359.KL	0.00960	0.00790	0.00900	0.01040	0.01420	0.01060	0.01130	0.01210
0005.KL	0.00780	0.00870	0.02480	0.01870	0.01080	0.01270	0.02780	0.02250
5005.KL	0.03790	0.04660	0.04560	0.03710	0.04250	0.06210	0.06020	0.05450
5292.KL	0.04610	0.03730	0.04520	0.05030	0.06180	0.04870	0.05990	0.06510
0060.KL	0.04860	0.04260	0.03890	0.02910	0.05720	0.07170	0.06750	0.05200
0069.KL	0.02210	0.02300	0.02120	0.02540	0.02320	0.02660	0.02500	0.03100
0120.KL	0.02160	0.01820	0.02170	0.03280	0.03480	0.02390	0.02780	0.04100
0097.KL	0.06290	0.09120	0.09530	0.09510	0.08970	0.12690	0.11500	0.13020
0066.KL	0.00810	0.01010	0.00830	0.00740	0.00880	0.01240	0.01100	0.00950
5162.KL	0.06090	0.02050	0.01100	0.01030	0.07930	0.03920	0.02470	0.02110
0008.KL	0.01950	0.00980	0.01110	0.01150	0.02450	0.01260	0.01340	0.01510
0086.KL	0.00740	0.00570	0.00690	0.00580	0.01170	0.00800	0.00800	0.00700
0094.KL	0.00380	0.00310	0.00500	0.00550	0.00440	0.00370	0.00830	0.00790
Average	**0.04023**	**0.02686**	**0.02711**	**0.03331**	**0.04752**	**0.03507**	**0.03596**	**0.04470**

Technology	Holt Winter							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
0181.KL	0.05110	0.05510	0.03260	0.03260	0.05860	0.06420	0.03900	0.03900
0209.KL	0.01300	0.01300	0.01300	0.01300	0.01520	0.01520	0.01520	0.01520
0079.KL	0.00720	0.03570	0.05070	0.05330	0.00900	0.04260	0.05990	0.06470
03011.KL	0.04630	0.03500	0.03500	0.03500	0.05380	0.04520	0.04520	0.04520
0119.KL	0.00210	0.01540	0.00400	0.04570	0.00250	0.01950	0.00590	0.04800
7181.KL	0.03830	0.06530	0.11240	0.15330	0.04300	0.07510	0.12670	0.16630
0068.KL	0.01580	0.06770	0.06160	0.14320	0.01860	0.07670	0.06950	0.16880
5204.KL	0.04050	0.28660	0.15680	0.63310	0.04720	0.31460	0.17600	0.67320
0191.KL	0.00990	0.04190	0.04190	0.04190	0.01220	0.05050	0.05050	0.05050
5195.KL	0.00870	0.00610	0.04180	0.01950	0.01010	0.00780	0.04380	0.02180
03001.KL	0.11120	0.07860	0.07860	0.07860	0.11480	0.10370	0.10370	0.10370
0051.KL	0.00920	0.01100	0.02030	0.10800	0.01230	0.01420	0.02780	0.11520
7204.KL	0.06160	0.06760	0.11690	0.12730	0.07750	0.07990	0.14340	0.15550
8338.KL	0.00710	0.00970	0.01860	0.04440	0.00860	0.01220	0.02120	0.05060
0131.KL	0.00550	0.01250	0.00900	0.04400	0.00700	0.01390	0.01120	0.04760
0152.KL	0.18380	0.09680	0.15900	0.22030	0.21400	0.12860	0.18710	0.26060
0029.KL	0.00700	0.05160	0.05060	0.03560	0.00740	0.05690	0.05970	0.04360
4456.KL	0.01320	0.02880	0.05040	0.12990	0.01540	0.03120	0.05330	0.14180
5216.KL	0.04850	0.23680	0.16500	0.10520	0.06420	0.26980	0.21840	0.12140
0154.KL	0.00250	0.00260	0.00250	0.00330	0.00270	0.00310	0.00290	0.00420
5036.KL	0.01560	0.02840	0.05040	0.12580	0.02100	0.03500	0.06770	0.15030
0107.KL	0.01520	0.06700	0.06200	0.03310	0.01700	0.07280	0.07400	0.03620
0065.KL	0.10200	0.31420	0.34410	0.09060	0.11460	0.34280	0.40530	0.11040

0090.KL	0.09890	0.21080	0.16410	0.27140	0.11350	0.23820	0.18860	0.30770
0174.KL	0.00250	0.00750	0.00980	0.03470	0.00290	0.01010	0.01290	0.03670
0128.KL	0.05260	0.20780	0.41900	0.42870	0.06540	0.25000	0.48810	0.51290
9377.KL	0.00000	0.01200	0.01570	0.09920	0.00000	0.01390	0.01810	0.10800
0104.KL	0.00430	0.02560	0.02670	0.02590	0.00480	0.02690	0.03040	0.03090
0021.KL	0.01670	0.09180	0.16860	0.21190	0.02050	0.09970	0.21140	0.26000
0045.KL	0.00350	0.00710	0.00790	0.02370	0.00450	0.00830	0.00980	0.02610
0208.KL	0.04070	0.04070	0.04070	0.04070	0.04490	0.04490	0.04490	0.04490
7022.KL	0.09130	0.50450	0.17750	0.65990	0.10290	0.59290	0.23120	0.72060
0041.KL	0.00290	0.00930	0.00740	0.02220	0.00320	0.01140	0.00850	0.02470
5028.KL	0.19860	0.14720	0.50920	0.31010	0.23720	0.18620	0.54450	0.37340
0023.KL	0.09740	0.05240	0.22760	0.07080	0.10700	0.06560	0.25090	0.08150
0166.KL	0.17600	0.24610	0.12210	0.80820	0.22330	0.28520	0.16260	0.86590
0010.KL	0.00410	0.00730	0.02980	0.00740	0.00560	0.00850	0.03330	0.01010
9393.KL	0.00860	0.01300	0.02270	0.04270	0.01020	0.01530	0.02370	0.05040
5161.KL	0.02700	0.06100	0.06320	0.03880	0.03980	0.07480	0.07100	0.04490
0146.KL	0.07700	0.03950	0.04210	0.03220	0.08300	0.05640	0.06250	0.03890
0127.KL	0.06070	0.09500	0.26290	0.11630	0.08250	0.12380	0.29150	0.15590
0111.KL	0.01120	0.04250	0.06110	0.03220	0.01230	0.04750	0.06390	0.04370
9334.KL	0.55190	2.60790	8.56530	9.07160	0.59310	3.19320	9.68540	9.81600
0143.KL	0.00530	0.02110	0.02420	0.09750	0.00590	0.02570	0.03300	0.10580
0036.KL	0.00510	0.01370	0.01140	0.01220	0.00580	0.01510	0.01320	0.01520
0176.KL	0.08920	0.11160	0.04440	0.04400	0.09520	0.12880	0.06120	0.06150
0018.KL	0.28270	0.16350	1.46310	1.69610	0.29050	0.20630	1.70130	2.15370
03022.KL	0.00210	0.00210	0.00210	0.00210	0.00260	0.00260	0.00260	0.00260
5286.KL	0.05770	0.29480	0.29480	0.29480	0.06740	0.32320	0.32320	0.32320
0126.KL	0.03020	0.06140	0.02380	0.02720	0.03850	0.07250	0.02960	0.03650

0112.KL	0.00670	0.00960	0.02160	0.11340	0.00950	0.01180	0.02500	0.12500
0085.KL	0.03520	0.03150	0.06320	0.02480	0.04250	0.03640	0.07240	0.03090
0034.KL	0.07540	0.07110	0.04820	0.03520	0.09040	0.09230	0.07210	0.05260
0113.KL	0.20610	0.17200	0.43580	0.38170	0.24720	0.21090	0.47590	0.42630
0156.KL	0.00670	0.00500	0.01820	0.06310	0.00880	0.00640	0.01960	0.06800
3867.KL	0.38790	2.22530	0.88590	1.65380	0.47090	2.68360	1.12280	1.84930
0070.KL	0.00190	0.00310	0.01450	0.00770	0.00210	0.00380	0.01680	0.00910
5011.KL	0.23630	0.17290	0.27020	0.22140	0.26050	0.20520	0.32160	0.28560
0138.KL	0.07530	0.06330	0.16330	0.08710	0.07950	0.07110	0.18850	0.10840
0108.KL	0.04060	0.02800	0.16620	0.41170	0.04730	0.04140	0.18200	0.46050
0020.KL	0.04200	0.08440	0.08420	0.07080	0.04410	0.10970	0.11660	0.08410
0083.KL	0.03390	0.20000	0.09440	0.11160	0.04050	0.25690	0.15060	0.14830
0026.KL	0.01480	0.01310	0.01450	0.08520	0.01750	0.01870	0.01850	0.09410
9008.KL	0.06750	0.04380	0.03870	0.04330	0.07550	0.06320	0.05870	0.06120
0040.KL	0.01400	0.01340	0.02680	0.00900	0.01610	0.01820	0.02970	0.01090
7160.KL	0.30170	0.62170	0.85190	0.57690	0.36080	0.72260	1.01880	0.78340
0006.KL	0.03790	0.02630	0.08950	0.06190	0.04140	0.03650	0.09670	0.07530
03002.KL	0.00470	0.00910	0.00910	0.00910	0.00570	0.01290	0.01290	0.01290
0200.KL	0.09460	0.10500	0.10500	0.10500	0.13970	0.14210	0.14210	0.14210
0106.KL	0.02310	0.01630	0.06280	0.06420	0.02800	0.02090	0.06960	0.07130
0202.KL	0.01050	0.04910	0.04910	0.04910	0.01210	0.05280	0.05280	0.05280
03008.KL	0.00030	0.01080	0.01080	0.01080	0.00030	0.01280	0.01280	0.01280
0203.KL	0.02220	0.04160	0.04160	0.04160	0.02550	0.04530	0.04530	0.04530
0117.KL	0.00210	0.00420	0.02120	0.03480	0.00270	0.00520	0.02340	0.03720
0169.KL	0.07930	0.05260	0.09120	0.09170	0.08770	0.06010	0.11250	0.12530
0093.KL	0.01390	0.02920	0.01410	0.06770	0.01700	0.03610	0.01670	0.07210
0050.KL	0.00990	0.02720	0.02210	0.09560	0.01160	0.02970	0.02590	0.10790

0132.KL	0.00610	0.00660	0.01860	0.03500	0.00690	0.00820	0.01960	0.04220
0145.KL	0.01050	0.02280	0.01530	0.01270	0.01300	0.02810	0.01900	0.01680
9075.KL	0.08180	0.06330	0.17820	0.09370	0.09990	0.07460	0.19090	0.11350
0118.KL	0.04060	0.10930	0.06830	0.05500	0.05260	0.12280	0.07820	0.06640
4359.KL	0.01630	0.00960	0.02790	0.01570	0.02380	0.01280	0.03200	0.01880
0005.KL	0.03370	0.05690	0.08730	0.22130	0.03440	0.06520	0.10700	0.24170
5005.KL	0.09840	0.09600	0.42640	0.18840	0.10690	0.11370	0.46470	0.21540
5292.KL	0.04360	0.04360	0.04360	0.04360	0.05070	0.05070	0.05070	0.05070
0060.KL	0.05230	0.40820	0.06410	0.21070	0.06690	0.41720	0.10200	0.23810
0069.KL	0.02240	0.14910	0.03030	0.03550	0.02470	0.16810	0.03760	0.04280
0120.KL	0.05390	0.07040	0.08010	0.21680	0.06050	0.07830	0.09060	0.23790
0097.KL	0.35020	0.10210	0.17360	0.21640	0.36350	0.13190	0.21140	0.29460
0066.KL	0.03460	0.07380	0.05830	0.04280	0.03990	0.09140	0.06430	0.05400
5162.KL	0.13630	0.08500	0.07300	0.04030	0.16020	0.12720	0.11110	0.07690
0008.KL	0.04670	0.02640	0.02170	0.09570	0.05950	0.03140	0.02740	0.10470
0086.KL	0.01540	0.00720	0.04280	0.03140	0.01720	0.00970	0.04770	0.03310
0094.KL	0.00350	0.01110	0.01540	0.01820	0.00410	0.01170	0.01830	0.02190
Average	**0.06068**	**0.12974**	**0.20876**	**0.24405**	**0.06999**	**0.15481**	**0.24231**	**0.27774**

Technology	Facebook Prophet							
	MAE				RMSE			
	1 year	3 years	5 years	7 years	1 year	3 years	5 years	7 years
0181.KL	0.05070	0.03640	0.10960	0.10960	0.05800	0.04300	0.11550	0.11550
0209.KL	0.08550	0.08550	0.08550	0.08550	0.15020	0.15020	0.15020	0.15020
0079.KL	0.02360	0.01750	0.07130	0.13410	0.02450	0.02130	0.07600	0.14300
03011.KL	0.04270	0.03490	0.03490	0.03490	0.04930	0.04460	0.04460	0.04460
0119.KL	0.00890	0.00960	0.03070	0.01780	0.00930	0.01320	0.03800	0.01970
7181.KL	0.06980	0.12720	0.08260	0.27740	0.07250	0.14690	0.10100	0.29030
0068.KL	0.00970	0.05830	0.10250	0.07510	0.01210	0.06310	0.13470	0.09190
5204.KL	0.04390	0.50460	0.26750	0.18950	0.05150	0.53120	0.31870	0.21780
0191.KL	0.01960	0.08900	0.08930	0.08730	0.02230	0.10160	0.10190	0.09970
5195.KL	0.01280	0.05040	0.03330	0.05600	0.01390	0.05400	0.04400	0.06240
03001.KL	0.11010	0.06510	0.06510	0.06510	0.11360	0.08930	0.08930	0.08930
0051.KL	0.01530	0.12110	0.10530	0.23610	0.02180	0.12310	0.12650	0.24360
7204.KL	0.11160	0.10540	0.17870	0.07850	0.11780	0.14470	0.19720	0.10120
8338.KL	0.03360	0.06860	0.07020	0.16090	0.03500	0.07820	0.07660	0.16610
0131.KL	0.00670	0.09340	0.02980	0.12960	0.00820	0.09760	0.03350	0.13730
0152.KL	0.30100	0.25340	0.30320	0.56790	0.32160	0.31500	0.33120	0.63750
0029.KL	0.00820	0.07770	0.07370	0.06580	0.01010	0.08050	0.08510	0.08170
4456.KL	0.00980	0.10510	0.02710	0.03290	0.01260	0.12460	0.03410	0.04100
5216.KL	0.05660	0.21850	0.30090	0.13620	0.07310	0.25270	0.35330	0.15660
0154.KL	0.00270	0.00790	0.00780	0.00520	0.00360	0.00920	0.00960	0.00630
5036.KL	0.03960	0.13120	0.05120	0.20290	0.04810	0.14650	0.06120	0.22710
0107.KL	0.04790	0.06050	0.08260	0.04410	0.04920	0.06490	0.09920	0.05510
0065.KL	0.13100	0.20290	0.62100	0.17340	0.14380	0.23180	0.67590	0.20470

0090.KL	0.10210	0.10690	0.38750	0.14450	0.11920	0.11650	0.40710	0.16990
0174.KL	0.00540	0.03080	0.04140	0.03750	0.00630	0.03300	0.04450	0.04000
0128.KL	0.12040	0.07980	0.29710	0.45230	0.14820	0.10080	0.35570	0.53190
9377.KL	0.00000	0.01760	0.03330	0.08090	0.00000	0.02050	0.03730	0.08820
0104.KL	0.00230	0.05500	0.02520	0.02440	0.00270	0.05810	0.02950	0.02930
0021.KL	0.01950	0.28190	0.18480	0.10980	0.02340	0.28860	0.21960	0.13480
0045.KL	0.00340	0.02970	0.03850	0.01890	0.00430	0.03130	0.04150	0.02140
0208.KL	0.05270	0.05270	0.05270	0.05270	0.06340	0.06340	0.06340	0.06340
7022.KL	0.06040	0.32880	0.18240	0.24320	0.07260	0.42990	0.22770	0.27960
0041.KL	0.00520	0.00670	0.00620	0.00670	0.00640	0.00840	0.00740	0.00850
5028.KL	0.19760	0.10570	0.64860	0.43060	0.23380	0.13240	0.69140	0.53150
0023.KL	0.18940	0.11680	0.21170	0.12390	0.19360	0.13700	0.23890	0.15350
0166.KL	0.21510	0.20890	0.21220	0.83910	0.27320	0.25080	0.25530	0.90710
0010.KL	0.00320	0.01570	0.03720	0.01540	0.00370	0.01790	0.04410	0.01960
9393.KL	0.00550	0.05070	0.01530	0.09970	0.00750	0.05160	0.01870	0.11560
5161.KL	0.04670	0.06900	0.06900	0.09550	0.05630	0.08270	0.09110	0.11400
0146.KL	0.09940	0.09620	0.07560	0.08800	0.10400	0.10630	0.09330	0.11160
0127.KL	0.12330	0.15690	0.42880	0.25830	0.15240	0.18880	0.47030	0.32450
0111.KL	0.03490	0.02220	0.09830	0.08530	0.03620	0.02730	0.10650	0.09720
9334.KL	1.53070	1.38020	2.62020	10.1433	1.59660	1.89020	3.11920	10.7784
0143.KL	0.00360	0.02510	0.01690	0.11810	0.00450	0.02850	0.02010	0.12680
0036.KL	0.00360	0.04410	0.02820	0.02970	0.00430	0.04570	0.03440	0.03320
0176.KL	0.09440	0.14410	0.14000	0.15320	0.10000	0.17170	0.16340	0.17780
0018.KL	0.19580	2.15020	0.51560	0.97100	0.21090	2.20000	0.59410	1.08620
03022.KL	0.11520	0.11520	0.11520	0.11520	0.23450	0.23450	0.23450	0.23450
5286.KL	0.05700	0.47820	0.48760	0.46610	0.06890	0.50280	0.51220	0.49030
0126.KL	0.02810	0.03570	0.06840	0.08920	0.03670	0.04230	0.08010	0.10180

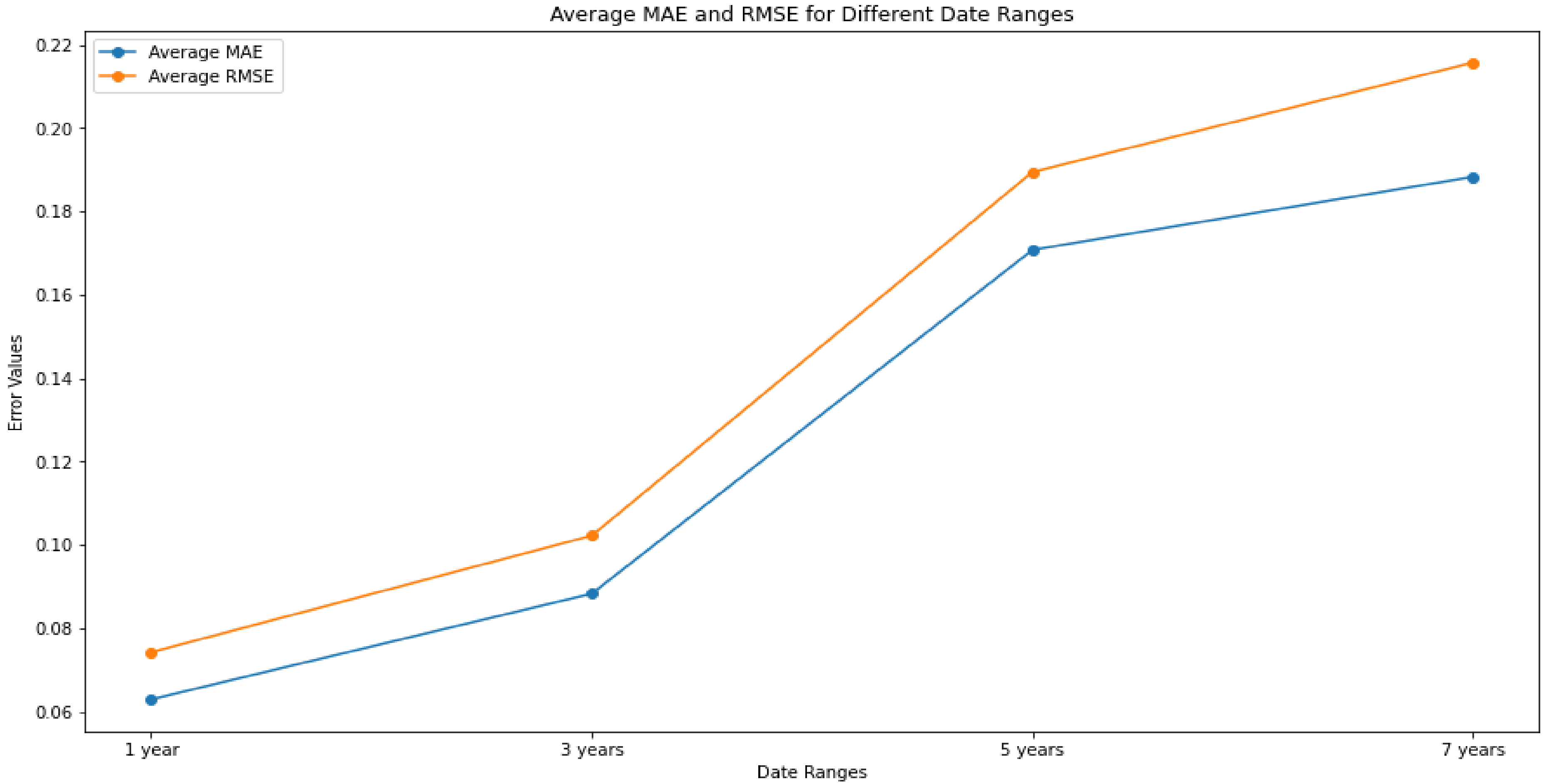
0112.KL	0.02450	0.03940	0.01760	0.05510	0.02600	0.04470	0.02340	0.06250
0085.KL	0.05690	0.14590	0.12470	0.28380	0.06510	0.19910	0.13520	0.31510
0034.KL	0.09710	0.05990	0.05220	0.04860	0.11840	0.07990	0.05960	0.06210
0113.KL	0.07860	0.24030	0.10590	0.28400	0.08380	0.25650	0.12990	0.30840
0156.KL	0.01110	0.01320	0.01680	0.01730	0.01290	0.01680	0.02030	0.02200
3867.KL	0.52490	1.04130	0.96090	3.24260	0.60100	1.23470	1.16830	3.43160
0070.KL	0.00210	0.00300	0.01580	0.03920	0.00230	0.00360	0.01880	0.04140
5011.KL	0.31360	0.51990	0.37290	0.19980	0.33880	0.54020	0.40750	0.24430
0138.KL	0.09150	0.28220	0.40130	0.12470	0.09310	0.30140	0.46370	0.16420
0108.KL	0.02490	0.11170	0.38360	0.38500	0.02830	0.12010	0.40840	0.44330
0020.KL	0.13320	0.16240	0.19170	0.12250	0.13870	0.18520	0.25750	0.16240
0083.KL	0.15370	0.26960	0.09970	0.38510	0.15920	0.32830	0.10810	0.41450
0026.KL	0.01370	0.02800	0.07090	0.05460	0.01500	0.03300	0.07890	0.06220
9008.KL	0.07300	0.06870	0.07070	0.04060	0.08110	0.08450	0.09240	0.05100
0040.KL	0.01330	0.01030	0.01700	0.05960	0.01540	0.01210	0.01960	0.06500
7160.KL	0.36600	0.44580	0.67200	0.66090	0.43150	0.51690	0.83030	0.87260
0006.KL	0.03650	0.06450	0.05390	0.16690	0.04570	0.07330	0.06190	0.17490
03002.KL	0.01260	0.02130	0.02130	0.02130	0.01390	0.02520	0.02520	0.02520
0200.KL	0.20490	0.09760	0.09760	0.09760	0.23910	0.13390	0.13390	0.13390
0106.KL	0.03470	0.03230	0.10940	0.09810	0.03820	0.03850	0.11830	0.11610
0202.KL	0.02110	0.03850	0.03790	0.03790	0.02350	0.04180	0.04110	0.04120
03008.KL	0.00610	0.00390	0.00390	0.00390	0.00650	0.00400	0.00400	0.00400
0203.KL	0.04320	0.03690	0.03690	0.03690	0.04590	0.04090	0.04090	0.04090
0117.KL	0.00280	0.02480	0.00880	0.03400	0.00350	0.02640	0.01070	0.04000
0169.KL	0.13300	0.11380	0.09830	0.05510	0.13490	0.12280	0.11680	0.08240
0093.KL	0.01930	0.02180	0.04450	0.04650	0.02210	0.02870	0.05040	0.05750
0050.KL	0.01140	0.07940	0.02250	0.02000	0.01440	0.09180	0.02770	0.02590

0132.KL	0.00960	0.01780	0.02910	0.01890	0.01150	0.02000	0.03250	0.02180
0145.KL	0.01180	0.01890	0.01780	0.02580	0.01380	0.02320	0.02070	0.02970
9075.KL	0.04220	0.09460	0.19850	0.07690	0.05140	0.10520	0.22070	0.08670
0118.KL	0.03440	0.17890	0.42760	0.70270	0.03970	0.20850	0.46840	0.80070
4359.KL	0.01640	0.01190	0.02340	0.01680	0.02330	0.01520	0.02840	0.02100
0005.KL	0.06540	0.08640	0.02340	0.32530	0.06650	0.10240	0.02740	0.34910
5005.KL	0.10030	0.41850	0.23600	0.48860	0.11340	0.45190	0.27610	0.50260
5292.KL	0.21290	0.21330	0.21280	0.21290	0.28160	0.28200	0.28150	0.28160
0060.KL	0.05050	0.06900	0.10650	0.26250	0.06170	0.07940	0.14200	0.29680
0069.KL	0.02080	0.05450	0.40240	0.15750	0.02490	0.06360	0.45360	0.18380
0120.KL	0.03710	0.17280	0.15690	0.17350	0.04460	0.18340	0.17180	0.18750
0097.KL	0.39470	0.39060	0.27980	0.28200	0.42210	0.41540	0.31660	0.33570
0066.KL	0.04530	0.08920	0.05850	0.04540	0.04940	0.11080	0.06930	0.05290
5162.KL	0.15050	0.10120	0.10600	0.11920	0.17500	0.13640	0.15090	0.16090
0008.KL	0.05960	0.09610	0.03220	0.09490	0.06990	0.13400	0.03990	0.10280
0086.KL	0.02200	0.01460	0.04960	0.02090	0.02520	0.01750	0.05610	0.02660
0094.KL	0.00250	0.00710	0.01370	0.01820	0.00310	0.00830	0.01780	0.02160
Average	**0.08868**	**0.15682**	**0.17388**	**0.29704**	**0.10065**	**0.18223**	**0.20346**	**0.33063**

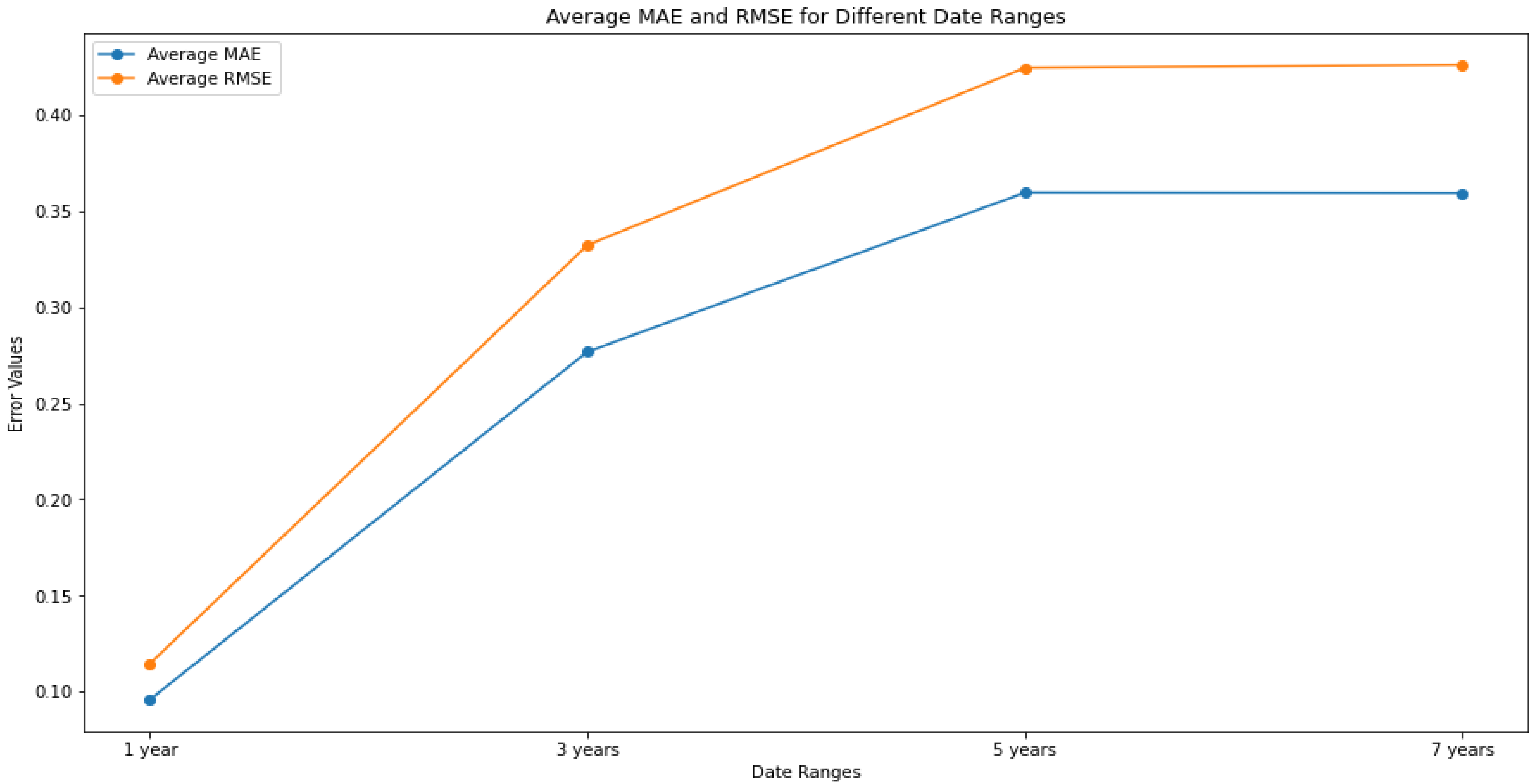
Appendix 1.3 Average MAE and RMSE for Different Date Ranges

Healthcare Stock

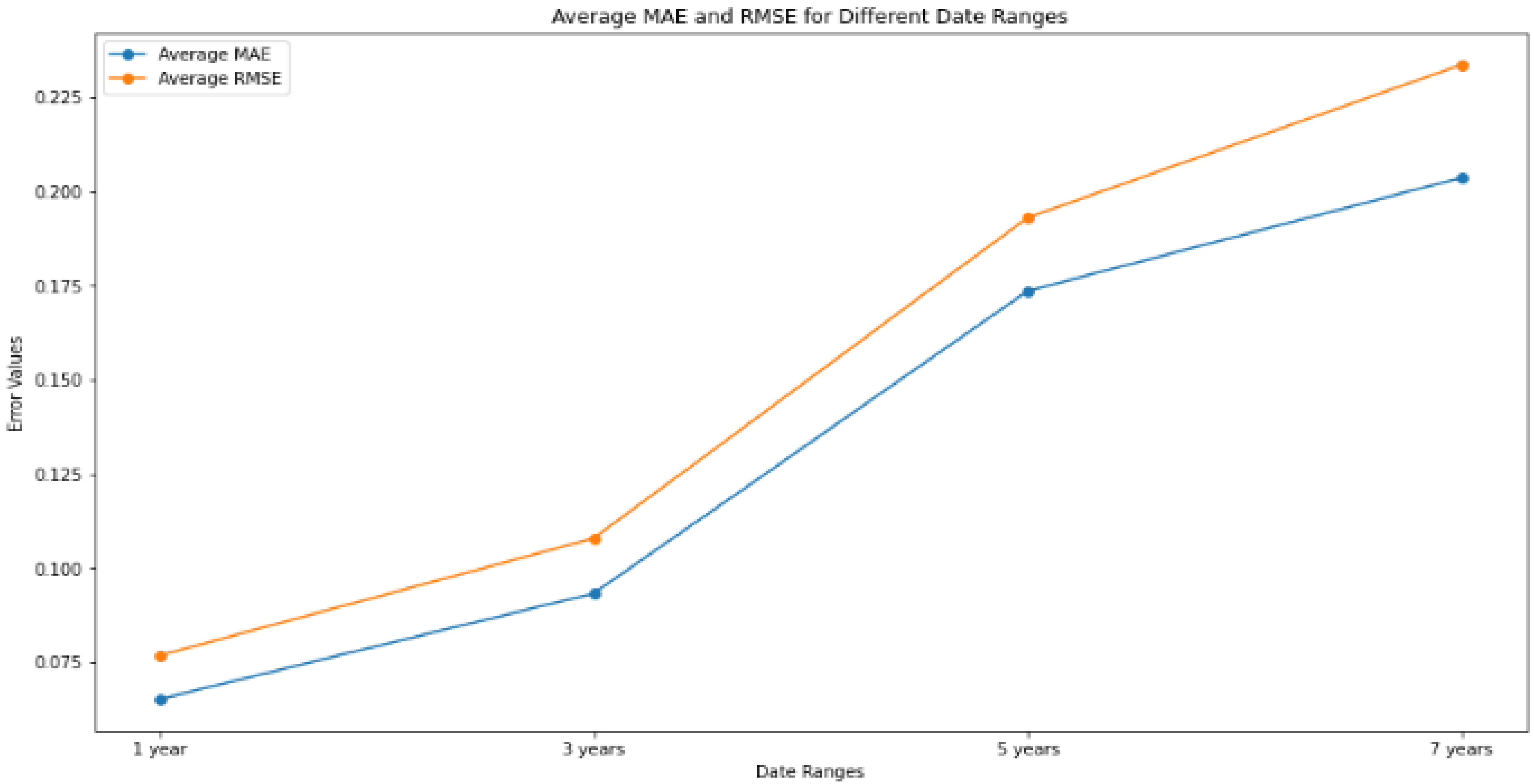
ARIMA Model



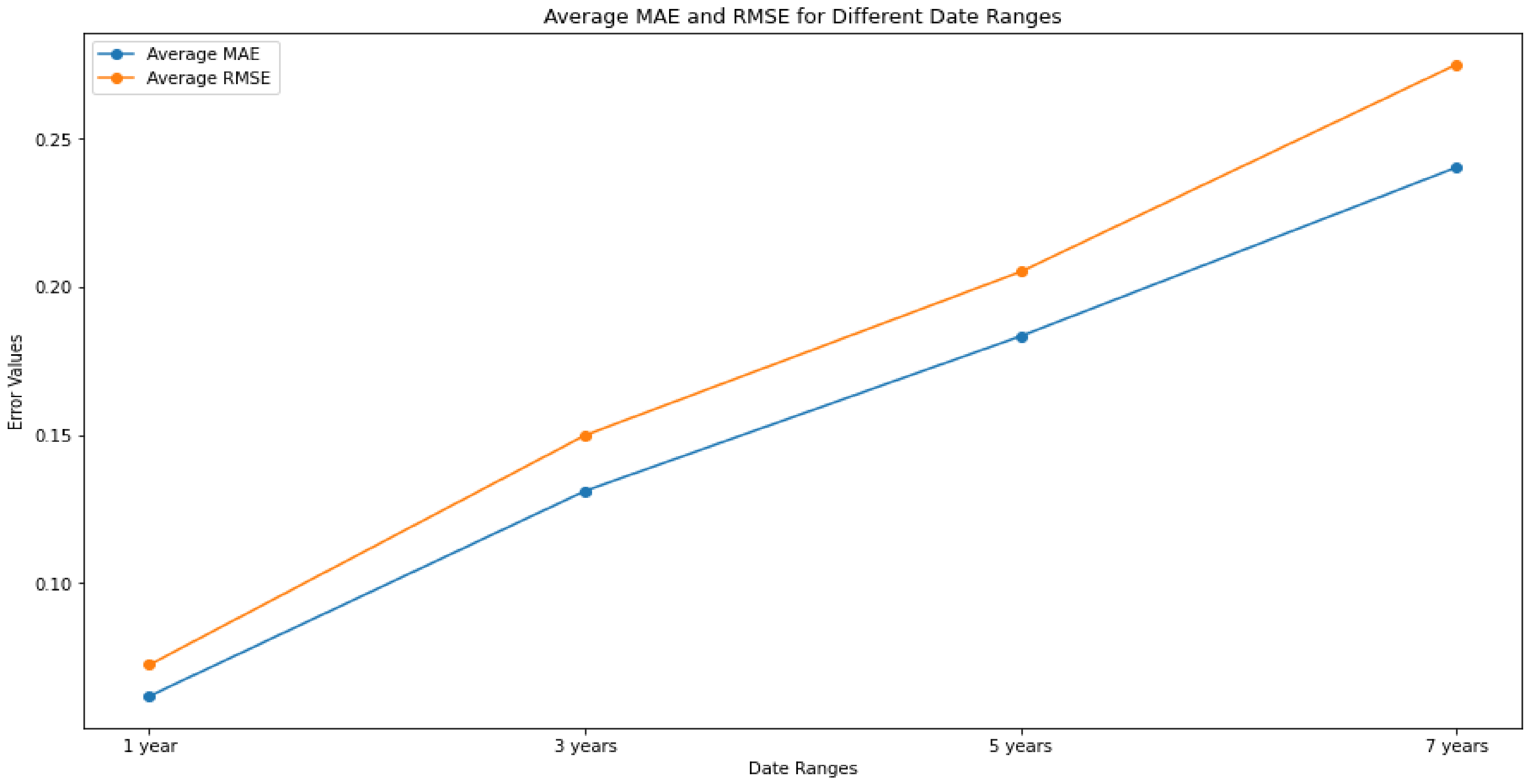
SARIMA Model



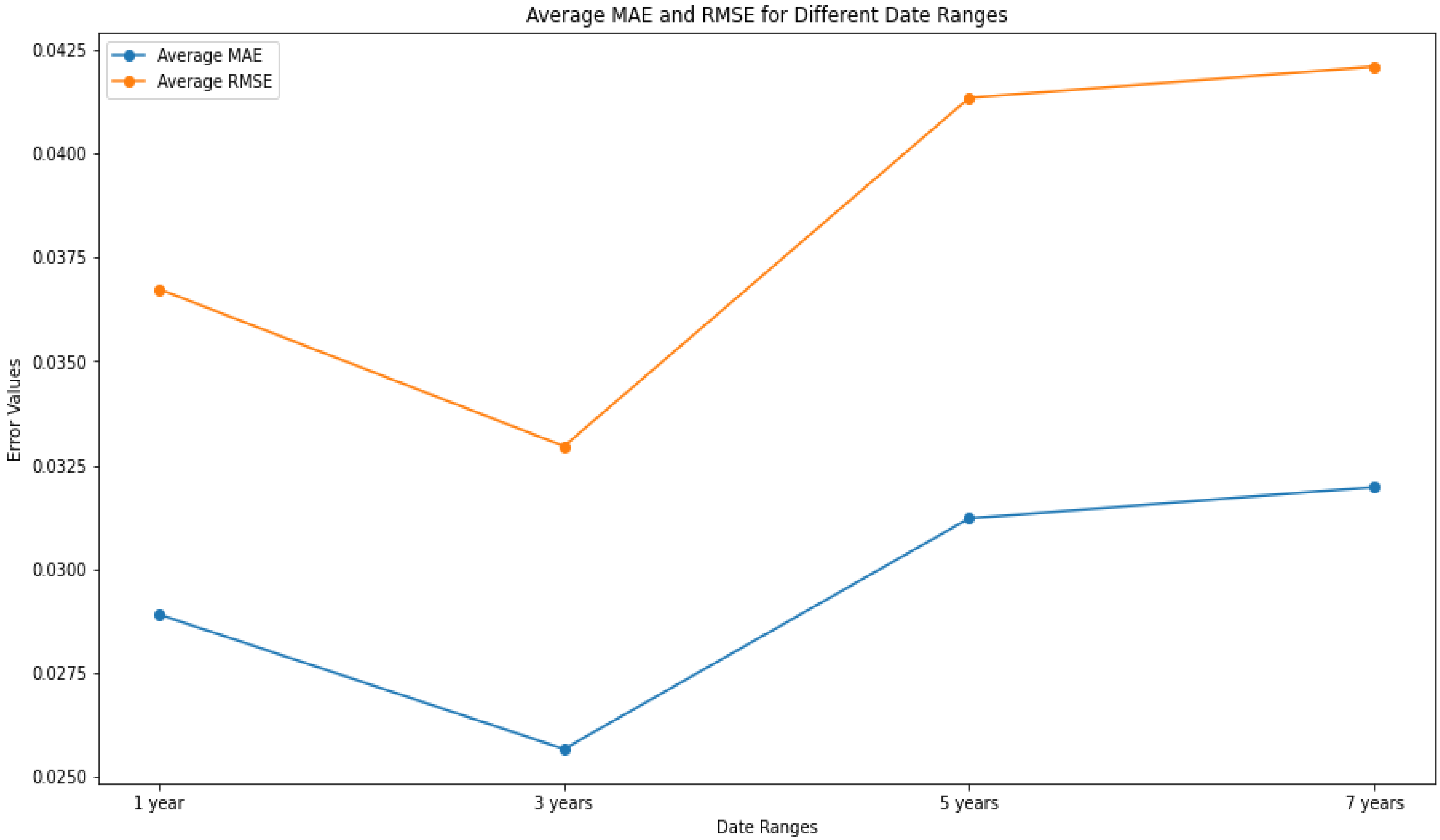
Holt Winter


Average MAE and RMSE for Different Date Ranges
Average MAE
Average RMSE
0.225
0.200
0.175
0.150
0.125
0.100
0.075
Error Values
1 year
3 years
5 years
7 years
Date Ranges

Prophet Model



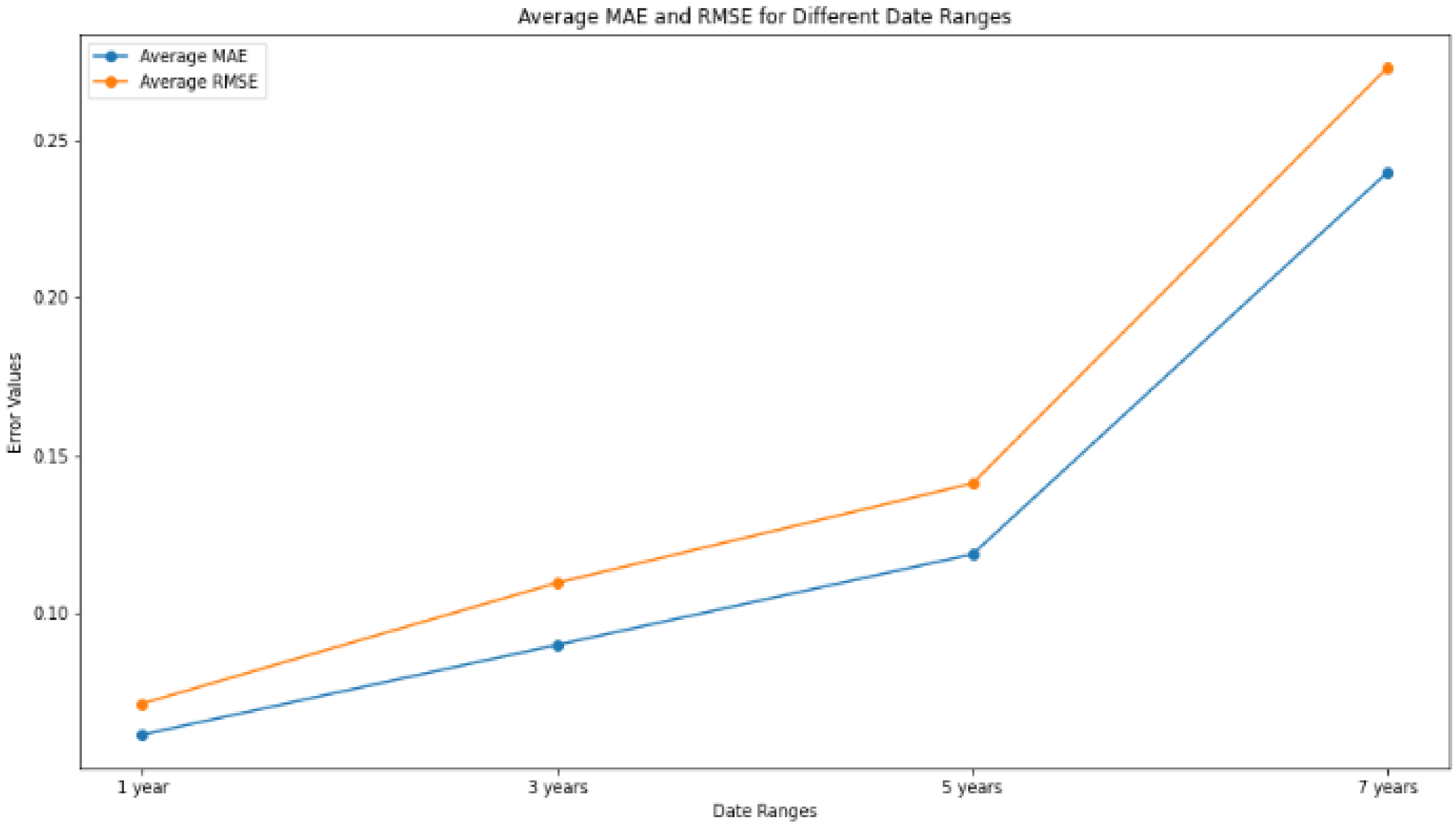
LSTM Model



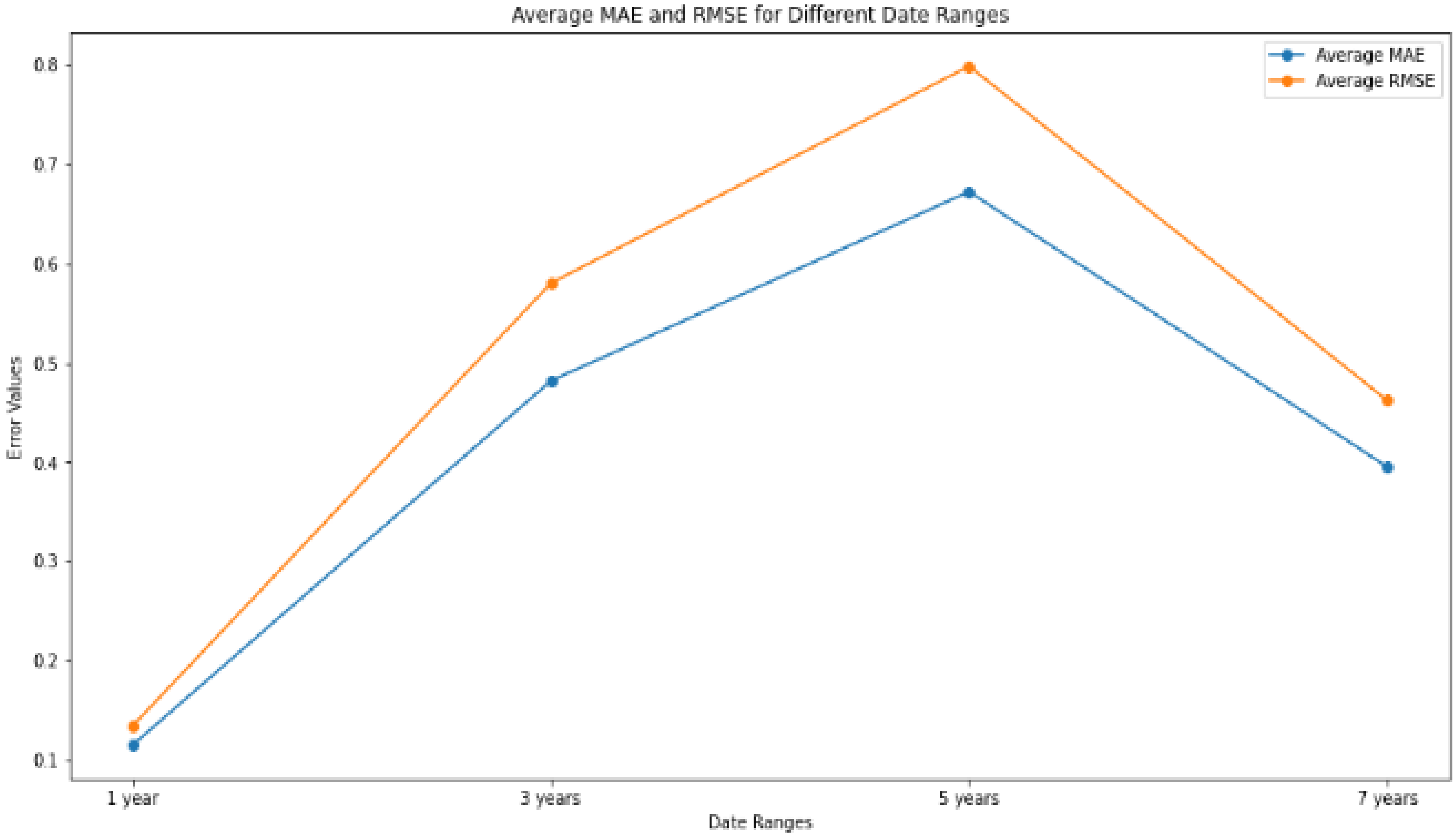
Average MAE and RMSE for Different Date Ranges
Average MAE
Average RMSE
0.0425
0.0400
0.0375
0.0350
0.0325
0.0300
0.0275
0.0250
Error Values
1 year
3 years
5 years
7 years
Date Ranges

<u>Technology Stock</u>

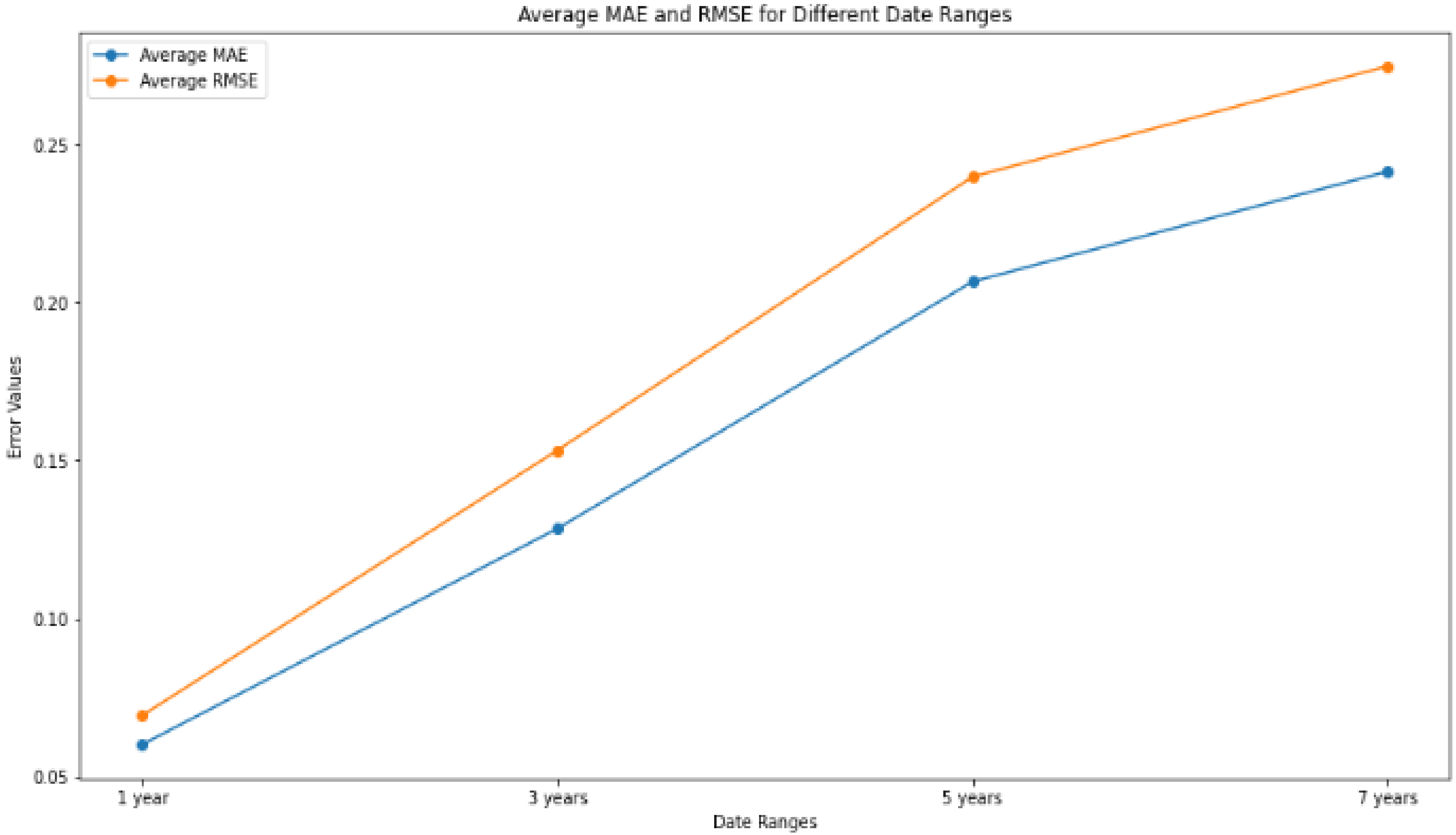
ARIMA Model



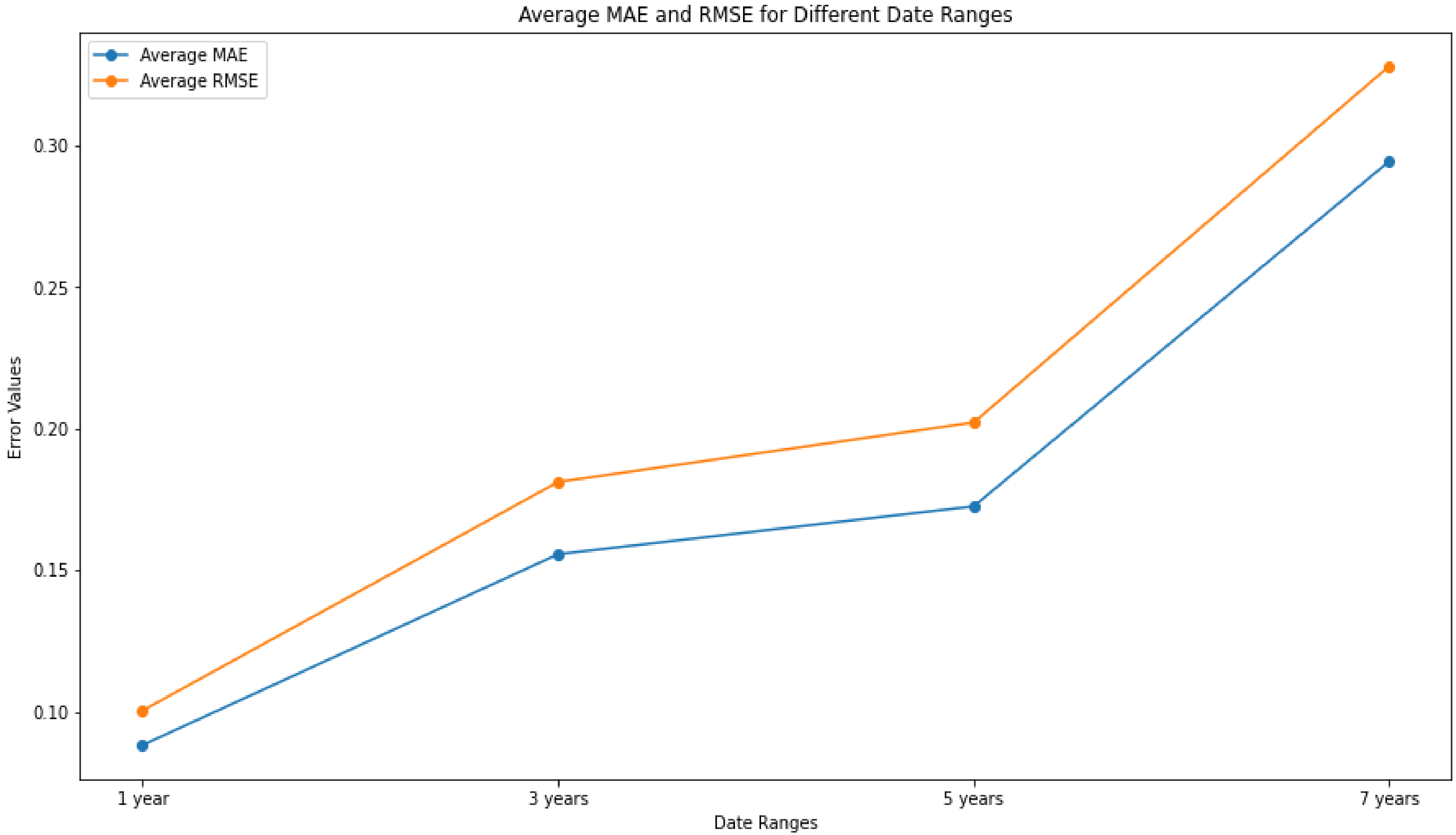
SARIMA Model



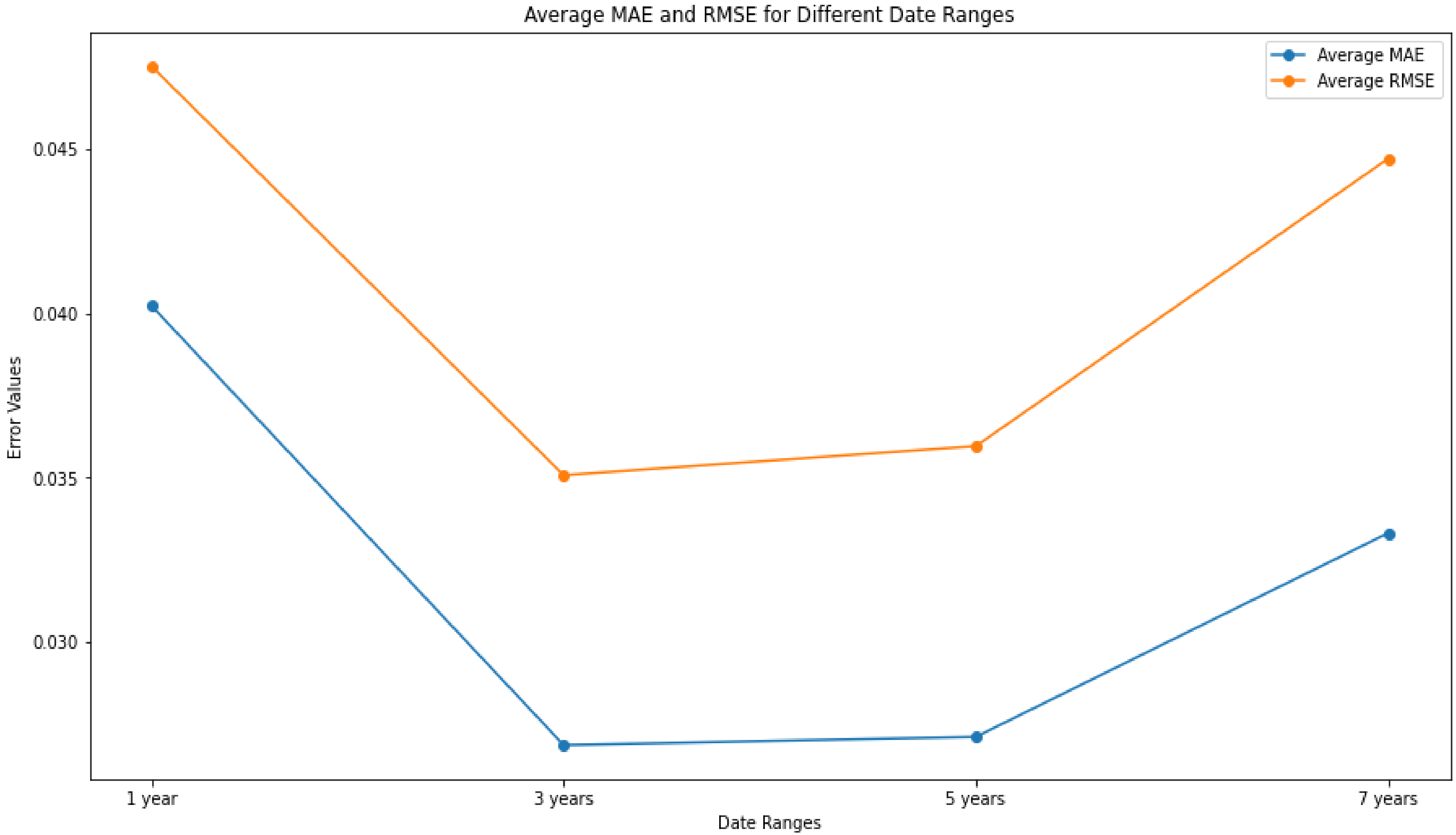
Holt Winter Model



Prophet Model



LSTM Model


Average MAE and RMSE for Different Date Ranges
Average MAE
Average RMSE
0.045
0.040
0.035
0.030
Error Values
1 year
3 years
5 years
7 years
Date Ranges