INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Revenue Forecasting for the "Magnificent Seven": Accuracy Comparison Between ANN, Prophet, and Traditional Econometric Models

Alexandre Antunes Rodrigues

Master's Degree in Monetary and Financial Economics

Advisor:
Full Professor José Dias Curto, PhD
ISCTE - University Institute of Lisbon

October, 2025

CIÊNCIAS SOCIAIS
E HUMANAS

Department of Political Economy

Revenue Forecasting for the "Magnificent Seven": Accuracy Comparison Between ANN, Prophet, and Traditional Econometric Models

Alexandre Antunes Rodrigues

Master's Degree in Monetary and Financial Economics

Advisor:
Full Professor José Dias Curto, PhD
ISCTE - University Institute of Lisbon

October, 2025

Acknowledgments

This work is one more step of my academic career and would not have been possible without the collaboration of countless people who helped me to overcome the various difficulties that arose over the years. I would therefore like to express my gratitude and appreciation to everyone who, directly or indirectly, made this possible.

My first words of thanks go to ISCTE and the entire teaching staff for the lessons they provided throughout my bachelor's and master's degrees.

Special thanks to Professor José Dias Curto, my friend and advisor, for his availability, for sharing his knowledge and assertive advice, for all his support, constant encouragement, and for the trust he has placed in me.

To my parents, whose steady encouragement, practical help, and example of hard work and humility have been my foundation.

To my sister, for her unconditional support in encouraging me every day with words of support even when she disagrees with me!

To my grandparents, for their help, affection, and kind words.

To my friends, for their enthusiasm and assistance.

Thank you all!

Resumo

No passado recente, com a expansão generalizada da inteligência artificial, tem-se verificado igualmente uma crescente utilização de métodos de machine learning para serem aplicados na previsão de receitas de empresas. A presente tese propõe uma análise comparativa entre as abordagens econométricas tradicionais e as abordagens de machine learning na previsão das receitas da Alphabet, Amazon, Apple, Meta, Microsoft, Nvidia e Tesla. O objetivo do presente estudo é, a partir de uma análise comparativa, determinar o método mais preciso. O estudo em questão recorre a uma diversidade de métodos estatísticos tradicionais, incluindo médias móveis simples, decomposição, exponencial Holt-Winters e modelos ARIMA, bem como abordagens mais sofisticadas, tais como Prophet do Facebook e redes neurais artificiais. A avaliação é realizada através da aplicação de métricas de erro padrão, tais como o erro quadrático médio (MSE), o erro quadrático médio raiz (RMSE) e o erro percentual absoluto médio (MAPE). O estudo verifica se a incorporação do efeito dos feriados no Prophet melhora a precisão do modelo e se a automatização do ajuste do modelo de aprendizagem automática, através de uma grid search de hiperparâmetros, produz um desempenho superior em comparação com a especificação manual dos parâmetros como seu tempo de relatório. Também é apresentada a arquitetura da ANN de forma transparente e são esclarecidos os ajustes que facilitam a execução e a reprodutibilidade das previsões. Os resultados obtidos indicam que, embora os modelos econométricos tradicionais forneçam bases úteis para a construção de métodos mais sofisticados, o Prophet e as redes neurais artificiais demonstram superioridade em termos de precisão de previsão.

Palavras-chave: Séries Temporais, Previsão de receitas, Métodos Econométricos Tradicionais, Machine Learning, Redes Neurais, Precisão.

Classificação JEL: C22, C53, C45, C52.

Abstract

In the recent past, with the widespread expansion of artificial intelligence, there has also been an increase in the application of machine learning methods for the purpose of forecasting company revenues. The present thesis proposes a comparative analysis of traditional econometric and machine learning approaches to forecasting revenues for Alphabet, Amazon, Apple, Meta, Microsoft, Nvidia, and Tesla. The aim of this study is to determine the most accurate method based on a comparative analysis. The study employs a range of traditional statistical methods, including simple moving averages, decomposition, Holt-Winters exponential smoothing, and ARIMA models, in addition to more sophisticated approaches such as Facebook's Prophet and Artificial Neural Networks. The performance of the models is evaluated by means of standard error metrics, including mean squared error (MSE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). The study ascertains whether the incorporation of holiday effects in Prophet enhances the accuracy of the model, and whether the automation of machine learning model tuning via hyperparameter grid search yields superior performance in comparison to the manual specification of parameters as its report time. It is also presented the architecture of the ANN in a transparent manner and elucidate the adjustments that facilitate the execution and reproducibility of forecasts. The findings indicate that while traditional econometric models furnish useful baselines for the development of more sophisticated methods, Prophet and Artificial Neural Networks consistently demonstrate superiority in terms of forecast accuracy.

Keywords: Time Series, Revenue Forecasting, Traditional Econometric Methods, Machine Learning, Neural Networks, Accuracy.

JEL Classification System: C22, C53, C45, C52.

Index

Index of Figures

Index of Tables

Index of Equations

Glossary of Abbreviations and Acronyms

ACF - Autocorrelation Function

ADF - Augmented Dickey–Fuller Unit Root Test

ANN - Artificial Neural Network

AR - Autoregressive

ARIMA - Autoregressive Integrated Moving Average

ARMA - Autoregressive Moving Average

HEGY - Hylleberg–Engle–Granger–Yoo Seasonal Unit Root Test

KPSS - Kwiatkowski–Phillips–Schmidt–Shin Stationarity Test

MA - Moving Average

MAPE - Mean Absolute Percentage Error

MSE - Mean Squared Error

NNAR - Neural Network Autoregression

PACF - Partial Autocorrelation Function

PP - Phillips–Perron Unit Root Test

RMSE - Root Mean Squared Error

SARIMA - Seasonal Autoregressive Integrated Moving Average

Introduction

In the recent past, companies have faced some significant financial vulnerabilities caused by an environment of heightened geopolitical and policy uncertainty, rising global trade tensions and the Covid-19 pandemic. A key challenge in the decision-making process pertains to the uncertainty surrounding future prospects and the associated risk. In order to overcome these challenges, it is essential for companies to concentrate on forecasts that predict the amount of money they will receive from selling their products and services, more commonly known as revenue.

The question of forecasting company revenues is a subject of critical importance in the domain of economical and financial analyses. The statistical literature is replete with a multitude of forecasting methods, techniques and models. The question is determined by a number of factors and reasons, including the required accuracy, cost and speed estimation. In this thesis, a rigorous and meticulous approach will be adopted to ensure a balanced consideration of most pertinent factors. This methodology is designed to provide reliable and accurate forecasts that meet the requisite level of accuracy and credibility.

The digital revolution, otherwise known as the third industrial revolution, took place in the second half of the 20th century marking the beginning of the digital era and the use of computers, mobile phones and the internet. Consequently, a multitude of companies were established, driven by technological innovation, research and development. Among these entities, some have distinguished themselves through their market dominance and financial performance, a distinction that has been recognised by investors over time. The companies that stand out the most from this era are: Alphabet, Amazon, Apple, Meta, Microsoft, Nvidia and Tesla.

The primary focus of this analysis is the quarterly revenue of those companies, given its recognised importance. As indicated by the financial media, the analysis of companies' performance focuses on several metrics, including revenue. This is due to the fact that revenue constitutes an essential first step in the valuation of any publicly traded company, as well as in the assessment of the company's financial health and future performance. Also gives the composition of the company's revenue, indicating its market reach and operational efficiency

(Padmanabhan & Nirmala, 2023; Cheng et al*.,* 2020; Whitfield & Duffy, 2013; Trueman et al*.,* 2001).

The revenue forecasting of the companies mentioned, will be conducted using the most common traditional econometric methods, such as Simple Moving Average, Decomposition Method, Exponential Holt-Winters Smoothing, Autoregressive Integrated Moving Average (ARIMA), and the machine learning methods Prophet (Facebook) and Artificial Neural Networks (ANN). The forecast is based exclusively on the historical behaviour of the variable. This methodology has previously been considered to provide a feasible estimate by Kożuch, et al. (2023), who compared ARIMA, Prophet and ANN; by Taylor & Letham (2018), who focused only on Prophet; by Makridakis & Hibon (2000), who compared simple moving average, decomposition, Holt-Winters, ARIMA and ANN and by Kuo & Xue (1999), who focused only on ANN.

The objective of this thesis is to investigate, through a comparative analysis of the methods employed, the efficacy and the respective forecast error and to highlight the benefits generated by an accurate forecast. The study also seeks to identify what are the improvements when using machine learning such as Prophet and ANN. It has been hypothesised that the most accurate methods for this type of forecast are ARIMA, Prophet and ANN. The prevailing opinion was that the ANN would have the best performance, a hypothesis that was confirmed in the empirical results and proved to be the one with the greatest predictive capacity.

Since no method can perfectly predict the future, forecasting methods provide companies and decision-makers with tools that help to mitigate risk in decision making by providing the most accurate estimates of the values of the quantities under consideration in a reasonable amount of time. This thesis provides a comprehensive explanation of the working principles of artificial neural networks, utilising solely the lags inherent within the series itself. The selection of lags is conducted automatically through the grid search of hyperparameters. Furthermore, a visualisation of a structure is presented, accompanied by the weight values. A study was conducted to forecast the revenue of the most dominant US companies comparing not only traditional methods but also with new machine learning methods, such as Prophet, which was launched in recent years, were the grid search was also applied. In the final analysis, the ANN demonstrated a superior performance in comparison to both traditional methods and Prophet.

The structure of the dissertation is organized into three chapters. The initial chapter is dedicated to a review of the related literature, in which an overview of the theoretical framework of the topic under discussion is provided, in addition to comparisons of a number of methods used in time series forecast. The second chapter provides a detailed exposition of the methodological framework employed in the study and the database that was utilised. In the third chapter, the empirical results and their analysis are presented. The conclusions of the study are presented in line with the analysis carried out and the theoretical framework of the work. The limitations of the study are then presented, with the aim of improving the study in future research.

1 - Literature Review

The literature on revenue forecasting encompasses a range of methodological approaches, each emphasizing the role of this metric in supporting business operations across different domains. This section reviews prior research in order to provide a deeper insight into its significance.

Chan (2024) and Hürtgen, et al. (2020) state that revenue forecasts shape the way a company thinks about its future and the business decisions it makes, as they are the critical first assumptions in setting a company's budget. The assumptions that shape a company's short and long-term goals and play an important role in preparing each organisation for the future and drive company decisions about investment and spending, investor perceptions of the company, and even the ability to attract talent. Being the primary objective of businesses, in the corporate world, to maximise sales and, by extension profit, in such a competitive and constantly fluctuating environment, the key to enhancing an organisation's commercial competitive advantage lies in its management relying on sophisticated information systems to make the right decision in time, depending on the information at hand (Thomassey, 2010; Kuo & Xue, 1999). This makes Chan (2024), Nguyen, et al. (2021), Ramos, et al. (2015), Whitfield & Duffy (2013), Chu & Zhang (2003) and Kuo & Xue (1999) place a particular emphasis on the importance of effective decision-making within a business context. They state that companies and decision-makers must be capable of making accurate, reliable and quick decisions based on forecasts, as this facilitates supply chain management, planning and controlling.

Nguyen, et al. (2021) and Ramos, et al. (2015) develop the microeconomic and organisational perspective, stating that by building an accurate forecast methodology, companies can improve their performance at various stages of the supply chain, such as due date management, financial planning, production planning and inventory management, evaluating the business's status, balancing demand and supply, identifying opportunities and risks, mapping out the best strategies for success and anomaly detection.

Ramos, et al. (2015) argues that accurate sales forecasts have the potential to increase retailers' profitability by improving the efficiency of chain operations and minimising waste. In addition, accurate retail sales forecasts can improve the ability of portfolio investors to predict share price movements.

Whitfield & Duffy (2013) brings with it a commitment to increasingly accurate forecasting to avoid the consequences of under-forecasting, including resource commitment, quality and service risk, while over-forecasting can be equally damaging in terms of excess resources, costs and eroded margins. Ramos, et al. (2015) points out that poor forecasts tend to lead to over- or under-inventory, which has a direct impact on the company's profitability and competitive position.

In the context of recent economic uncertainty and market volatility, it is imperative for companies to invest in advanced analytics, automation and machine learning algorithms. This permits the production, analysis and forecast of real time series data that reflects effective responses to constantly changing market conditions (Hürtgen, 2020; Siami-Namini et al., 2018). Despite the associated costs, which include the necessity of experts in the field, a substantial amount of time for pre-processing the data and training the model, and computational power, the potential benefits of such an approach are significant when compared with manual, intuition-driven, non-automated practices (Hürtgen, 2020; Alon et al., 2001).

In the recent years, many studies have been conducted by academics seeking to establish a consensus on the most accurate method for revenue forecasting. Pundir, et al. (2020) conducted a comparative study using 4-period Moving Average, Exponential Smoothing, ARIMA, Vector Autoregression (VAR) and Random Forest, evaluated via Mean Absolute Percentage Error (MAPE), to predict the revenue of a retail chain business. The study was also based on historical sales values and several other external factors, including economic performance, the consumer price index, the unemployment rate, oil prices, ambient temperature and price markdowns. They found that the VAR method performed best, followed by the Random Forest, ARIMA, 4-period Moving Average and Exponential Smoothing, with and MAPE of 2.58%, 4.36%, 6.49%, 7.01% and 8.68% respectively. It is emphasised that multivariate time series machine learning methods can increase the forecast accuracy given the importance of internal and external variables, as they are capable of capturing patterns in the data, comparing to traditional time series methods and univariate time series regression methods.

Ramos, et al. (2015) investigated the forecasting performance of Error-Trend-Seasonal (ETS) state-of-space models in a comparative analysis with seasonal ARIMA models. The data set was employed in a case study of retail sales of five distinct categories of women's footwear from the portuguese retailer Foreva. Both one-step and multiple-step forecasts were produced. The model that demonstrated the lowest value of Akaike's information criteria for the in-sample period was selected for further evaluation via RMSE, MAE and MAPE in the out-of-sample period. The results demonstrate that the ETS and ARIMA models achieve quite similar overall out-of-sample forecasting performance for both one-step and multi-step forecasts, with multi-step forecasts performing slightly better than one-step forecasts, which is not surprising since multi-step forecasts incorporate more recent information.

The increasing availability of data and the advancement of computational power have resulted in a rise in the utilisation of machine learning techniques, such as Prophet and ANN, among others. These techniques are frequently employed in comparison with or in conjunction with classical methods such as Coimbra, et al. (2024); Suryawan, et al. (2024); Kożuch, et al. (2023); and Siami-Namini, et al. (2018); Gong, et al. (2018); Lin, et al. (2013); Wong & Guo, (2010); Zhang & Qi, (2005); Chu & Zhang, (2003); Alon, et al. (2001); by Makridakis & Hibon, (2000) and Kuo & Xue, (1999) being examples of it. This tendency is indicative of a pursuit for more adaptable and robust methodologies, which are able to capture nonlinear patterns that are frequently by conventional statistical techniques.

Motivated by the promising performance in the field of modelling and pattern recognition of the ANN and the Fuzzy Neural Networks (FNN), Kuo & Xue (1999) carried out a study in which they forecast the sales of a supermarket chain company applying the ANN and the Autoregressive Moving Average (ARMA) to the data. It was proposed a system where they integrated the ANN and fuzzy logic in one (Integration ANN), where they use the ANN to predict the general pattern of sales and the FNN to predict the promotion effect. They argue that the application of conventional methods in sales forecasting is a complex process, due to the influence of both internal and external factors. That these methods are only efficient for data which are seasonal or cyclical. In scenarios where the sales data are influenced by a sudden change in the pattern, such as a promotional campaign, they are not feasible. The Integration ANN outperforms other forecasting methods because it prioritises the promotion effect on the sales pattern.

Kożuch, et al. (2023) compare feed-forward artificial neural networks with Facebook's Prophet and with traditional methods such as ARIMA, ETS, BATS and TBATS. The aim was to evaluate the suitability of the methods for forecasting timber prices in Poland. The study material comprised quarterly time series of net nominal prices of roundwood for the years 2005 to 2021. The evaluation was utilising the MAE and RMSE. The findings revealed that the ANN model exhibited superior performance in forecasting price changes and levels when compared to other models. ANN models demonstrated a superior fit to minimum and maximum values in comparison to classical models, which exhibited a propensity to smooth price trends and generate forecasts that were biased towards average values. The Prophet resulted in the most inferior quality of projections. It was also observed that traditional time series methods, such as ARIMA and ETS, which have been developed with the intention of addressing single seasonality, have been found to be ineffective in the handling of multiple seasonality effects. The distinctive properties of ANN models, including their adaptive capacity and nonlinear characteristics, make them highly effective for predicting a wide range of phenomena, including economic ones and that they replace programming with a learning process, due to which they can rapidly and accurately adapt to empirical data.

Furthermore, Alon, et al. (2001) also compares the traditional methods of Winters exponential smoothing, Box-Jenkins ARIMA model, multivariate regression with Artificial Neural Networks applied to US aggregate retail sales. The robustness of the forecasting methods was tested in two different periods, each one characterised by different economic conditions and evaluated via MAPE. The first marked with supply push inflation, high interest rates, and high unemployment. The second, characterized by less macroeconomic instability. Their findings indicate that, in volatile economic environments, ANN typically deliver more accurate forecasts compared to the traditional methods, although the ARIMA model remains a strong contender. In relatively stable macroeconomic conditions, both the ARIMA and the Winters exponential smoothing methods demonstrated viable performances, though the ANN consistently outperformed them. They also pointed out that the ANN are difficult to compute, as they require special software, a lot of computing time and a great deal of in-house expertise to understand them.

Suryawan, et al. (2024) utilises a single univariate series of daily white-bread sales from a bakery business in Indonesia to evaluate the performance of ARIMA, Long Short-Term Memory (LSTM), and Prophet. The data, consisting of 822 observations from March 2021 to May 2023, has been divided into a training set and a testing set in an 80/20 ratio. The performance of the models is assessed using the MAPE, MSE, and RMSE. The ARIMA (1,0,2) model demonstrated the highest level of point-accuracy, with a MAPE of 4.548%. The optimal configuration of LSTM attained a MAPE of 7.3275%. The performance of Prophet was found to be sub-optimal in comparison to both methods, with a mean absolute percentage error MAPE of 7.402%. In the context of the study, the choice of LSTM and Prophet was determined by their ability to process complex time-series data and produce accurate predictions, and it was hypothesised that, given the choice, LSTM and Prophet would match or surpass ARIMA. However, the data was found to be short, univariate and relatively regular, whereas LSTM/Prophet used relatively basic configurations with limited hyperparameter exploration and no exogenous signals. These conditions were found to make the conclusion that a well-specified ARIMA outperforms the tested LSTM and Prophet.

Research on revenue forecasting has introduced a wide range of methodological approaches, reflecting the relevance of this metric for business operations across different domains. For instance, Chu & Zhang (2003) study compared the accuracy of various linear and nonlinear models in forecasting aggregate monthly retail sales data compiled by the US Bureau of the Census, evaluated via RMSE, MAE and MAPE. The linear models that were examined included the ARIMA model, regression with dummy variables, and regression with trigonometric variables. The non-linear models that were investigated included artificial neural networks. The study investigated if seasonal dummy variables and trigonometric variables were beneficial in forecasting seasonal time series with both linear and nonlinear models, and also if seasonal adjustment was advantageous in improving the forecasting accuracy of neural networks. The findings indicated that nonlinear models exhibited superior out-of-sample forecasting capabilities in comparison to their linear counterparts. It was also observed that prior seasonal adjustment of the data led to a substantial improvement in the forecasting performance of the neural network model. Conversely, the dummy regression models demonstrated a consistent tendency to underestimate forecasts, while the forecasts derived from trigonometric models lacked stability. Moreover, trigonometric models were found to be ineffective in capturing the seasonal pattern in out-of-sample data.

Padmanabhan & Nirmala (2023) proposes a global revenue forecasting panel of 2000 large companies listed by Forbes, covering the period from 2010 to 2022. The study employed a set of regression models incorporating Ordinary Least Squares, Artificial Neural Networks, and Bayesian Neural Networks. The selection of the methods was made with the specific purpose of elucidating non-linear relationships. Within each method are further variants, such as the Auto regressive (AR) model, in which the previous year's financial outcome is used exclusively as an input variable; the Auto regressive model with exogenous inputs (ARX), in which the previous year's financial outcome, along with a group of additional variables, is employed as an input variable; and the Non-Auto regressive model (REG), in which exogenous inputs are derived exclusively from the group of variables from the previous year. The study demonstrates that incorporating exogenous variables (ARX) enhances average MAPE by 13.5% in comparison with a pure AR. Furthermore, on the global average MAPE, OLS-ARX emerges as the most effective model with a MAPE of 0.284, surpassing ARX-ANN with 0.291 and ARX-BNN with 0.315. It was hypothesised that, given that the OLS models only fit linear relationships between financial outcomes and input variables, the neural network models would demonstrate greater accuracy, due to the ability of neural networks to identify nonlinear and segmented relationships, and to store them as weights and biases. However, this was not observed.

Wong & Guo (2010) proposed a different type of model, a hybrid intelligent (HI) model. This model integrates an improved harmony search algorithm and an extreme learning machine (ELM) with a purpose to improve the network generalisation performance. In order to evaluate the performance of the proposed model via RMSE, MAPE and Mean Absolute Scaled Error (MASE), a series of experiments were conducted. The medium-term sales in fashion retail supply chains of four cities and four fashion item categories were the subject of monthly, quarterly and annual forecasts, respectively. The experimental results demonstrate that the performance of the proposed model is significantly superior to that of traditional ARIMA models and two recently developed neural network models, such as Elman Neural Network (ENN) and ELM. It was also demonstrated that the forecasting performance is contingent upon the accuracy measure employed. For instance, the proposed model generates minimal MAPE but almost maximal RMSE and MASE when forecasting the category on a monthly basis. Nevertheless, the proposed HI model demonstrates the optimal forecasting performance for 14 out of 16 forecasting cases when the RMSE is utilised as the accuracy measure. It is also concluded that data pre-processing is helpful to improve forecasting performance and that the proposed model is broadly applicable to time series with irregularity and seasonality.

Lin, et al. (2013) developed a fuzzy least-squares support vector regression model with genetic algorithms (FLSSVRGA) to forecast the online monthly revenue of MediaTek Corporation in Taiwan. The study was conducted in an environment characterised by uncertain economic factors and government policies. The proposed FLSSVRGA model is a rolling forecasting model, with data updated on a monthly basis. The purpose of the model is to predict revenue for the coming month. Four additional forecasting models are also employed and evaluated via MAPE and RMSE: the seasonal autoregressive integrated moving average

(SARIMA), the generalized regression neural networks (GRNN), the support vector regression with genetic algorithms (SVRGA), and the least-squares support vector regression with genetic algorithms (LSSVRGA). The empirical results demonstrate that the FLSSVRGA model exhibits superior forecasting accuracy in comparison to the four other models. Furthermore, the GRNN, SVRGA and SARIMA models could not effectively capture the trend of the data. The superiority of the proposed FLSSVRGA model is primarily attributable to its capacity to address monthly variance through the utilisation of membership functions.

Siami-Namini, et al. (2018) conducted an empirical investigation into the performance of deep learning-based algorithms, such as LSTM, and traditional algorithms, including the ARIMA model. This research was motivated by concerns regarding high market volatility in the recent years. In order to ensure the robustness of the study, a range of monthly data from diverse economic sectors was considered, including financial data, medical data, housing data and transport data. The data was segmented into two subsets: a training dataset and a test dataset. The training dataset was allocated 70% of the data, while the test dataset was allocated the remaining 30%. The objective of the test data set was to evaluate the accuracy of the models via RMSE. The findings demonstrated that the LSTM model exhibited superior performance in comparison to the ARIMA model, with an average reduction in error rates of between 84 and 87 per cent. One reason for this is the assumption of the ARIMA models that there is a constant standard deviation in errors. However, this may not be upheld in practical applications.

To summarise, a number of studies have been conducted over the last few decades in order to determine the most accurate method. A variety of methods have been employed in this field, including Moving Average, Holt-Winters Exponential Smoothing, ARIMA, Multiple Regression, VAR, Random Forest, Artificial Neural Networks, Fuzzy Neural Networks, ETS state of space, SVG, ELM and LSTM. The wide adoption of these methods is due to their proven ability to model trend and seasonal fluctuations. However, all these methods have shown difficulties and limitations, necessitating further investigation into how to improve the quality of forecasts. According to the reviewed articles, machine learning methods exhibited superior predictive capacity compared to traditional ones in Kożuch, et al. (2023), Pundir, et al. (2020), Siami-Namini, et al. (2018), Lin, et al. (2013), Wong & Guo (2010), Chu & Zhang (2003), Alon, et al. (2001), and Kuo & Xue (1999). Ramos, et al. (2015) indicated that the two groups were equivalent. And as demonstrated in the works of Suryawan, et al. (2024) and Padmanabhan & Nirmala (2023).

The following findings are of particular significance and will be investigated in the present study: (1) The application of methodologies in revenue forecasting is a complex process, due to the influence of both internal and external factors. It is evident that the efficacy of these methodologies is contingent upon the presence of seasonal or cyclical data. In circumstances where data is subject to sudden fluctuations in patterns, such as the implementation of a promotional campaign or holidays, machine learning methods and non-linear models have been shown to outperform conventional statistical approaches. (2) In volatile economic environments, machine learning algorithms has been shown to produce more accurate forecasts than traditional methods, although the ARIMA model remains a strong contender. In relatively stable macroeconomic conditions, both the ARIMA and the Holt-Winters Exponential Smoothing methods demonstrated viable performances, though the machine learning algorithms consistently outperformed them. (3) Although machine learning methods in general have greater predictive capacity, they have a much slower estimation speed, which requires greater computing power. In this sense, they will be compared according to these factors to determine which is best for the purpose of revenue forecasting.

The objective of this study is to estimate and compare traditional and modern methods for forecasting the quarterly revenue of the following US companies: Alphabet, Amazon, Apple, Meta, Microsoft, Nvidia, and Tesla. The present study is supported by a comprehensive literature review, which has led to the identification of particular methods, with emphasis on ARIMA, Prophet, and ANN. The subsequent investigation will ascertain whether the incorporation of holiday effects in Prophet enhances the accuracy of the model, and whether the automation of machine learning model tuning via hyperparameter grid search yields superior performance in comparison to the manual specification of parameters as its report time. We present the architecture of the ANN in a transparent manner and elucidate the adjustments that facilitate the execution and reproducibility of forecasts.

2 - Methodology and Data

Time series forecasting methods are arguably the most widely employed techniques for predicting revenue data. These statistical approaches include several well-known methods, such as the Moving Average, Decomposing, Holt-Winters, Autoregressive Integrated Moving Average (ARIMA), Prophet and Artificial Neural Networks. It should be noted that the efficiency of these methods is contingent on various factors, including the field of application, the forecasting objective, the time horizon, and the user's experience and preferences (Armstrong, 2001). Nonetheless, these methods have been implemented in diverse domains and have produced satisfactory outcomes (Kożuch et al*.,* 2023; Makridakis & Hibon, 2000; Kuo & Xue, 1999)

The objective of this research is to conduct a comparative analysis of traditional time series methods and machine learning algorithms. The primary objective is to identify differences between these methods and to conclude which one is more appropriate to revenue forecast. In order to achieve the outlined objective, several tests will be conducted. A 5% significance level ($\alpha = 0.05$) is used by default. The forecasting methods will be based exclusively on the historical behaviour of the variable.

However, it should be noted that the statistical significance observed in this research may not necessarily imply economic significance, as more accurate forecasts do not always lead to more accurate realized values.

2.1 - Database

The dataset employed in this thesis consists of revenue the of Alphabet, Amazon, Apple, Microsoft and Nvidia. The data has been obtained from Bloomberg. It should be noted that all the time series are quarterly and in millions of US dollars.

Table 1 presents the time span and size of the revenue datasets for the companies included in this study. For each company, the table reports the starting date, ending date, and the total number of observations.

Table 1 - Revenue Database

	Revenue Database		
	Starting Date	Ending Date	Total Number of Observations
Alphabet	Q4 of 2003	Q1 of 2025	86
Amazon	Q1 of 1997	Q1 of 2025	113
Apple	Q2 of 1995	Q1 of 2025	120
Meta	Q1 of 2010	Q1 of 2025	61
Microsoft	Q2 of 1995	Q1 of 2025	120
Nvidia	Q1 of 1999	Q2 of 2025	106
Tesla	Q4 of 2009	Q1 of 2025	62

The time series will be split into two subsets: a training set and a test set; 80% of the data will be used for training, while the remaining 20% will be reserved for testing and comparing the accuracy of the models. Subsequently, the most suitable model will be selected for forecasting purposes.

2.2 - Methodology

This section provides a detailed description of the various methods and evaluation criteria employed in the study.

2.2.1 - Simple Moving Average

The simple moving average method is a smoothing technique that is used in time series analysis which the primary objective is to mitigate the short-term volatility engendered by random fluctuations, thereby facilitating enhanced identification of structural patterns, such as trend and seasonality. The simplicity and non-parametric nature of the method make it a useful tool both in exploratory analysis and as a preliminary step in the construction of more sophisticated econometric models.

The moving average for a specified period t (mm_t), is defined as the simple arithmetic mean of m adjacent observations, with each observation being centred in the period t. The term m is the moving average period and represents the number of observations to be included in the respective calculation. The moving average period is conventionally considered to be equivalent to the seasonal cycle which in our case, in the context of quarterly data, $m = 4$.

The value of the moving average must correspond to the most central observation in the group of observations involved in the respective calculation. Consequently, there will be no values for either the first two quarters or the last two quarters of the time series, as there are no two observations on either side.

$$mm'_t = \frac{y_t + y_{t+1} + y_{t+2} + y_{t+3}}{4} \quad (1)$$

However, given the period of the moving average is even ($m = 4$), since the central value lies between two more central observations and the value of the simple moving average does not correspond directly to an observation, it is necessary to calculate the centred moving average, equation (2). The centred moving average is defined as the simple arithmetic mean of each pair of observations in the original moving average.

$$mm_t = \frac{mm'_{t-0,5} + mm'_{t+0,5}}{2} \quad (2)$$

In order to evaluate the forecast, the last value of the centred moving average will be used as a constant forecast for the test set.

2.2.2 - Decomposition

Decomposition plays a fundamental role in the analysis of time series, with the aim of isolating and analysing separately the effects of each component that make up and contribute to explaining the evolution of the series over time. The decomposition of a time series is based on the premise that the observed values can be expressed as a combination of four main components: Trend (t_t), Seasonal (s_t), Cyclical (c_t), and Irregular (ε_t):

$$y_t = f(t_t, c_t, s_t, \varepsilon_t), \qquad t = 1, 2, \ldots, T \tag{3}$$

where T represents the total number of observations and y_t is the value of the series in period *t*.

It is therefore hypothesised that the variations in a time series are the result of a set of effects, and the decomposition aims to isolate the contribution of each component. The forecasting method based on this type of analysis is divided into two stages. Firstly, each of the components is identified and isolated, and the respective forecast is made. Then, the individual forecasts are combined to obtain the overall forecast (Armstrong, 2001).

The manner in which the components are combined is contingent upon the decomposition method that is adopted, which can be either additive or multiplicative. In the additive decomposition method, the various components are added together:

$$y_t = t_t + c_t + s_t + \varepsilon_t \tag{4}$$

The multiplicative decomposition method assumes that the time series is the result of the product between the components:

$$y_t = t_t \times c_t \times s_t \times \varepsilon_t \tag{5}$$

When the amplitude of the seasonal variations around the trend is relatively constant, the additive method is the most appropriate, and it can be concluded that seasonal influences are not related to the level of the series. The multiplicative method is justified if the amplitude of the variations increases or decreases progressively over time. Armstrong (2001) also states that the multiplicative decomposition should be used when uncertainty is high and avoided when uncertainty is low. The revenue of all the companies under study behaves as described in the multiplicative approach.

The first step in decomposing a time series is to extract the trend-cyclical component using a smoothing method, also known as filtering. This is achieved through the implementation of a simple moving average method, which serves to mitigate the effects of random fluctuations in the data.

The second step is to remove this component. To do this, the original series is divided by the values obtained in the previous step.

$$t_t \times c_t = x_t \tag{6}$$

$$\frac{y_t}{\hat{x}_t} = s_t \times \varepsilon_t \tag{7}$$

The seasonal component is then estimated based on the original scale values of the series excluding the trend-cyclical component. Given that the data is collected on a quarterly basis, it is necessary to estimate a seasonal coefficient for each quarter. In order to estimate the seasonal coefficient for the first quarter, the average of all the values in the series is calculated, with the trend-cyclical component for the first quarter of each year being excluded. This process is then repeated for the subsequent quarters. The seasonal coefficients are generally subjected to correction so that the mean is equivalent to 1. With regard to the interpretation of the estimated seasonal coefficients, a value below 1 indicates that, on average, the value of y_t is below the trend-cycle series.

Due to the fact that some of the revenue of the companies studied since around 2010 have grown exponentially, the logarithm was applied using linear regression over time. This adjustment was implemented to ensure the trend-cyclical component exhibited exponential behaviour in the forecasting framework, when it was appropriate and when it improved the forecast, otherwise it was not applied. In order to forecast the seasonal component, it is assumed that seasonality is fixed and periodic. Furthermore, the vector of seasonal coefficients is reused as many times as is necessary to cover h forecasting quarters.

The final step in this process is to estimate the irregular component:

$$\hat{\varepsilon}_t = \frac{y_t}{\hat{x}_t \times \hat{s}_t} \quad (8)$$

The final forecasts were generated by multiplying the forecasts of the seasonal component with the trend-cyclical component. Fixed seasonality is assumed so the seasonal coefficients are recycled over the forecast horizon, and the exponential was computed to the trend-cyclical to convert back the trend to level form.

2.2.3 - Holt-Winters Exponential Smoothing

The Exponential Holt–Winters method, Holt (1957, 2004) and Winters (1960), is a statistical forecasting method that is employed in time series univariate methods. The application of this method is appropriate in circumstances where the data exhibit trend and seasonal patterns, or random fluctuations, making it versatile for different forecasting scenarios. Exponential smoothing forecasting methods base predictions on a weighted sum of past observations. These methods assign greater importance to more recent observations, as they give more weight to them, with the importance decreasing exponentially as the observations become more distant in time. This gives them the advantage of being able to respond efficiently to sudden changes or shifts in the data, making them particularly useful for short-term forecasting or when there are quick fluctuations in the time series.

The Holt-Winters method is a technique that offers two distinct approaches, like the decomposition method, depending on the seasonal component: the additive and multiplicative approaches. The additive approach is employed when seasonal variations maintain a relatively constant amplitude throughout the duration of the dataset. The multiplicative approach is employed when seasonal patterns increase proportionately through the overall level of the series, such as those currently under study. The equations for the multiplicative approach are as follow:

Level $$l_t = \alpha\left(\frac{y_t}{s_{t-m}}\right) + (1-\alpha)(l_{t-1} + b_{t-1}) \quad (9)$$

Trend $$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \quad (10)$$

Seasonality $$s_t = \gamma \left(\frac{y_t}{l_t} \right) + (1 - \gamma) s_{t-m} \quad (11)$$

Forecast $$\hat{y}_{t+h} = (l_t + h \times b_t) \times s_{t+h-m(k+1)} \quad (12)$$

The parameters for the Holt-Winters equations are as follows: y_t is the actual value in period t; α is the smoothing constant and determines the weight given to the most recent past observations and therefore controls the rate of smoothing (it ranges between 0 and 1). When the value is close to 1, it indicates a greater reliance on recent past values. Conversely, when the value is close to 0, it signifies a greater focus on past observations; s_t is the smoothed value of the seasonal component at period *t* after adjusting for seasonality; β is the smoothing constant used for the trend effects b_t, ranging between 0 and 1; γ is the smoothing constant used for the effects of seasonality, ranging between 0 and 1; b_t is the smoothed value of the trend for period *t*; m is the length of the seasonal cycle; $k = \left[\frac{h-1}{m}\right]$ number of complete seasonal cycles before $t + h$.

2.2.4 - ARIMA

The Autoregressive Integrated Moving Average (ARIMA) is widely regarded as one of the most popular and significant linear models in the field of time series forecasting over the past three decades. The ARIMA model is renowned for its versatility and statistical properties, as well as the well-known Box–Jenkins methodology.

The ARIMA model is a generalized model of ARMA that combines Autoregressive (AR) and Moving Average (MA) processes. It can be represented as ARIMA (p, d, q). The Autoregressive (AR) process is defined as the dependencies between an observation and a number of lagged observations (p). The Moving Average (MA) process is defined as the dependency between observations and the error terms (q). The series is said to be integrated of order I(d) when it converts a non-stationary time series into a stationary one by differencing. In this case, d denotes the number of times the differences of the series need to be made to achieve stationarity. This step enables the model to deal with non-stationary time series.

The model in question has the capacity to represent different types of time series. These include, but are not limited to, pure autoregressive (AR), pure moving average (MA), and mixed AR and MA processes. If q = 0, then it becomes a pure autoregressive (AR) model of order p. When p = 0, the model reduces to a pure moving average (MA) model of order q. One central task of the ARIMA model building is to is selecting an appropriate model order (p, d, q).

The coefficients ϕ_1, …, ϕ_p are fixed parameters that tell us how y_t is related to the past values y_{t-1}, y_{t-2}, y_{t-p}. The coefficients θ_1, …, θ_q are parameters that represents the relationship between y_t and lagged random shocks.

$$AR(p) = y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \quad (13)$$

$$MA(q) = y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (14)$$

$$ARMA(p,q) = c + \phi_1 y_{t-1} + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_q \varepsilon_{t-q} \quad (15)$$

Given that the time series data under consideration is seasonal in nature, it is reasonable to hypothesise that short-run non-seasonal components may contribute to the model. It is therefore necessary to estimate a seasonal ARIMA model, incorporating both non-seasonal and seasonal components in a multiplicative framework. The general form of a seasonal ARIMA model is denoted as SARIMA (p, d, q) × (P, D, Q), where p is the non-seasonal AR order, d is the non-seasonal differencing, q is the non-seasonal MA order, P is the seasonal AR order, D is the seasonal differencing, and Q is the seasonal MA order.

The theory of ARIMA models has been developed by numerous researchers, with its extensive application being attributable to the contributions of Box, et al. (1976, 2015) who developed a systematic and practical model-building method. The efficacy of the Box–Jenkins methodology as a practical time series modelling approach has been demonstrated by an iterative three-step model-building process. This process involves model identification, parameter estimation and model diagnosis. After these steps, the `best' model is used to forecast.

The initial step in estimating a seasonal ARIMA model is the identification of an appropriate model, by finding the values of (p, d, q) and (P, D, Q). The following steps are employed in order to identify a model:

The time series is plotted. As the variance grows with time, a logarithmic transformation is applied. Unit root tests are applied to the original time series to draw conclusions about its stationarity, according to the test statistics and the critical values associated with them. In this study we apply the ADF, PP, KPSS and HEGY tests. The Augmented Dickey-Fuller (ADF) test (Fuller, 1995) and the Phillips-Perron (PP) test (Perron, 1988) are tests used to detect the presence of a unit root in the time series. The presence of a unit root indicates that the series is non-stationary. The weak stationarity of a time series means that its statistical properties (mean, variance, covariance) remain constant over time. The respective hypotheses are as follows:

H_0: The series has a unit root (is non-stationary).

H_1: The series is stationary.

The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test (Kwiatkowski et al., 1992) checks for stationarity and not for the presence of a unit root. The respective hypotheses are as follows:

H_0: The series is stationary.

H_1: The series is not stationary.

The Hylleberg-Engle-Granger-Yoo (HEGY) test (Hylleberg et al., 1990) is a specialized unit root test for seasonal time series, allowing you to detect seasonal and non-seasonal unit roots separately. It's essential for diagnosing stationarity in quarterly data before fitting seasonal time series models. The respective hypotheses are as follows:

H_0: Unit roots at frequency 0 and/or seasonal frequencies (π, $\pi/2$, $3\pi/2$).

H_1: The series is stationary and/or there is no unit root at all frequency.

Based on the conclusions of the ADF, PP and KPSS tests, depending on the necessity of differencing, we apply regular differences, d. We compute and analyse the autocorrelation function (ACF) to conclude about the linear dependence between observations in a time series, leading to the value of q, and the partial autocorrelation function (PACF) to determine how many autoregressive terms p to include. Based on the conclusions of the HEGY test and if there are marked peaks at multiple lags of 4 in the ACF, it strongly suggests the existence of seasonal unit roots. Therefore, we apply the seasonal differences so we can identify the values of seasonal parameters P, D and Q. Finally, the tests are applied again to confirm the stationarity of the differenced time series.

Post-identification of a pertinent model, the subsequent task is to verify the satisfaction of the model assumptions. The fundamental premise of the models under consideration is that ε_t is a zero mean Gaussian white noise process. Consequently, the process of model diagnostic checking is achieved through a methodical analysis of the residuals: The first step involves the examination of the plot of the residuals to ensure that the variance remains constant. The third step involves the estimation of the sample autocorrelation function (ACF) and the sample partial autocorrelation function (PACF) of the residuals with the objective of checking if the lagged peaks are not statistically significant (until $\alpha = 0.05$). In addition, Box-Pierce and Ljung–Box tests are computed on residuals.

In the event of substantial spikes in the sample ACF and/or PACF of the residuals, and if the model failed the Box-Pierce test and Ljung–Box test, it is recommended that an alternative model be considered. In the absence of such issues, we can proceed to the forecast.

2.2.5 - Prophet

Prophet is an algorithm developed by Facebook (Meta) that has been distinguished by its robustness, performance in the presence of missing data and shifts in trend, and effective handling of outliers. It is characterised by its intuitive interpretation and ability to deal with non-linear trends, multiple seasonality and public holidays (Ensafi et al., 2022; Taylor & Letham, 2018). The distinguishing feature of this model is its flexibility in handling different types of component interactions. The components are combined linearly as follows:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{16}$$

where $g(t)$ is the trend component; $s(t)$ is the periodic seasonality component; $h(t)$ is the effect of holidays or special events; and ε_t the random error term, assumed to be normally distributed.

The trend component $g(t)$ assumes linear growth with change points. Where k is the growth rate, δ is vector that has the rate adjustments with variations in slope at each change point, m is the offset parameter, $a(t)$ is a vector showing which change points are active and γ is a vector with adjustments to maintain the function continuous.

$$g(t) = (k + a(t)^T\delta)(t - t_0) + (m + a(t)^T\gamma) \tag{17}$$

The Periodic seasonality component $s(t)$ is modelled using harmonic functions (Fourier). P is the period we expect the time series to have seasonality, in our case, we take the trimesters as 91.25 days. N is the number of Fourier terms which controls the smoothness.

$$s(t) = \sum_{n=1}^{N}\left(a_n\,cos\left(\frac{2\pi nt}{P}\right) + b_n sin\left(\frac{2\pi nt}{P}\right)\right) \tag{18}$$

Holidays and special events $h(t)$, represents events that occur on occasions which are not regular, but which nevertheless tend to have consistent impacts on the time series. To introduce them on the model, a customised list of past and future dates, D_i, for each event i, must be made, where a matrix of indicators $Z(t)$ is constructed that takes the value 1 when t coincides with the event and 0 otherwise.

Each event i has a parameter k_i, which quantifies the expected variation in the forecast during that event, making it possible to include windows of previous or subsequent days to capture residual effects.

$$h(t) = Z(t)k \tag{19}$$

The holidays and events considered are Christmas from 24 to 30 of December and New Year from 30 December to 1 January. In the event that their impact does not improve accuracy, they are not considered.

The hyperparameters can be selected manually but given the absence of guidelines for this choice and the objective of selecting the most suitable model, a grid search is employed. A grid search is a hyperparameter optimisation technique that is used to identify the optimal combination of parameters for a given model. In the context of Prophet, the process involves exhaustively testing all possible combinations of values defined for each parameter. The model's performance in each of these combinations is then evaluated based on the MAPE criteria, and the combination that generates the best result is selected.

The hyperparameters considered are: changepoint_prior_scale, with values of [0.001, 0.01, 0.1, 0.5, 1, 2, 5, 10, 20], which controls the smoothness δ of the changes in the slope of the trend, with the increase in the hyperparameter value: increases the flexibility of the model in trend changes, allowing it to detect more abrupt variations. High values can lead to overfitting; seasonality_prior_scale, with values of [0.1, 1, 5, 10, 20, 30, 50, 100], regulates the coefficients a_n and b_n in the Fourier series, responsible for the strength of seasonality, allowing for greater or lesser amplitude in seasonal fluctuations. When increased, the seasonality becomes stronger and with greater amplitude, as there is less regularisation of the Fourier series coefficients. There is the risk of exaggerated or unrealistic seasonality if the value is too high; seasonality_mode, with the options ["additive", "multiplicative"], which defines whether seasonality is added to the trend (constant effect) or whether seasonality is proportional to the trend level (greater effect when the series is at high values); changepoint_range with values of [0.8, 0.85, 0.9, 0.95, 1.0], delimits the initial fraction of the series in which change points may occur. Note that 1.0 = 100% of the series. It is acknowledged that including the entire series may generate change points near the end, which may affect long-term forecasts; and fourier_order, with values of [5, 10, 15, 20, 25], defines the complexity of the representation of seasonality through the Fourier series. The higher the value, the more complex the seasonal modelling. Very high values can cause overfitting.

These settings are explored combinatorially to identify the parameters that result in the best fit of the model to historical data.

2.2.6 - Artificial Neural Networks

In the field of forecasting, considerable attention has recently been given to Artificial Neural Networks following their promising performances in the domains of control and pattern recognition. ANN are a class of generalized nonlinear non-parametric models, meaning that the structure of the model is not fixed in advance as it can be adapted to the complexity of the data (Ensafi et al., 2022; Hyndman & Athanasopoulos, 2018; Chu & Zhang, 2003; Alon et al., 2001; Zhang et al., 1998).

The system can be considered to consist of a collection of simple nonlinear computing elements (nodes or neurons), with inputs and outputs that are linked together to form a network. In theory, it has been demonstrated that, with an adequate number of nonlinear processing units, neural networks can learn from experience and accurately estimate complex functional relationships (Zhang & Qi, 2005). ANN have the capacity to learn from data and experience, identify patterns or trends, and make generalisations about the future. The popularity of the neural network model can be attributed to its capability to simulate a wide variety of underlying nonlinear behaviours (Chu & Zhang, 2003).

The comparative advantage of ANN over more traditional econometric models is that they can model complex nonlinear relationships without any prior assumptions about the underlying data generating process. This flexible, data-driven modelling property has made it an attractive tool for many forecasting tasks and solving complex real-world forecasting problems (Zhang & Qi, 2005; Chu & Zhang, 2003; Alon et al., 2001).

Artificial neural networks models have been shown to be capable of overcoming the limitations of traditional forecasting methods such as misspecification, biased outliers, the assumption of linearity, and re-estimation (Alon et al., 2001). Moreover, although ANN are inherently nonlinear models, they are also capable of modelling linear processes (Zhang & Qi, 2005).

ANN are structurally constituted by a minimum of three layers, namely: an input layer, a hidden layer, and an output layer. The input layer is composed by input nodes, at which data is initially received. These nodes are linked to the nodes created in the hidden layer through "synapses". The synapses links carry a certain weight for each node. The weights are the model's trainable parameters. These weights are learned by minimizing a loss function, typically MSE, and determine how much influence each input has on the output. The weights adjust the strength of the signal passed between nodes in the network. The number of nodes in the hidden layer defines the model's complexity, while the weights reflect the strength of synapses between those nodes. In essence, a neural network has the capacity to acquire knowledge through the adjustment of weights associated to each synapse, by means of a nonlinear optimization algorithm (Zhang & Qi, 2005).

In the hidden layers, the nodes apply an activation function, so that the model captures the nonlinear structure in the data and the network can model complex patterns. In the present study, the function under consideration is a sigmoid function. This function transforms the inputs into outputs, or predicted values, by applying the weighted sum of inputs. The selection of an appropriate number of hidden nodes for a given application is a challenging task. Despite the indications of the neural network universal approximation theory that a substantial number of hidden nodes may be necessary for achieving a satisfactory approximation, empirical evidence has demonstrated that a limited number of hidden nodes frequently yields effective outcomes in the context of out-of-sample forecasting in numerous practical applications (Chu & Zhang, 2003). This phenomenon can be explained by the overfitting problem, which is a common challenge in neural network modelling processes. An overfitted model demonstrates a satisfactory fit to the sample used for model building, however, it exhibits limited generalisation capability for data out of the sample.

It is widely accepted that a three-layer feedforward network has the capacity to approximate any continuous function arbitrarily well, given a sufficient number of hidden layer nodes (Alon et al., 2001). Despite the proposal of many types of neural network models, the most popular for time series modelling and forecasting is the feedforward network model. As illustrated in Figure 1, a conventional three-layer feedforward model is employed for forecasting applications (Zhang & Qi, 2005).

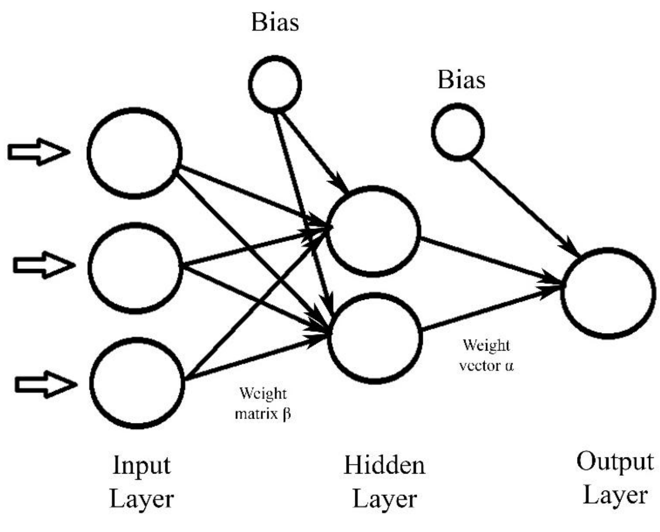

Figure 1 - Three-layer feedforward network Source: Own Source

The objective of the output layer is to produce a continuous numeric forecast, with minimum error rate or cost, which is to minimize the MSE between the predicted and actual values. It is possible that the assignments to the weights vector, and consequently the errors obtained through the network training, are not optimal. In order to determine the optimal weight values, the errors are backpropagated through the network from the output layer toward the hidden layers. Consequently, the weights are adjusted, using an optimization algorithm like gradient descent. This procedure is repeated over multiple iterations (epochs), during which the weights are continually updated to improve the predicted outputs and reduce the cost. The training process continues until the cost function is minimized, at which point the model is considered to have reached its optimal performance.

The relationship between the output (y_t) and the inputs $(y_{t-1},\ y_{t-2},\dots,y_{t-p})$ can be mathematically represented as follows:

$$y_t = \alpha_0 + \sum_{j=1}^{n} \alpha_j f\left(\sum_{i=1}^{m} \beta_{ij} y_{t-i} + \beta_0 j\right) + \varepsilon_t \tag{20}$$

where m is the number of input nodes, n is the number of hidden nodes, f is an activation function, that by default is a sigmoid function: $f(x) = \frac{1}{1+exp(-x)}$. $\{\alpha_j, j = 0,1, \dots ,n\}$ is a vector of weights from the hidden nodes to the output nodes and $\{\beta_{ij}, i = 0, 1, \dots, m;\ j = 0,1, \dots ,n\}$ is a matrix of weights from the input nodes to the hidden nodes. α_0 and $\beta_0 j$ are weights of the synapses leading from the bias terms. The bias term is a trainable parameter added to the weighted sum of inputs in a neural network node, prior to applying the activation function. Its purpose is to allow the model to fit the data more flexibly and efficiently by adjusting the activation threshold, enabling the neuron to learn to favour or suppress activation even when the input values are zero. The employment of bias terms is in both the hidden and output nodes.

In seasonal time series, observations separated by multiples of the seasonal period are frequently found to be correlated. Quarterly time series data is likely to exhibit correlation with observations made four quarters apart, that is, with past and future observations $(y_{t-4}, y_{t+4}, y_{t-8}, y_{t+8}, \dots, y_{t\mp p})$. It is imperative that these seasonal observations be incorporated into the neural network model as potential input variables, when direct seasonality modelling is being considered. It is important to note that, although theoretically, many seasonal lagged observations may be included in the model, in practice, the number of seasonal observations required in model can be fairly small (Zhang & Qi, 2005). The fundamental difference between models lies in their ability to directly model seasonal patterns and the necessity of seasonal adjustment.

In this study, the nnetar() function is employed to fit an $NNAR(p, P, k)[m]$ model, thereby indicating that there are p lagged input nodes and k nodes in the hidden layer, where k is the "size" parameter in R. In the case of seasonal data, it is also considered beneficial to incorporate seasonal observations as inputs, designated by P. In the context of our study, given that the time series is quarterly, m = 4. When P and k are zero, the model is equivalent to a pure $AR(p)$. An $NNAR(p, P, 0)[m]$ model is equivalent to an $ARIMA(p, 0,0)(P, 0,0)[m]$, but both with nonlinear functions and without the restrictions on the parameters that ensure stationarity (Hyndman & Athanasopoulos, 2018).

Regardless of the dimensionality of the input layer, the input vector for the time series forecast problem will be composed of a moving window of fixed length along the series. The data is divided into a training set and a test set. The training set is utilised for the estimation of weights, while the test set is employed to evaluate the generalisation capability of the network. In the context of the training set, which contains N observations $(y_1, y_2, \dots, y_N)$, it is assumed that all values utilised correspond to the observed values, and one step ahed forecasting is required. To this end, the nnetar() function, which incorporates p input nodes, is employed. This function generates input-output pairs from the training data, thereby yielding $N - p$ training patterns. The initial training pattern will comprise the inputs $y_1, y_2, \dots, y_p$, with y_{p+1} designated as the target output. The second training pattern will comprise $y_2, y_3, \dots, y_{p+1}$ as inputs and y_{p+2} as the target. The final training pattern will be $y_{N-p}, y_{N-P+1}, \dots , y_{N-1}$ for inputs and y_{N-1} for targets. In the context of P seasonal input, it is crucial to incorporate it into the training pattern inputs. In the present study, given the quarterly nature of the data, if $P = 1$, the resulting input includes the previous seasonal value $y_{(p+1)-4}$ (Hyndman & Athanasopoulos, 2018; Zhang et al., 1998).

During the forecast of the test set, after the last observed value y_N, the prediction of $\hat{y}_{N+1}$ is based on $y_{N-P+1}, \dots , y_N$. Subsequent predictions are made using the recursive multi-step forecast process, where predicted values are fed back into the model inputs. For instance, the prediction of $\hat{y}_{N+2}$, uses $y_{N-p+2}, \dots, y_n, \hat{y}_{N+1}$, and so on, until the final target forecast (Hyndman & Athanasopoulos, 2018; Zhang et al., 1998).

The selection of NNAR hyperparameters is a complex process. As with Prophet, due to the absence of guidelines and assumptions, a grid search was implemented in order to ascertain the optimal combination of hyperparameters. The selection of the most appropriate model was guided by the same principles as in Prophet, with the objective being to minimise errors in the test set according to MAPE.

The grid was over the following hyperparameters: the number of non-seasonal lagged inputs $p \in [1,17]$; the number of seasonal lagged inputs $P \in [0,4]$; the number of nodes in the hidden layer $size \in [1,17]$; the number of repeated trainings of the network with different random weight initializations $repeats \in [1,50]$. The $repeats$ argument in nnetar() specifies how many times the neural network is trained independently, each time starting with a different set of random initial weights. The final forecast is obtained by averaging the forecast from all the repetitions (Hyndman & Caceres, 2025). This mechanism addresses a known issue in neural network training, the sensitivity to weight initialization, and helps stabilise the forecasts by reducing variance across runs.

In order to guarantee reproducibility, a fixed seed value was established by means of the set.seed() function. This ensured that the weight initialisation for each model repetition was consistent across runs, thus allowing results such as forecast accuracy and the network to be reproduced.

The objective of this study is to estimate and compare traditional and machine learning methods for forecasting the quarterly revenue of the following US companies: Alphabet, Amazon, Apple, Meta, Microsoft, Nvidia, and Tesla. The present study is supported by a comprehensive literature review, which has led to the identification of particular methods, with emphasis on ARIMA, Prophet, and ANN. The subsequent investigation will ascertain whether the incorporation of holiday effects in Prophet enhances the model's accuracy, whether any data processing enhances the performance of the ANN, and whether automating the process via a hyperparameter grid search yields superior performance in comparison to manually specifying parameters in terms of its report time. It is imperative that the architecture of the artificial neural network ANN be presented in a transparent manner, and that the adjustments which render forecasting both operational and reproducible be explained.

2.2.7 - Evaluation Criteria

The estimates obtained in this study are subject to a certain degree of measurement error. In order to evaluate the efficacy of a prediction, it is necessary to have in place a series of metrics that can be used to judge the accuracy of the results. It is important to note that different accuracy measures can provide different results, and consequently, different conclusions may be drawn as to which forecast method is most effective (Ramos et al*.,* 2015; Wong & Guo, 2010).

Let be denoted the actual observation for time period *t* by y_t and the predicted value for the same period by $\hat{y}_t$. The error associated to the forecast is denoted by:

$$e_t = y_t - \hat{y}_t \tag{21}$$

The present study will evaluate and compare the accuracy of forecasting models based on scale-dependent measures such as the Mean Squared Error (MSE), and the Root Mean Squared Error (RMSE). Furthermore, the percentage error and scale-independent measure, the mean absolute percentage error (MAPE), is also considered. Where n represents the number of observations.

2.2.7.1 - MSE

The MSE is calculated by computing the average of the squared errors, that is, the difference between the actual values and the estimated, squared. The formula for computing MSE is as it follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{22}$$

This metric has historically been applied because of is theoretical relevance in statistical modelling. This measure is more sensitive to outliers than other measures. Furthermore, it assigns disproportionately greater weight to large errors (Ramos et al., 2015; Hyndman & Koehler, 2006).

2.2.7.2 - RMSE

The RMSE is computed by taking the square root of the MSE. The formula for computing RMSE is as it follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{23}$$

The RMSE is also sensitive to outliers but scales the scores in the same units as the forecast values and by that, is preferred to the MSE. The main benefit of using RMSE is that it penalizes large errors (Siami-Namini et al., 2018; Hyndman & Koehler, 2006).

2.2.7.3 - MAPE

Percentage errors have the advantage of being scale-independent, and as such, they are frequently used to compare forecast performance between different data sets. The formula for computing MAPE is as it follows:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{24}$$

This measure has the disadvantage of being infinite or undefined if $y_t = 0$ for any t in the period of interest and having extreme values when any y_t is close to zero (Hyndman & Koehler, 2006) and it is less sensitive to large errors than RMSE (Wong & Guo, 2010).

3 - Empirical Findings

The empirical findings are consistent with what is described the methodology section, where the different methods were replicated in the same way for the seven companies under study. For each company, the sample is divided chronologically into an 80% training set and a 20% test set. All estimation and parameter tuning is based on the training data, and the accuracy is assessed on the test set. The following forecasting methods are then implemented for each company: Moving Average, Decomposition, Holt–Winters, ARIMA, Prophet (with and without a holidays component), and Artificial Neural Networks. The final architecture specification is reported for ARIMA, Prophet and ANN. The estimated parameters of decomposition and Holt–Winters methods, can be found in the Appendices A and B, respectively. For the Simple Moving Average, there are no parameter estimates. The evaluation of comparative performance is conducted on the test set using the following metrics: MSE, RMSE and MAPE. Finally, for each company, the model that achieves the best out-of-sample accuracy is used to generate eight-quarter-ahead forecasts.

The Box-Jenkins methodology was followed meticulously as delineated in section 2.2.4. The results of the tests performed are to be found in the Appendices C to S. The final models were maintained when stationarity and the presence of non-autocorrelated residuals were achieved, and are as follows in Table 2:

Table 2 - ARIMA Final Models

ARIMA Final Models	
Alphabet	SARIMA (0,1,0) (0,1,2)
Amazon	SARIMA (4,1,0) (1,1,1)
Apple	SARIMA (2,1,0) (3,1,3)
Meta	SARIMA (6,1,0) (0,1,6)
Microsoft	SARIMA (4,1,0) (0,1,0)
Nvidia	ARIMA (1,1,1)
Tesla	SARIMA (6,1,1) (4,1,3)

It should be noted that all series have a seasonal component except for Nvidia.

According to section 2.2.5, the best combinations of hyperparameters of Prophet are as follows in Table 3:

Table 3 - Prophet hyperparameters

	Prophet hyperparameters					
	changepoint_ _prior_scale	seasonality_ _prior_scale	seasonality_ _mode	changepoint_ _range	fourier_ _order	holidays
Alphabet	10	20	multiplicative	0,8	15	No
Amazon	5	30	additive	0,9	5	No
Apple	0,05	10	multiplicative	0,8	15	Yes
Meta	5	1	multiplicative	0,95	10	No
Microsoft	2	5	multiplicative	0,9	5	No
Nvidia	1	0,10	multiplicative	0,9	25	No
Tesla	5	20	additive	0,8	25	No

It should be noted that only in Apple the holidays improved accuracy in the forecast of the test set. In accordance with the estimated seasonal parameters calculated in previous methods, it was anticipated that a greater number of companies would be influenced by these events in Q4. Given the superior performance of multiplicative methods in previous methods, it was to be expected that all companies would gravitate towards multiplicative seasonality. However, two exceptions were observed. Amazon and Tesla, for instance, have incorporated additive seasonality.

According to section 2.2.6, the best combinations of hyperparameters of the ANN are as follows in Table 4:

Table 4 - ANN Hyperparameters

	ANN Hyperparameters			
	NNAR type	Averages by network	Network	Total Weights
Alphabet	(p = 16, P = 0, size = 2, repeats = 1) [4]	1	16-2-1	37
Amazon	(p = 15, P = 0, size = 7, repeats = 2) [4]	2	15-7-1	120
Apple	(p = 12, P = 1, size = 2, repeats = 1) [4]	1	12-2-1	29
Meta	(p = 5, P = 4, size = 2, repeats = 1) [4]	1	8-2-1	21
Microsoft	(p = 4, P = 3, size = 15, repeats = 6) [4]	6	6-15-1	121
Nvidia	(p = 16, P = 4, size = 15, repeats = 2) [4]	2	16-15-1	271
Tesla	(p = 6, P = 1, size = 4, repeats = 5) [4]	5	6-4-1	33

To represent the architecture of ANN models, the structure for Meta is illustrated below in Figure 2, as it is the simplest and most understandable for the study. The trained NNAR model was decomposed in order to explicitly extract the network from each repetition. For each network that was fitted, the complete weight vector was reorganised according to the architecture. The first set is associated with the weighting matrix β, which encompasses the weights that establish the connection between the input layers to the hidden layer. Second, the hidden bias vector β_0, stores the weight for the bias terms in the hidden layer. The third, corresponds to the weight vector α, the weights from the hidden layer to the output layer. Finally, the output bias term α_0. These weights were isolated and stored in structured lists, allowing the visualization of the network.

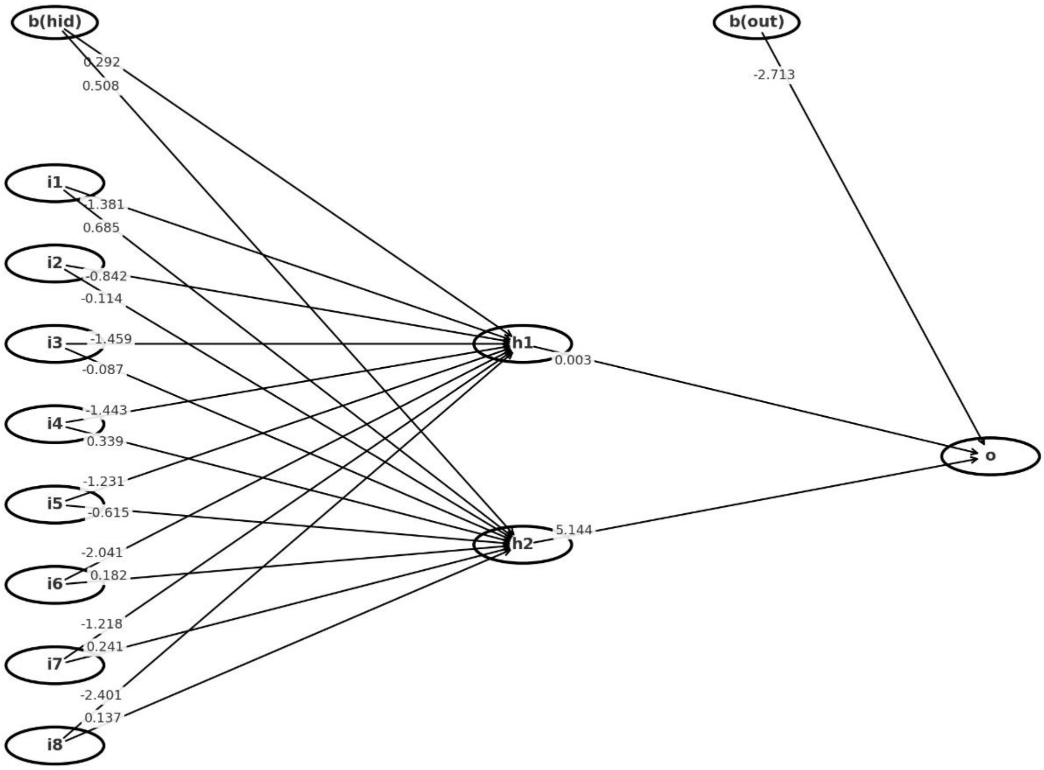

Figure 2 - Meta Weights ANN Source: Own Source

As illustrated in Figure 2, the model comprises eight input nodes, of which five are non-seasonal lagged inputs p $(y_t - 1, y_t - 2, y_t - 3, y_t - 4, y_t - 5)$ and four are seasonal lagged inputs P $(y_t - 4, y_t - 8, y_t - 12, y_t - 16)$.

Finally, the results for the evaluation criteria for predicting the test set of the revenue variable, can be found in Table 5.

Table 5 - Evaluation Criteria

	Alphabet			Amazon			Apple		
	MSE	RMSE	MAPE	MSE	RMSE	MAPE	MSE	RMSE	MAPE
Moving Average	1,18E+09	34375,0	42,2	4,98E+09	70583,3	47,87	9,85E+08	31386,5	26,96
Decomposition	1,54E+09	39196,1	41,7	7,74E+09	87982,1	64,69	9,22E+08	30362,7	25,21
Holt-Winters	6,44E+08	25382,7	31,5	75773118	8704,8	6,05	4,12E+08	20298,7	19,84
ARIMA	1,02E+08	10077,1	12,62	1,44E+08	11999,0	8,42	1,23E+08	11107,5	9,82
Prophet	67099362	8191,4	9,65	74636122	8639,2	5,74	1,4E+08	11823,2	9,25
ANN	**45707064**	**6760,7**	**7,86**	**30870136**	**5556,1**	**3,36**	**63736081**	**7983,5**	**6,91**

	Meta			Microsoft			Nvidia		
	MSE	RMSE	MAPE	MSE	RMSE	MAPE	MSE	RMSE	MAPE
Moving Average	8,1E+07	8997,1	17,01	5,3E+08	23077,2	36,89	2,9E+08	17029,2	65,74
Decomposition	8,2E+07	9043,5	22,58	2,2E+07	4709,3	8,38	2,4E+08	15491,0	51,52
Holt-Winters	2,0E+07	4492,1	12,54	2,6E+08	16256,0	25,25	2,7E+08	16570,7	59,35
ARIMA	4,2E+06	2051,1	5,09	3,4E+06	1847,5	2,78	2,1E+08	14587,4	43,78
Prophet	**4,1E+06**	**2035,5**	5,07	5,4E+06	2324,4	4,19	**1,4E+08**	**11817,6**	**28,23**
ANN	5,8E+06	2399,3	**4,21**	**1,9E+06**	**1372,5**	**1,86**	1,8E+08	13560,7	33,49

	Tesla		
	MSE	RMSE	MAPE
Moving Average	1,1E+08	10505,5	43,65
Decomposition	1,7E+08	12945,2	55,62
Holt-Winters	2,4E+07	4881,1	17,05
ARIMA	1,4E+07	3705,8	12,98
Prophet	1,1E+07	3288,0	11,67
ANN	**4,7E+06**	**2165,7**	**6,79**

According to the data presented in the Table 5 and assigning greater significance to MAPE due to its nature as an unscaled measure, it can be concluded that, in six out of seven companies, ANN demonstrates a superior predictive capacity in comparison to alternative methods. However, in the case of Nvidia, Prophet does demonstrate a slightly superior predictive capability, exhibiting a slightly more accurate capture of the exponential revenue pattern observed in recent years. Nonetheless, a MAPE of 28,23 indicates that, on average, the predictions are still approximately 28,23% inaccurate. It is noteworthy that the ANN model exhibited a remarkable performance in the Microsoft example, with a MAPE of 1,86. The ARIMA method is a well-established approach that can still be regarded as a strong contender in this field. The findings indicate that the Simple Moving Average, Decomposition and Holt-Winters methods are significantly less effective due to their inability to capture non-linearities and irregularities. However, it should be noted that the Exponential Holt-Winters method achieved an MAPE of 6,05 in the case of Amazon and the Decomposition method achieved an MAPE of 8,38 in the case of Microsoft. Moreover, an analysis of the evaluation criteria for Prophet was conducted to ascertain whether the incorporation of the Holidays component would enhance the accuracy. Evidence for this can be found in Table 5 and Appendix T, where Table 5 is the best evaluation criteria (with or without the component) and Appendix T is only with the component. The evidence presented indicates that its incorporation has resulted in an increase only in Meta. We can find all the forecasts in the Appendices U to BJ.

Finally, as an exercise to be confirmed at a future date, an ex-ante forecast from the ANN is presented below in Table 6, for the revenue series over a two-year time interval. It is worth noting that the values are expressed in millions of US dollars.

Table 6 - ANN Forecast in Millions of US dollars

	ANN Forecast							
	2025 Q2	2025 Q3	2025 Q4	2026 Q1	2026 Q2	2026 Q3	2026 Q4	2027 Q1
Alphabet	88169.08	104175.28	97847.45	92352.69	99015.27	112399.88	101681.91	92902.27
Amazon	154421.8	161954.1	201535.9	154131.8	159416.1	166071.6	208744.7	155813.2
Apple	94842.12	112925.90	125297.71	98333.28	98700.08	116087.93	125515.50	100475.11
Meta	38567.31	38484.95	41542.98	39177.16	39834.69	40379.08	43497.83	41751.97
Microsoft	71527.40	71411.57	74713.18	72683.33	76779.95	76379.16	79371.93	77633.04
Nvidia	-	9387.47	9158.72	8884.23	8860.38	9170.98	9217.55	8759.26
Tesla	23888.41	23655.75	23460.78	22650.66	21910.65	21805.51	21735.31	21235.12

Conclusion

The objective of the present study was to conduct a comparative analysis of the performance of traditional time series methods and machine learning algorithms in the context of revenue forecasting. Considering that the variable under study, revenue, displays seasonal, recurring and systematic behaviour over time, it is therefore hypothesised that both traditional forecasting methods and machine learning-based models would have the potential to effectively capture and model these patterns. The observed repetitiveness in the majority of time series indicates that, within the specific context of the analysed companies, the most sophisticated classical approaches, such as ARIMA, would already be sufficiently robust to predict revenue trends. However, and in accordance with prior research and contemporary trends in sales/revenues forecasting, it was hypothesised that machine learning algorithms would demonstrate superior performance in comparison to time series due to their capacity to manage complex data patterns and non-linear relationships.

It is also noteworthy that all results obtained were based exclusively on the time series history itself, without incorporating exogenous variables that could also influence revenue behaviour. It is acknowledged that external factors, including macroeconomic conditions, marketing campaigns, regulatory changes and specific seasonal events, exert a considerable influence on revenue dynamics. However, it was found that, even when considering only endogenous information, the three most effective methods had reduced forecast errors, demonstrating consistent and reliable performance. This outcome reinforces the robustness of the approaches examined, and their suitability for contexts in which external data is not available.

It was also observed that, in the initial phase, the training set and the test set were divided into 2/3 and 1/3, respectively. However, unsatisfactory results from the machine learning algorithms necessitated a modification of the division, which was changed to 80% and 20%, respectively. This alteration has resulted in increased capacity for training the models, which has consequently improved the accuracy of all implemented methods.

As demonstrated in Table 7, where we can see the percentage improvement between methods accuracy based on MAPE, ANN emerge as the most effective approach for revenue forecasting. A substantial improvement in its utilisation has been observed. It has been demonstrated that Prophet is a superior model when compared with ARIMA. In only one instance is the Prophet model outperformed by ARIMA, and in one instance the ANN model is outperformed by the Prophet model, supporting the idea formulated by Alon, et al. (2001) that machine learning algorithms are capable of producing more accurate forecasts in volatile economic environments than traditional methods. Nevertheless, the ARIMA model continues to be a strong contender. In relatively stable conditions, the ARIMA demonstrated viable performance. However, the machine learning algorithms consistently outperformed.

Table 7 - Percentage improvement between methods based on MAPE

	ANN x Prophet	Prophet x ARIMA	ANN x ARIMA
Alphabet	18,55 %	23,53 %	37,72 %
Amazon	41,46 %	31,83 %	60,10 %
Apple	25,26 %	5,79 %	29,59 %
Meta	16,98 %	0,41 %	17,32 %
Microsoft	55,53 %	-50,99 %	32,85 %
Nvidia	-18,62 %	35,52 %	23,52 %
Tesla	41,82 %	10,12 %	47,71 %

Additionally, it should be noted that more recent methods, such as Prophet and ANN, involve a substantially more computationally demanding estimation process. In both cases, a grid search procedure was applied to automatically select the optimal parameter combinations within a predefined range. While this ensured more rigorous tuning of the models, it resulted in an average processing time of approximately 12 hours per company. By contrast, the ARIMA model used classic methodology guidelines for parameter selection had a significantly shorter estimation time. Therefore, while machine learning models have demonstrated superior predictive capabilities, the decision to adopt them must be carefully considered, taking into account factors such as the non-necessity to verify certain assumptions such as stationarity, autocorrelation of errors or the presence of white noise errors, estimation time, computing power costs, and operational feasibility in business contexts requiring rapid and recurring forecasts.

Limitations of the Study

Nevertheless, the study was subject to certain limitations. Initially, the hyperparameter selection for Prophet and ANN relied on grid search, which, given the search space and hardware constraints, took approximately 12 hours per company to converge. Secondly, the forecasting setup was univariate, with models trained exclusively on each series own history, excluding exogenous drivers. Consequently, the models demonstrated limited capacity to incorporate information known to affect revenue, such as macroeconomic indicators, marketing expenditure, seasonality shocks, and holidays. The incorporation of these factors would likely result in an increase in the predictive power of the model. It should be noted that the information on holidays was incorporated only on Prophet, as it is already part of the model structure. Thirdly, the present study was lacking in the implementation of tests for equal predictive accuracy, such as the Diebold–Mariano test. Such tests require a sequence of out-of-sample forecast errors obtained from multiple forecast origins via 1-step-ahead forecast at every point across the entire test set to form a long error series. Consequently, model comparisons were reported only via MSE, RMSE and MAPE rather than statistically significant DM tests. The limitations are indicated to assist future research, thereby contributing to an enhancement in the methodology employed and the results obtained. Finally, and following on from the fact that ANN outperformed the other methods, implement or combine more advanced and sophisticated methods.

Bibliography

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. *Journal of retailing and consumer services*, *8*(3), 147-156. https://doi.org/10.1016/S0969-6989(00)00011-4

Armstrong, J. S. (Ed.). (2001). *Principles of forecasting: a handbook for researchers and practitioners* (Vol. 30, pp. 1-2). Boston, MA: Kluwer Academic.

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

Chan, A. (2024, maio, 10). Revenue forecasting explained. *Oracle*. https://www.oracle.com/erp/cfo/revenue-forecasting/#what

Cheng, C. A., Chu, K. K., & Ohlson, J. (2020). Analyst forecasts: sales and profit margins. *Review of Accounting Studies*, *25*, 54-83. https://doi.org/10.1007/s11142-019-09521-z

Chu, C. W., & Zhang, G. P. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of production economics*, *86*(3), 217-231. https://doi.org/10.1016/S0925-5273(03)00068-9

Coimbra, T., Boccato, L., Siqueira, H. V., & Attux, R. (2024). Comparative Analysis of Deseasonalization and Detrending Methods in Energy Consumption Forecasting. http://dx.doi.org/10.21528/CBIC2023-105

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning–A comparative analysis. *International Journal of Information Management Data Insights*, *2*(1), 100058. https://doi.org/10.1016/j.jjimei.2022.100058

Fuller, W. A. (1995). *Introduction to statistical time series*. John Wiley & Sons.

Gong, Y., Wang, Z., Xu, G., & Zhang, Z. (2018). A comparative study of groundwater level forecasting using data-driven models based on ensemble empirical mode decomposition. Water, 10(6), 730. https://doi.org/10.3390/w10060730

Holt, C. C. (1957). Forecasting seasonals and trends by exponentially weighted moving averages (ONR Memorandum No. 52). Office of Naval Research.

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. International Journal of Forecasting, 20(1), 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015

Hürtgen, H., Plaschke, F., Sauer-Sidor, K., & Wittmann, N. (2020, agosto). Predictive sales forecasting: Is your finance function up to code?. *McKinsey & Company.* https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/predictive-sales-forecasting-is-your-finance-function-up-to-code#/

Hylleberg, S., Engle, R. F., Granger, C. W., & Yoo, B. S. (1990). Seasonal integration and cointegration. *Journal of econometrics*, *44*(1-2), 215-238. https://doi.org/10.1016/0304-4076(90)90080-D

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

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, *22*(4), 679-688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Hyndman, R. J., & Caceres, G. (2025). nnetar: Neural network time series forecasts (forecast package version 8.24.0) [R package documentation]. RDocumentation. https://www.rdocumentation.org/packages/forecast/versions/8.24.0/topics/nnetar

Kożuch, A., Cywicka, D., & Adamowicz, K. (2023). A comparison of artificial neural network and time series models for timber price forecasting. *Forests*, *14*(2), 177. https://doi.org/10.3390/f14020177

Kuo, R. J., & Xue, K. C. (1999). Fuzzy neural networks with application to sales forecasting. *Fuzzy Sets and Systems*, *108*(2), 123-143. https://doi.org/10.1016/S0165-0114(97)00326-6

Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. *Journal of econometrics*, *54*(1-3), 159-178. https://doi.org/10.1016/0304-4076(92)90104-Y

Lin, K. P., Pai, P. F., Lu, Y. M., & Chang, P. T. (2013). Revenue forecasting using a least-squares support vector regression model in a fuzzy environment. *Information Sciences*, *220*, 196-209. https://doi.org/10.1016/j.ins.2011.09.003

Makridakis, S., & Hibon, M. (2000). The M3-Competition: results, conclusions and implications. International journal of forecasting, 16(4), 451-476. https://doi.org/10.1016/S0169-2070(00)00057-1

Nguyen, H. D., Tran, K. P., Thomassey, S., & Hamad, M. (2021). Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, *57*, 102282. https://doi.org/10.1016/j.ijinfomgt.2020.102282

Padmanabhan, V., & Nirmala, D. S. (2023). PREDICTABILITY OF TOTAL REVENUE IN GLOBAL LARGE COMPANIES FROM FINANCIAL AND ECONOMIC PARAMETERS. International Development Planning Review, 22(2023), 862-879.

Perron, P. (1988). Trends and random walks in macroeconomic time series: Further evidence from a new approach. *Journal of economic dynamics and control*, *12*(2-3), 297-332. https://doi.org/10.1016/0165-1889(88)90043-7Get rights and content

Pundir, A. K., Ganapathy, L., Maheshwari, P., & Kumar, M. N. (2020, November). Machine learning for revenue forecasting: A case study in retail business. In *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0201-0207). IEEE. https://doi.org/10.1109/IEMCON51383.2020.9284819

Ramos, P., Santos, N., & Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and computer-integrated manufacturing*, *34*, 151-163. https://doi.org/10.1016/j.rcim.2014.12.015

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, December). A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)* (pp. 1394-1401). Ieee. https://doi.org/10.1109/ICMLA.2018.00227

Suryawan, I. G. T., Putra, I. K. N., Meliana, P. M., & Sudipa, I. G. I. (2024). Performance Comparison of ARIMA, LSTM, and Prophet Methods in Sales Forecasting. *Sinkron: jurnal dan penelitian teknik informatika*, *8*(4), 2410-2421. https://doi.org/10.33395/sinkron.v8i4.14057

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

Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics*, *128*(2), 470-483. https://doi.org/10.1016/j.ijpe.2010.07.018

Trueman, B., Wong, M. F., & Zhang, X. J. (2001). Back to basics: Forecasting the revenues of Internet firms. Review of Accounting Studies, 6(2), 305-329. https://doi.org/10.1023/A:1011623111051

Whitfield, R. I., & Duffy, A. H. B. (2013). Extended revenue forecasting within a service industry. *International Journal of Production Economics*, 141(2), 505-518. https://doi.org/10.1016/j.ijpe.2011.11.015

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, *6*(3), 324-342. https://doi.org/10.1287/mnsc.6.3.324

Wong, W. K., & Guo, Z. X. (2010). A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm. *International Journal of Production Economics*, *128*(2), 614-624. https://doi.org/10.1016/j.ijpe.2010.07.008

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European journal of operational research*, *160*(2), 501-514. https://doi.org/10.1016/j.ejor.2003.08.037

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International journal of forecasting*, *14*(1), 35-62. https://doi.org/10.1016/S0169-2070(97)00044-7

Appendices

Appendix A

Table 8 - Estimates of Seasonal Coefficients for the Multiplicative Decomposition Method

	Seasonal Coefficients			
	Q1	Q2	Q3	Q4
Alphabet	0.9926119	0.9769186	0.9791275	1.0513420
Amazon	0.9617433	0.8749766	0.8753059	1.2879742
Apple	0.9587870	0.9015207	0.9508055	1.1888867
Meta	0.9395576	0.9825179	0.9624374	1.1154871
Microsoft	0.9735688	0.9914404	0.9393998	1.0955910
Nvidia	1.0184569	0.9892735	0.9565155	1.0357542
Tesla	0.9920500	0.9282865	0.9940564	1.0856071

Appendix B

Table 9 - Multiplicative Holt-Winters

	Holt-Winters Estimates								
	α	β	γ	Level (a)	Trend (b)	s1	s2	s3	s4
Alphabet	0,181	1	0,340	41387,19	583,45	1,082	0,966	0,948	1,005
Amazon	0,566	1	1	61428,07	3886,47	0,945	1,182	0,938	0,933
Apple	0,646	0,072	1	42998,29	513,70	0,814	0,915	1,335	0,936
Meta	0,876	0,131	1	29784,46	1216,50	0,931	1,000	0,974	1,094
Microsoft	0,531	0,034	0,227	30140,68	395,07	0,999	0,937	1,099	0,962
Nvidia	0,905	0	1	2841,66	21,01	0,970	0,974	0,998	1,058
Tesla	0,540	0,285	0,381	14252,91	1245,05	0,902	0,939	1,052	1,107

Appendix C

Table 10 - ADF Unit Root Test Before Differentiation (type="none")

ADF Unit Root Test Before Differentiation (type="none")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	2,1899	-1,95	Non-stationary Has a unit root
Amazon	-0.822	-1.95	Non-stationary Has a unit root
Apple	1.6061	-1.95	Non-stationary Has a unit root
Meta	2.6183	-1.95	Non-stationary Has a unit root
Microsoft	3.9343	-1.95	Non-stationary Has a unit root
Nvidia	-0.0705	-1.95	Non-stationary Has a unit root
Tesla	-0.1276	-1.95	Non-stationary Has a unit root

Appendix D

Table 11 - ADF Unit Root Test Before Differentiation (type="drift")

ADF Unit Root Test Before Differentiation (type="drift")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	1.9228	tau2= -2,89	Non-stationary
	3.1719	phi1= 4.71	Has a unit root
Amazon	-0.782	tau2= -2.88	Non-stationary
	1.1787	phi1= 4.63	Has a unit root
Apple	0.277	tau2= -2.88	Non-stationary
	2.3507	phi1= 4.63	Has a unit root
Meta	1.8736	tau2= -2,89	Non-stationary
	9.3442	phi1= 4.71	Has a unit root
Microsoft	3.4348	tau2= -2.88	Non-stationary
	8.0802	phi1= 4.63	Has a unit root
Nvidia	1.2054	tau2= -2.88	Non-stationary
	1.2717	phi1= 4.63	Has a unit root
Tesla	-0.4954	tau2= -2,89	Non-stationary
	1.8255	phi1= 4.71	Has a unit root

Appendix E

Table 12 - ADF Unit Root Test Before Differentiation (type="trend")

ADF Unit Root Test Before Differentiation (type="trend")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-0.4299	tau 3= -3.45	Non-stationary Has a unit root
	3.3136	phi2= 4.88	
	3.591	phi3= 6.49	
Amazon	-1.5415	tau 3= -3.43	Non-stationary Has a unit root
	2.3261	phi2= 4.75	
	2.5768	phi3= 6.49	
Apple	-2.8273	tau 3= -3.43	Non-stationary Has a unit root
	4.6926	phi2= 4.75	
	4.5284	phi3= 6.49	
Meta	-1.8738	tau 3= -3.45	Non-stationary Has a unit root
	8.4671	phi2= 4.88	
	4.2562	phi3= 6.49	
Microsoft	1.9045	tau 3= -3.43	Non-stationary Has a unit root
	5.3449	phi2= 4.75	
	5.8586	phi3= 6.49	
Nvidia	2.6168	tau 3= -3.43	Non-stationary Has a unit root
	2.691	phi2= 4.75	
	3.4547	phi3= 6.49	
Tesla	-2.066	tau 3= -3.45	Non-stationary Has a unit root
	1.8842	phi2= 4.88	
	2.1498	phi3= 6.49	

Appendix F

Table 13 - PP Unit Root Test Before Differentiation (lags="short")

PP Unit Root Test Before Differentiation (lags="short")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-0.5124	-3.462585	Non-stationary Has a unit root
Amazon	-0.8981	-3.450084	Non-stationary Has a unit root
Apple	-5.7015	-3.4478	Stationary (trend)
Meta	-2.1721	-3.484869	Non-stationary Has a unit root
Microsoft	0.9878	-3.4478	Non-stationary Has a unit root
Nvidia	8.1124	-3.452684	Non-stationary Has a unit root
Tesla	-1.8221	-3.483605	Non-stationary Has a unit root

Appendix G

Table 14 - PP Unit Root Test Before Differentiation (lags="long")

PP Unit Root Test Before Differentiation (lags="long")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-0.5619	-3.462585	Non-stationary Has a unit root
Amazon	-1.0735	-3.450084	Non-stationary Has a unit root
Apple	-6.9681	-3.4478	Stationary (trend)
Meta	-2.3557	-3.484869	Non-stationary Has a unit root
Microsoft	0.8538	-3.4478	Non-stationary Has a unit root
Nvidia	12.3545	-3.452684	Non-stationary Has a unit root
Tesla	-2	-3.483605	Non-stationary Has a unit root

Appendix H

Table 15 - KPSS Unit Root Test Before Differentiation (lags="short")

KPSS Unit Root Test Before Differentiation (lags="short")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	0.5313	0.146	Non-stationary
Amazon	0.5441	0.146	Non-stationary
Apple	0.52	0.146	Non-stationary
Meta	0.3717	0.146	Non-stationary
Microsoft	0.4809	0.146	Non-stationary
Nvidia	0.2913	0.146	Non-stationary
Tesla	0.3642	0.146	Non-stationary

Appendix I

Table 16 - KPSS Unit Root Test Before Differentiation (lags="long")

KPSS Unit Root Test Before Differentiation (lags="long")			
	Statistic	Critical Value (5%)	Conclusion
Alphabet	0.2119	0.146	Non-stationary
Amazon	0.2381	0.146	Non-stationary
Apple	0.2371	0.146	Non-stationary
Meta	0.1846	0.146	Non-stationary
Microsoft	0.2219	0.146	Non-stationary
Nvidia	0.1779	0.146	Non-stationary
Tesla	0.1671	0.146	Non-stationary

Appendix J

Table 17 - HEGY Seasonal Unit Root Test Before Differentiation

HEGY Seasonal Unit Root Test Before Differentiation				
	Seasonal Frequency	Statistic	p-value	Conclusion (5%)
Alphabet	0 (t_1)	0.284	0.9794	Unit root present
	π (t_2)	–0.2531	0.9833	Unit root present
	±π/2 ($F_{3:4}$)	0.3725	0.9915	Unit root present
	π & ±π/2 ($F_{2:4}$)	0.2704	0.5239	Unit root present
	All ($F_{1:4}$)	0.2207	0.0098	No joint unit root
Amazon	0 (t_1)	–1.4894	0.7286	Unit root present
	π (t_2)	1.4682	0.9710	Unit root present
	±π/2 ($F_{3:4}$)	12.8597	0.0000	No unit root
	π & ±π/2 ($F_{2:4}$)	9.2736	0.0000	No unit root
	All ($F_{1:4}$)	7.4114	0.0062	No joint unit root
Apple	0 (t_1)	–2.2161	0.4801	Unit root present
	π (t_2)	–1.0055	0.8247	Unit root present
	±π/2 ($F_{3:4}$)	0.9429	0.9287	Unit root present
	π & ±π/2 ($F_{2:4}$)	0.9740	0.5742	Unit root present
	All ($F_{1:4}$)	1.9857	0.6771	Joint unit root
Meta	0 (t_1)	–2.0055	0.0000	No unit root
	π (t_2)	0.8239	0.6160	Unit root present
	±π/2 ($F_{3:4}$)	0.0450	0.0054	No unit root
	π & ±π/2 ($F_{2:4}$)	0.2599	0.0130	No unit root
	All ($F_{1:4}$)	1.2446	0.0197	No joint unit root
Microsoft	0 (t_1)	2.3654	1.0000	Unit root present
	π (t_2)	–2.3123	0.2088	Unit root present
	±π/2 ($F_{3:4}$)	13.9079	0.0015	No unit root
	π & ±π/2 ($F_{2:4}$)	11.0942	0.0942	Unit root present
	All ($F_{1:4}$)	10.0519	0.0652	Joint unit root
Nvidia	0 (t_1)	6.2593	1.0000	Unit root present
	π (t_2)	–5.4743	0.0000	No unit root
	±π/2 ($F_{3:4}$)	29.2713	0.0000	No unit root
	π & ±π/2 ($F_{2:4}$)	31.1891	0.0000	No unit root
	All ($F_{1:4}$)	29.0314	0.0000	No joint unit root
Tesla	0 (t_1)	–2.3781	1.0000	Unit root present
	π (t_2)	–1.8276	1.0000	Unit root present
	±π/2 ($F_{3:4}$)	2.8157	0.9964	Unit root present
	π & ±π/2 ($F_{2:4}$)	2.7910	0.3996	Unit root present
	All ($F_{1:4}$)	3.6888	0.6413	Joint unit root

Appendix K

Table 18 - ADF Unit Root Test After Differentiation (type="none")

	ADF Unit Root Test After Differentiation (type="none")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-6.7592	-1.95	Stationary No Unit Root
Amazon	-6.9355	-1.95	Stationary No Unit Root
Apple	-6.3465	-1.95	Stationary No Unit Root
Meta	-4.5455	-1.95	Stationary No Unit Root
Microsoft	-7.4736	-1.95	Stationary No Unit Root
Nvidia	-5.0877	-1.95	Stationary No Unit Root
Tesla	-5.0265	-1.95	Stationary No Unit Root

Appendix L

Table 19 - ADF Unit Root Test After Differentiation (type="drift")

	ADF Unit Root Test After Differentiation (type="drift")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-6.6957	tau2= -2,89	Stationary
	22.4459	phi1= 4.71	No Unit Root
Amazon	-6.9287	tau2= -2.88	Stationary
	24.0146	phi1= 4.63	No Unit Root
Apple	-6.3077	tau2= -2.88	Stationary
	19.9331	phi1= 4.63	No Unit Root
Meta	-3.5594	tau2= -2,89	Stationary
	6.3735	phi1= 4.71	No Unit Root
Microsoft	-7.4343	tau2= -2.88	Stationary
	27.6351	phi1= 4.63	No Unit Root
Nvidia	-5.7749	tau2= -2.88	Stationary
	16.6824	phi1= 4.63	No Unit Root
Tesla	-4.9872	tau2= -2,89	Stationary
	12.459	phi1= 4.71	No Unit Root

Appendix M

Table 20 - ADF Unit Root Test After Differentiation (type="trend")

	ADF Unit Root Test After Differentiation (type="trend")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-6.6541 14.8226 22.2044	tau 3= -3.45 phi2= 4.88 phi3= 6.49	Stationary No Unit Root
Amazon	-6.9145 15.9527 23.9179	tau 3= -3.43 phi2= 4.75 phi3= 6.49	Stationary No Unit Root
Apple	-6.3484 13.5261 20.2501	tau 3= -3.43 phi2= 4.75 phi3= 6.49	Stationary No Unit Root
Meta	-3.5872 4.321 6.4435	tau 3= -3.45 phi2= 4.88 phi3= 6.49	Stationary No Unit Root
Microsoft	-7.3916 18.218 27.3266	tau 3= -3.43 phi2= 4.75 phi3= 6.49	Stationary No Unit Root
Nvidia	-6.1449 12.5918 18.8799	tau 3= -3.43 phi2= 4.75 phi3= 6.49	Stationary No Unit Root
Tesla	-5.2103 9.0678 13.578	tau 3= -3.45 phi2= 4.88 phi3= 6.49	Stationary No Unit Root

Appendix N

Table 21 - PP Unit Root Test After Differentiation (lags="short")

	PP Unit Root Test After Differentiation (lags="short")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-6.3905	-3.465873	Stationary No Unit Root
Amazon	-9.6968	-3.451905	Stationary No Unit Root
Apple	-16.3067	-3.449402	Stationary No Unit Root
Meta	-4.7564	-3.491931	Stationary No Unit Root
Microsoft	-16.6832	-3.449402	Stationary No Unit Root
Nvidia	-7.9115	-3.453085	Stationary No Unit Root
Tesla	-4.9996	-3.490411	Stationary No Unit Root

Appendix O

Table 22 - PP Unit Root Test After Differentiation (lags="long")

	PP Unit Root Test After Differentiation (lags="long")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	-6.0725	-3.465873	Stationary No Unit Root
Amazon	-9.7929	-3.451905	Stationary No Unit Root
Apple	-21.9369	-3.449402	Stationary No Unit Root
Meta	-4.1476	-3.491931	Stationary No Unit Root
Microsoft	-23.8651	-3.449402	Stationary No Unit Root
Nvidia	-7.8718	-3.453085	Stationary No Unit Root
Tesla	-4.9187	-3.490411	Stationary No Unit Root

Appendix P

Table 23 - KPSS Unit Root Test After Differentiation (lags="short")

	KPSS Unit Root Test After Differentiation (lags="short")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	0.0451	0.146	Stationary
Amazon	0.2051	0.146	Non-stationary
Apple	0.0275	0.146	Stationary
Meta	0.0409	0.146	Stationary
Microsoft	0.0233	0.146	Stationary
Nvidia	0.196	0.146	Non-stationary
Tesla	0.0242	0.146	Stationary

Appendix Q

Table 24 - KPSS Unit Root Test After Differentiation (lags="long")

	KPSS Unit Root Test After Differentiation (lags="long")		
	Statistic	Critical Value (5%)	Conclusion
Alphabet	0.0964	0.146	Stationary
Amazon	0.1532	0.146	Non-stationary
Apple	0.0688	0.146	Stationary
Meta	0.1171	0.146	Stationary
Microsoft	0.0761	0.146	Stationary
Nvidia	0.1901	0.146	Non-stationary
Tesla	0.088	0.146	Stationary

Appendix R

Table 25 - HEGY Seasonal Unit Root Test After Differentiation

	HEGY Seasonal Unit Root Test After Differentiation			
	Seasonal Frequency	Statistic	p-value	Conclusion (5%)
Alphabet	0 (t_1)	-5.1546	0	No unit root
	π (t_2)	-4.6518	0	No unit root
	±π/2 ($F_{3:4}$)	26.7578	0	No unit root
	π & ±π/2 ($F_{2:4}$)	29.9117	0.2884	Possible unit root
	All ($F_{1:4}$)	27.026	0	No unit root
Amazon	0 (t_1)	-4.212	0.0259	No unit root
	π (t_2)	-6.8682	0	No unit root
	±π/2 ($F_{3:4}$)	38.0482	0	No unit root
	π & ±π/2 ($F_{2:4}$)	40.1006	0.032	No unit root
	All ($F_{1:4}$)	36.9063	0.0007	No unit root
Apple	0 (t_1)	-7.592	0	No unit root
	π (t_2)	-7.9963	0	No unit root
	±π/2 ($F_{3:4}$)	63.6228	0	No unit root
	π & ±π/2 ($F_{2:4}$)	59.0684	0.0005	No unit root
	All ($F_{1:4}$)	76.9699	0	No unit root
Meta	0 (t_1)	-3.3717	0	No unit root
	π (t_2)	-4.2078	0.0292	No unit root
	±π/2 ($F_{3:4}$)	9.191	0.0128	No unit root
	π & ±π/2 ($F_{2:4}$)	14.2207	0.0156	No unit root
	All ($F_{1:4}$)	21.4143	0.0185	No unit root
Microsoft	0 (t_1)	-7.9890	0	No unit root
	π (t_2)	-8.4434	0	No unit root
	±π/2 ($F_{3:4}$)	69.4430	0	No unit root
	π & ±π/2 ($F_{2:4}$)	66.1780	0.0002	No unit root
	All ($F_{1:4}$)	87.7503	0	No unit root
Nvidia	0 (t_1)	-5.2607	0.0054	No unit root
	π (t_2)	-5.1588	0.0057	No unit root
	±π/2 ($F_{3:4}$)	31.4100	0	No unit root
	π & ±π/2 ($F_{2:4}$)	35.2653	0.0717	Possible unit root
	All ($F_{1:4}$)	32.2632	0.0164	No unit root
Tesla	0 (t_1)	-5.2643	0	No unit root
	π (t_2)	-2.8721	0	No unit root
	±π/2 ($F_{3:4}$)	24.0937	0	No unit root
	π & ±π/2 ($F_{2:4}$)	21.8843	0.0955	No unit root
	All ($F_{1:4}$)	23.345	0.0092	No unit root

Appendix S

Table 26 - P-values of Box-Ljung test and Box-Pierce test

	P-values of the Serial correlation tests		
	Box-Ljung test	Box-Pierce test	Conclusion (5%)
Alphabet	0.3006	0.505	No evidence of serial correlation
Amazon	0.8221	0.8974	No evidence of serial correlation
Apple	0.9991	0.9996	No evidence of serial correlation
Meta	0.7458	0.9279	No evidence of serial correlation
Microsoft	0.4795	0.6343	No evidence of serial correlation
Nvidia	0.7959	0.8884	No evidence of serial correlation
Tesla	0.9896	0.9991	No evidence of serial correlation

Appendix T

Table 27 - Prophet Evaluation Criteria with Holidays

	Prophet Evaluation Criteria with Holidays		
	MSE	RMSE	MAPE
Alphabet	126856437	11263,06	14,41
Amazon	80257223	8958,64	6,16
Apple	139788727	11823,23	9,25
Meta	6509103,1	2551,29	6,43
Microsoft	65300636	8080,88	10,46
Nvidia	133107074	11537,20	33,51
Tesla	13436239	3665,55	13,76

Appendix U

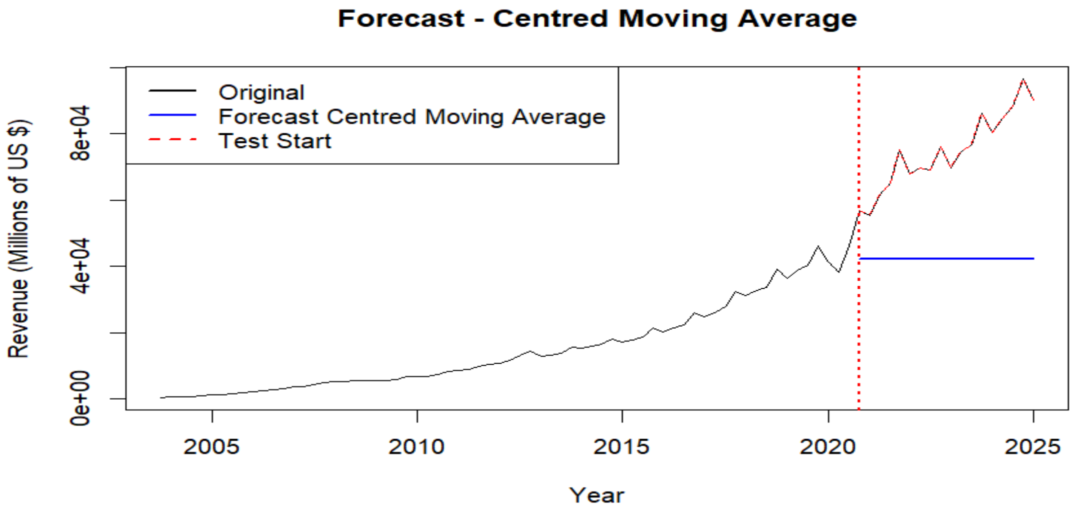

Figure 3 - Alphabet Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix V

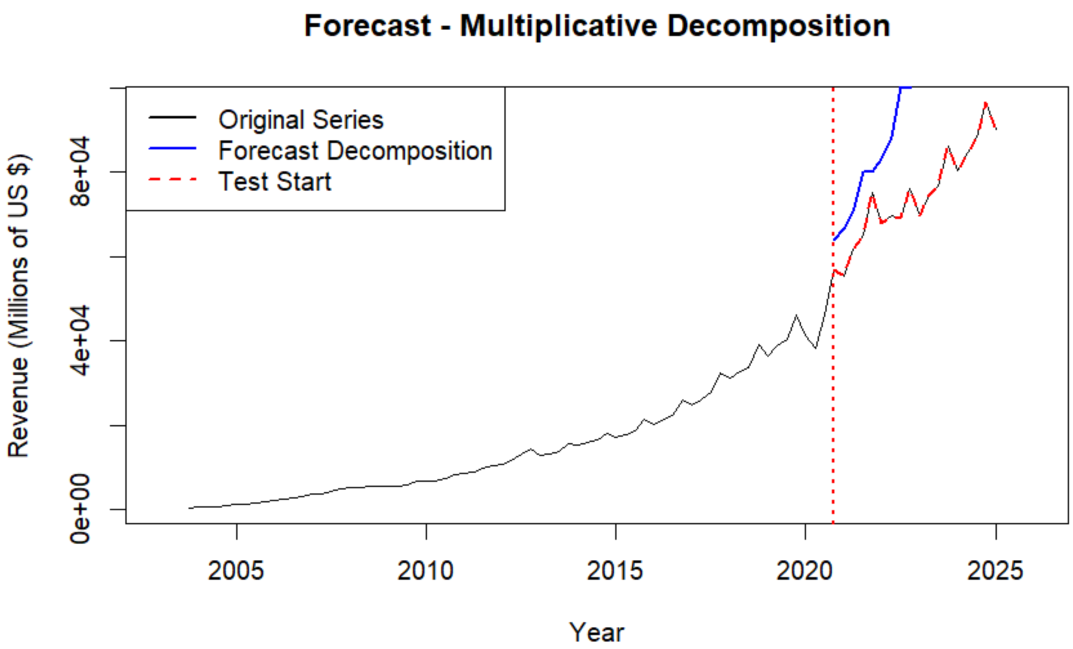

Figure 4 - Alphabet Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix W

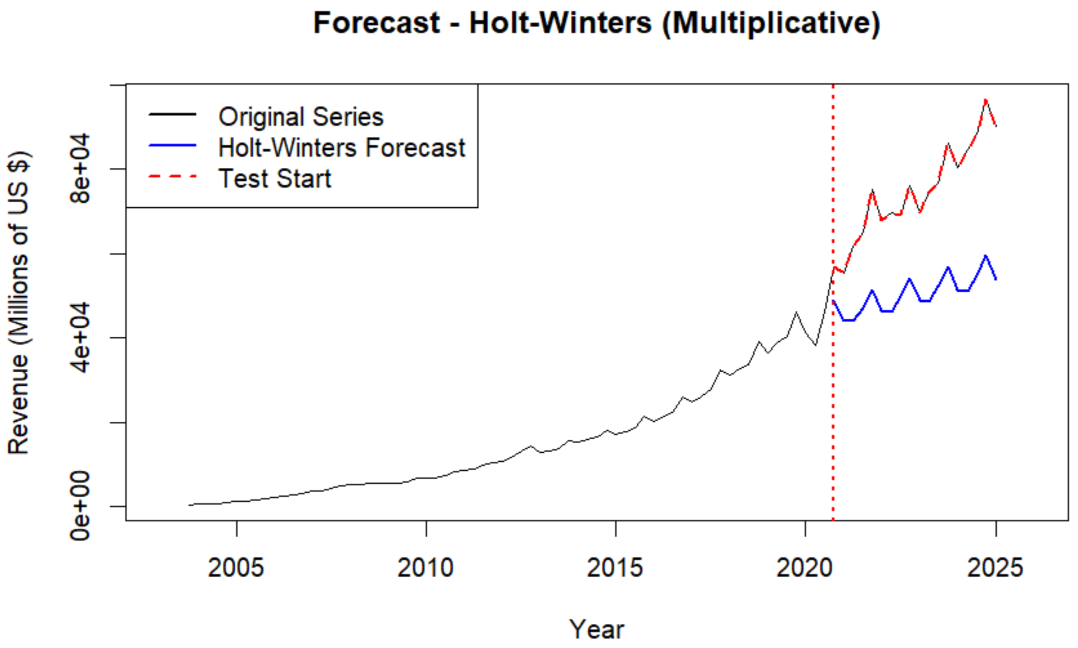

Figure 5 - Alphabet Exponential Holt-Winters Test Set Forecast Source: Own Source via R Studio

Appendix X

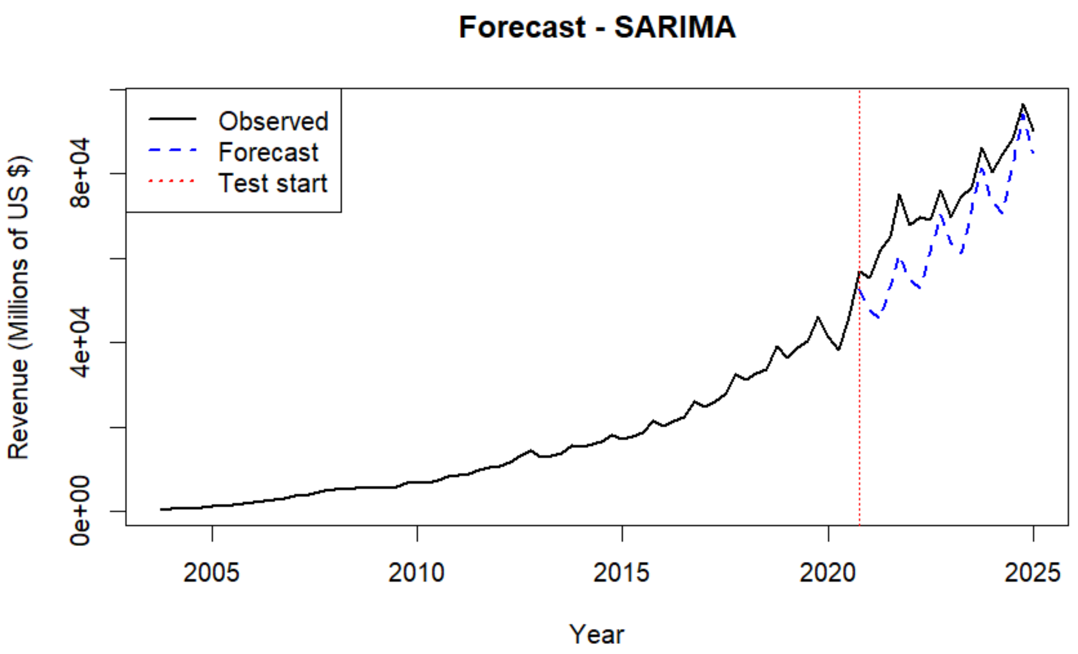

Figure 6 - Alphabet SARIMA Test Set Forecast Source: Own Source via R Studio

Appendix Y

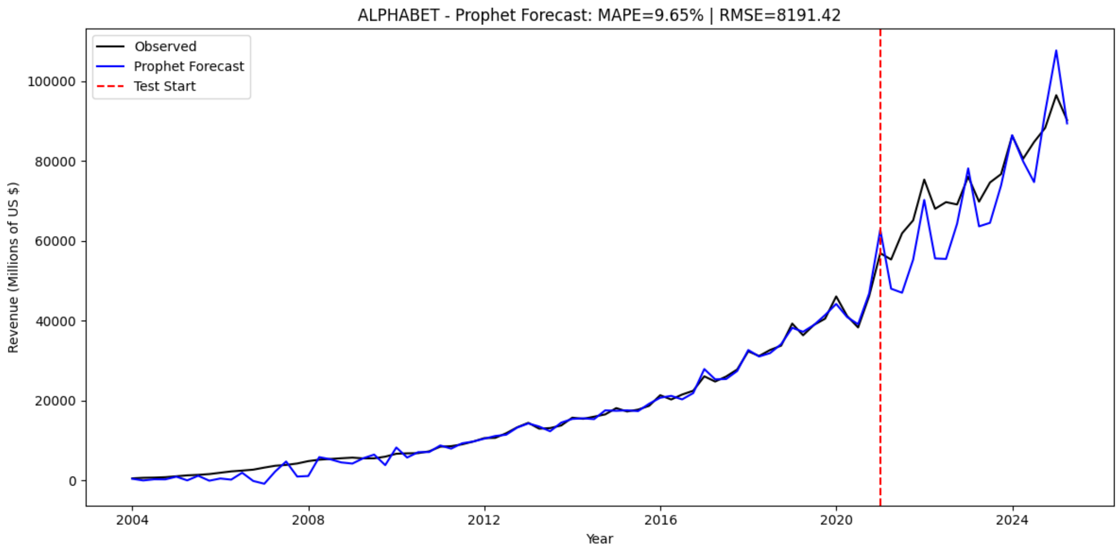

Figure 7 - Alphabet Prophet Forecast Source: Own Source via Python

Appendix Z

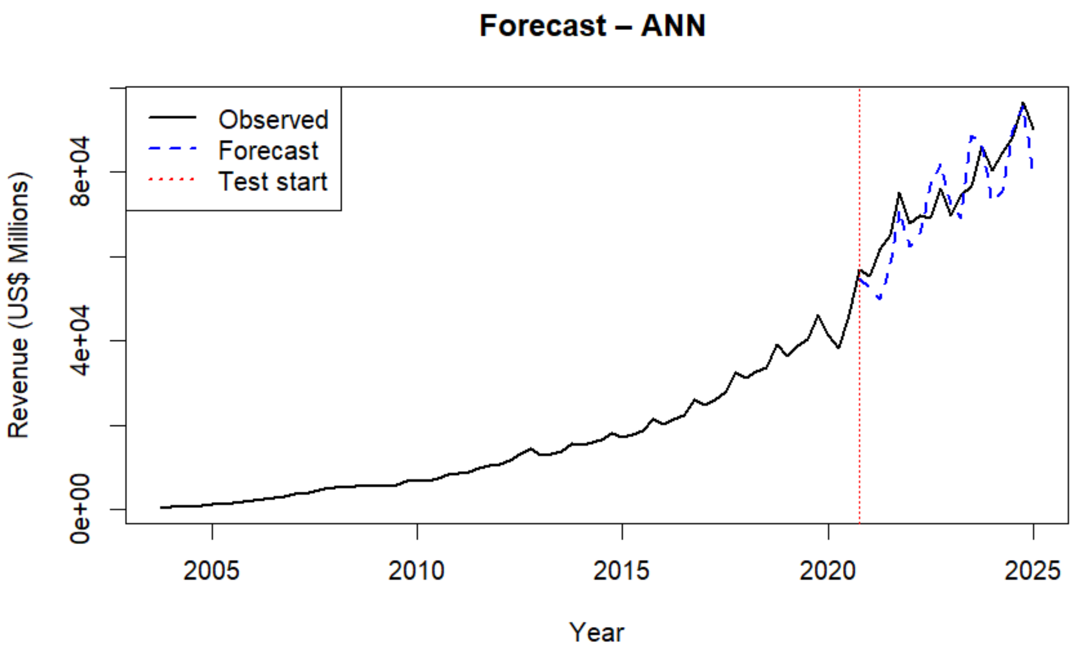

Figure 8 - Alphabet ANN Test Set Forecast Source: Own Source via R Studio

Appendix AA

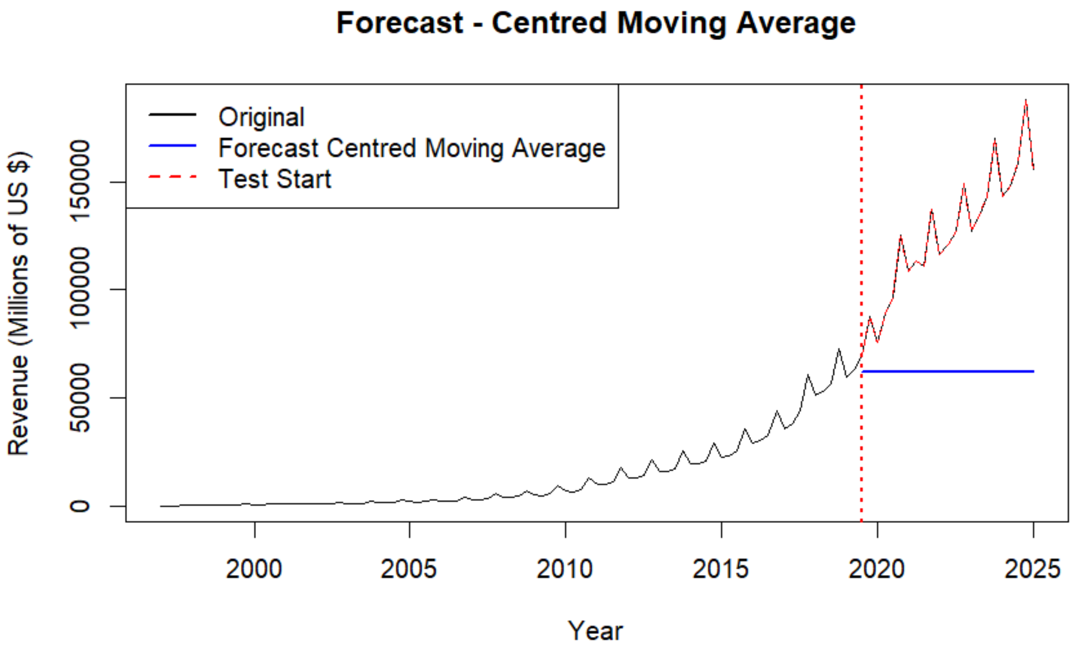

Figure 9 - Amazon Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix AB

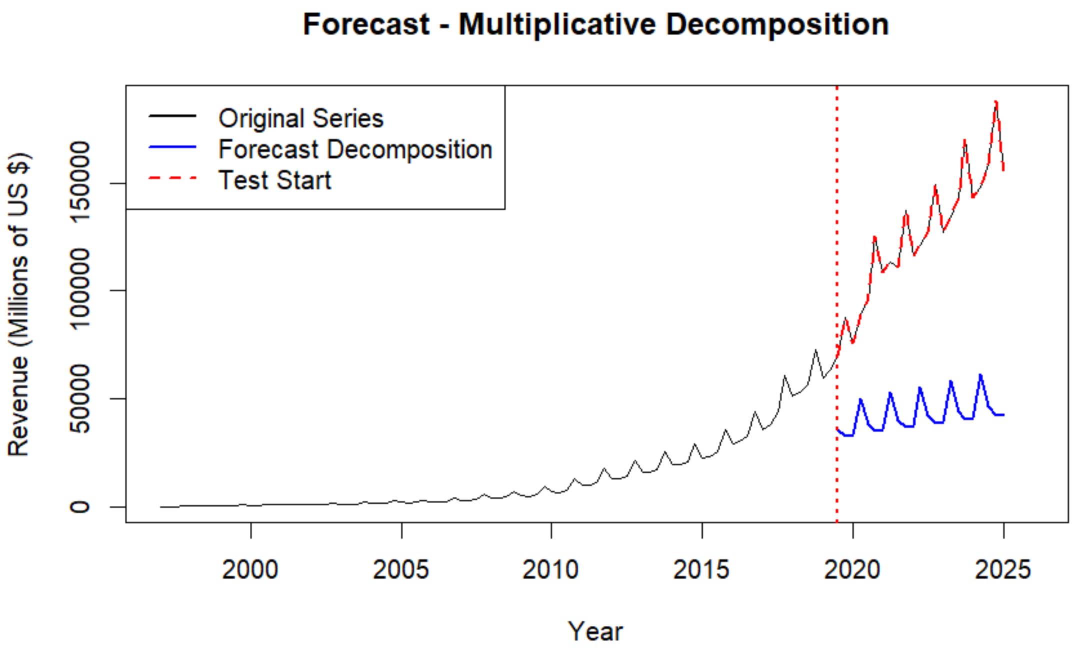

Figure 10 - Amazon Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix AC

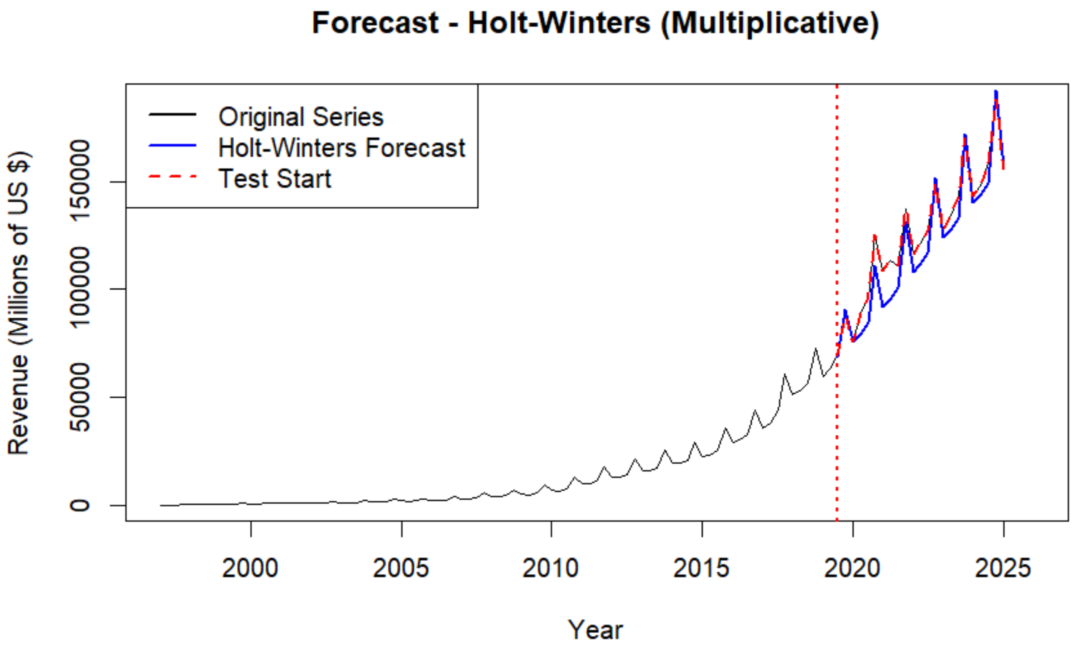

Figure 11 - Amazon Exponential Holt-Winters Test Set Forecast Source: Own Source via R Studio

Appendix AD

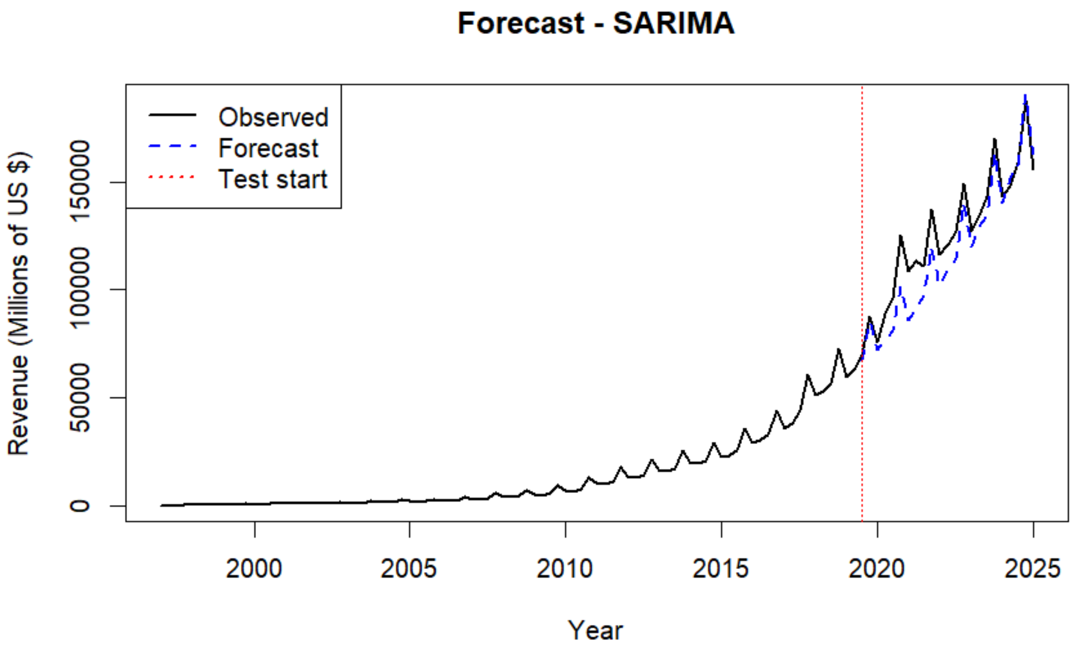

Figure 12 – Amazon SARIMA Test Set Forecast Source: Own Source via R Studio

Appendix AE

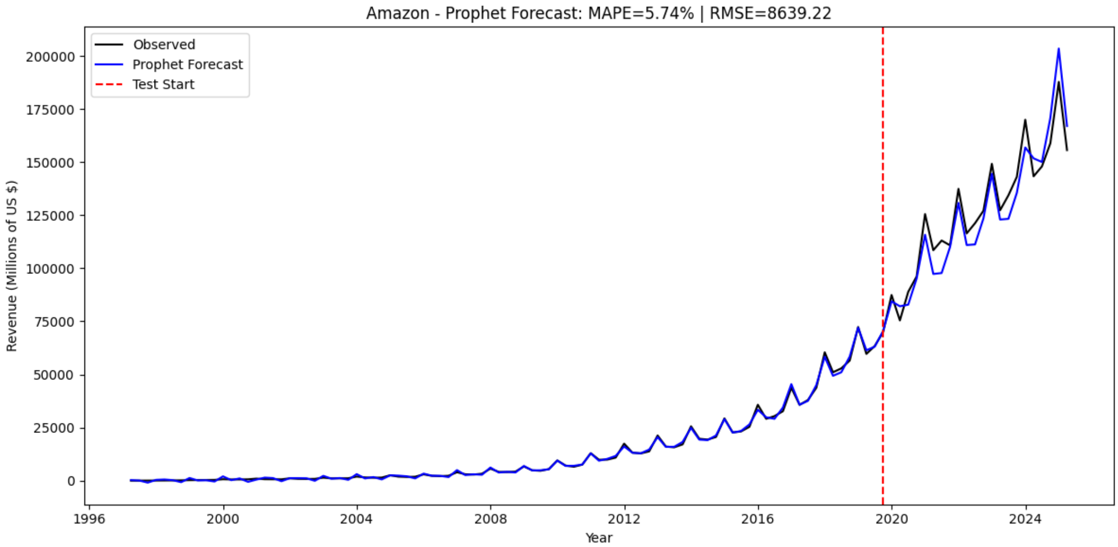

Figure 13 - Amazon Prophet Forecast Source: Own Source via Python

Appendix AF

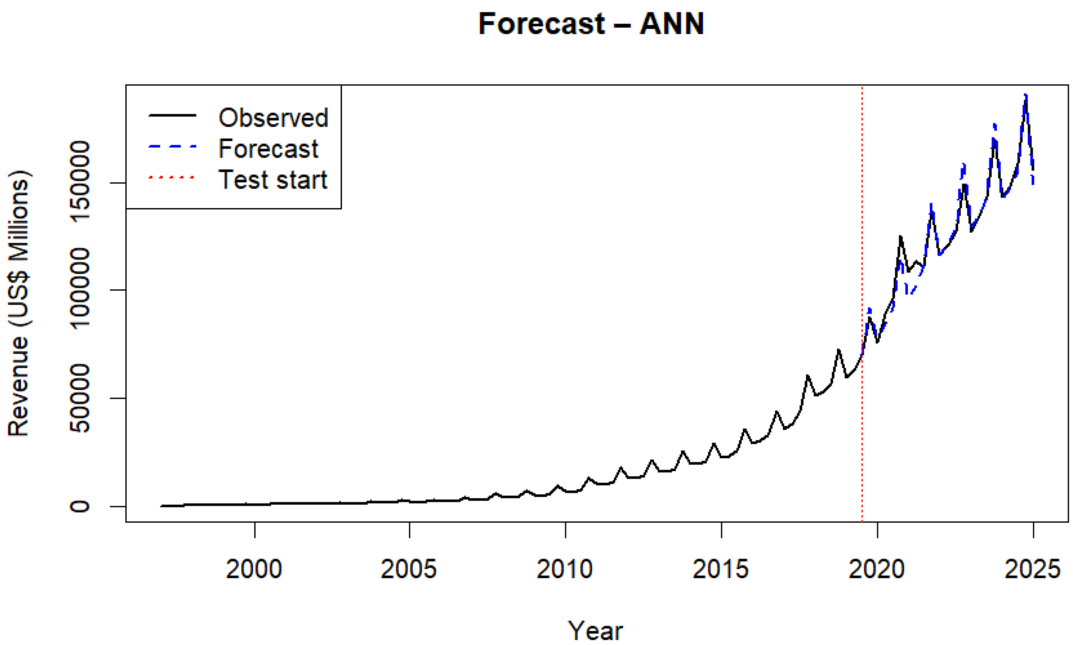

Figure 14 - Amazon ANN Test Set Forecast Source: Own Source via R Studio

Appendix AG

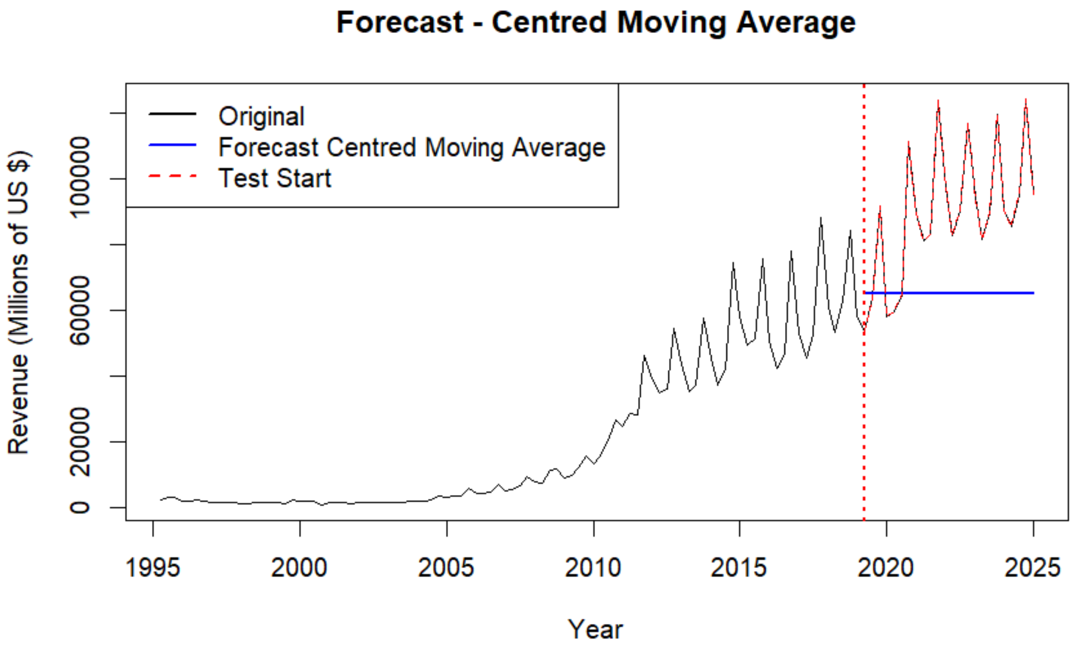

Figure 15 - Apple Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix AH

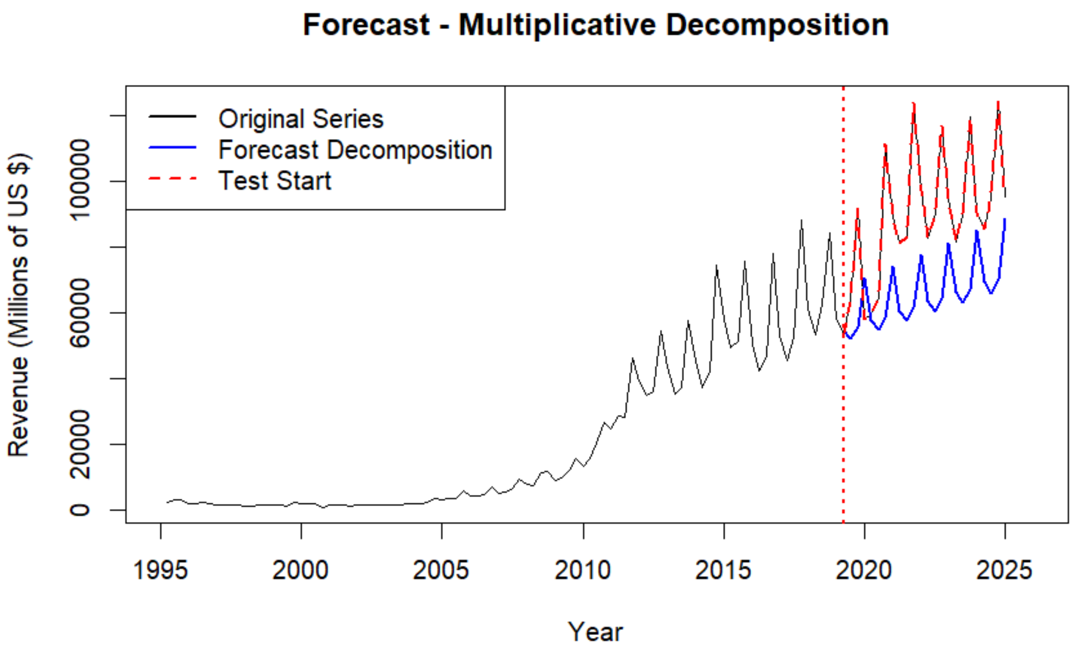

Figure 16 - Apple Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix AI

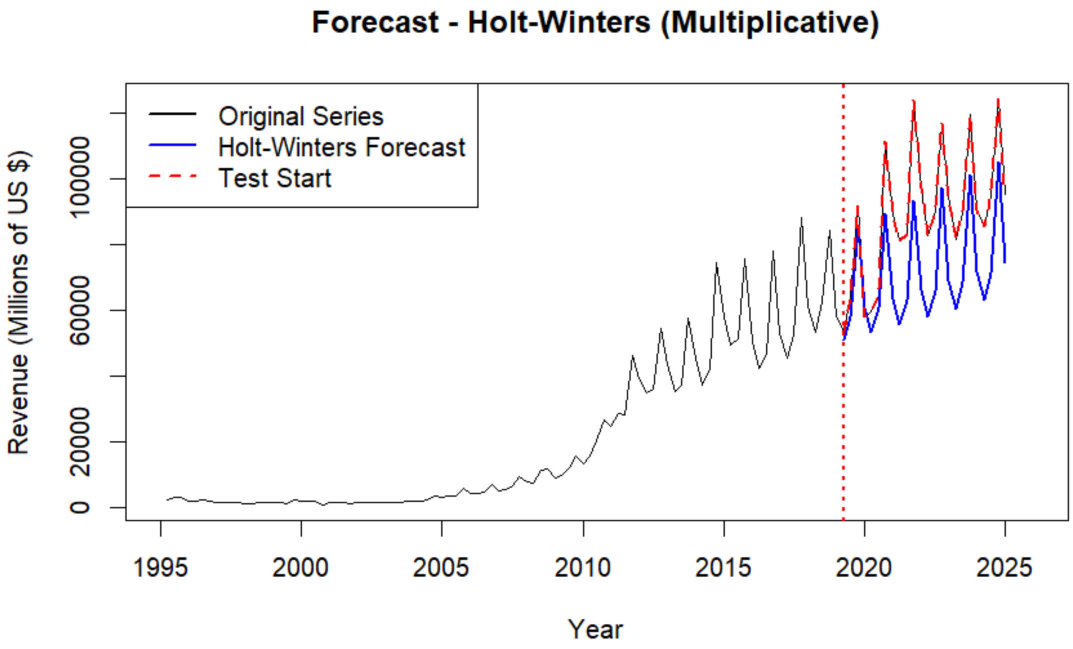

Figure 17 - Apple Exponential Holt-Winters Test Set Forecast Source: Own Source via R Studio

Appendix AJ

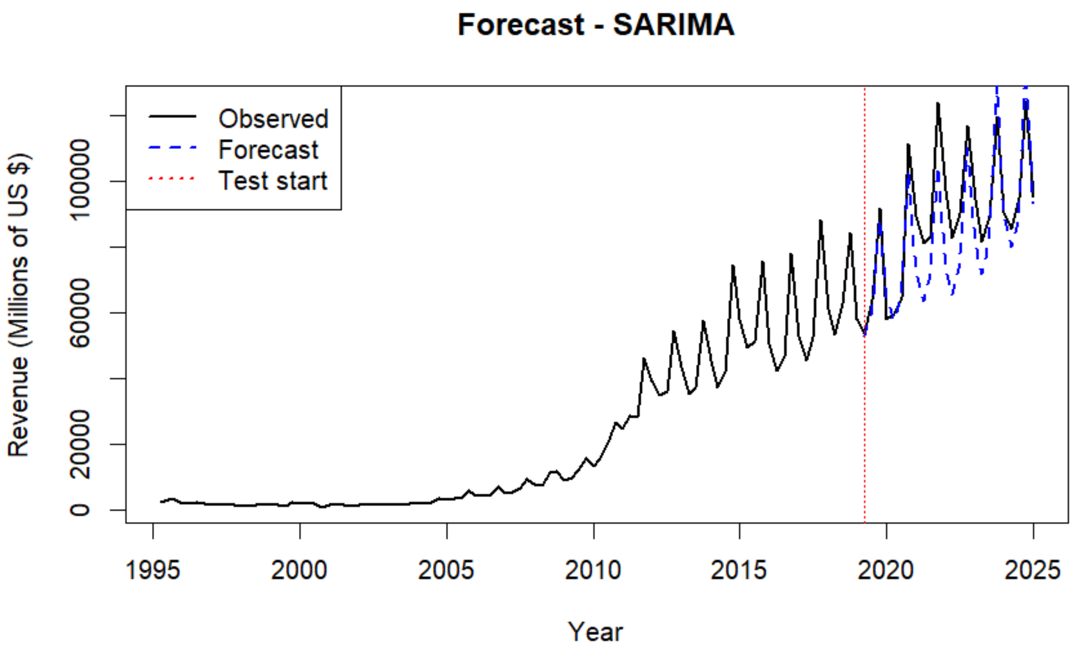

Figure 18 - Apple SARIMA Test Set Forecast Source: Own Source via R Studio

Appendix AK

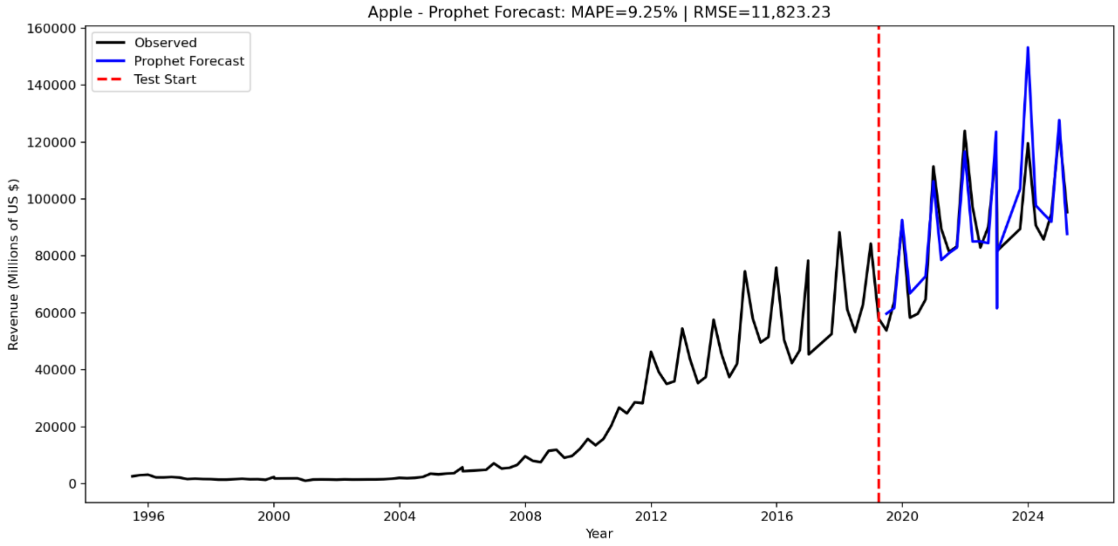

Figure 19 - Apple Prophet Forecast Source: Own Source via Python

Appendix AL

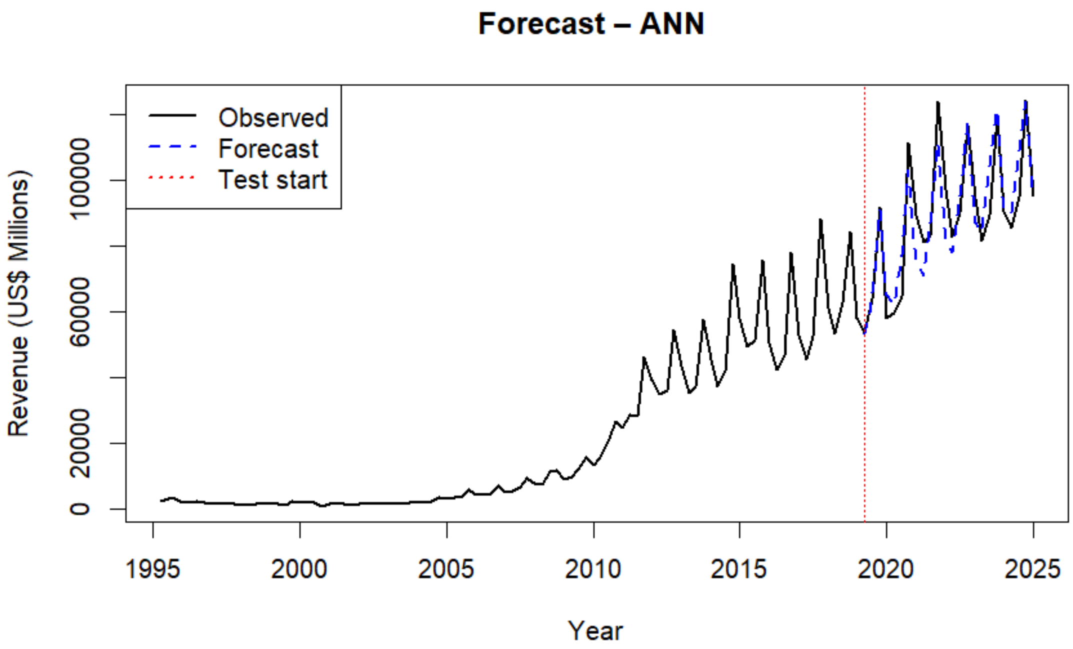

Figure 20 - Apple ANN Test Set Forecast Source: Own Source via R Studio

Appendix AM

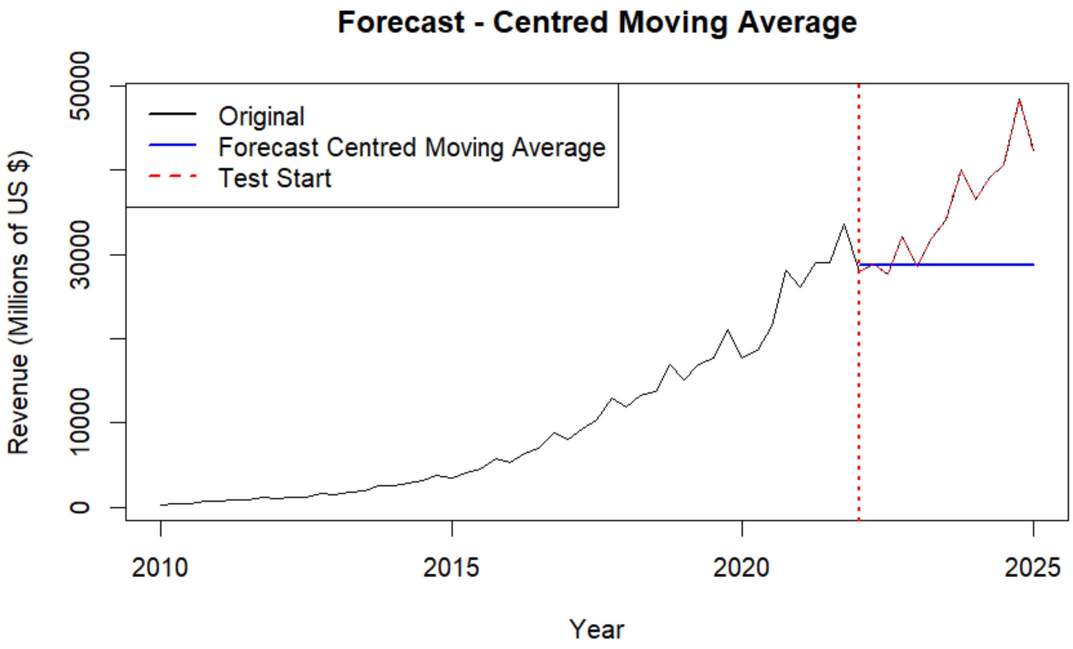

Figure 21 – Meta Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix AN

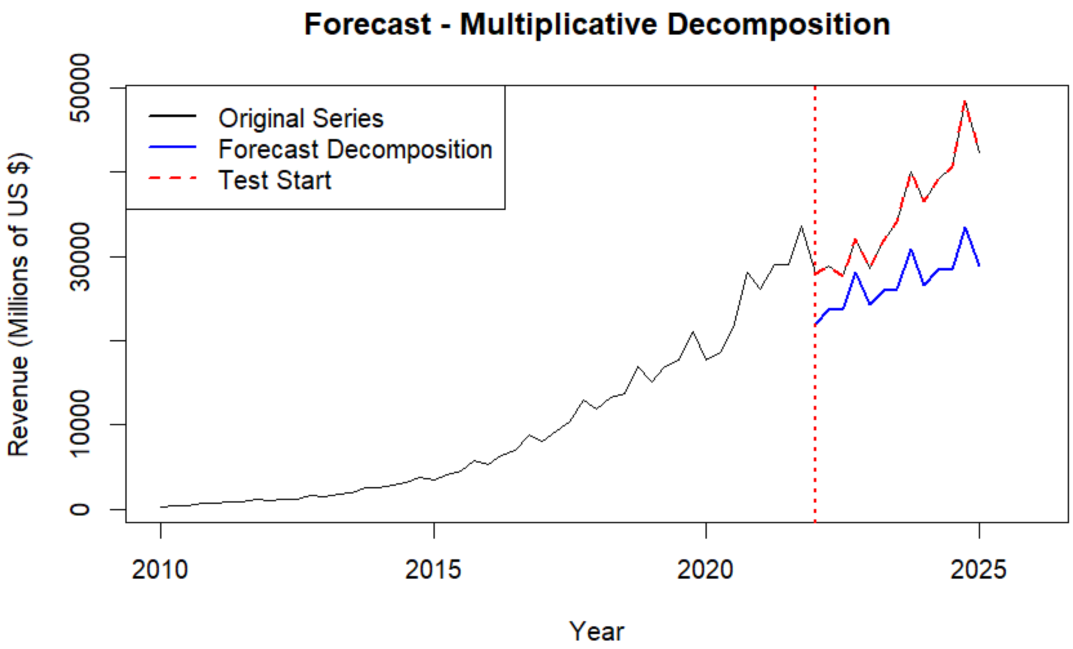

Figure 22 - Meta Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix AO

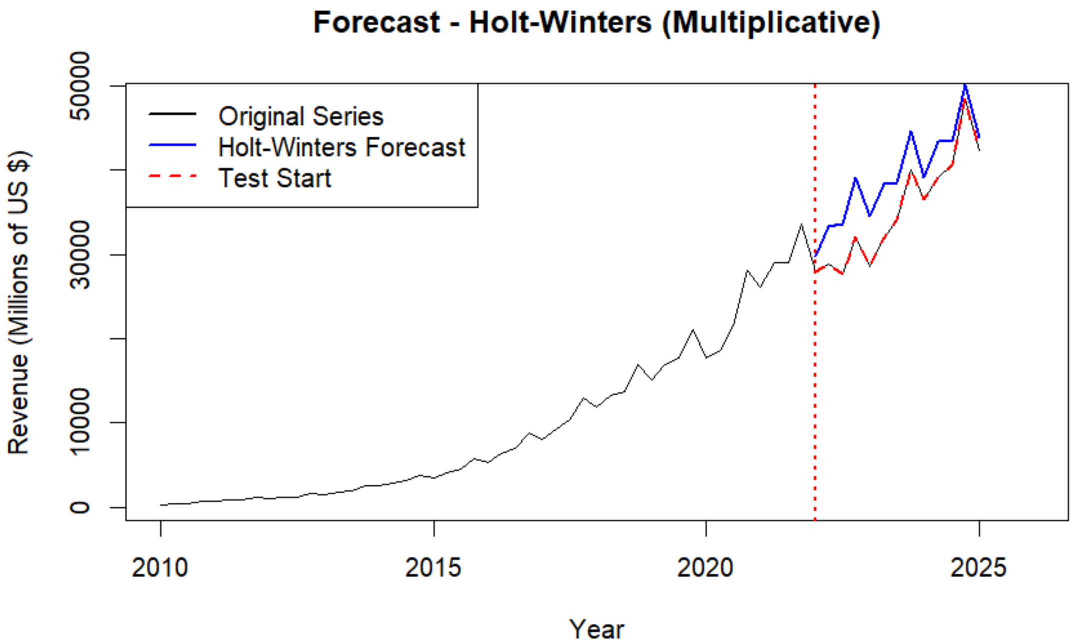

Figure 23 - Meta Exponential Holt-Winters Test Set Forecast Source: Own Source via R Studio

Appendix AP

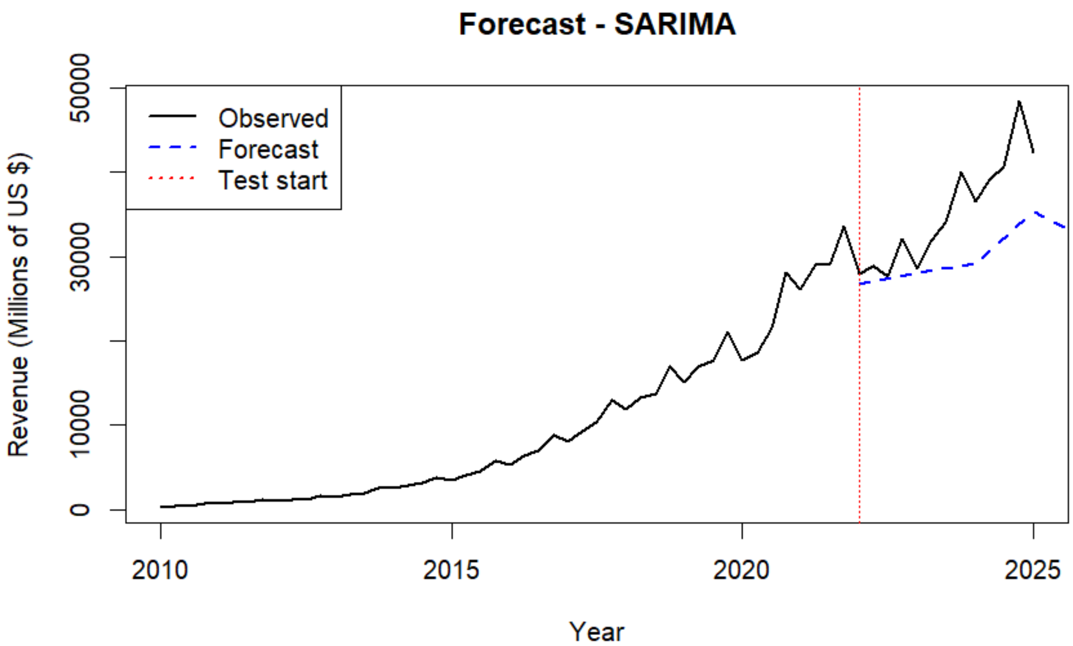

Figure 24 - Meta SARIMA Test Set Forecast Source: Own Source via R Studio

Appendix AQ

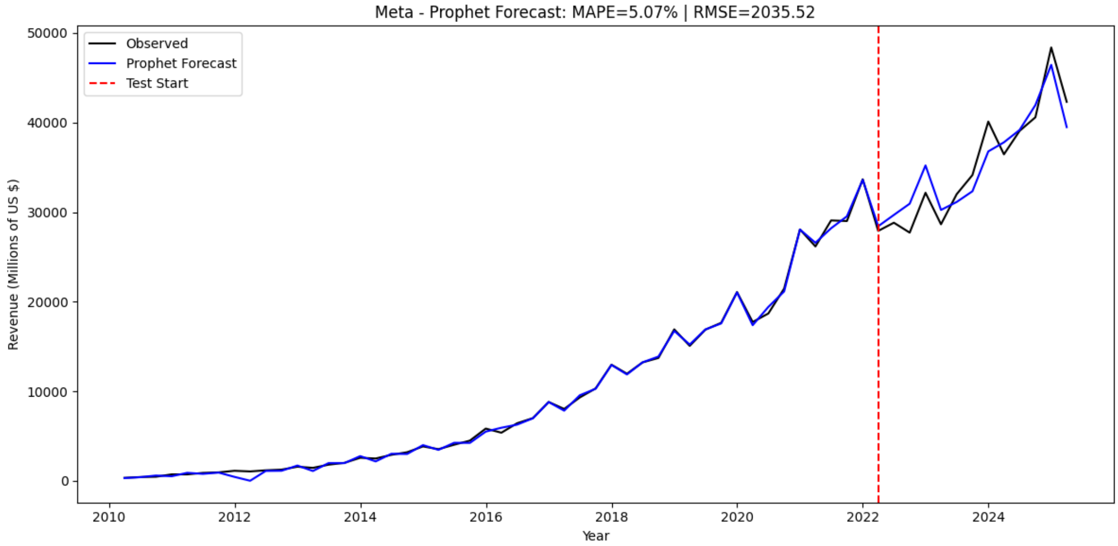

Figure 25 - Meta Prophet Forecast Source: Own Source via Python

Appendix AR

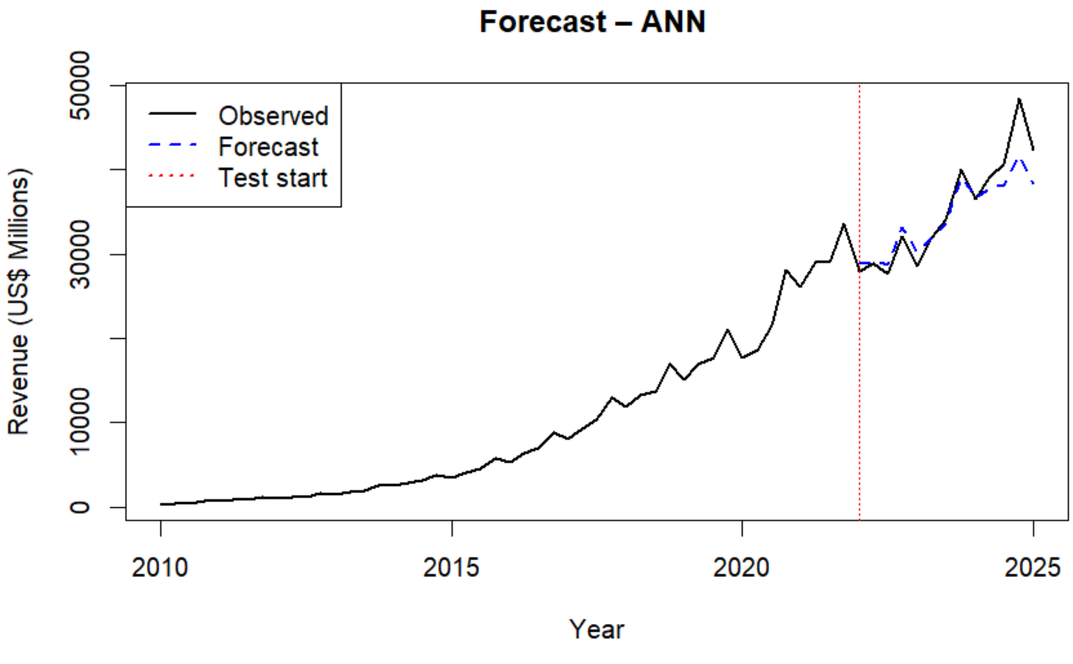

Figure 26 - Meta ANN Test Set Forecast Source: Own Source via R Studio

Appendix AS

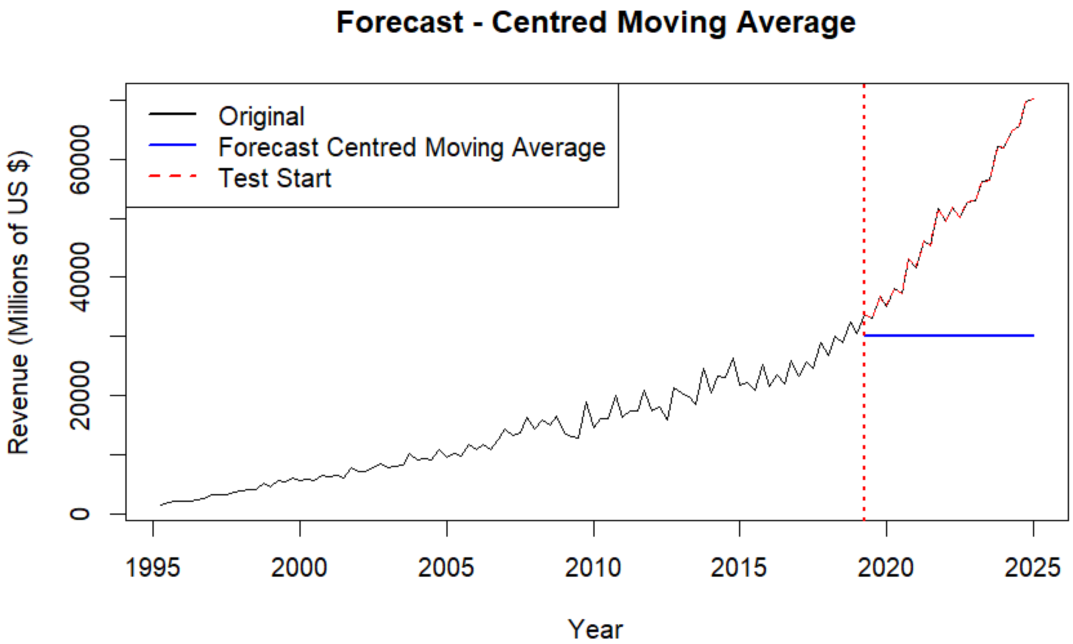

Figure 27 - Microsoft Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix AT

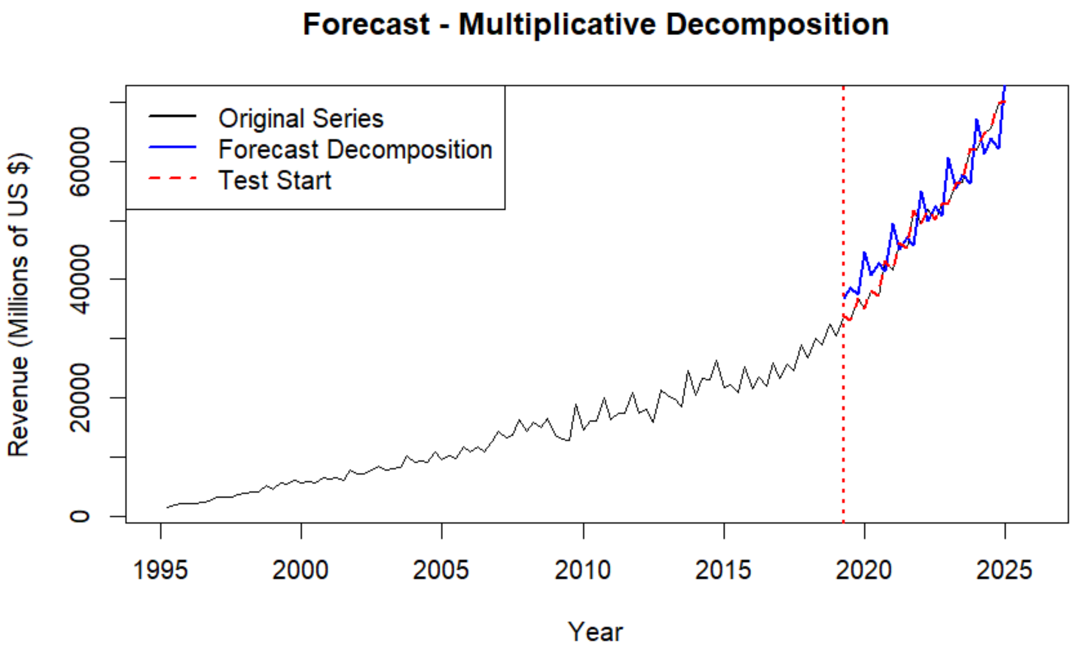

Figure 28 - Microsoft Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix AU

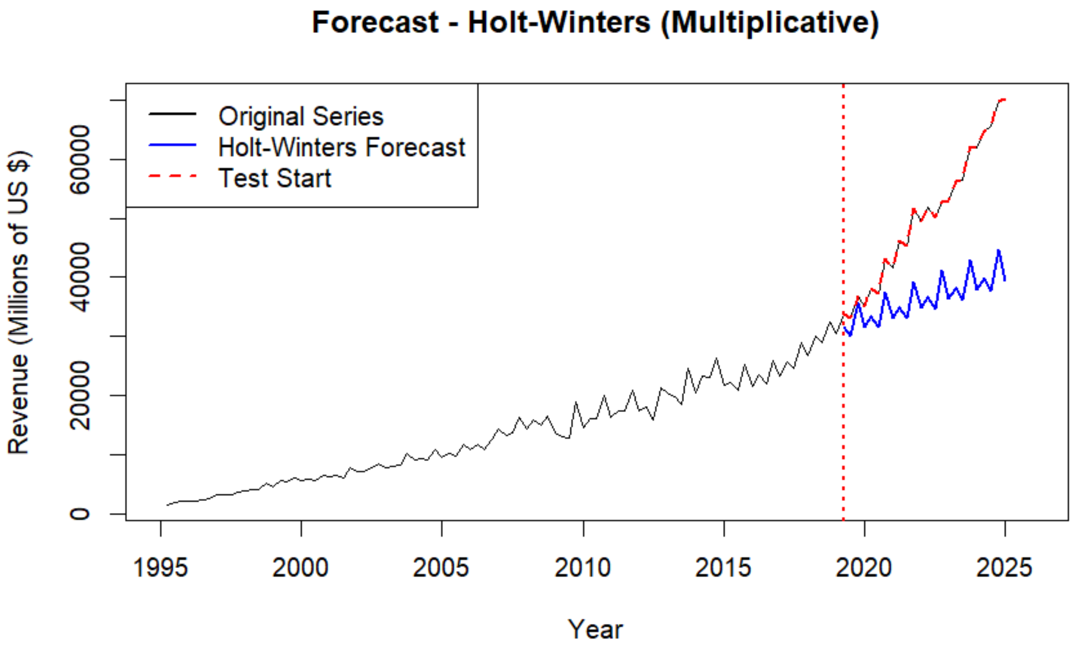

Figure 29 - Microsoft Exponential Holt-Winters Test Set Forecast Source: Own Source via R Studio

Appendix AV

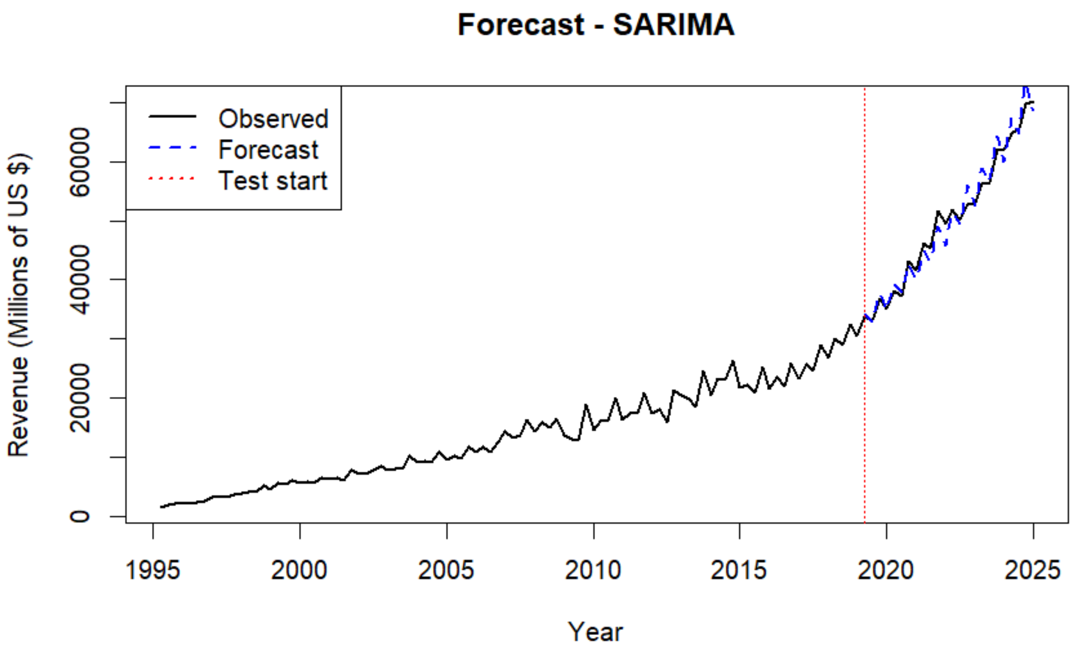

Figure 30 - Microsoft SARIMA Test Set Forecast Source: Own Source via R Studio

Appendix AW

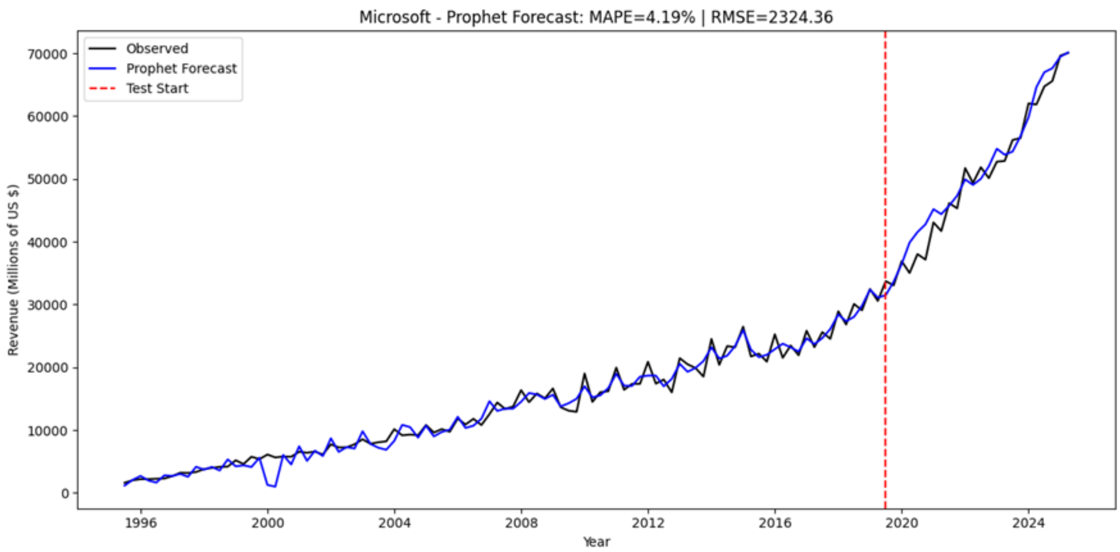

Figure 31 - Microsoft Prophet Forecast Source: Own Source via Python

Appendix AX

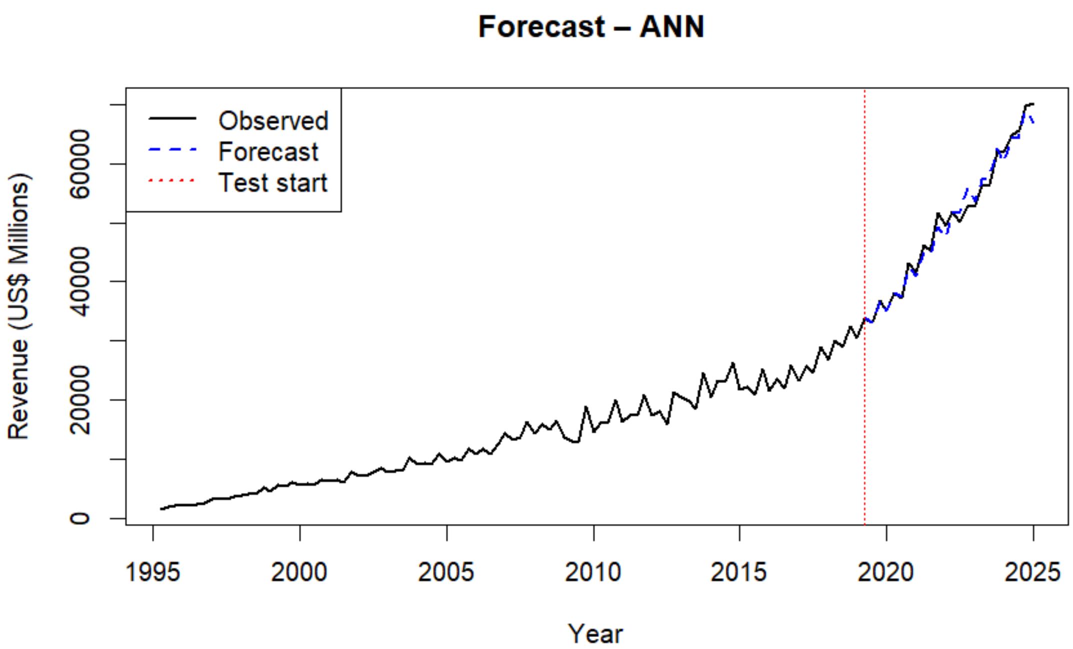

Figure 32 - Microsoft ANN Test Set Forecast Source: Own Source via R Studio

Appendix AY

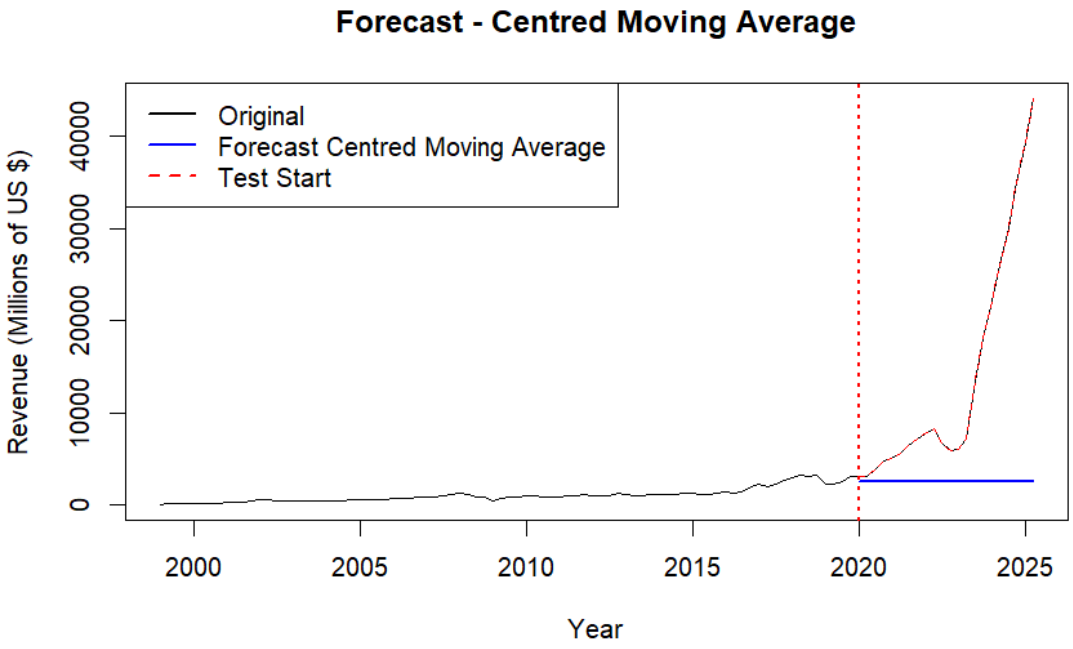

Figure 33 - Nvidia Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix AZ

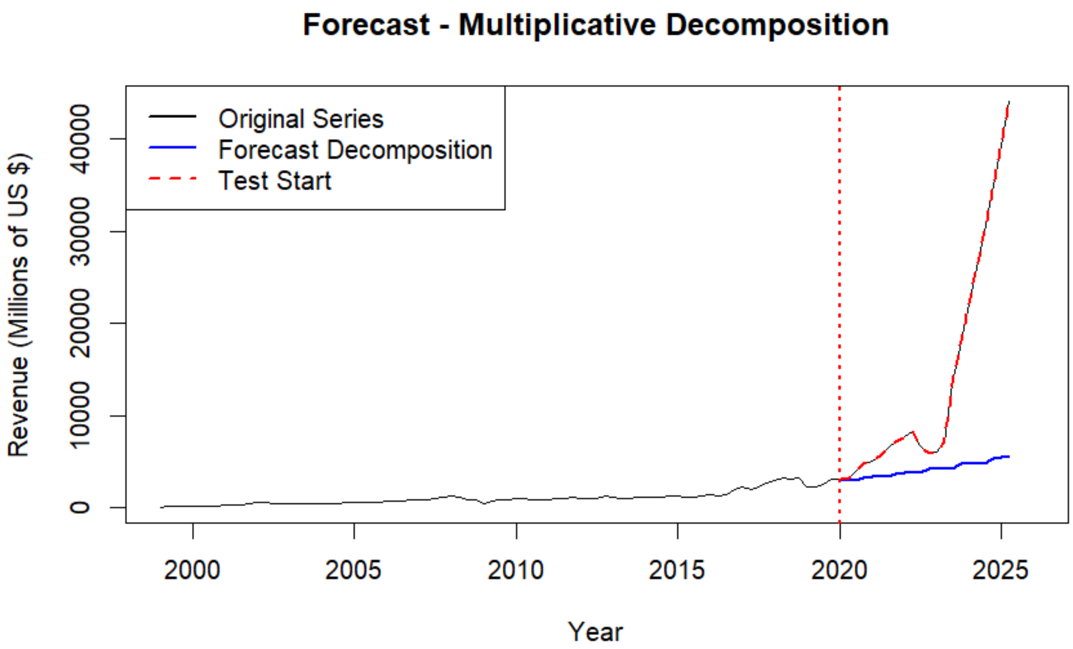

Figure 34 - Nvidia Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix BA

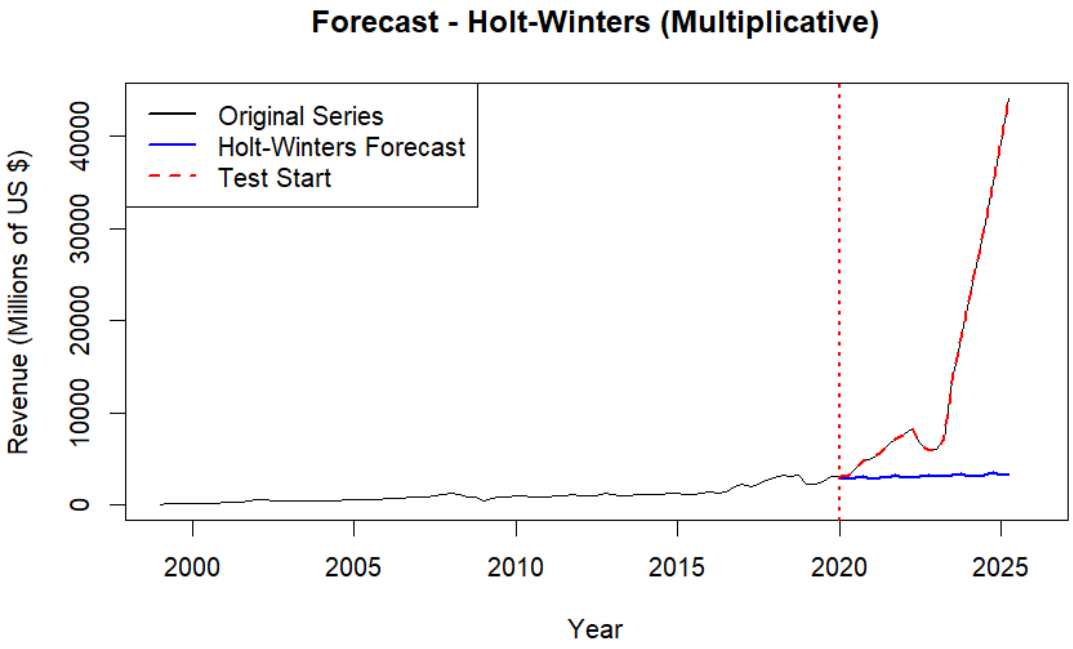

Figure 35 - Nvidia Exponential Holt-Winters Test Set Forecast Source: Own Source via R Studio

Appendix BB

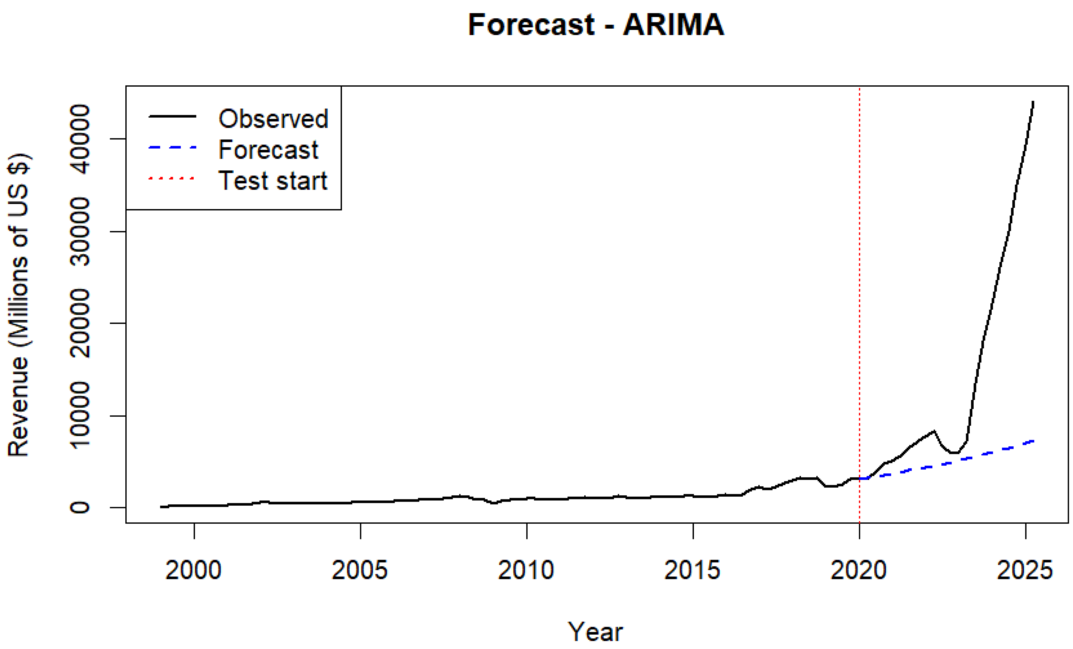

Figure 36 - Nvidia ARIMA Test Set Forecast Source: Own Source via R Studio

Appendix BC

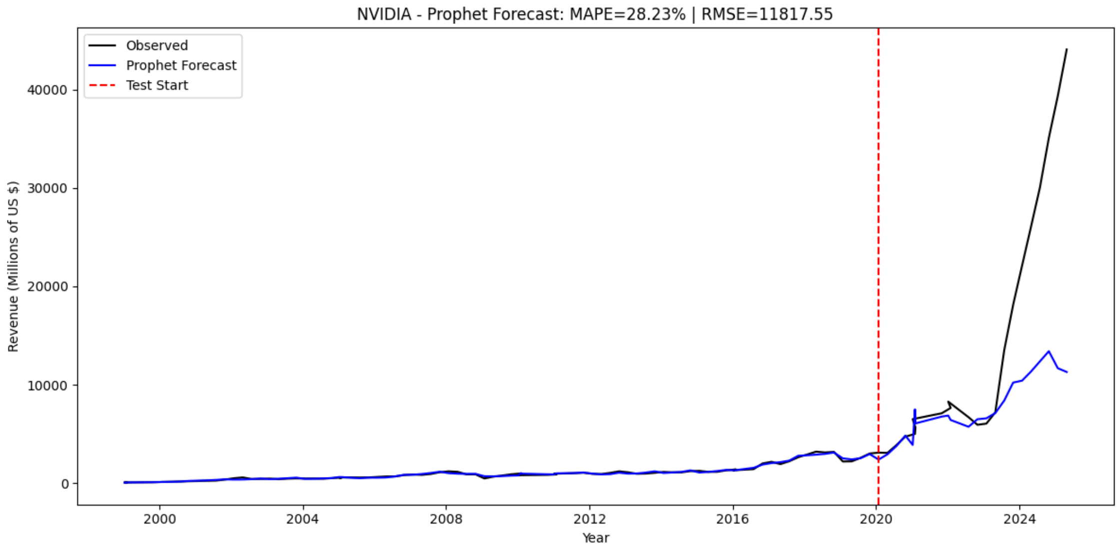

Figure 37 - Nvidia Prophet Forecast Source: Own Source via Python

Appendix BD

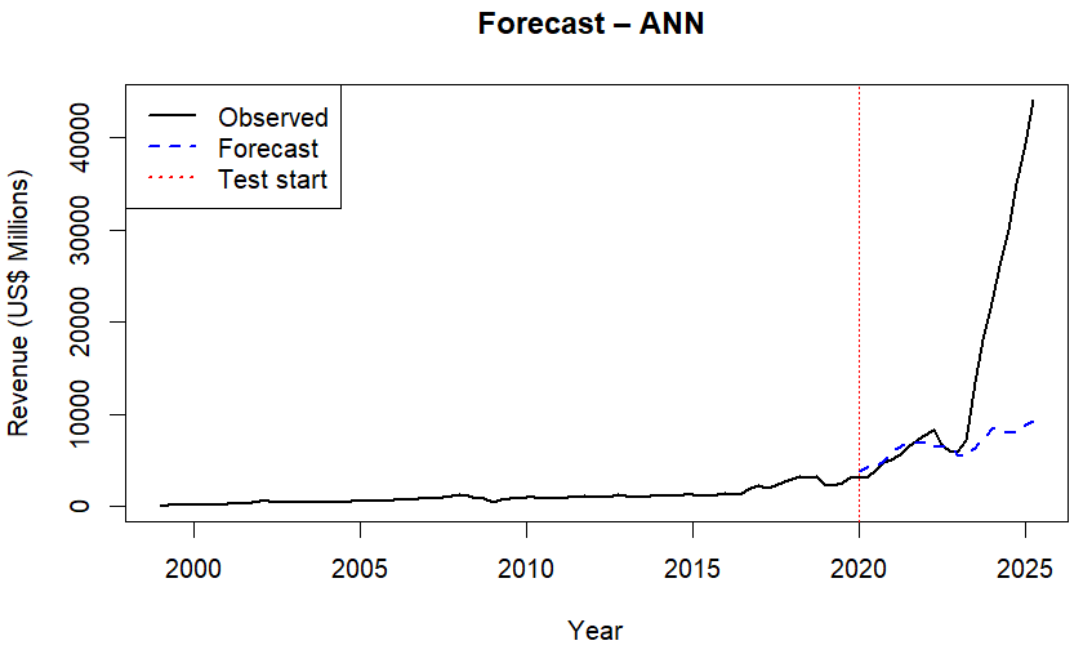

Figure 38 - Nvidia ANN Test Set Forecast Source: Own Source via R Studio

Appendix BE

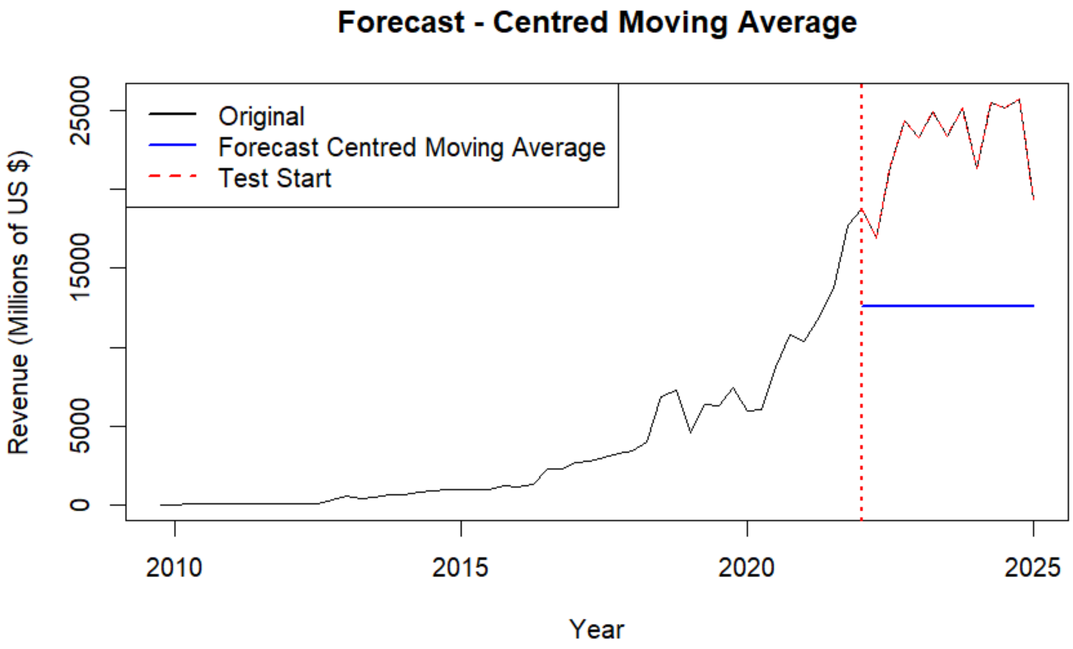

Figure 39 - Tesla Simple Moving Average Test Set Forecast Source: Own Source via R Studio

Appendix BF

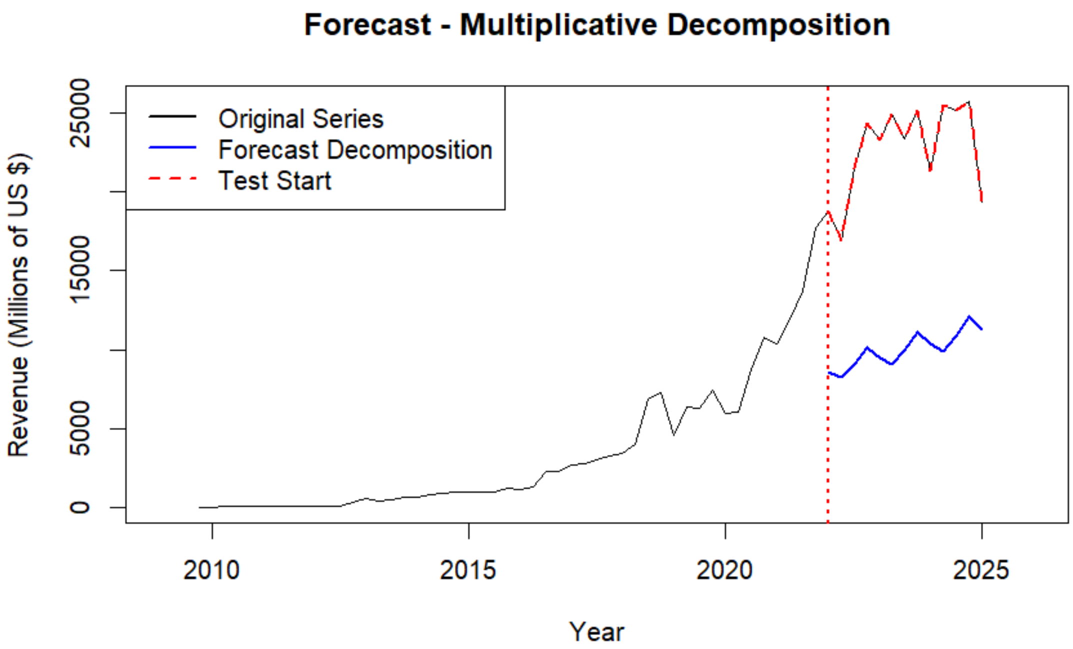

Figure 40 - Tesla Multiplicative Decomposition Test Set Forecast Source: Own Source via R Studio

Appendix BG

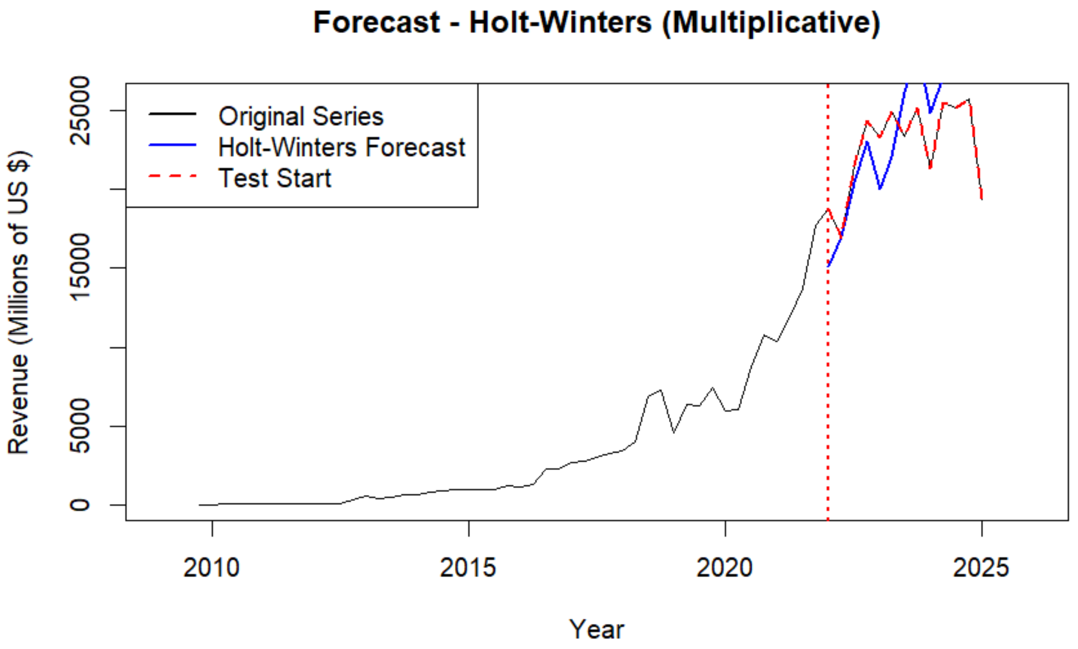

Figure 41 - Tesla Exponential Holt-Winters Test Set Forecast Own Source: Own Source via R Studio

Appendix BH

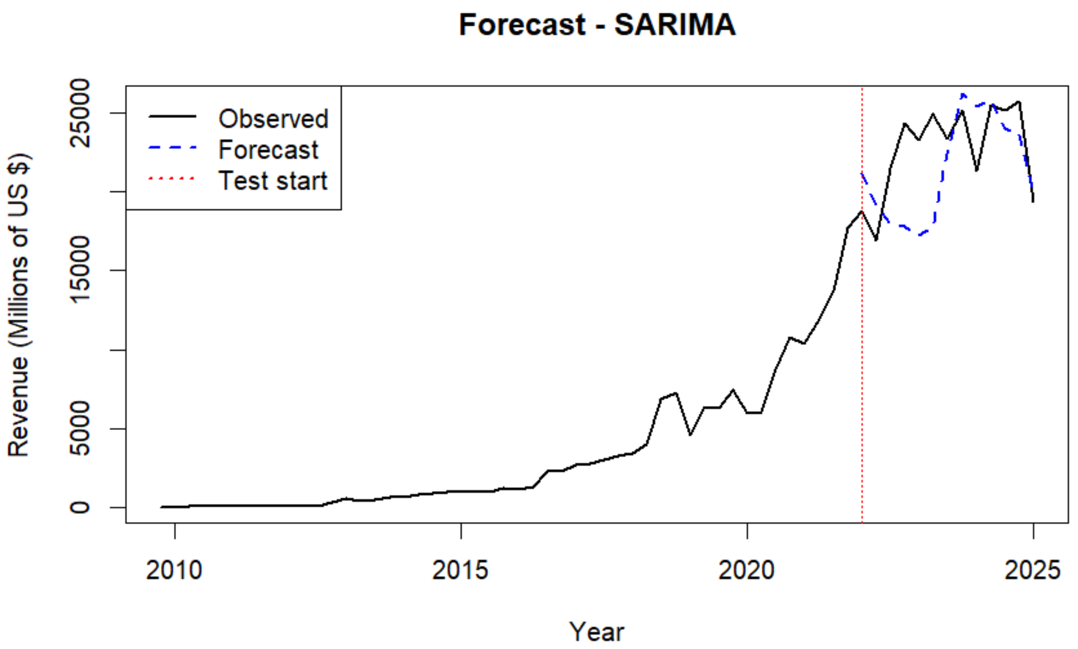

Figure 42 - Tesla SARIMA Test Set Forecast Source: Own Source via R Studio

Appendix BI

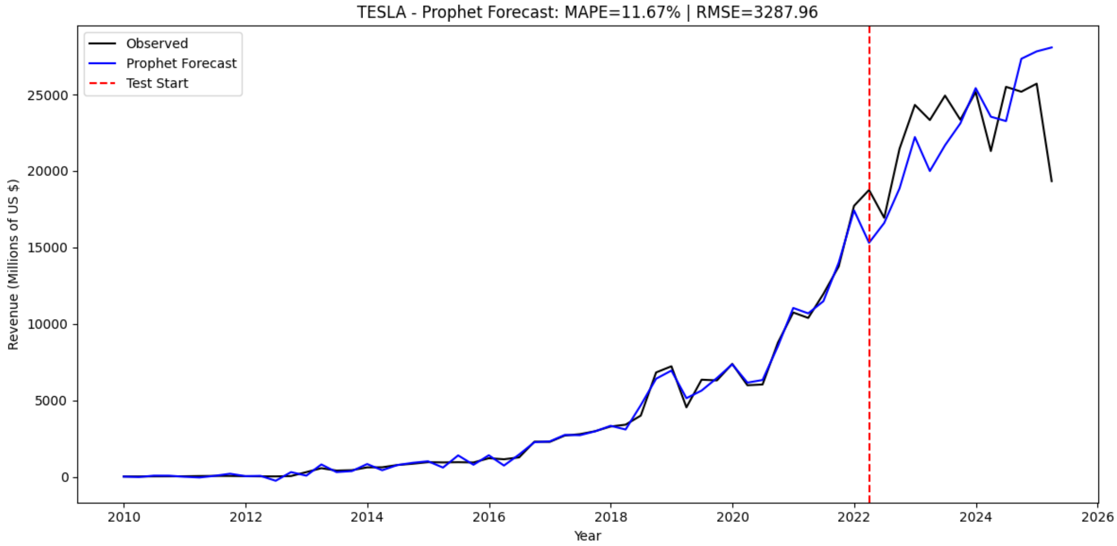

Figure 43 - Tesla Prophet Forecast Source: Own Source via Python

Appendix BJ

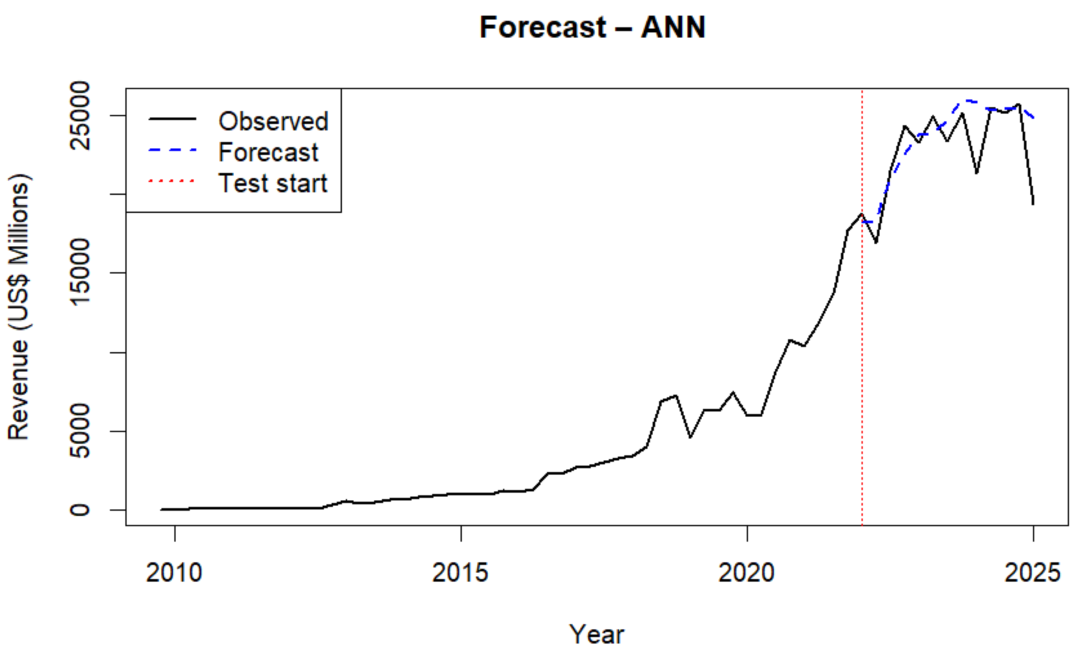

Figure 44 - Tesla ANN Test Set Forecast Source: Own Source via R Studio