Application and Comparison of Traditional Time-Series Models to Machine Learning Models for Stock Price Forecasting

Candidate no: 215939

BSc. Computer Science & Artificial Intelligence
Department of Informatics

2021
Supervisor: Dr Adam B. Barrett

Statement of Originality

This report is submitted as part requirement for the degree of Computer Science & Artificial Intelligence at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Acknowledgments

I would like to express my gratitude to Dr. Adam Barrett for his invaluable guidance and inspiration throughout this project. I would also like to thank my family for their unwavering support. Your hopes and prayers for me is what keeps me going.

Everything everyone knows, they learnt. Therefore, I can learn too. -Vusi Thembekwayo

Professional Considerations

This project serves as an investigation and does not possess any ethical consequences. In order to ensure compliance, the rules stated in the BCS Code of Conduct were abided by from the onset and throughout the study.

Section 1 — Public Interest

In compliance with section 1 of the BCS code of Conduct,

This project requires the use of securities data which is largely available to the public. In accordance with the rules, I will ensure the data used in this study is obtained from a credible source and is complete and untampered with. The work done does not intend to portray any individual or organization referred to in this report negatively.

Section 2 — Professional Competence and Integrity

In compliance with section 2 of the BCS code of Conduct,

The significance of this project is to better understand the efficiency of forecasting techniques for predicting stock prices. All of the work produced was done to the best of my ability and the opinions of the researchers cited and other parties involved. I also value feedback on any oversights observed.

Section 3 — Duty to Relevant Authority

In compliance with section 3 of the BCS code of Conduct,

I ensured no false information was provided, nor was this project made for personal gains. I have reported my findings to the best of my knowledge.

Abstract

Time series forecasting using computational intelligence techniques has become widely popular recently due to the continuous advancements in methods and the growing opportunity for use cases. Individuals and organisations aim to use these tools to attain a better understanding of their data, plan for the future, and gain an edge over competitors. One of the use cases is in the field of finance and economics. With the rise of machine learning techniques, industry professionals believe structural models could compete with traditionally used statistical (econometrics) methods for forecasting. It is often debated how accurate the forecasts of these models could be, especially in unpredictable areas such as equity price forecasting. Market analysts believe it is close to impossible to predict the stock market following the strong opinion of the Efficient Market Hypothesis (EMH). However, a number of previous successful works have proven the efficacy of both machine learning and traditional time series models for predicting market trends in multiple circumstances. They are claimed to serve as an effective tool for hedging against market risk and for making effective decisions. This project will examine the efficacy of the selected structural and statistical models in different market scenarios for performing short term stock price trend prediction. A total of 10 models from the two classes were tested and evaluated. The predictive performance of the models was analysed on two variants of daily stock price data between 2010 and 2018. The empirical results of all models were compared, and they provide evidence that classical time series models do still outperform machine learning models, although the results from the machine learning methods are also encouraging when using the walk forward validation strategy. Moreover, a robust ensemble model approach combining the top performing models from both classes was proposed. This combination achieved a better prediction accuracy than all individual models.

Contents

List of Figures

List of Abbreviations

ACF Autocorrelation function.

AIC Akaike Information Criterion.

ANN Artificial Neural Networks.

AR Autoregression.

AR-Net AutoRegressive Neural Network.

ARIMA Autoregressive Integrated Moving Average.

EDA Exploratory Data Analysis.

LASSO Least Absolute Shrinkage and Selection Operator.

LSTM Long Short Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage error.

MCHP Microchip Technology Inc.

ML Machine Learning.

MSE Mean Square Error.

RMSE Root Mean Square Error.

RNN Recurrent Neural Networks.

S&P500 Standard & Poors 500.

VZ Verizon Communications Inc.

1 Introduction

The Equities (stock) market is the second largest financial market by size after the fixed-income market(2021). It enables organizations to raise capital, compete with industry competitors, and spread ownership by selling shares (part ownership) to individuals and organizations. The stock market is both beneficial to retail investors and the listed companies on the exchanges. The advantage to the retail investor is that they are given the ability to own part of a company and earn from the possible increase in value of the securities in possession. The company value is the price of the security (stock price) multiplied by the number of shares available to the market. This is also referred to as a market capitalisation. The company also benefits from the market, as it is able to raise money from the public to fund its innovation and other activities.

Investing in the stock market is regarded as risky, as investors are betting on the success or failure of the company invested in with no guarantee of profits (Securities and Commission n.d.). However, the reward for the uncertainty is the possible returns earned when asset values increase or when dividends are shared to investors. When a company stock rises, the companys market cap increases, and investors also gain from the rise in value of their investments. This opportunity has spawned individuals and organisations to look for ways of reducing market uncertainty, which can eventually reduce the risk of their decisions. One of the ways individual investors and organisations try to do this is by forecasting future performance to help guide their decisions and to make bets.

Forecasting in markets is not a new topic. Techniques for forecasting have previously been developed and utilised by economists and statisticians for decades to try to gain an edge over the rest of the market. Although the accuracy of these forecasts is still questioned, they are claimed to serve as an effective tool for individuals and organizations in making decisions. With recent advancements and acceptance of artificial intelligence, there has been curiosity into the role it has to play in financial markets. Over the years, there has been a greater acceptance of machine learning methods for use in forecasting and economics. However, there has also been curiosity about how accurate and usable these machine learning models would perform in real-world scenarios. With machine learning providing innovative strategies and better calculation accuracy, it is believed that ML models could eventually compete with traditional time series forecasting techniques.

The aim of this project is to investigate the efficiency of both traditional time series and machine learning forecasting methods for short-term stock price trend prediction. Models yield results that are objective in comparison to predictions done by humans. Humans may easily be biased, which may explain why many analysts fail to predict accurately. This tendency for bias also proves the need for assistive technical tools to promote objective thinking

1.1 Real-World Application and Motivation

As mentioned by(Mallikarjuna and Rao, 2019)(Mallikarjuna and Rao 2019), the better you can accurately predict the market, the greater your profitability. Forecasting using Time series and Machine Learning models could serve as effective tools in areas like portfolio management and corporate decision making. Markets are constantly evolving, thus the need to constantly evaluate forecasting methods in order to keep up with trends and reinforce predictions.

A large percent of retail investors lose money in the market due to speculation. Thus, forecasting serves as a tool for tackling investor irrationality. These models could also prove to be useful to professional analysts as a secondary tool for making price estimates and corporate decisions and for detecting change in market trends. The extent to which these forecasts can be useful is what this study explores. However, the use of forecasting for betting on short-term market trades is not encouraged because past prices may not be fully indicative of future performance, which makes forecasting a secondary tool and not a primary decision maker. Also, the lower the prediction window, the higher the accuracy that is needed to be able to trust these systems. This is because, in the short term, price movements in markets are closer to random, however in the long term, they are more predictable due to generalised trends. That said, markets are never 100% certain.Longer-term investors are not usually worried about precise prices. Rather, they are more concerned about overall price trends.

1.2 Objectives

Primary:

1. Research and selection of currently used statistical and machine learning methods
2. Exploratory data analysis to find a range of stocks data with certain criteria (volatile, non-volatile, low/high volume, causes of data non-linearities)
3. Model building and optimisation on chosen data.
4. Model evaluation: Investigation of the error metrics of different models on the same data and how certain market dynamics (eg. volatility) can affect model performance.

Extension:

5. Ensemble forecasting: Combination of models for better predictive power
6. Implementation of other models: Prophet, NeuralProphet
7. Model optimization

1.3 What influences stock price movements?

To predict stock prices, there is the need to understand what constitutes a stock price and why it changes. The difficulty in predicting the stock market comes from the fact that stock prices are influenced by multiple factors. Fundamentally, market prices are heavily influenced by the actions of market participants. The value of a stock at a given moment is determined by its current demand and future expectations in the market, which is also known as investor enthusiasm. When there is high demand for a stock, the prices tend to rise as it becomes more expensive to purchase due to its supply and vice versa. When investors believe the stock price is undervalued and expect it to rise in the future, they buy more, which eventually raises the price of the asset. The economy also has an influence on the growth of the stock market(Harvey 1989).

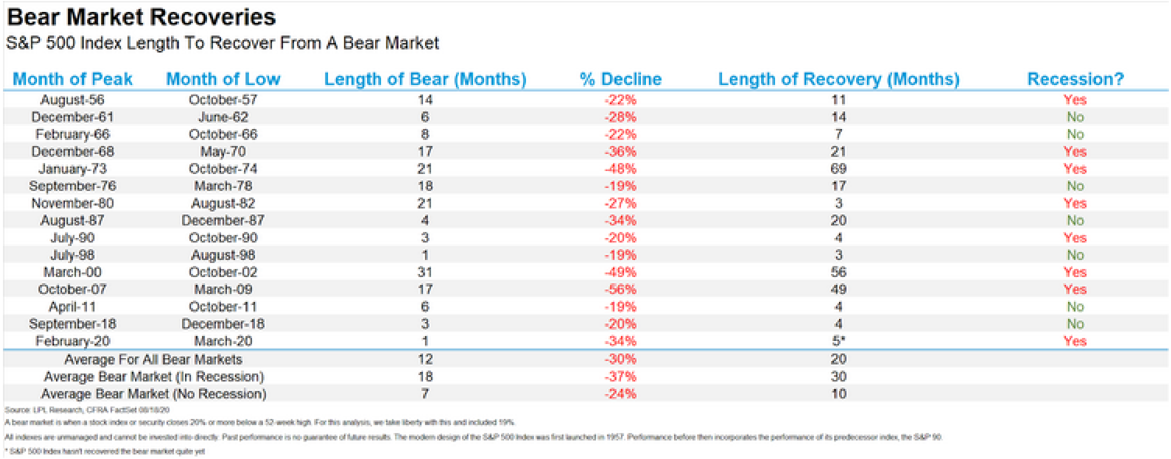
Two other major factors that affect stock prices are macroeconomic activities such as inflation,interest rates, and the political climate and available market information (Weng, Martinez et al. 2018). It is believed that when new information is available, the market incorporates the information (either negative or positive) into the stock price. Examples of information include earning reports, market sentiments, news articles, and trends. Macroeconomic indicators usually affect the longer term, while news could affect both short and long term(Weng, Martinez et al. 2018). A study carried out by (Homa and Jaffee 1971) concluded that there is a link between stock prices and external factors such as money supply. It is also important to note that the major determinants of stock prices change and also portray different effects over time. For example, a stock price could be highly correlated with the predicted results of a certain election, and over time, this correlation could be lost or strengthened. (Homa and Jaffee 1971) also stated that structurally specifying all the determinants of stock prices is not possible, as they are dependent on expectational factors. However, the more determinants to which we have access, the better we can make more accurate predictions.

Traditionally, two main approaches have been used for forecasting future performance of companies. The first is technical analysis, which is the use of quantitative variables, such as past prices, Volume of sales of the stock, and company revenue. The second is fundamental analysis, which is the use of qualitative information, such as income statements, market sentiments, and company earnings reports, to make decisions. For forecasting, technical analysis is usually used. However, in recent studies, a combination of both have been used to try to make stronger predictions.

(Anhar 2015) studied the influence of stock prices and concluded that factors could also be classified into direct and indirect influences. Examples of indirect influences that can influence stock prices are the general performance of stocks in the same market sector and interest and inflation rates. Direct influences include company performance (e.g. earnings and debt) and investment risk.

1.4 Market Trends and Cycle Durations

The stock market has two main trend cycles. The bull market (uptrend) refers to when there is generally an upward growth of the asset or market over prolonged periods of time. The bear market (downtrend) is the opposite- when an asset or market has generally been declining in value over a prolonged period of time. Research carried out by an American investment and business solutions company - LPL Research (lplresearch 2020) - found that Market cycles are typically long. As seen in figure 1.1 cycles have lasted on average between 1 to 5 years over the past few decades. Market trends play an important role in this study, and a goal of this paper is to understand which models can pick up change in trends more efficiently.



Bear Market Recoveries

S&P 500 Index Length To Recover From A Bear Market

Month of Peak	Month of Low	Length of Bear (Months)	% Decline	Length of Recovery (Months)	Recession?
August-56	October-57	14	-22%	11	Yes
December-61	June-62	6	-28%	14	No
February-66	October-66	8	-22%	7	No
December-68	May-70	17	-36%	21	Yes
January-73	October-74	21	-48%	69	Yes
September-76	March-78	18	-19%	17	No
November-80	August-82	21	-27%	3	Yes
August-87	December-87	4	-34%	20	No
July-90	October-90	3	-20%	4	Yes
July-98	August-98	1	-19%	3	No
March-00	October-02	31	-49%	56	Yes
October-07	March-09	17	-56%	49	Yes
April-11	October-11	6	-19%	4	No
September-18	December-18	3	-20%	4	No
February-20	March-20	1	-34%	5*	Yes
Average For All Bear Markets		12	-30%	20	
Average Bear Market (In Recession)		18	-37%	30	
Average Bear Market (No Recession)		7	-24%	10	

Source: LPL Research, CFRA FactSet 08/18/20

A bear market is when a stock index or security closes 20% or more below a 52-week high. For this analysis, we take liberty with this and included 19%.

All indexes are unmanaged and cannot be invested into directly. Past performance is no guarantee of future results. The modern design of the S&P 500 Index was first launched in 1957. Performance before then incorporates the performance of its predecessor index, the S&P 90.

* S&P 500 Index hasn't recovered the bear market quite yet

Figure 1.1: Historical Length of bull and bear markets (lplresearch 2020)

1.5 Efficient Market Hypothesis and Random Walk

In a paper about the efficient market hypothesis, (Malkiel 2003) defined a random walk as a term used in characterising a price series in which future prices of an asset represent random departures without correlation from its previous prices. The efficient market hypothesis assumes that stock prices follow this random walk model (Fama 1965). This means that changes in stock prices are independent of one another, and past prices and trends are not indicative of future performance. It also suggests that the flow of market information is instantly incorporated into stock prices and that future prices and information are not affected by previously available information (Malkiel 2003, p. 3). This implies that as unseen events happen, they all have unique, instantaneous effects on stock prices, and information from previous time periods provide no connection to future prices. An example is the most recent stock market crash which was influenced by the coronavirus pandemic.

In early studies of the EMH, studies like that of (Malkiel and Fama 1970, p. 197) inferred that models based on past price changes and patterns could not be used to infer future price movements and that they were inaccurate. (Malkiel and Fama 1970, p.197) classified the hypothesis into three forms: The strong, semi-strong and weak forms. If the markets were fully efficient, then creating models from past data to make future predictions using technical analysis would be useless, thereby making this study pointless. The complete efficiency and unpredictability was regarded as the strong view of the hypothesis. However, over the years the notion of the efficient market hypothesis has been challenged by analysts due to several anomalies in the market, disproving the strong hypothesis. This gave way for weaker views of this theory. (Darrat n.d.) used models to investigate whether Chinese stock prices follow a random walk process. Their experiment results proved to be at odds with the random walk theory. They found a positive correlation between the data and lagged versions of itself which proved that predictability is possible to an extent.

The weak hypothesis believes that markets may not reflect information that is not available publicly. For example, despite previous crashes in the market, the 2020 crash was still captured by the market. If the markets were fully efficient, then the prices would have immediately reacted when the information about the coronavirus was released. In theory, if the strong hypothesis was fully valid, then it would have been impossible to find undervalued assets and outperform the market. However, multiple traders and investment funds have proved the flaws in this theory by continuously outperforming the market. For example, between the years of 1964 and 2014 Bekshire Hathway managed to outperform the overall market significantly with returns of 1,826,163% in the span of 50 years, which is much higher than that of the S&P500 index(2021).

Most modern-day funds use either intelligent trading algorithms, artificial intelligence, or both to find price arbitrage opportunities - which are occasional inefficiencies found asset prices, in which the difference in prices can be exploited to make profits.(Harvey 1989) explains how predicting stock prices can be compared to using only the rear-view mirrors to drive a car forward on an unpredictable road. He states that it is not impossible, however, it is difficult. Although others believe that stock price changes cannot be predicted, some believe that they can be partially determined by analyzing historical market fluctuations and charts. Recent studies have also shown promising results in predicting stock prices by incorporating modern forecasting techniques. The studies claim that stock prices can be partially predicted in order to gain some useful information if the right techniques and information are applied. The main purpose of this paper is to develop those techniques to capture the nonlinearities in stock price data in order to pick up patterns in the form of trends.

2 Research and Related Work

There has been significant work done to predict stock prices especially, in the econometrics space. Methods used have also evolved over the years. Research into previous works was done to understand the type of techniques previously applied and which ones worked best.

(Adebiyi, Adewumi and Ayo 2014, p. 5) investigated the efficiency of ARIMA(Autoregressive Integrated Moving Average) models for stock price prediction, which exhibited substantial potential. They used the previous stock price data of two well-known companies from different industries: Nokia which is in the technology industry, and Zenith Bank, which is in banking. They concluded that the ARIMA models proved to be an effective forecasting method for understanding short-term price movements and could be a useful tool for making market decisions. The forecast was carried out based solely on the closing price variable. No comparison was made to other methods of forecasting which could have better quantified the ARIMA models performance in respect to other forecasting techniques.

(Mallikarjuna and Rao 2019, p. 9) tested out multiple models on 24 stock indices. They concluded that different markets could have different ideal forecasting approaches used due to the difference in behavior of markets. Indexes show patterns much better than individual stocks due to their grouping. They also found that there was no optimal method for forecasting that worked for all markets. Techniques needed to be reinforced from time to time to keep up with market changes.

(Lee et al. 2008, p. 4) also compared the efficiency of the neural network (NN) to the Seasonal Autoregressive Integrated Moving Average (SARIMA) model on the Korea Composite Stock Price Index (KOSPI). This analysis imitates this research, but by only comparing one linear and non-linear model. It was found that the SARIMA model showed greater performance with better error rates than the neural network on the tested dataset. (Lee et al. 2008) also inferred that the NN model performed considerably well in predicting certain Korean equity market variables. For this project, the deep learning approach will be considered with the use of long short-term memory (LSTM) networks.

(Shen and Shafiq 2020, p. 21) also proposed a deep learning approach for short-term price prediction with emphasis on using LSTM networks. They found that an increase in features increased the prediction accuracy of the models. The features they used were non-correlated variables which asserts the studies of (Guidolin et al. 2014) and (Xiaohua Wang et al. 2003). They also realised that preprocessing does not largely impact training efficiency, however, it does extremely affect the model accuracy. This precaution will be taken into account when building the LSTM models in this project to ensure the preprocessing techniques proposed will allow for the models to perform better on the testing dataset used. (Shen and Shafiq 2020) also compared the accuracy of their proposed model to that of other authors, e.g. (Adebiyi, Adewumi and Ayo 2014). However, the comparison of the models proposed in the paper was not based on the same data, as both researchers used different datasets. For this study, the models will be compared on the same dataset, and the same evaluation metrics will be applied to all models to ensure a fair comparison.

There are also novel methods that have been created and studied, such as that by (Eapen, Bein and Verma 2019), who proposed a new hybrid model using a pipeline inclusive of a convolutional neural network and a bidirectional LSTM model. They found that compared to the single deep learning approach, the results of their forecast largely improved. They also concluded that combining multiple models is an effective way of reducing overfitting, which allows for a model that generalises better.

A number of studies have also studied the effect of using macroeconomic variables as features for predicting price returns. (Guidolin et al. 2014) attempted to use variables such as inflation and interest rates as model features. (Xiaohua Wang et al. 2003) used volume as a feature for predicting stock prices and concluded that it has little to no effect on the trend of the forecast and can sometimes lead to overfitting of data. From the papers, it can be inferred that using variables that are highly correlated, such as stock prices with their sales volumes, provide little to no increase in performance, although, including external uncorrelated variables such as interest rates provided much better results.

Another study conducted by (Weng, Lu et al. 2018), on creating hybrid models for short-term stock price prediction created an expert system based on four ensemble models of which the best performing model is chosen for forecasting. The study utilised methods such as neural networks, support vector regressors, and boosted regression trees. This robust strategy produced good results. A mean MAPE of 1.07% was achieved across all the tested models.

(Rather, Agarwal and Sastry 2015) also created a hybrid model that utilises the power of both linear and nonlinear models. They combined the ARIMA and exponential smoothing models with a recurrent neural network to achieve better predictions. Their model also performed well in comparison to the single model approaches. They proved that ensemble methods do sometimes provide better forecasting accuracy if used with the right combination of models.

3 Exploratory Analysis and Methodology

3.1 Exploratory Data Analysis

EDA(Exploratory Data Analysis) is an important step to ensure the thorough understanding of the data before modelling. The type of data used can largely affect the accuracy of models. For example, when data sequences are relatively linear, it could be easier to forecast. This is because linear-like data are easier to fit in a model in comparison to random data. The datasets used in this study were chosen based on several criteria, such as volatility, correlation and Industry. This was done to allow for an in-depth analysis of the models with respect to the type of data uses. As reported by (Mallikarjuna and Rao 2019), models perform differently when exposed to different types of data. The criteria were chosen to better understand model performance and to ensure an in-depth evaluation.

3.2 Time Series in Relation to Stock Data

A time series can be defined as a series of observations recorded over successive time points(Nist.gov 2019). This differs from cross-sectional data, where there is no dependence on time as a component.

$$Y_t = constant_t + error_t \tag{3.1}$$

$$y_t = y_{t-1} + w_t \tag{3.2}$$

The general Time series formula above shows that the current variable (y_t) is influenced by a lagged version of itself (y_{t-1}) with an addition of noise (w_t). Time series data are assumed to have memory. This explains why lagged versions of a time series have influence on the current variable. Time series can be split into two main components, which are the systematic and non-systematic components. The systematic components include the important information, which are trend, seasonality, and cyclicity, while the non-systematic component consists of noise or irregularity, which makes the patterns harder to predict. The size of time series intervals can vary, however they have to be fixed throughout the data. They could range from secondly all the way to yearly intervals.

Time series analysis is the estimation of how a target variable and any other explanatory variable changes with respect to time. It assumes that a certain time series exhibits a combination of a systematic pattern and noise. There are a range of techniques used for time series forecasting, each of which have their specific use cases. Time series forecast models provide a better estimate for how much trends will continue based on passed information. Time series forecasting can be done with both univariate (single variable) and multivariate(multiple dependent variables) datasets. For this project, a combination of univariate and multivariate models will be tested.

3.3 Are Stock Price Series Different from Other Time Series?

For stock price data, the current day price is influenced by previous prices, including noise. Stock price series are made up of useful underlying patterns, which are referred to as the signal, and insignificant information, which is called noise. To predict, we have to separate the signal from the noise. An example of a signal in stock data is trends. The essence is to pick out the useful signal (such as possible trends) from the past data and make models based on past behavior to predict future values. Stock prices are usually more predictable in the longer term, as there is less noise and more about the long-term trends.

Stock price data are considered to not be randomly generated values. Stock price data can be treated as a time series, as the values are given with respect to time. Also, for stock price data, we will prove that previous prices have an influence on future prices. This is called the autocorrelation. However, price series are usually non-stationary. The steps mentioned below were taken to ensure that stock data could be used for time series forecasting.

$$\text{Today's stock price} = \text{Information from previous prices} + \text{noise} \tag{3.3}$$

The signal-to-noise ratio also plays an important role. This represents the amount of useful underlying information compared to the unuseful market noise. It is worth noting that with the advancements in technology and exposure of markets to more participants, including retail traders and automated trading systems, signal-to-noise ratios, may continue to decrease as prices will get more distorted from the underlying trends due to higher volatility.

3.4 Data Choice, Frequency, and Length

As stated by (Makridakis, Hibon and Moser 1979), model accuracies can largely be affected by the type of data used. In this study, the characteristics of the datasets were closely examined to accurately build up on previous research. The datasets used on the models was chosen based on multiple criteria including the type of stock, length of data, and frequency of observations.

In an article written by(Kumar, Jain and Singh 2020) where a survey was taken of the length of data used in previous research studies for stock price forecasting, there was a consensus in the use of 1500 (4 years) or more daily observations. For this study, an 8-year time range between 2010 and 2018 was chosen. The timeframe of the two price series was specifically chosen to avoid irregularities in price behavior, for example, the previous stock market crashes (e.g. the 2008 stock market crash, the dot com crash). Between 2010 and 2018, there were no significant irregular events for either of the datasets that would affect the quality of the forecasts. If large irregularities were present, they may have adversely affected the accuracy of the forecasted results.

The price series were sourced directly through the Yahoo finance API. Multiple variables were collected, including the open, high, low, close, adjusted close price and volume of each stock. The adjusted close prices of both stocks were used as the main forecasting variable for all the models to reflect the true value of the stocks prices which accounts for any previous corporate actions that were taken, such as stock splits. Two multivariate models used the additional features as exogenous variables and the rest of the models were trained using only the adjusted close as the dependent variable.

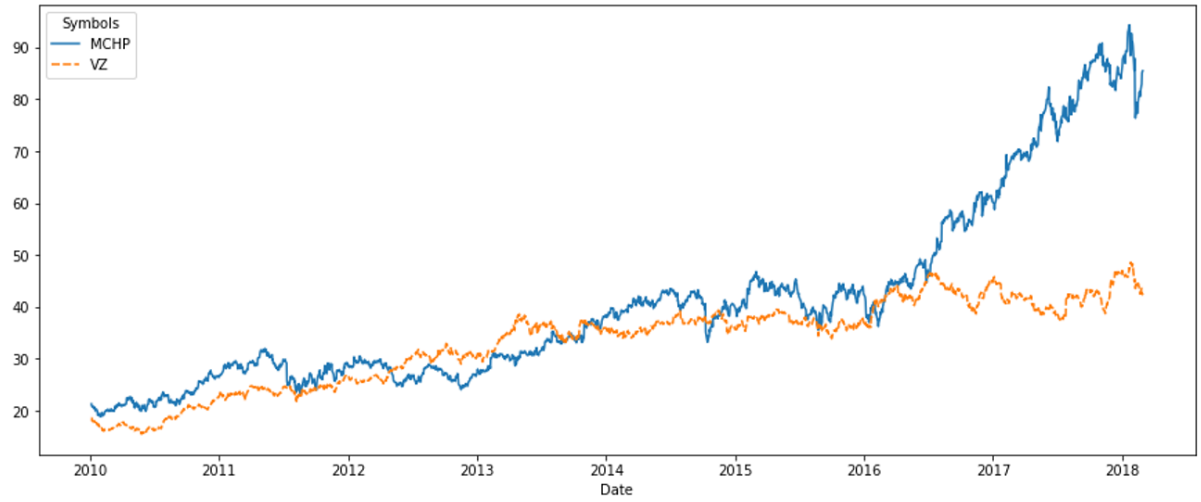


Figure 3.1: Microchip (MCHP) and Verizon (VZ) stock price 2010-2018

The data were chosen based on multiple criteria to ensure that they were distinct in order to allow for a more complex analysis of the models in different scenarios.

A short-term forecast horizon was chosen to imitate a real-world forecasting scenario. Ideally, in the short term, analysts would be looking at predicting the next quarter of the year, which is 3 months long. Daily observed data was utilised, as this was the interval commonly used in most of the researched studies. Where (n) is the size of each individual dataset, the last ($n-90$) observations from both datasets were excluded for the evaluation and comparison of all models. The rest of the data were used for training the models with the walk forward validation. The same split was applied on all models to allow for a fair comparison. Also, as markets are continually evolving over time, it is not efficient to use data from long periods of time as it becomes irrelevant. This leaves a tradeoff. It may be assumed that the more data you have, the better. However, the more irrelevant the data becomes over time due to loss of autocorrelation and constant change in market.

3.5 Data Volatility with Respect to Predictability and Growth Patterns

Stock volatility can be defined as the rate of dispersion of a stock price from its expectation. There is a general belief that the higher the volatility of a stock, the lesser it follows a pattern, which makes it harder to predict. Volatility is usually calculated as the standard deviation or variance of the returns of the stock, or it can also be compared with the dispersion of the overall market index. A higher volatile stock means that the asset has a greater chance of fluctuating in price over shorter periods of time. To accurately measure the volatility of the chosen datasets, two evaluative criteria were considered: the deviation(standard deviation) of the securities with respect to their respective calculated means and also the deviation in comparison to the total market.

1. Long-term Beta Values:

Beta can be defined as a measure of correlation of an asset to a certain benchmark. It measures the relative strength of a company in comparison to the benchmark. The benchmark used for comparison is usually the whole market or an index, in this case, the S&P 500, which is an index containing the top-500 performing companies on the US stock exchanges. It is widely considered a measure of performance of the overall market. Beta can also explain how volatile an asset is with respect to the benchmark. For example, a stock with a beta value of 1.3 means it moves 30% times more than the index (S&P 500). The 5-year (long-term) beta values were considered for each security. The chosen datasets, Microchip (MCHP) and Verizon (VZ) , had 5-year beta values of 1.6 and 0.39, respectively. This shows that Microchip is much more volatile than the market and Verizon, on the other hand, is less.

2. Measurement of 10-,15-, and 30-Day Volatility

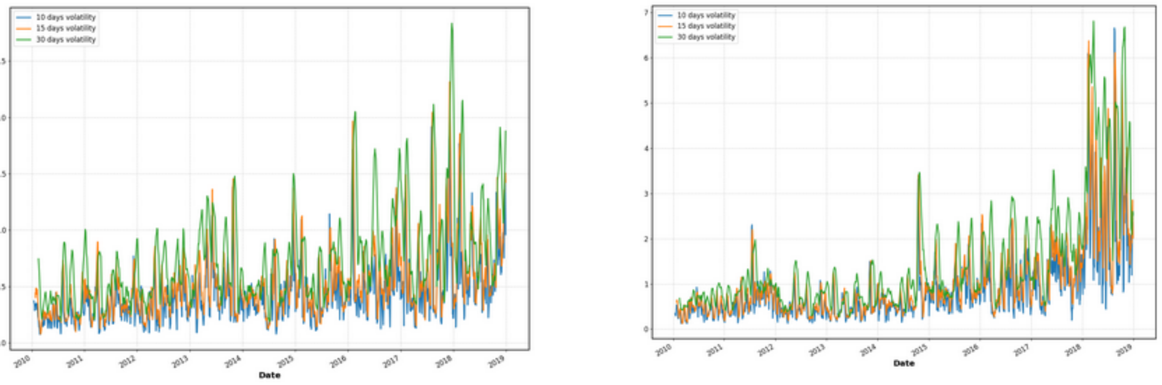


Figure 3.2: 10, 20, and 30 day rolling volatility

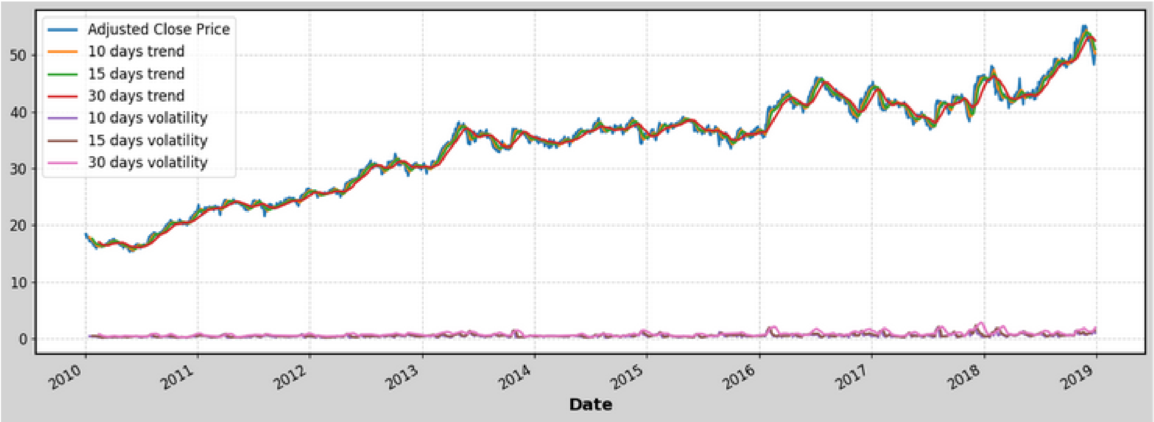


Figure 3.3: Moving averages and Volatility Verizon(VZ)

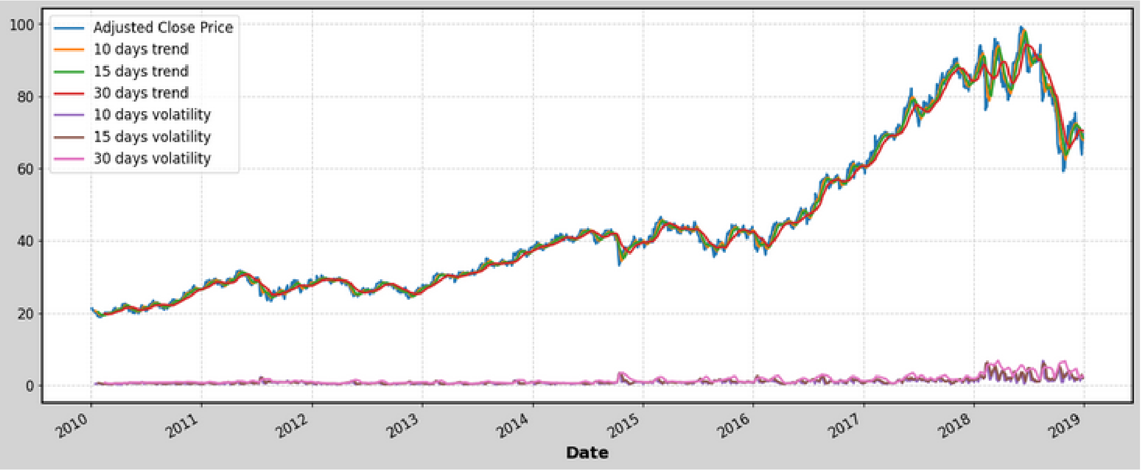


Figure 3.4: Moving averages and Volatility Microchip(MCHP)

Figure 3.3 and 3.4 show the 10-,15-, and 30-day rolling volatility of the two price series in comparison to the moving averages when measured over the 8-year period. The rolling mean is also reffered to as the moving average. The moving average is a technical indicator that helps to better understand change in trends visible of the data. The rolling volatilities were achieved by calculating the rolling standard deviation for the three time periods. The rolling standard deviation measures the dispersion from the rolling mean(moving averages) of the series. The higher the standard deviation from the mean, the more volatile the asset is.

MCHP exhibits higher longer-term standard deviation changes. In figure 3.2, it can be seen that verizon's rolling volatility patterns focus around a relatively constant mean as opposed to Michrochip on the right hand side which displays and increase in volatility over time. MCHP also displays suprises in volatility values towards observations in the final year. It is quite clear that MCHP overall is a more volatile asset than VZ. It possesses a higher continual long-term volatility and also on average shows higher rates in its 30-day volatility patterns which hints that it has a longer term volatility.

3.5.1 Growth patterns

Both datasets possess an upwards slope in trend over the 8-year period of the dataset. However, on a smaller scale, the datasets are quite diverse in their characteristics. The Verizon (VZ) data also exhibit a different growth pattern in respect to that of Microchip (MCHP). As seen in figure 3.1 MCHP displays exponential growth over the 8-year period, while VZ displays a more linear trend and a more predictable growth pattern.

3.6 Time Series Stationarity

A time series can be regarded as stationary when its statistical properties, such as its mean and variance, are constant over time. Stationary time series are mean reversing. This means whenever there is a large deviation, the series always reverses back, and fluctuates closer to the mean. A time series that contains trends and seasonality or has a unit root can be assumed to be non-stationary. A time series possesses a unit root if it shows an unpredictable systematic pattern. Stationary models, such as the ARIMA require the input time series to be stationary and have a no unit root to produce good results. For stationarity, the goal is to have the distributions depend only on the difference in time (lags) rather than the location of the variable in time.

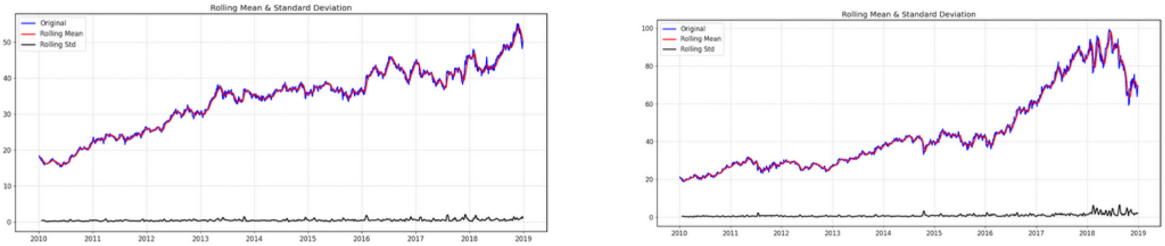
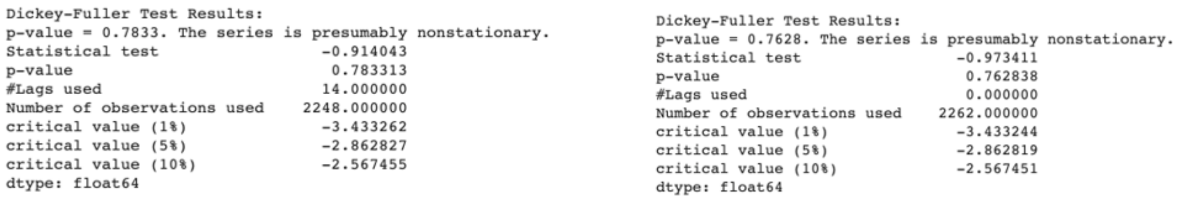


Figure 3.5: Rolling Mean and Standard Deviation of both price series

From both plots in Figure 3.5, it can be seen that there is an increasing mean and increasing standard deviation over time. This also confirms that the price series is non-stationary.

Tests were carried out on both datasets to prove the non-stationary nature of stock data. From the figures above, it is clear that both datasets possess an upwards trend with a continuous increase in mean and variance over the time series.

The Augmented Dickey Fuller (ADF) test is a unit root test to prove the stationarity of a time series. The test was performed on both datasets. The rule is that if the ADF t-statistic is greater than the critical values, then it can be said to have failed to reject the null hypothesis, therefore rendering the time series as non-stationary. Verizon and Microchip obtained t-statistic values of -0.973411 and -0.740900, respectively The P-values for both datasets are 0.762838 and 0.835869, which are both above 5%. and both confirms that both datasets possess a unit root.



```
Dickey-Fuller Test Results:
p-value = 0.7833. The series is presumably nonstationary.
Statistical test                -0.914043
p-value                          0.783313
#Lags used                      14.000000
Number of observations used   2248.000000
critical value (1%)             -3.433262
critical value (5%)             -2.862827
critical value (10%)            -2.567455
dtype: float64
```

```
Dickey-Fuller Test Results:
p-value = 0.7628. The series is presumably nonstationary.
Statistical test                -0.973411
p-value                          0.762838
#Lags used                       0.000000
Number of observations used   2262.000000
critical value (1%)             -3.433244
critical value (5%)             -2.862819
critical value (10%)            -2.567451
dtype: float64
```

Figure 3.6: Augmented Dickey fuller Test

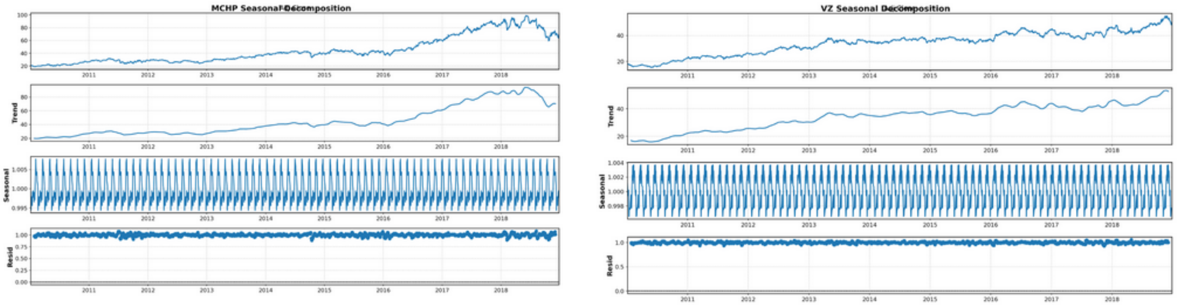


Figure 3.7: Seasonality Decomposition Plot

The seasonality decomposition plot provides us with a decomposition of the main components of a time series.The seasonality of the two price data and their trends are dependent on the trend over time. This is why we go with the multiplicative method for decomposition.

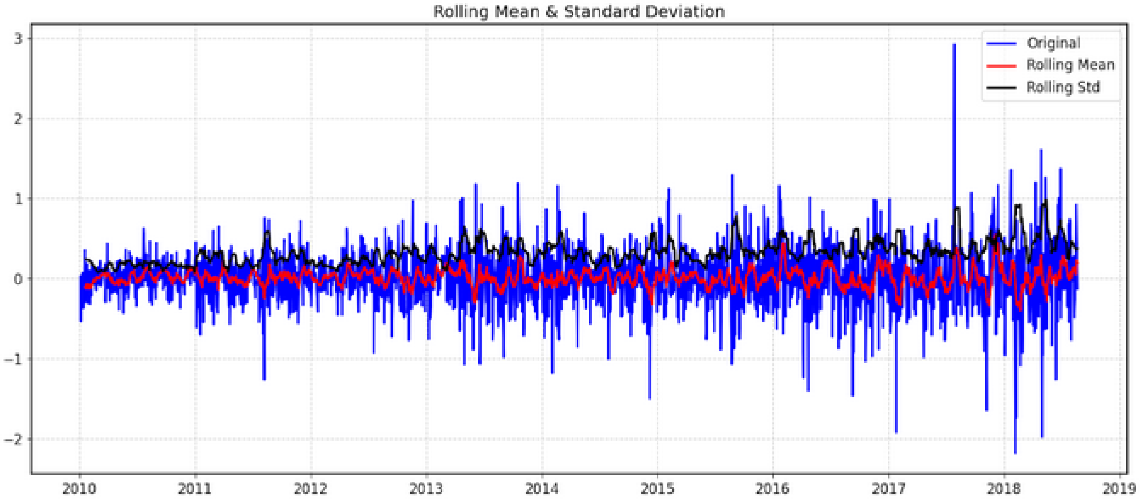
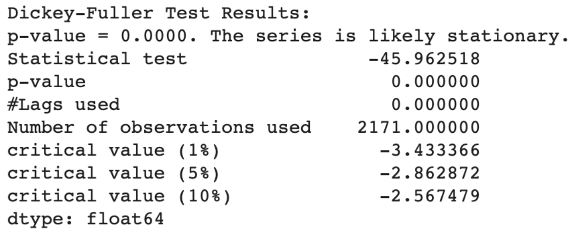
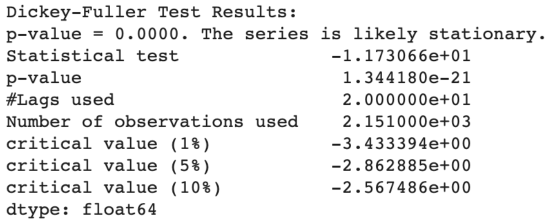


Figure 3.8: VZ after Differencing



```
Dickey-Fuller Test Results:
p-value = 0.0000. The series is likely stationary.
Statistical test                 -45.962518
p-value                            0.000000
#Lags used                         0.000000
Number of observations used     2171.000000
critical value (1%)               -3.433366
critical value (5%)               -2.862872
critical value (10%)              -2.567479
dtype: float64
```

(a) VZ ADF test after First order Differencing



```
Dickey-Fuller Test Results:
p-value = 0.0000. The series is likely stationary.
Statistical test              -1.173066e+01
p-value                        1.344180e-21
#Lags used                     2.000000e+01
Number of observations used    2.151000e+03
critical value (1%)           -3.433394e+00
critical value (5%)           -2.862885e+00
critical value (10%)          -2.567486e+00
dtype: float64
```

(b) MCHP ADF test after First order Differencing

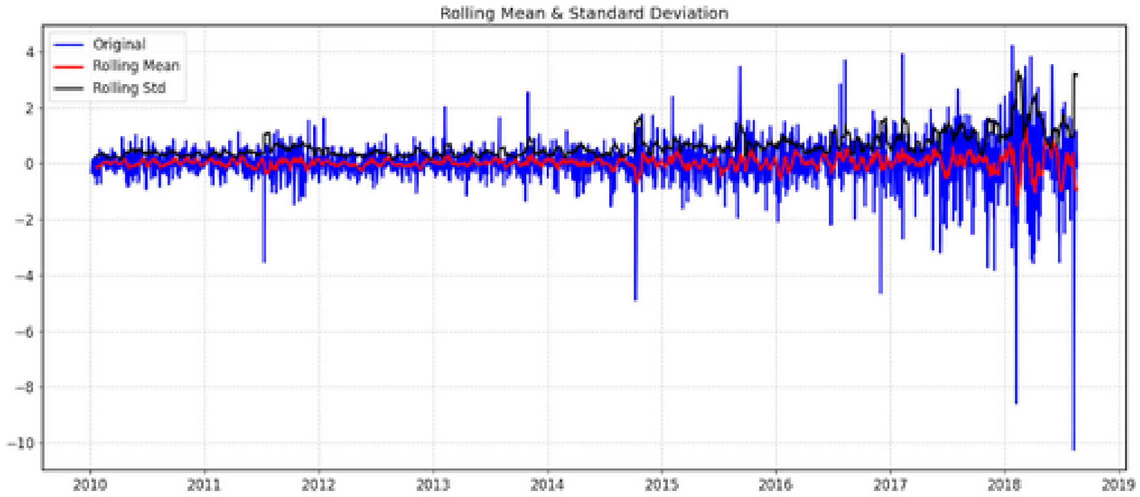


Figure 3.9: MCHP after Differencing

3.7 Transformations to Achieve Stationarity

To make a time series stationary enough for the statistical models, the mean and standard deviation need to be made constant over time. Differencing was applied to both datasets in order to achieve stationarity. Figure 3.8 and 3.9 show the first order differencing of the time series which provides the returns of the stock price. A stationarity test was done on the differenced datasets, which succeeded in rejecting the null hypothesis. The stationary models will be set to provide first order differencing.

3.8 Autocorrelation and Industry Correlation

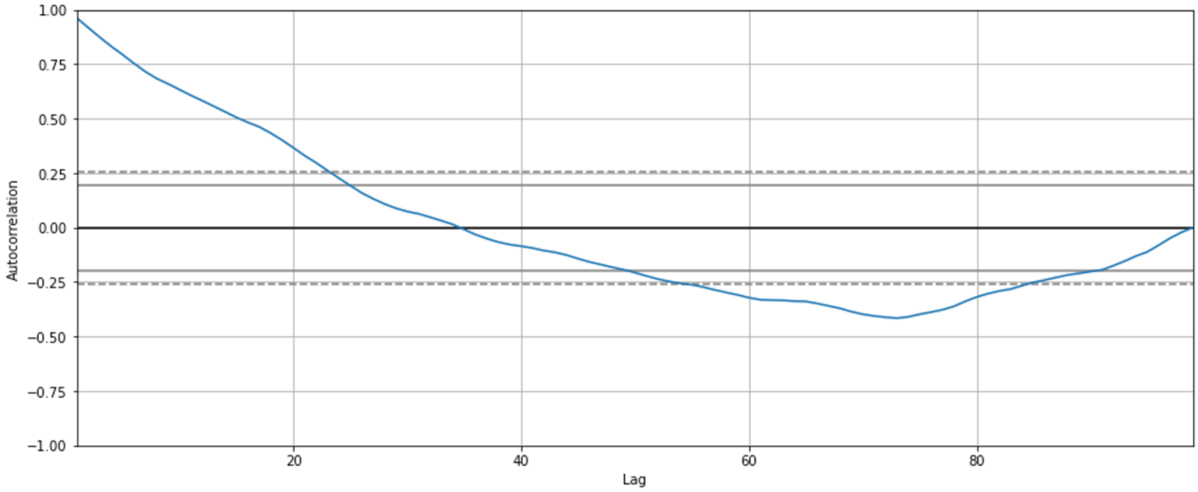


Figure 3.11: Autocorrelation of Verizon(VZ)

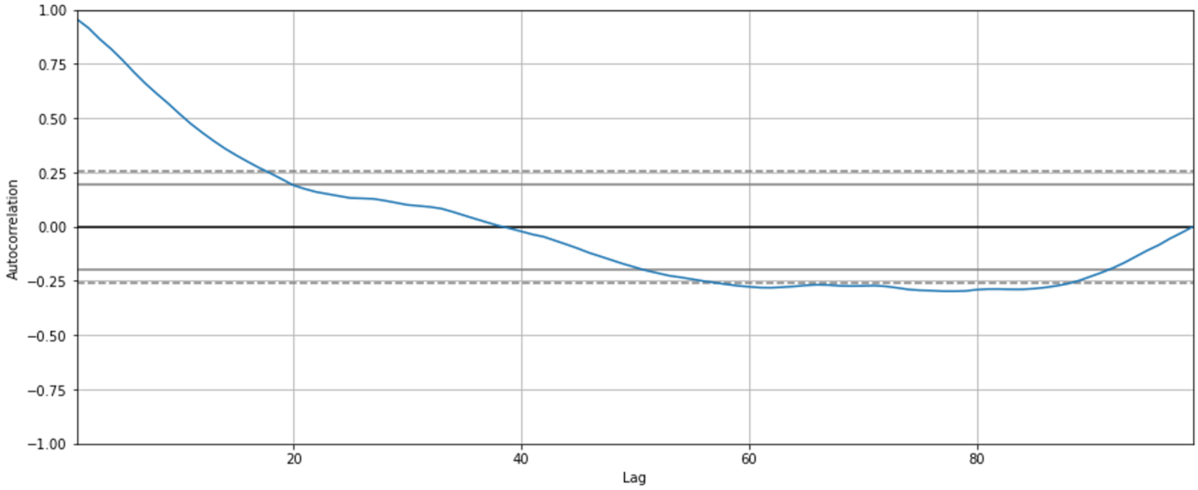
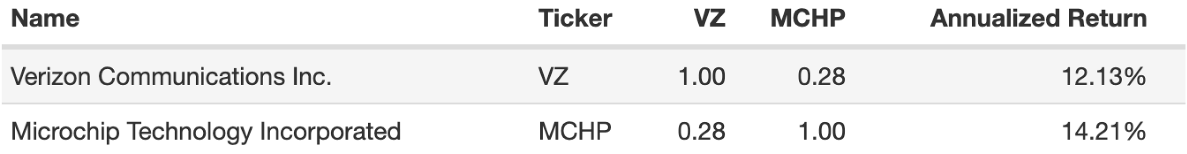


Figure 3.12: Autocorrelation of Microchip(MCHP)

The auto-correlation of each of the datasets were taken into account. Data autocorrelation can be defined as the correlation of a variable (current price) with past versions of itself. Autocorrelation helps to understand the correlation of lagged values in a variable with the specific value of the same variable. The autocorrelation plot also proves that the time series is not random, as a time series with highly correlated lags cannot be random. The traditional time series models depend on the autocorrelation of the data to understand patterns. Stock price time series data should have correlation with lagged versions of themselves because previous prices do influence future prices, but that influence fades over time.

To check the autocorrelation, the data were resampled into the mean of every business month, and the ACF plots were computed. The ACF graph also assists in examining and deciding on an appropriate lag value to be used on the time series models. Any correlation that is below the 95% confidence intervals can be regarded as statistically non-significantBox et al. 2015. This is because random data has very low autocorrelation to lagged values of itself. If sizeable ACF values are achieved, It can be confirmed that the values are serially correlated. From the ACF graph, it is clear that both datasets exhibit good autocorrelation with their lagged values.It is also clear that an increase in lag equates to a decrease in the autocorrelation over time. The autocorrelation plot of VZ shows about 22 lags that are above the 95% confidence line. The MCHP dataset shows lesser autocorrelation to its lagged values due to the data possessing higher volatility. Studies done by (LeBaron 1992) & (Faff and McKenzie 2007) show that when there happens to be an increase in volatility, there is a decrease in autocorrelation with past values.

For stock price data, the effect of lagged prices over current prices decays over time as behavior of prices changes with time. This means that the stock price from two months ago has a greater chance of having a lower autocorrelation with the current stock price than that of the previous day price.



Name	Ticker	VZ	MCHP	Annualized Return
Verizon Communications Inc.	VZ	1.00	0.28	12.13%
Microchip Technology Incorporated	MCHP	0.28	1.00	14.21%

Figure 3.13: Asset Correlation matrix between 2010 to 2018 based on daily returns (AssetCorr 2021)

In terms of data correlation by industry, Microchip represents the integrated circuit and semiconductor industry and Verizon represents the telecommunications industry. Although these can be classified as similar industries, companies in the integrated circuit sector are tolerably more volatile than the average market stocks due to greater activity. This has ensured relatively lesser correlation, which means the data are more distinct.

4 Models

4.1 Selection Process

The methods used for forecasting were selected based on popular models used in past research papers. A total of 4 machine learning models and 5 traditional time series models were tested with an addition of a new approach to forecasting being proposed.

4.2 Machine learning Models

Over the years, machine learning has grown in terms of use cases. It can be used to perform classification, regression, segmentation, and clustering tasks. In this paper, machine learning models will be used for regression tasks, although machine learning models are relatively new to time series forecasting in comparison to traditional time series models. In this study, the idea is to replicate the method of forecasting used by the traditional models. That is why the forecasting pipeline has been prepared differently for ML and LSTM models. The following techniques guided the creation of the models.

4.3 Conversion of Univariate Time Series to Supervised Learning

Time series models are dependent on past(lagged) values for future predictions by default, which explains the search for autocorrelation. However the ML models are not geared this way. Machine learning models assume that each datapoint is independent of the next timestep. In order to use machine learning models for time series forecasting scenarios, the data had to be reframed from a single time series input into a pair of supervised time series learning tasks. Supervised learning is the foundation of machine learning predictive modelling techniques. The data were replicated and then lagged by one time step to mimic the autocorrelation considered by the statistical models. In this case a lag of 1 was used so that the target variable is set as the next day's close price. The restructured data is then fed into the models as a supervised task.

4.4 Recursive Multi-Step Forecasting with the Walk Forward Optimisation Approach

In time series forecasting problems, typically one step is predicted. To be able to achieve the 3-month forecast, multi-step forecasting techniques must be applied. The technique used in this study is called recursive multi-step forecasting. This technique involves forecasting a single time step, then recursively appending prior forecasted values as part of the input to the next recursive forecast until the period desired is achieved. This strategy allows for multiple predictions in time and mimics the behavior of the traditional time series models. However, model performance degrades rapidly as errors from individual predictions accumulate (Brownlee 2017). This is why it could be suggested to use this technique for shorter term forecasts. A study carried out by (Marcellino, Stock and Watson 2006) found that iterated forecasting outperformed direct forecasting techniques for both univariate and multivariate models.

Since multi-step forecasts are being produced, a walk forward validation method was used to produce the forecast results. The walk forward validation technique works hand in hand with the recursive forecasting methods by assimilating new data as a new step in the prediction is made. After every one step prediction, the actual result is appended back into the dataset, and the one step model is retrained towards predicting the next step. This happens continuosly until the last forecast is done. This allows models to have access to the best available data at each forecasting step in the recursion.

The walk forward validation process splits the model optimization into multiple (individual) stages. Usually models are optimised wholly at once on the in-sample data and then applied on the out-of-sample (test) data. Compared to the other methods of cross validation, this approach prevents overfitting and provides a realistic approach of how the model would perform in a real-life scenario. This approach also takes into account the sequential nature of time series data and aids in continuously finding more robust parameters as each time step occurs. This gives higher confidence to out-of-sample predictions and prevents model overfitting. These techniques were applied to all the models.

1. Linear Regression Model

   Linear regression is a predictive modelling technique that works by finding a linear relationship between the forecast variable and any other explanatory variables that fits a line (best fit). In this case, we use it to find the trend patterns in the stock series over time with the use of time as the independent variable. For example, it may be the case that for a certain period in time, the stock closing price increases positively. The model should be able to capture that positive increase in trend. Usually, this linear technique is used to model cross-sectional data.

   However, it can also be applied to model non linear time series if it is utilised properly. It is worth noting that the linear model is a trend model. It works by capturing linear global trends with the use of a trend line. A basic linear regression will not predict the points accurately due to the continuous non-linearity of stock price data. However, when the recursive multi-step forecasting with walk forward validation technique is used, we are able to capture both local and global trends. This is because the model continuously updates the trend after each recursion and when new values become available.

   The simple linear model can be explained as:

   $$y = \varphi_0 + \varphi_1 x_1 + \cdots + \epsilon \tag{4.1}$$

   Where
   y is the forecast(dependent) variable and
   x serves as the independent variables

φ_0 is the intercept
φ_1 is the regression coefficient

In this case, we are using lagged values of our forecast and walk forward validation to predict future values. If the linear model were to be done without using walk forward validation, the forecast would have been less accurate. Instead of obtaining the patterns of the prices, we would have gotten a linear trend line.

2. Extreme Gradient Boost(XGBoost):
   Extreme Gradient Boost (XGBoost), is a relatively new gradient boosting approach that is well known for its efficiency in predictions. The method was discovered by (Chen and Guestrin 2016). It is well known for consistently outperforming single algorithm models and is efficient and scalable. It is a form of an ensemble method that combines a set of weaker independent regression trees in parallel to create a stronger predictive model to yield better results. The method of combining weak learners sequentially is known as boosting. They are combined sequentially to allow each new tree to correct the errors of the previous one. This process is repeated until no improvements in results can be found. The supervised datasets were passed into the model with walk forward validation. The loss used for this problem was the regression with squared loss.

3. Support Vector Regression (SVR):
   Support Vector Machines (SVM) is a boundary-based supervised learning approach that was discovered by Vladimir N. Vapnik in 1963 (Drucker et al. n.d.). SVM is a supervised machine learning model that utilises a hyperplane for performing classification tasks. Its objective is to classify the provided data by obtaining an optimal hyperplane that maximises the distance between the points on each class of the data while minimising margin violations. Support vector regression(SVR) was derived from SVM. Support vector machines are usually used for classification. It was noticed that the support vector technique could be used for regression tasks. SVR tackles regression tasks by trying to fit as many points as possible within the margins while minimising the number of points on both sides that violate the margins. While in usual regression tasks, the goal is to minimise the errors, for SVR, the goal is to find the values that fit within a certain threshold of the error.

The width of the hyperplane boundaries is controlled by the epsilon value. The kernel function transforms the predictors and the target values into a higher dimensional space to establish the boundaries. The kernel used for this model is the radial basis function due to its good generalisation. The C value was specified to be 1 and epsiolon was set to be 0.2.

The equation that minimises :

$$\text{MIN } 1/2 \; ||W||^2 \tag{4.2}$$

With the constraints given as:

$$|y_i - w_i x_i| <= \epsilon \tag{4.3}$$

4. Artificial Neural Networks: LSTM Network

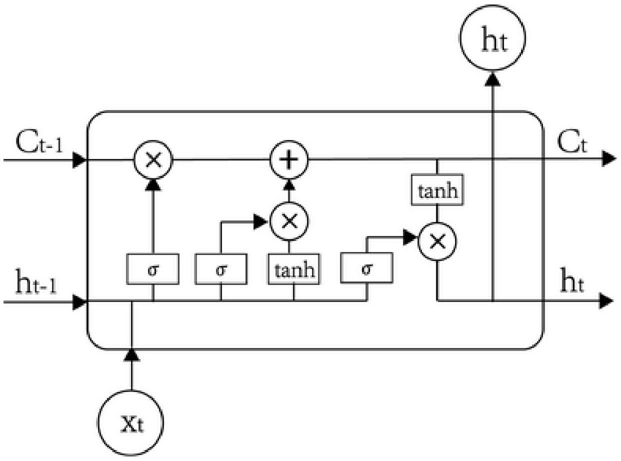


Figure 4.1: Structure of LSTM cell Qiu, Wang and Zhou 2020

LSTM networks are a form of powerful RNN (Recurrent Neural Networks) that use a theory called Long Short-Term Memory Cells. The cells have three important states: the input state, the forget state, and the output state. The forget state decides which time points need to be forgotten and which ones will be kept. The input state decides what information can be newly incorporated, and the output state decides what the next hidden state will be. LSTM networks are capable of storing information about previous occurrences over time. This has proved helpful when it comes to time series analysis, as the past prices have an impact on future prices.

Long Short-Term Memory (LSTM) networks handle the issue of unstable gradients, better known as the vanishing gradient problem, which affects basic ANN(Artificial Neural Networks). This issue affects earlier layers in the network during training as their gradients continue to drastically shrink, thus making the update to the weights of layers very minute. This deters the weights from reaching their optimal value, which hinders learning. This only then allows for short-term memory. LSTM solves the issue of vanishing gradients by using the gates to regulate the relevant information learnt by the model. These gates decide which information is relevant and which can be forgotten.

The data were first transformed to ensure stationarity by applying a differencing function. Differencing was applied once, which provided the returns of the stock. The LSTM model was then built using the combination of the supervised learning method and walk forward validation strategy, which was applied to the other machine learning methods. Any forecasted value was reloaded into the model to produce the next forecast. The data were replicated and lagged by a one-time frame to mimic the strategy of the traditional time series methods. The data were split for training and testing, and the minmaxscaler was applied to scale the respective datasets. The processed data were then fed into the neural network for training.

A sequential architecture that was motivated by (Mehtab, Sen and Dutta 2020) consisting of the input layer, one LSTM layer, and three fully connected dense layers was implemented to generate the predictions. The function for the stepwise forecast was inspired by an article on how to update the data fed into LSTM models during training by (Brownlee 2017).

4.5 Econometrics Models

In order to utilise the traditional time series (econometrics) models, extensive research was carried out into understanding the mechanisms of the models and the specific situations in which to use them. Traditional time series models are recursive, which means the initial forecasted value is used to generate the next forecast, compared to the machine learning approach, which directly produces the forecasts depending on the horizon, which is without dependence on the new forecasted values. Most of the statistical models only work well with stationary time series data. Auto-regressive (AR) models predict future values by regressing through past values and errors in the series (Pai and Lin 2005). In this paper, different types of auto-regressive models were examined.

Linear regression models work best when the explanatory variables are independent of one another to avoid multicollinearity(Xiaohua Wang et al. 2003). This explains why for most models, only the adjusted close price was used. The use of variables like volume are not independent of stock price.

1. Basic AutoRegressive AR(1) Model:
   An autoregressive model is a type of long memory forecasting model that forecasts future values of a time series based on previous values which are referred to as lags. For example, a model that depends only on the previous day price as a lag can be referred to as an AR(1) model. The formula can be explained as:
   $$y_t = \omega + \varphi_1 y_{t-1} + \epsilon_t \tag{4.4}$$
   Where y_t is the target variable and y_{t-1} is the previous day (lagged) target variable. φ is the vector coefficient and e_t, stands as the error term, which is assumed to be stationary white noise. This can be extended for multiple lags in the time series.

The formula can then be generalised as:

$$y_t = c + \varphi_1 y_{t1} + \varphi_2 y_{t2} \boxed{+} + \varphi_p y_{tp} + \epsilon_t \tag{4.5}$$

Where P is the number of lags considered.

2. ARIMA and ARIMAX:
   ARIMA
   The ARIMA model, which stands for Autoregressive Integrated Moving Average, is a type of statistical model that was introduced in 1970 by George Box and Gwilym Jenkins. It combines the power of two distinct models. The simple autoregressive(AR) model and the moving average(MA) model. The model works with stationary data by using lagged values of themselves as predictors. The ARIMA model caters to data that are non-stationary. It does this by performing differencing of the observations to achieve a level of stationarity. The model takes care of the moving average of the data.

   The I stands for integration, which is the amount of differencing to achieve stationarity. Instead of predicting the time series itself, ARIMA predicts the differences in the time series from the previous to the next time stepin this case, the return between the previous day and the next day. However, most economic series are non-stationary. This is why the stock price data needed differencing to ensure stationarity. The ARIMA model works only with non- seasonal time series data that is non random. The simple ARIMA is a univariate model, which means it only takes in single variable input. In this case, the adjusted daily close price was used as the input to the model.

   The ARIMA(p,d,q) model can be defined as:

$$Y = (\text{Auto-Regressive Parameters}) + (\text{Moving Average Parameters}) \tag{4.6}$$

   The AR(p) auto-regression term represents the autoregressive term. This represents the lag value needed to be used.

   The I(d) integration represents the non-seasonal differencing term. This represents the amount of differencing required to allow the dataset to be able to be fitted to the ARIMA model.

   The MA(q) moving average is the order of the moving average term. This represents the number of lagged forecast errors.

   ARIMAX

   The ARIMAX model is a variant of the ARIMA model that utilises exogenous explanatory variables as inputs into the model. It is a multivariate alternative to the basic ARIMA Model. The

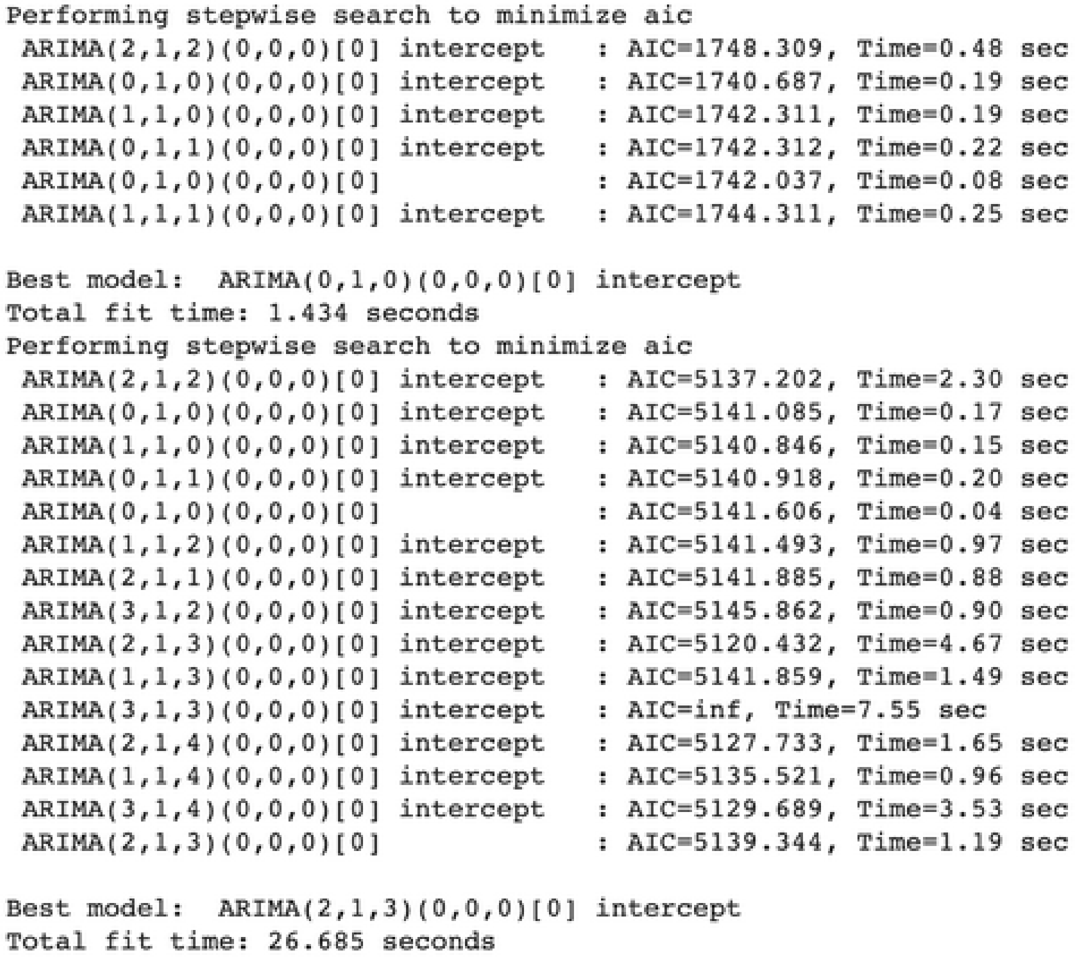
ARIMAX model uses lagged values of the dependent variable and the other explanatory variable to forecast future values.

4.5.1 Using Auto-ARIMA for Parameter Estimation

The auto-ARIMA function was used to find an appropriate order of terms for the models being tested. It traverses through different ARIMA configurations and finds the optimal parameters according to a specified criterion for the p, q, and d values of a dataset. It performs a stepwise search through multiple values until it finds a combination that minimizes the criterion.

This study uses the Akaike information criterion (AIC) as an estimator for prediction error to determine suitable parameters that would be used for the models . It is a model selection tool that works by estimating the prediction error of a dataset on different statistical models.

he process was used to estimate the best parameters for the amount of differencing required (d) and the lag length of both the autoregressive term (p) and the moving average term (q) that minimise the AIC for both the ARIMA and ARIMAX models. Figure 4.2 shows the combinations that were tested for each dataset and the chosen terms that reduced the AIC.



```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=1748.309, Time=0.48 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=1740.687, Time=0.19 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=1742.311, Time=0.19 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=1742.312, Time=0.22 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=1742.037, Time=0.08 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=1744.311, Time=0.25 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0] intercept
Total fit time: 1.434 seconds
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=5137.202, Time=2.30 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=5141.085, Time=0.17 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=5140.846, Time=0.15 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=5140.918, Time=0.20 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=5141.606, Time=0.04 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=5141.493, Time=0.97 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=5141.885, Time=0.88 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=5145.862, Time=0.90 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=5120.432, Time=4.67 sec
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=5141.859, Time=1.49 sec
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=inf, Time=7.55 sec
 ARIMA(2,1,4)(0,0,0)[0] intercept   : AIC=5127.733, Time=1.65 sec
 ARIMA(1,1,4)(0,0,0)[0] intercept   : AIC=5135.521, Time=0.96 sec
 ARIMA(3,1,4)(0,0,0)[0] intercept   : AIC=5129.689, Time=3.53 sec
 ARIMA(2,1,3)(0,0,0)[0]             : AIC=5139.344, Time=1.19 sec

Best model:  ARIMA(2,1,3)(0,0,0)[0] intercept
Total fit time: 26.685 seconds
```

Figure 4.2: Auto-ARIMA stepwise parameter search

Differencing: It was concluded in previous sections that stock price data are not stationary. In this case, the auto-ARIMA suggested that a first-order differencing of both datasets was enough to make the data usable by the models. This was also tested out in fig 3.8 and 3.9 which the differenced data successfully rejected the null hypothesis. Differencing a stock price series produces the daily returns($y_2 - y_1$) which could sometimes be stationary but.

The auto-ARIMA suggested an ARIMA(0,1,0) for the VZ dataset and ARIMA(2,1,3) for the MCHP dataset. ARIMA(0,1,0) represents a time series that is not stationary, and the order of the p and q terms of value 0 mean the model is a random walk. It turned out that modelling the data as a random walk did not provide the best results. It is valuable to note that the auto-ARIMA does not always provide the best combination of terms. The first order differencing of the series may not guarantee that the combination gotten from the auto-ARIMA will be the best throughout the time series. This is why the autocorrelation plots and diagnostic plots were re-examined and other terms were tested in order to achieve better forecasts.

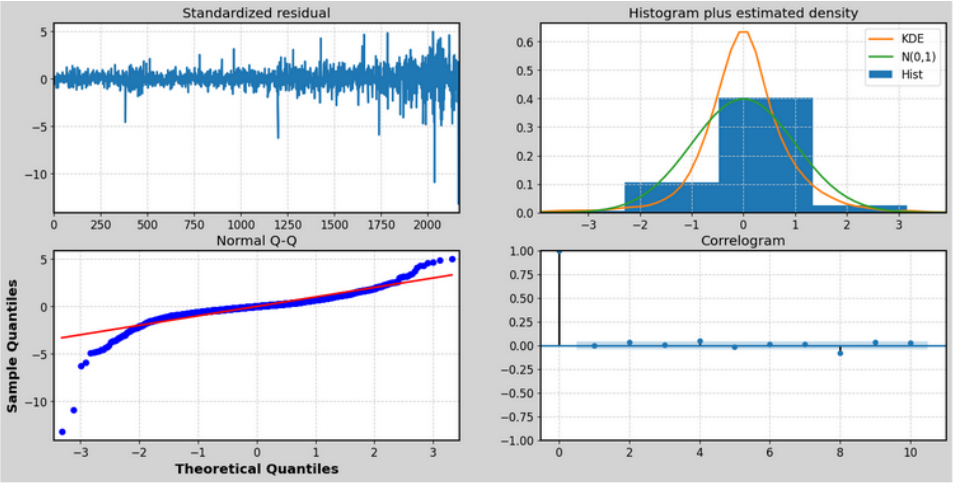


Figure 4.3: MCHP Diagnostic Plots

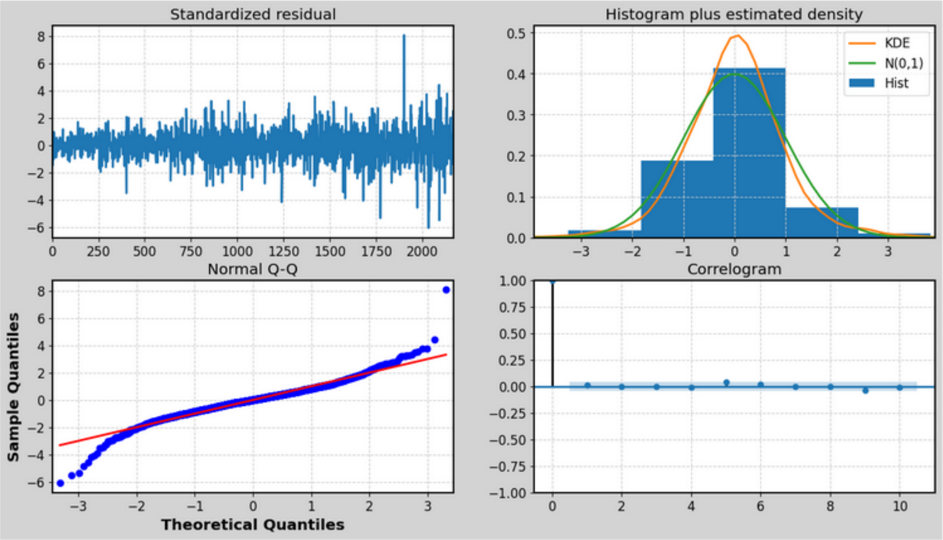


Figure 4.4: VZ Diagnostic Plots

The diagnostic plots in Figures 4.4 and 4.3 help to understand the model fit, residuals, density, and auto-correlation of fitted values.

The standardised residual plots (top left) are mostly contained between 2 to -2, centered around 0, and have the same variance throughout, which suggests that the data may be stationary.

The Q-Q plot allows for comparison of two probability distributions. It should ideally have all its points along the red line. From the plot, it can be seen that most blue dots align with the red line, except after the -2:2 range, though there does not seem to be any heavy outliers which may raise questions.

The density plot looks at residuals distribution, which seems to be normally distributed for both datasets.

The ACF plot or correlogram shows the correlation between the residuals. If the blue dots values were above zero, it would suggest that there might be a need for other predictors for better modelling.

An interesting observation is that the residuals seem to not have constant variance, but rather a periodical fluctuation. That pattern can also be seen in Q-Q plot and density plot, though the ACF looks normal.

The models were eventually fit based on the terms of (1,1,1) for the Verizon datasets and (2,1,3) for the Microchip datasets.

3. Vector Autoregressive model:
   The Vector Autoregressive model is a multivariate time series model. The VAR model builds on the univariate AR model to allow for multiple variables. The model captures the relationship between an independent variable of the time series and lagged versions of itself and also the lagged values of other dependent variables. It looks at how lagged values of the explanatory variables can affect the future values of the main variable. In this case, the model predicts all variables, including volume, open price, close price, and adjusted price.

   The VAR(1) model formula can be defined as:

$$y_t = \alpha + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \epsilon_t \tag{4.7}$$

   The formula multiplies to how many variables are available

   where:
   yt (y1, y2, ⋯, yn) : an (n x 1) vector of time series variables
   αv : is the vector of the intercepts
   φ_p(p=1, 2, ... , p) : coefficient matrices
   ϵt: Is the vector for the error term(white noise)

4. Facebook Prophet and Neural Prophet Models:
   Facebook developed an open-source library for time series analysis, which has been acknowledged as a good forecasting alternative due to its relative ease of use and positive performance. It works similarly to the AR models. The prophet model leverages its ability to detect change points in trends and also account for seasonality. The model utilises three main components: trends, seasonality, and holiday effects. The advantage of prophet is that it can automatically detect change points in trends. It utlises the fourier series as a function of time for the seasonality function.

   The Neural prophet model builds up on the prophet model. It utilises the use of the AR-Net and Facebook Prophet libraries to forecast using neural networks.

   The Prophet model formula can be described as:

$$y(t) = g(t) + s(t) + h(t) + e(t) \tag{4.8}$$

   where:
   g(t) : represents the trend factor
   s(t) : represents the repeated seasonality component
   h(t) : represents the holiday component
   e(t) : represents the error term(white noise)

   An issue faced with the prophet model was, unlike other time series methods, it did not consider the stock prices series are only available 5 out of 7 days a week. It made 7-day predictions including the weekends. The data was restructured to remove the weekend prices so as to be in line with the other models

   The predictions, however, should not be affected by this issue. The weekends were eventually extracted from the prediction to allow for equality in predictions.

4.6 Ensemble Model

The Ensemble Model approach was proposed to provide a better predictive model accuracy by combining the power of individual learning algorithms to create a more robust model. Ever so often, purely linear or non-linear models may not perform very efficiently due to their individual flaws. The base models all make unique assumptions of their respective predictions. The ensemble method strengthens the prediction by extracting the best part of all the chosen base models, thereby offsetting the biases of the individual models. Unlike the XGBoost method, the Ensemble Method does not focus only on combining weak learners. Any model output can be combined with another to make an ensemble. The Ensemble Model is made by the combination of multiple models in order to achieve better predictive power. In a real-world scenario, this would be a more appropriate method to try.

The chosen Ensemble Method for this project was the Stacked Generalisation Method. This is a popular Ensemble Method that builds off of the Bagging Method used by the Random Forest Algorithm.

Unlike bagging, the Stacking Method allows us to give weights to each of the models based on a certain specified criteria. This allows for the better-performing models to be weighed stronger compared to the poor-performing models, which could provide better predictive accuracy. Stacked Ensemble Methods have also been proven to perform better than the individual pure linear and nonlinear models due to the bias offset. The structure of the code for the ensemble methods was suggested by (*Stock Prediction with ML: Ensemble Modeling The Alpha Scientist* 2021). Different combinations of base models were considered for this study in order to achieve better performances. The Ensemble Methods were chosen and experimented with using the following criteria:

1. Combining pairs of machine learning and traditional time series models
2. Combining models based on individual performance

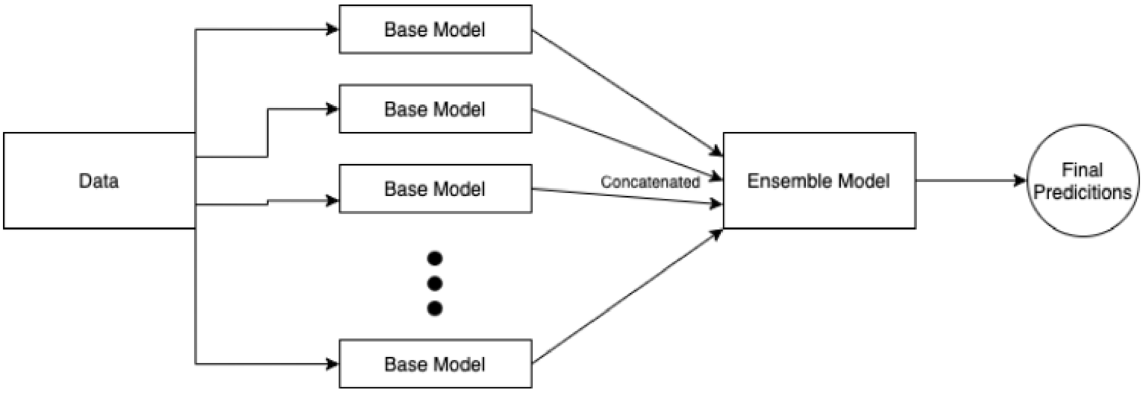


Figure 4.5: Ensemble learner flowchart

As seen in Figure 6.2, ensemble forecasting uses the idea of feeding different predictions into another meta model that handles the ensembling. The meta model used in this study is the Least Absolute Shrinkage and Selection Operator (LASSO) method, which is a regularisation method that performs selection and regularization to help to shrink inputted variables towards a central point. It is also one of the few linear learners that allows us to assign only positive weightings to all models. The results from the already made base models were concatenated and fed into the stacked (LASSO) model to achieve the final prediction. The same walk forward validation approach used in the other model was used for the Ensemble Model also.

5 Evaluation

In order to ensure a fair and unbiased evaluation process, the same metrics were used across all models to measure the accuracy of the empirical results produced. For this study, the models were evaluated based on multiple error metrics and also with respect to generalisation with type of data also taken into account. Evaluating forecasting methods is usually focused on the predictive power (generalisation) rather than the model fit. This is because the model could fit well on the training data but may not be able to generalise well on the unseen data. The strength of the models should also be judged by performance with respect to the type of data used. This section analyses the performance of the models based on two criteria:

1. Goodness of fit and model generalisation (error metrics)
2. Model performance with respect to the dataset used (volatility)

5.1 Error Metrics

The four error metrics used were Mean Absolute Error (MAE), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). These criteria were commonly used in the literature (Vijh et al. 2020) to assess time series model performance when used for stock price forecasting.

An error can be defined as:

$$e_t = X_t - Y_t \tag{5.1}$$

Where X_t is the true value, Y_t is the predicted value.

The models were tested on both datasets (VZ & MCHP) and the error values were computed for both instances.

1. Root Mean Square Error(RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2} \tag{5.2}$$

   The Root Mean Square Error (RMSE) proves to be a good measure, as it penalises larger errors. RMSE is also easily interpretable, as it has the same units as the variable being estimated. The RMSE of all models will be computed and used for evaluation. The RMSE is not useful in absolute terms. It is beneficial when compared to the values passed onto the model.

2. Mean Absolute Error(MAE):

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{5.3}$$

The Mean Absolute Error (MAE) provides an average of the summation of the absolute difference between the forecasted and the actual values. MAE is useful for understanding the magnitude of the errors, although it doesn't consider the direction of the error, but rather the magnitude of deviation from the real values.

3. Mean Absolute Percentage Error(MAPE):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|Pe_t| \tag{5.4}$$

The Mean Absolute Percentage Error (MAPE) works like the MAE, but it provides an error in terms of percentages. MAPE has its limitations, as it does not work well with data points of which there are zero, and is also quite biased towards negative forecast errors (Makridakis, Hibon and Moser 1979). However, it allows for a percentage comparison between models. A lower MAPE means there is a lower percentage of errors.

4. Mean Square Error(MSE):
The Mean Square Error (MSE) measures the average squared difference of the forecasted values to the actual. The lower the MSE, the more accurate the line of fit is to the actual. However, this metric does not work well with noisy data due to the squaring of errors.

5.2 Results

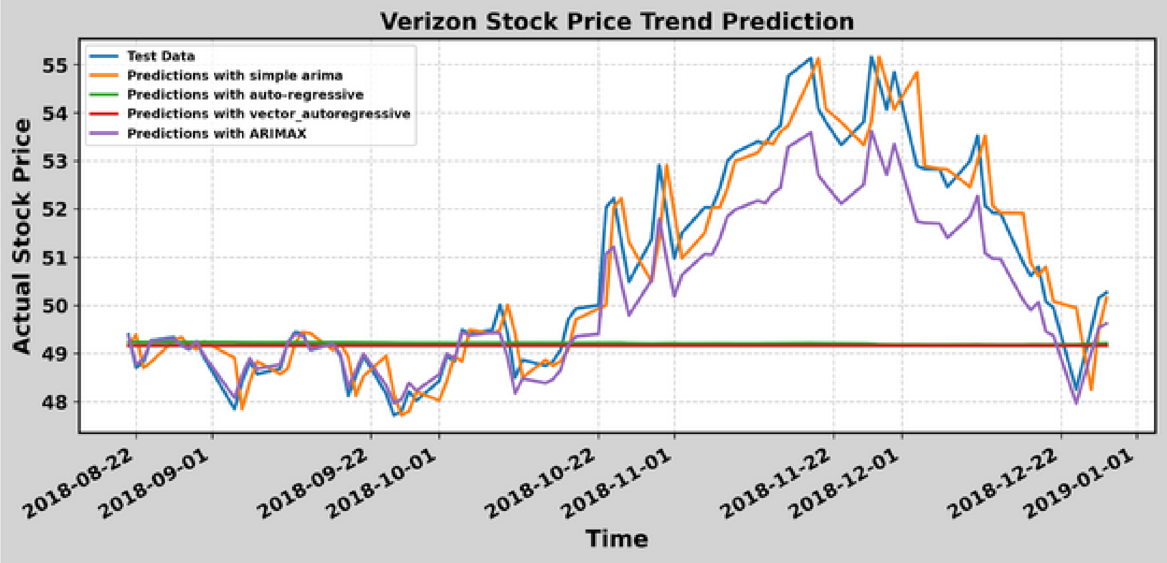


Figure 5.1: Time Series Models performance: VZ

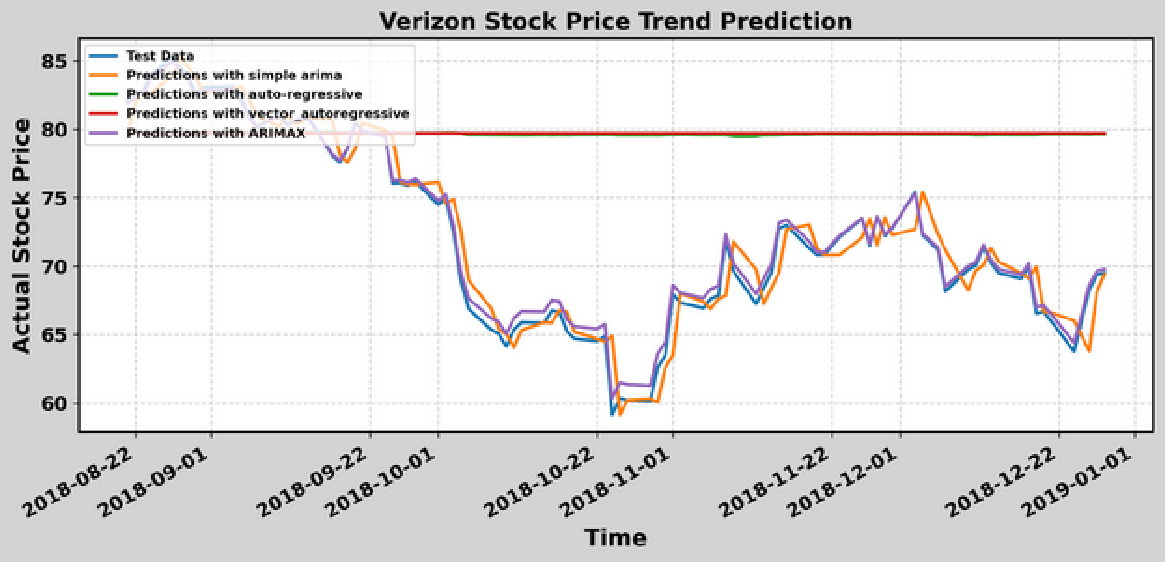


Figure 5.2: Time Series Models performance: MCHP

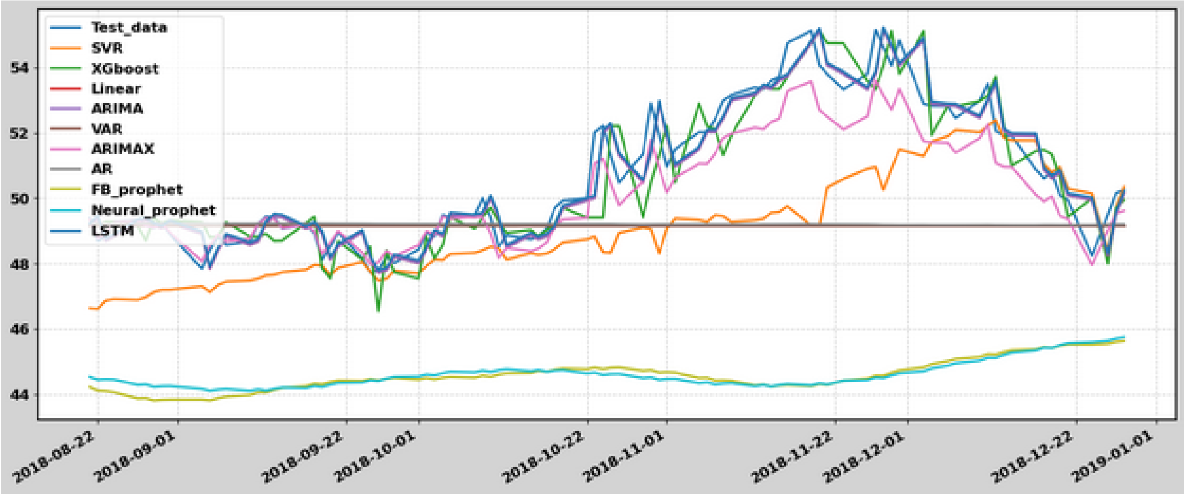


Figure 5.3: Verizon All Models

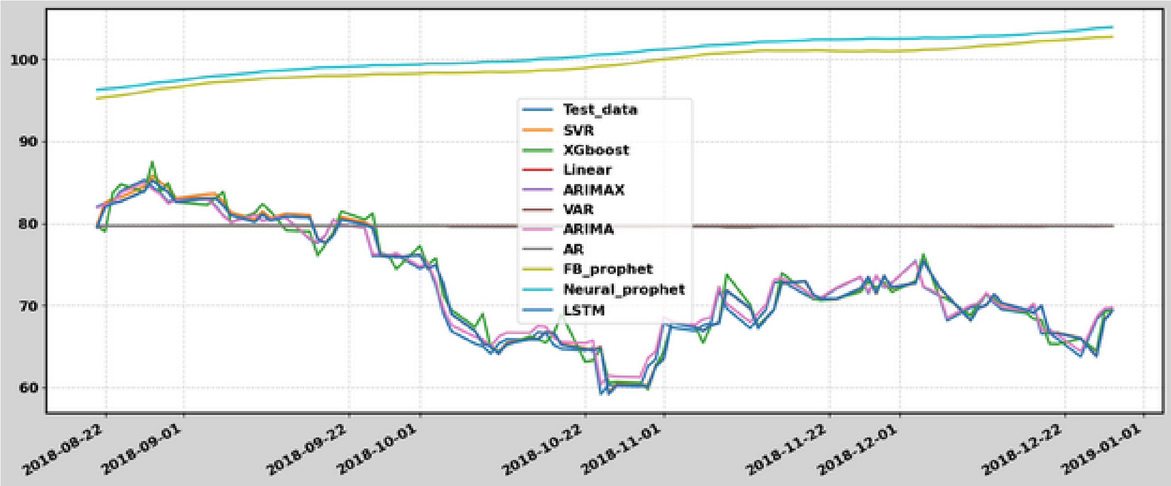


Figure 5.4: Microchip All Models

Table 5.1: Verizon Errors

	MSE	MAE	RMSE	MAPE
ARIMAX	0.625174	0.614676	0.790679	0.011769
AR	6.622628	1.886823	2.573447	0.035981
VAR	6.715572	1.894561	2.591442	0.036112
ARIMA	0.435858	0.495963	0.660195	0.009738
XGBOOST	0.798307	0.652950	0.893480	0.012753
Linear	0.436485	0.496249	0.660670	0.009744
SVR	5.135176	1.808547	2.266093	0.034974
LSTM	0.439805	0.489644	0.663178	0.009616
Facebook Prophet (Verizon)	41.512701	6.108388	6.443035	0.119121
Facebook NN (Verizon)	41.619812	6.091421	6.451342	0.118715

Table 5.2: MCHP Errors

	MSE	MAE	RMSE	MAPE
ARIMAX	0.277082	0.412605	0.526386	0.006124
AR	100.436901	8.536501	10.021821	0.125826
VAR	102.218314	8.612218	10.110307	0.126945
ARIMA	2.988795	1.273039	1.728813	‘0.018031
XGBOOST	3.207820	1.349863	1.791039	0.019007
Linear	2.976159	1.268015	1.725155	0.017957
SVR	3.011030	1.293838	1.735232	0.017987
LSTM	2.979673	1.270768	1.726173	0.017987
Facebook Prophet (Microchip)	802.558110	27.172844	28.329457	0.389811
Facebook NN (Microchip)	872.549927	28.392254	29.538956	0.407029

Figure 5.3 and 5.4 show the visual performance of the forecasts of all models for the last 90 days in the two datasets. It can be seen that most models accurately forecasted the overall change in trends. However, the AR, VAR and Facebook Prophet models did not perform well.

Tables 5.1 and 5.2 display the error metrics of all the models on the two datasets. The best values for each error metric are highlighted. In general, the models performed weaker on the more volatile dataset due to the increase in difficulty of prediction. All machine learning models displayed a relatively good performance across both datasets. The XGBoost model performed poorly in comparison to the rest of the ML models on the more volatile dataset across all four metrics. For the ML models, the difference in accuracy between the training and test sets were negligible, which hints that the models were generalising well and were not overfitted to the train set. There was a relatively significant increase in error between training and testing for the XGBoost model(see 6.5 and 6.6).

Furthermore, it can also be observed that the two ARIMA model variations outperformed all other machine learning models on both datasets. The VAR and AR models performed poorly with relatively high error in all four error categories compared to other models. The AR and VAR models are not usually used for stock price prediction due to their inability to capture non-linear movements. It can be seen from Figure 5.1 that both models predicted linear trend lines. The trend lines forecasted contradict the overall trend of the actual values. For example, in Figure 5.2, when the real values move on a downtrend, both AR models predict a level trend.

The Prophet models also performed poorly in comparison to the other traditional time series and machine learning models. This may Comparing the results, we see that the models performed considerably better on the less volatile dataset (VZ) as compared to the more volatile dataset (MCHP). This can be seen by comparing the general MAPE of the VZ data to the MCHP data. The LSTM performed considerably well, however, it failed to outperform the simple linear models. As stated in a study by (Makridakis, Spiliotis and Assimakopoulos 2018), It would be expected that the more complex methods such as the LSTM would outperform simpler methods like the ARIMA, however that is not the case. The LSTM however provided a good forecast in terms of capturing overlying trends. It achieved the best MAE when applied to the Verizon dataset.

5.3 Ensemble Model Approach:

Based on the performance of all individual models, the following combinations were considered as base models for use to create the ensemble forecasts. These are a combination of performance models and linear and non-linear models.

1. ARIMA SVR Ensemble
2. Linear ARIMA Ensemble
3. Linear ARIMAX Ensemble
4. ARIMAX XGBoost Ensemble

Table 5.3: Comparison of Ensemble errors to base model errors: MCHP

	MSE	MAE	RMSE	MAPE
BASE MODELS				
ARIMAX	0.277082	0.412605	0.526386	0.006124
AR	100.436901	8.536501	10.011821	0.125826
VAR	102.218314	8.612218	10.110307	0.126945
ARIMA	2.988795	1.273039	1.728813	0.018031
XGBOOST	3.207820	1.349863	1.791039	0.019007
Linear	2.976159	1.268015	1.725155	0.017957
SVR	3.011030	1.293838	1.735232	0.018279
LSTM	2.979673	1.270768	1.726173	0.017987
Facebook Prophet (Microchip)	802.558110	27.172844	28.329457	0.389811
Facebook NN (Microchip)	872.519927	28.392254	29.538956	0.407029
ENSEMBLE MODELS				
Arimax_Linear	2.906908	1.264499	1.704966	0.017902
Linear_Arima	0.024145	0.134615	0.155385	0.001942
Arima_SVR	0.024145	0.134615	0.155385	0.001942
Arimax_LSTM	2.906908	1.264499	1.704966	0.017902

Table 5.4: Comparison of Ensemble errors to base model errors: Verizon

	MSE	MAE	RMSE	MAPE
ENSEMBLE MODELS				
Arimax_Xgboost	0.043371	0.181661	0.208257	0.003652
ARIMAX_LSTM	0.043371	0.181661	0.208257	0.003652
ARIMA_Linear	0.423771	0.484903	0.650977	0.009510
Linear_ARIMAX	0.423810	0.484967	0.651007	0.009511
BASE MODELS				
LSTM	0.439805	0.489644	0.663178	0.009616
ARIMA	0.435858	0.495963	0.660195	0.009738
Linear	0.436485	0.496249	0.660670	0..009744
ARIMAX	0.625174	0.614676	0.790679	0.011769
XGBOOST	0.798307	0.652950	0.893480	0.012753
SVR	5.135176	1.808547	2.266093	0.034974
AR	6.622628	1.886823	2.573447	0.035981
VAR	6.715572	1.894561	2.591442	0.036112
Facebook NN (Verizon)	41.619812	6.091421	6.451342	0.118715
Facebook Prophet (Verizon)	41.512701	6.108388	6.443035	0.119121

From the Ensemble Models error metric, It can be seen that predictions greatly improved. The ensemble forecasts provide more consistency and accuracy of predictions compared to the single methods. The usage of the LASSO Algorithm proved significant because the input (X) variables are a combination of the predicted (Y) values based on other traditional time series models and machine learning models. With the LASSO method, we can reduce the variance of estimators(reducing errors) while not greatly increasing model bias(overfitting). Other models, like the Random Forest (Tree-Based) Model, could have been used to combine the ensembles, but after testing several options, it was clear that the LASSO method produced the best results. Other models produce the same results for different X variables, and those did not look feasible.

The linear-ARIMA and ARIMA-SVR ensembles achieved the lowest errors across all four metrics for MCHP. The ARIMAX-XGBoost and ARIMAX-LSTM attained the lowest errors For VZ. From intuition, The expectation was for the ARIMAX-Linear to perform best out of all the ensemble models as it had the lowest errors for its base models. However this was not the case

It is worth noting that ensemble models do not automatically guarantee better results. However when the right combinations are implemented, forecast results can be highly reinforced.

5.4 Further Work and Robustness:

This study effectively serves as a foundation for more robust forecasting systems. There are a variety of possibilities for further research:

1. The first could be the combination of technical and fundamental analysis for utilising alternative data and macroeconomic indicators as feature inputs. The more causal features that are available, the greater chance for boosting accuracy. It will be important to see how the multivariate models perform with these features.

2. The second expansion would be the Application of models to data from different markets(eg. Emerging markets). As concluded by (Mallikarjuna and Rao 2019), models behave differently depending on the type of data used. Especially data from different markets.

3. The third suggestion is to build a multiple ensemble forecasting technique that utilises an expert system to pick the best ensemble for each forecast step while using the same multi step forecast and walk forward validation approach applied in this paper.

6 Conclusion:

Predicting stock prices is complex due to the plethora of factors that influence prices in their different ways. Some factors which are unpredictable. Using models for forecasting stock prices allows for making more informed decisions. Both traditional time series and machine learning methods are geared towards the same goal: to create predictions while minimising the specified error between the actual and the forecasted. The main importance is to have a model that works efficiently for the type of forecasting needed. Based on the results, the time series models still outperform the machine learning and deep learning methods for multi-step forecasting on univariate datasets. The two variants of the ARIMA models outperformed all singular machine learning models across the four metrics, although other time series models like the VAR and AR model failed to capture the patterns correctly, thus leading to relatively high errors. There are limited choices in terms of the statistical models that can handle forecasting nonlinear patterns like stock prices. It should be noted that the machine learning models in general performed considerably well. Although they didnt have the lowest errors, non of the ML methods produced bad forecasts.

Furthermore, there also stands some difficulties when comparing accuracies of multiple models. The first is most research done in this field have applied their proposed models on one, or a few sets of data. A model might perform considerably well on a dataset, and end up not as good on another dataset, as shown in this paper. This shows the importance of applying further statistics based on the forecasting objective such as statistical significance tests to accurately gauge the quality of forecasts. Another thing to note is that market cycles change, and trends reverse. If the models are not able to identify the surprises, then the confidence in these models will be lost.

It was also realised that combining models in ensembles could provide much more efficient forecast results. It proves a great way of capturing the best attributes of each model. Forecasting only using base models for stock price forecasting is limited in its use cases. The markets are way too complex to be forecasted only using single base models. More robust approaches, like the ensemble and hybrid methods would need to be considered in order to achieve more efficient results. In an applicative scenario, these models would need to be constantly reevaluated as markets and trends continue to evolve.

Bibliography

[1] 2021. URL: https://www.bloomberg.com/professional/product/bloomberg-market-concepts/.

[2] 2021, p. 2. URL: https://www.berkshirehathaway.com/letters/2014ltr.pdf.

[3] Ayodele Adebiyi, Aderemi Adewumi and Charles Ayo. "Stock price prediction using the ARIMA model". In: Mar. 2014. DOI: 10.1109/UKSim.2014.67.

[4] Khaled A. Althelaya, El-Sayed M. El-Alfy and Salahadin Mohammed. "Evaluation of bidirectional LSTM for short-and long-term stock market prediction". In: *2018 9th International Conference on Information and Communication Systems (ICICS)*. Irbid: IEEE, Apr. 2018, pp. 151–156. ISBN: 978-1-5386-4366-2. DOI: 10.1109/IACS.2018.8355458. URL: https://ieeexplore.ieee.org/document/8355458/ (visited on 07/05/2021).

[5] Muhammad Anhar. "The Direct and Indirect Influence of Company Performance, Investors Expectation and Investment Risk on Individual Stock Price Index at Indonesia Stock Exchange". en. In: *Research Journal of Finance and Accounting* 6.1 (2015), p. 108. ISSN: 2222-1697. URL: https://iiste.org/Journals/index.php/RJFA/article/view/19155 (visited on 02/05/2021).

[6] *ARIMA Time Series Data Forecasting and Visualization in Python*. en. URL: https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3 (visited on 09/11/2020).

[7] AssetCorr. *Asset Correlation*. 2021. URL: https://www.portfoliovisualizer.com/asset-correlations.

[8] Gianluca Bontempi, Souhaib Ben Taieb and Yann-Aël Le Borgne. "Machine Learning Strategies for Time Series Forecasting". en. In: *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*. Ed. by Marie-Aude Aufaure and Esteban Zimányi. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer, 2013, pp. 62–77. ISBN: 978-3-642-36318-4. DOI: 10.1007/978-3-642-36318-4_3. URL: https://doi.org/10.1007/978-3-642-36318-4_3 (visited on 15/04/2021).

[9] George EP Box et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[10] Jason Brownlee. *4 Strategies for Multi-Step Time Series Forecasting*. en-US. Mar. 2017. URL: https://machinelearningmastery.com/multi-step-time-series-forecasting/ (visited on 08/05/2021).

[11] ANTÓNIO CERQUEIRA. "Autocorrelation in Daily Stock Returns". In: Working Paper, 4th PFN Conference. 2006.

[12] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. en. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, Aug. 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: https://dl.acm.org/doi/10.1145/2939672.2939785 (visited on 05/05/2021).

[13] J. Christy Jackson et al. “Stock market analysis and prediction using time series analysis”. en. In: *Materials Today: Proceedings* (Jan. 2021), S2214785320389823. ISSN: 22147853. DOI: 10.1016/j.matpr.2020.11.364. URL: https://linkinghub.elsevier.com/retrieve/pii/S2214785320389823 (visited on 07/05/2021).

[14] Zhong Darrat. “On Testing the Random-Walk Hypothesis: A Model-Comparison Appr”. en. In: (), p. 31.

[15] Harris Drucker et al. “Support Vector Regression Machines”. en. In: (), p. 7.

[16] Jithin Eapen, Doina Bein and Abhishek Verma. “Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction”. In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. Las Vegas, NV, USA: IEEE, Jan. 2019, pp. 0264–0270. ISBN: 978-1-72810-554-3. DOI: 10.1109/CCWC.2019.8666592. URL: https://ieeexplore.ieee.org/document/8666592/ (visited on 07/05/2021).

[17] Robert W. Faff and Michael D. McKenzie. “The relationship between implied volatility and autocorrelation”. en. In: *International Journal of Managerial Finance* 3.2 (Apr. 2007), pp. 191–196. ISSN: 1743-9132. DOI: 10.1108/17439130710738736. URL: https://www.emerald.com/insight/content/doi/10.1108/17439130710738736/full/html (visited on 04/05/2021).

[18] Eugene F. Fama. “The Behavior of Stock-Market Prices”. en. In: *The Journal of Business* 38.1 (Jan. 1965), p. 34. ISSN: 0021-9398, 1537-5374. DOI: 10.1086/294743. URL: https://www.jstor.org/stable/2350752 (visited on 07/05/2021).

[19] Massimo Guidolin et al. “Does the Macroeconomy Predict UK Asset Returns in a Nonlinear Fashion? Comprehensive Out-of-Sample Evidence”. en. In: *Oxford Bulletin of Economics and Statistics* 76.4 (Aug. 2014), pp. 510–535. ISSN: 03059049. DOI: 10.1111/obes.12035. URL: http://doi.wiley.com/10.1111/obes.12035 (visited on 20/04/2021).

[20] Michael van Gysen, Chun-Sung Huang and Ryan Kruger. “The Performance Of Linear Versus Non-Linear Models In Forecasting Returns On The Johannesburg Stock Exchange”. en. In: *International Business & Economics Research Journal (IBER)* 12.8 (July 2013). Number: 8, pp. 985–994. ISSN: 2157-9393. DOI: 10.19030/iber.v12i8.7994. URL: https://www.clutejournals.com (visited on 05/04/2021).

[21] Campbell R. Harvey. “Forecasts of Economic Growth from the Bond and Stock Markets”. In: *Financial Analysts Journal* 45.5 (1989), pp. 38–45. DOI: 10.2469/faj.v45.n5.38.

[22] Osman Hegazy, Omar S. Soliman and Mustafa Abdul Salam. “A Machine Learning Model for Stock Market Prediction”. In: *arXiv:1402.7351 [cs]* (Feb. 2014). arXiv: 1402.7351. URL: http://arxiv.org/abs/1402.7351 (visited on 20/04/2021).

[23] Kenneth E. Homa and Dwight M. Jaffee. “THE SUPPLY OF MONEY AND COMMON STOCK PRICES”. en. In: *The Journal of Finance* 26.5 (Dec. 1971), pp. 1045–1066. ISSN: 00221082. DOI: 10.1111/j.1540-6261.1971.tb01747.x. URL: http://doi.wiley.com/10.1111/j.1540-6261.1971.tb01747.x (visited on 19/04/2021).

[24] Gourav Kumar, Sanjeev Jain and Uday Pratap Singh. “Stock Market Forecasting Using Computational Intelligence: A Survey”. en. In: *Archives of Computational Methods in Engineering* (Feb. 2020). ISSN: 1134-3060, 1886-1784. DOI: 10.1007/s11831-020-09413-5. URL: http://link.springer.com/10.1007/s11831-020-09413-5 (visited on 09/11/2020).

[25] Blake LeBaron. “Some Relations Between Volatility and Serial Correlations in Stock Market Returns”. In: *The Journal of Business* 65.2 (1992). Publisher: University of Chicago Press, pp. 199–219. ISSN: 00219398, 15375374. URL: http://www.jstor.org/stable/2353162 (visited on 04/05/2021).

[26] Kyung Joo Lee et al. “Forecasting korean stock price index (kospi) using back propagation neural network model, bayesian chiaos model, and sarima model”. In: *Journal of Management Information and Decision Sciences* 11.2 (2008).

[27] lplresearch. *Examining Bear Market Bounces*. en-US. Mar. 2020. URL: https://lplresearch.com/2020/03/31/examining-bear-market-bounces/ (visited on 09/05/2021).

[28] Spyros Makridakis, Michele Hibon and Claus Moser. “Accuracy of Forecasting: An Empirical Investigation”. In: *Journal of the Royal Statistical Society. Series A (General)* 142.2 (1979), p. 97. ISSN: 00359238. DOI: 10.2307/2345077. URL: https://www.jstor.org/stable/10.2307/2345077?origin=crossref (visited on 07/05/2021).

[29] Spyros Makridakis, Evangelos Spiliotis and Vassilios Assimakopoulos. “Statistical and Machine Learning forecasting methods: Concerns and ways forward”. In: *PLOS ONE* 13.3 (Mar. 2018), pp. 1–26. DOI: 10.1371/journal.pone.0194889. URL: https://doi.org/10.1371/journal.pone.0194889.

[30] Burton G. Malkiel. “Efficient Market Hypothesis”. In: *Finance*. Ed. by John Eatwell, Murray Milgate and Peter Newman. London: Palgrave Macmillan UK, 1989, pp. 127–134. ISBN: 978-1-349-20213-3. DOI: 10.1007/978-1-349-20213-3_13. URL: https://doi.org/10.1007/978-1-349-20213-3_13.

[31] Burton G. Malkiel. “The Efficient Market Hypothesis and Its Critics”. en. In: *Journal of Economic Perspectives* 17.1 (Mar. 2003), pp. 59–82. ISSN: 0895-3309. DOI: 10.1257/089533003321164958. URL: https://www.aeaweb.org/articles?id=10.1257/089533003321164958 (visited on 10/11/2020).

[32] Burton G. Malkiel and Eugene F. Fama. “EFFICIENT CAPITAL MARKETS: A REVIEW OF THEORY AND EMPIRICAL WORK*”. en. In: *The Journal of Finance* 25.2 (May 1970), pp. 383–417. ISSN: 00221082. DOI: 10.1111/j.1540-6261.1970.tb00518.x. URL: http://doi.wiley.com/10.1111/j.1540-6261.1970.tb00518.x (visited on 07/05/2021).

[33] M. Mallikarjuna and R. Prabhakara Rao. “Evaluation of forecasting methods from selected stock market returns”. en. In: *Financial Innovation* 5.1 (Dec. 2019), p. 40. ISSN: 2199-4730. DOI: 10.1186/s40854-019-0157-x. URL: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-019-0157-x (visited on 08/11/2020).

[34] Massimiliano Marcellino, James H. Stock and Mark W. Watson. “A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series”. en. In: *Journal of Econometrics* 135.1-2 (Nov. 2006), pp. 499–526. ISSN: 03044076. DOI: 10.1016/j.jeconom.2005.07.020. URL: https://linkinghub.elsevier.com/retrieve/pii/S030440760500165X (visited on 07/05/2021).

[35] Sidra Mehtab, Jaydip Sen and Abhishek Dutta. *Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models*. 2020. arXiv: 2009.10819 [q-fin.ST].

[36] Aditi Mittal. *Understanding RNN and LSTM*. en. Oct. 2019. URL: https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e (visited on 09/11/2020).

[37] *Multivariate Time Series Prediction with LSTM and Multiple features (Predict Google Stock Price)*. July 2020. URL: https://www.youtube.com/watch?v=gSYiKKoREFI&list=PLZJWvxgqdHl4iqNAeitr-jFw8-uvm_i5S&index=15 (visited on 08/11/2020).

[38] David MQ Nelson, Adriano CM Pereira and Renato A de Oliveira. “Stock market’s price movement prediction with LSTM neural networks”. In: *2017 International joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 1419–1426.

[39] Yibin Ng. *Machine Learning Techniques applied to Stock Price Prediction*. en. Oct. 2019. URL: https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001 (visited on 08/11/2020).

[40] Apostolos Nicholas Refenes, Achileas Zapranis and Gavin Francis. “Stock performance modeling using neural networks: A comparative study with regression models”. en. In: *Neural Networks* 7.2 (Jan. 1994), pp. 375–388. ISSN: 08936080. DOI: 10.1016/0893-6080(94)90030-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0893608094900302 (visited on 07/05/2021).

[41] Nist.gov. *6.4. Introduction to Time Series Analysis*. 2019. URL: https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm.

[42] Lucas Nunno. “Stock Market Price Prediction Using Linear and Polynomial Regression Models”. en. In: (), p. 6.

[43] Ping-Feng Pai and Chih-Sheng Lin. “A hybrid ARIMA and support vector machines model in stock price forecasting”. en. In: *Omega* 33.6 (Dec. 2005), pp. 497–505. ISSN: 03050483. DOI: 10.1016/j.omega.2004.07.024. URL: https://linkinghub.elsevier.com/retrieve/pii/S0305048304001082 (visited on 06/05/2021).

[44] W Penny and L Harrison. “Chapter 40: Multivariate autoregressive models”. en. In: (), p. 15.

[45] Jiayu Qiu, Bin Wang and Changjun Zhou. “Forecasting stock prices with long-short term memory neural network based on attention mechanism”. In: *PloS one* 15.1 (2020), e0227222.

[46] Akhter Mohiuddin Rather, Arun Agarwal and V.N. Sastry. “Recurrent neural network and a hybrid model for prediction of stock returns”. en. In: *Expert Systems with Applications* 42.6 (Apr. 2015), pp. 3234–3241. ISSN: 09574174. DOI: 10.1016/j.eswa.2014.12.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417414007684 (visited on 11/04/2021).

[47] U.S Securities and Exchange Commission. *Risk and return*. URL: https://www.investor.gov/additional-resources/information/youth/teachers-classroom-resources/risk-and-return.

[48] Ani Shabri. “Comparison of time series forecasting methods using neural networks and Box-Jenkins model”. In: *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics* 17 (2001), pp. 25–32.

[49] Jingyi Shen and M. Omair Shafiq. “Short-term stock market price trend prediction using a comprehensive deep learning system”. en. In: *Journal of Big Data* 7.1 (Dec. 2020), p. 66. ISSN: 2196-1115. DOI: 10.1186/s40537-020-00333-6. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6 (visited on 07/05/2021).

[50] Han Shih and Suchithra Rajendran. “Comparison of Time Series Methods and Machine Learning Algorithms for Forecasting Taiwan Blood Services Foundations Blood Supply”. en. In: *Journal of Healthcare Engineering* 2019 (Sept. 2019), pp. 1–6. ISSN: 2040-2295, 2040-2309. DOI: 10.1155/2019/6123745. URL: https://www.hindawi.com/journals/jhe/2019/6123745/ (visited on 10/11/2020).

[51] *Short-term stock market price trend prediction using a comprehensive deep learning system | SpringerLink*. URL: https://link.springer.com/article/10.1186/s40537-020-00333-6 (visited on 09/11/2020).

[52] *Stock Prediction with ML: Ensemble Modeling The Alpha Scientist*. URL: https://alphascientist.com/ensemble_modeling.html (visited on 07/05/2021).

[53] *Stock Prediction with ML: Walk-forward Modeling The Alpha Scientist*. URL: https://alphascientist.com/walk_forward_model_building.html (visited on 07/05/2021).

[54] *Stock price prediction and forecast using LSTM and ARIMA. | by Dhiraj Reddy | Medium*. URL: https://medium.com/@dhirajreddy13/stock-price-prediction-and-forecast-using-lstm-and-arima-52db753a23c7 (visited on 09/11/2020).

[55] *Stock Price Prediction Using Machine Learning | Deep Learning*. Oct. 2018. URL: https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/ (visited on 08/11/2020).

[56] P. K. Trivedi. “Time Series versus Structural Models: A Case Study of Canadian Manufacturing Inventory Behavior”. In: *International Economic Review* 16.3 (Oct. 1975), p. 587. ISSN: 00206598. DOI: 10.2307/2525998. URL: https://www.jstor.org/stable/2525998?origin=crossref (visited on 08/11/2020).

[57] Mehar Vijh et al. “Stock Closing Price Prediction using Machine Learning Techniques”. en. In: *Procedia Computer Science* 167 (2020), pp. 599–606. ISSN: 18770509. DOI: 10.1016/j.procs.2020.03.326. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050920307924 (visited on 07/05/2021).

[58] Bin Weng, Lin Lu et al. “Predicting short-term stock prices using ensemble methods and online data sources”. en. In: *Expert Systems with Applications* 112 (Dec. 2018), pp. 258–273. ISSN: 09574174. DOI: 10.1016/j.eswa.2018.06.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417418303622 (visited on 07/05/2021).

[59] Bin Weng, Waldyn Martinez et al. “Macroeconomic indicators alone can predict the monthly closing price of major U.S. indices: Insights from artificial intelligence, time-series analysis and hybrid models”. en. In: *Applied Soft Computing* 71 (Oct. 2018), pp. 685–697. ISSN: 15684946. DOI: 10.1016/j.asoc.2018.07.024. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494618304125 (visited on 11/05/2021).

[60] Xiaohua Wang et al. “Stock market prediction using neural networks: does trading volume help in short-term prediction?” In: *Proceedings of the International Joint Conference on Neural Networks, 2003.* Vol. 4. Portland, Oregon USA: IEEE, 2003, pp. 2438–2442. ISBN: 978-0-7803-7898-8. DOI: 10.1109/IJCNN.2003.1223946. URL: http://ieeexplore.ieee.org/document/1223946/ (visited on 20/04/2021).

[61] Wen-Jun Xue and Li-Wen Zhang. “Stock return autocorrelations and predictability in the Chinese stock marketEvidence from threshold quantile autoregressive models”. In: *Economic Modelling* 60 (2017), pp. 391–401.

Appendices

Appendix A

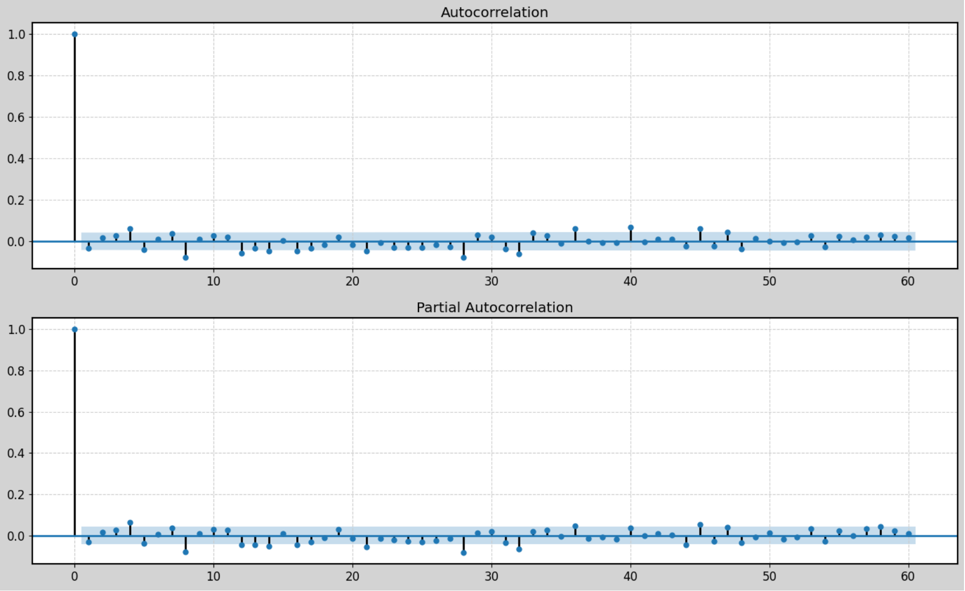


Figure 6.1: ACF and PACF plot MCHP

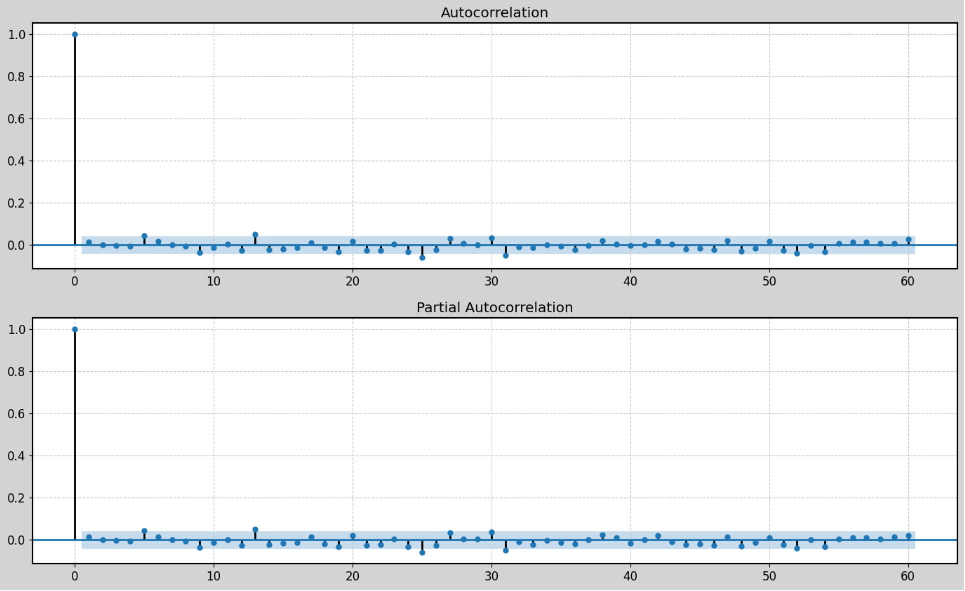


Figure 6.2: ACF and PACF plot VZ

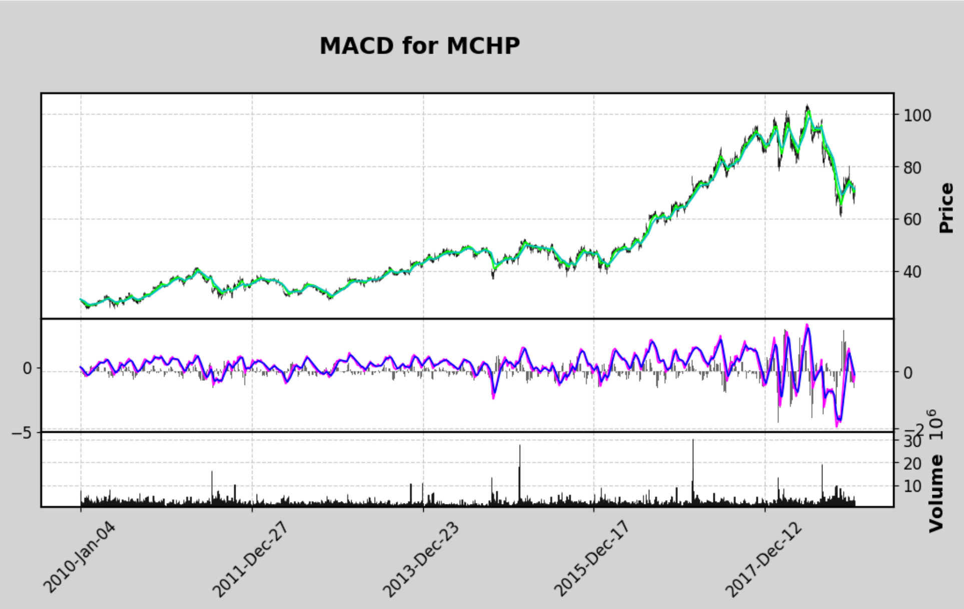


Figure 6.3: MACD Technical Indicator MCHP

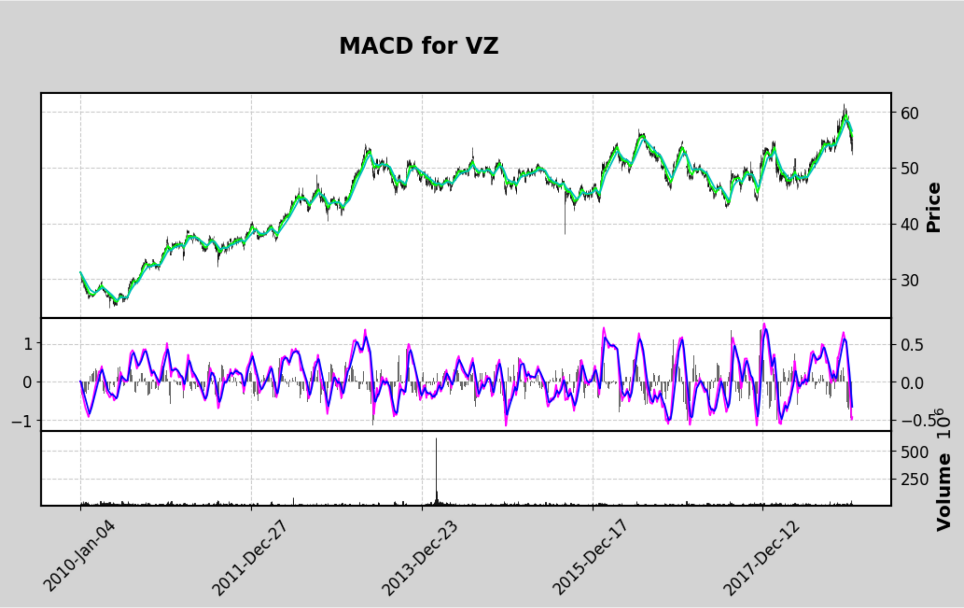
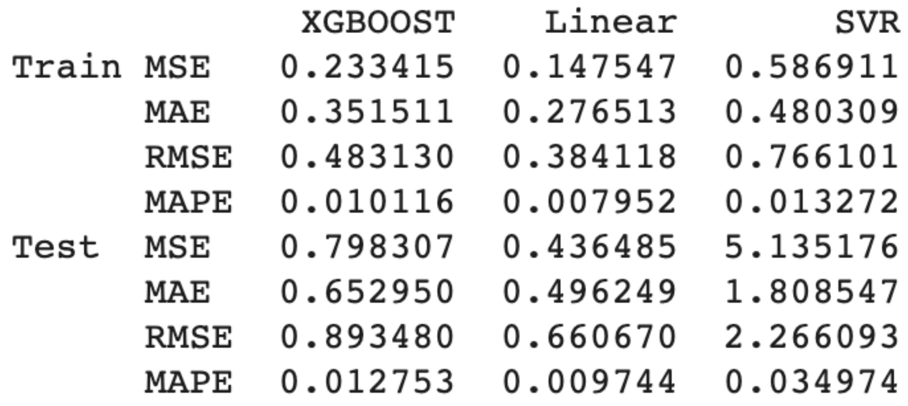
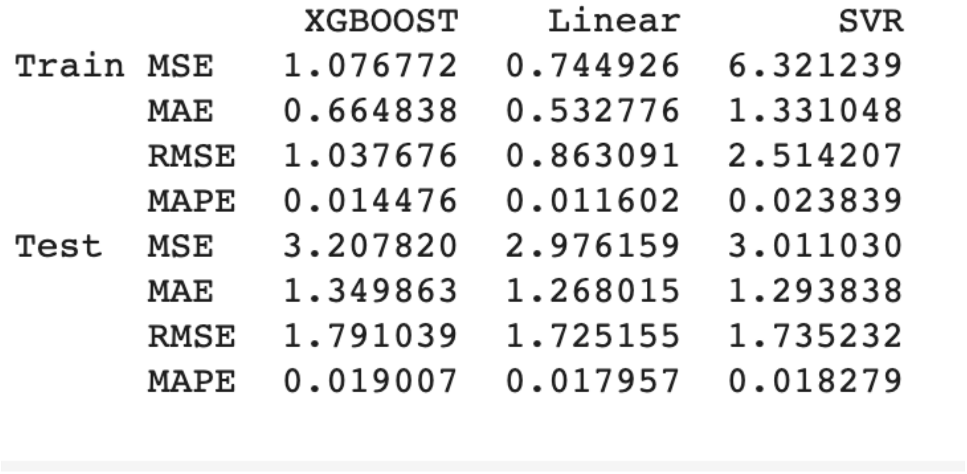


Figure 6.4: MACD Technical Indicator MCHP



```
            XGBOOST    Linear       SVR
Train MSE   0.233415  0.147547  0.586911
      MAE   0.351511  0.276513  0.480309
      RMSE  0.483130  0.384118  0.766101
      MAPE  0.010116  0.007952  0.013272
Test  MSE   0.798307  0.436485  5.135176
      MAE   0.652950  0.496249  1.808547
      RMSE  0.893480  0.660670  2.266093
      MAPE  0.012753  0.009744  0.034974
```

Figure 6.5: Train vs Test errors VZ



```
            XGBOOST    Linear       SVR
Train MSE   1.076772  0.744926  6.321239
      MAE   0.664838  0.532776  1.331048
      RMSE  1.037676  0.863091  2.514207
      MAPE  0.014476  0.011602  0.023839
Test  MSE   3.207820  2.976159  3.011030
      MAE   1.349863  1.268015  1.293838
      RMSE  1.791039  1.725155  1.735232
      MAPE  0.019007  0.017957  0.018279
```

Figure 6.6: Train vs Test errors MCHP