University of East London

(UNICAF)

UEL-CN-7000 Mental Wealth; Professional Life
(Dissertation)

Time Series Analysis: Hybrid Econometric-Machine Learning Model for Improved Financial Forecasting

Joanna Marie Diaz

student ID: u2237195

Teacher: Muhammad Arshad

Abstract

This thesis aims to develop and evaluate hybrid models that combine financial econometric techniques namely Auto-Regressive Integrated Moving Average (ARIMA) model and convolutional neural network (CNN). The proposed model enhanced the accuracy and interpretability of financial forecasting results in the context of time series analysis. The stock market dataset from 2015 to 2023 has been collected and undergone data wrangling. Resulting of 1787 row of time series dataset with 21 features, and a target variable for up and down trend. The ARIMA model was used to analyze the "Close" variable as a univariate variable of the dataset. The residuals of the ARIMA model were used as an added feature to train the CNN model and increase the accuracy and achieve better prediction. Lastly, used support vector machine (SVM) as the final classifier of the trend output of both models. MSE was used to evaluate the models and measure the error occur while training the model. The F1 score was used to measure the accuracy of the model and support model evaluation. It determines the best cycle of the model and learns the direction of change for the next day trend. Experimental results and comparison were conducted, and the study demonstrates that the proposed method outperforms the performance of the stand-alone models.

Table of contents

List of Tables and Figures

List of Tables

List of Figures

1. Introduction

Time series is defined as a time-oriented or chronological sequence of observations on a variable of interest (*Montgomery et al., 2015)*. Explicitly, time series is a series of data points that occurs over time. Time series analysis provides significance and insights of the given dataset's features that change over time as well as bestow assumptions what factors affect a certain variable at different point of time (Pandian, S., 2023). The purpose of time series analysis is generally denoted in two ways: to understand or model the stochastic mechanism that gives rise to an observed series and to predict or forecast the future values of a series based on the history of that series and, possibly, other related components or factors (Chan & Cryer, 2008). Basically, it is the process of using past data to predict future events (Kumar, 2023). According to Tsay (2010), compared to other time series analysis, financial time series analysis involves an element of uncertainty. Such as asset volatility, complexity, and randomness of financial markets. For a stock return series, the volatility is not directly observable. Many investors and analysts use a combination of fundamental analysis, technical analysis, and sentiment analysis, along with machine learning models, to make informed predictions. Moreover, past performance is not always indicative of future results, and that is why it was mentioned that there is an inherent risk in investing in the stock market. Hence, it is of great importance to be mindful in choosing effective methods and models when analyzing financial time series data.

Econometric methods for data analysis are used in many branches, including in finance, when understanding patterns and predicting trends in terms of existing economic conditions. The ARIMA model is one of the considered models for statistical time series prediction. It captures the patterns, trends, and seasonality of the data using a combination of past values, differences, and errors (Shweta, 2022). On the other hand, CNN or convolutional neural network model automatically detects patterns from a large dataset and produces highly accurate predictions. While traditional econometric methods based its assumption on statistical and mathematical techniques, machine learning models focuses on data-driven approaches and automatically learn patterns and relationship without explicitly conveying the underlying process (Janiesch et al., 2021). Both approaches were proven effective by various studies and research when analyzing financial time series data despite the fact that the differences in techniques and results are indisputable.

The focus of this study is the exploration of financial data in terms of time series analysis and the methodologies and challenges of integrating traditional statistical analysis (ARIMA model) and machine learning model (CNN model). One noted challenge in combining the models involves analyzing both model's complexity and determining an effective strategy for combining their outputs or features. The challenges also involve data preprocessing, data compatibility, hyperparameter tuning, overfitting and underfitting, error analysis and benchmarking. Hence, the resulting hybrid model produces less error rate with accordance to mean squared error and F1 score than the evaluation result for each independent model.

2. Literature Review

Time series analysis was popularized in the 1970s by George Box and Gwilym Jenkins when they write a book entitled "Time Series Analysis" and introduced the Box-Jenkins method. The method applies moving average (ARMA) or autoregressive integrated moving average (ARIMA) models to analyze past values and find the best fit for a time series model (Nielsen, 2019). The ARIMA model has become one of the conventional approaches in time series analysis, especially for univariate (one variable) analysis, as it takes long-term trends and performs well with moderate amount of data (Macias, 2022). A few statistical methods were influenced by this model such as SARIMA, ARCH and GARCH method. A study made by Ariyo et al., (2015) to use ARIMA models for short-term stock price prediction on different stock market datasets. They used statistical techniques to determine the best ARIMA model, such as Bayesian or Schwarz Information Criterion (BIC), Relatively small standard error of regression (S.E. of regression), relatively high of adjusted R2, and Q-statistics and correlogram that shows significant pattern left in the autocorrelation functions (ACFs) and partial autocorrelation functions (PACFs) of the residuals. Devi et al., (2013) also used ARIMA model to predict next day trend predictions. They utilized Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) to identify the best ARIMA model for trend prediction. In addition, they used Mean Absolute Percentage Error (MAPE), Percent Mean Absolute Deviation (PMAD) and Error accuracy for model evaluation. Both studies show that the ARIMA model can be effectively used for linear data and short-term prediction.

Eventually in the 1980s, the financial industry leverages the usage of machine learning techniques for business decisions. From financial planning systems, credit scoring formula, anti-money laundering to fraud detection (Sharma, 2023). In essence, the idea of utilizing machine learning algorithm to gain significant profit in financial industry by discovering the future value of company stock and other financial assets is inevitable. Artificial neural network (ANN) has been applied to stock market prediction since the early 1990s. These neural network models were relatively simple compared to modern deep learning networks but demonstrated the potential of neural networks in capturing non-linear patterns in stock price data (Strader et al., 2020). ANN was implemented on stock market data analysis in the early 2000s by the study made by Jasic & Wood (2004). They develop an ANN model that is based on univariate neural networks. Using untransformed data inputs to provide short-term stock market index return prediction and predict daily stock market index returns, using data from several global stock markets. In the current era of machine learning, convolutional neural network (CNN) is one of the machine learning methods that is now used for stock market prediction. It is a deep learning neural network that can construct non-linear and complex functions that map input and output from the dataset (Durairaj & Mohan, 2019). It is more commonly used for image feature detection.

Wu et al., (2020) and Hoseinzade & Haratizadeh (2019) developed a CNN model for feature extraction and trend prediction using financial time series dataset. Various data was collected from the financial market, that were thought relevant and can affect the specific stock indices. The dataset includes technical analysis and other market financial information. The collected data were then normalized and used as an image like matrix that can be fed and trained to a CNN model. Both get much better accuracy compared to a traditional model. An experimental study conducted by Selvin et al., (2017) determines which deep learning models perform best in a time series dataset. The deep learning models are CNN, Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM).

The result shows that CNN model gives more accurate prediction than the other two models. CNN enables the understanding of the dynamical changes and patterns occurring in the current window and not depending on the previous information. This makes the volatility of the stock market dataset for prediction less crucial. However, in the case of RNN and LSTM, it uses information from previous lags to predict future instances.

Several studies conducted comparison which models will achieve the better performance in time series datasets and produce close to accurate prediction. Pérez-Pons et al., (2021) and Shobana and Umamaheswari (2021) carry out research on various articles relevant to traditional econometric and machine learning models. It was determined that econometric models are not efficient enough to identify complex relationships and process diverse and huge amounts of data. Furthermore, it verifies that machine learning helps increase the accuracy of prediction algorithms with regards to traditional econometric methods. Asokan, M. (2022) and Moshiri and Cameron (2000), conducted a study on financial stock prices. A study to compare forecasting error between LSTM, SARIMA, and Hybrid ARIMA-GARCH models using a financial time series dataset and determine which model perform best. The study proves that neural network model (LSTM) performs best and has lower forecasting error than the two traditional models. Herrera et al. (2019), presume in their study the possible energy consumption in the world by the year 2040. Traditional econometric model was used such as ARIMA, SARIMA, Hybrid ARIMA-GARCH, multinomial logit (MNL) and ordinal logit (OL) whereas long, short-term memory (LSTM), artificial neural network (ANN), decision tree (DT), random forest (RF), support vector machine (SVM) methods were used as machine learning models. The study investigates the performance of the data on each model and if it will produce a close to accurate prediction. Result found that machine learning approaches consistently surmount the classical approach. Thereafter, it was confirmed that machine learning models overcome the limitations of econometric models mainly in making prediction. And that CNN model can have a better performance in financial dataset since it based its assumption in the current changes of pattern rather than the previous information. Hence, the utilization of both methods on developing a framework for more effective data forecasting has motivated me to develop a Hybrid model using traditional econometric ARIMA model and CNN machine learning model. By the period of creating this research paper, the combination of ARIMA and CNN for stock market prediction are not yet published.

3. Methodology

This section explains the theory and methods used in this paper. Most mathematical ideas and theory are excerpted from the book of time series analysis by Tsay (2010), Montgomery et al. (2015) and scholastic article from the web. The ML Python libraries are utilized for dataset analysis and model development using Jupyter Notebook environment.

3.1 Financial Time Series Analysis

Financial time series analysis is concerned with the theory and practice of asset valuation over time. (Tsay, 2010). It is considered unpredictable because of its volatility in changes that occur because of several factors that affect its behavior. Due to this, financial analysts have encountered few adversities on how to better analyze and make assumptions on its future behavior. One of the propositions is by analyzing the data trend and pattern with respect to time and evaluate possible risk and data inflation to make a better prediction.

3.1.1 Stock Market Dataset

Stock prices are one form of financial time series that are recorded at regular intervals, typically daily or minute-by-minute during trading hours (Hayes, 2022). According to yahoo finance, stock markets produce capital for companies, promote economic growth, can raise capital for investors and mobilize savings by short-term and long-term investment for stock traders. Stock prices are unpredictable because of many factors that influence its behavior such as, company's earnings reports, economic indicators, market sentiments, supply and demand, dividends, political events, commodity prices, currency exchange rate, etc.

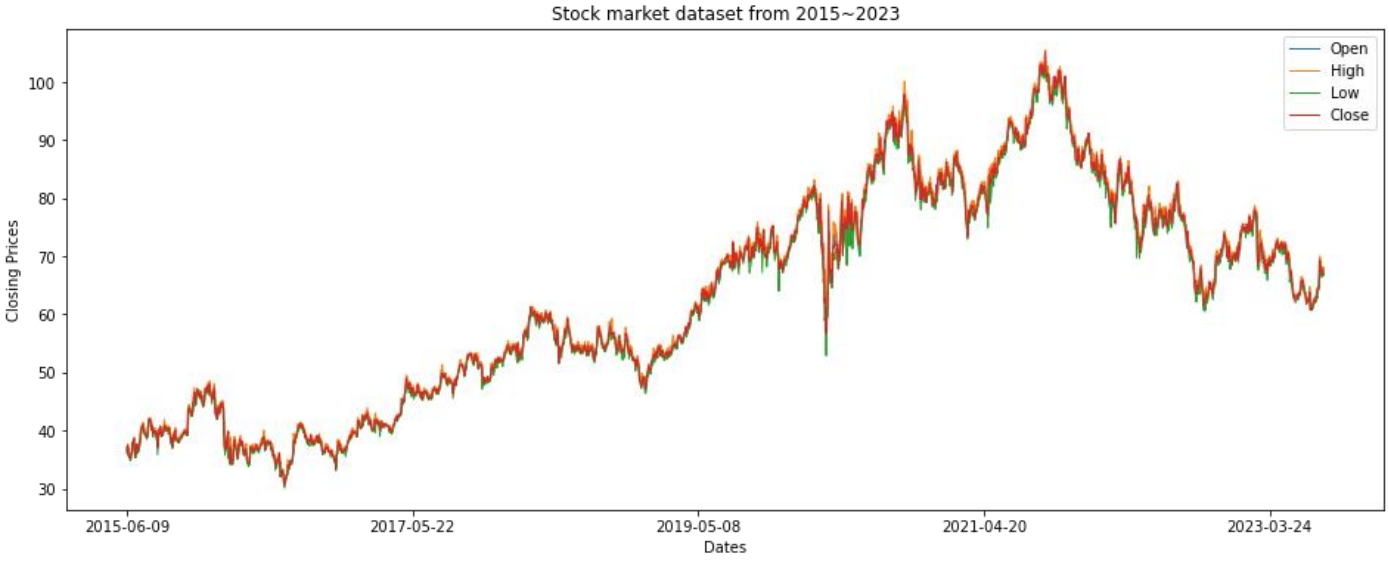

Figure 1. Euronext dataset from 2015 ~ 2023

3.1.2 Simple Moving Average

Developing a forecasting model was said to be best achieved by starting with graphical display and analysis of the available data. Plotting the moving average of the data helps to make variability in the plot less and smooth the noises. This makes pattern and trend more evident. The closing price is what

determines the final price for each trading day, therefore the basis of all technical analysis in this project is the close column of the dataset. Below is the graph of close columns with smoothed line for 10 days moving average.

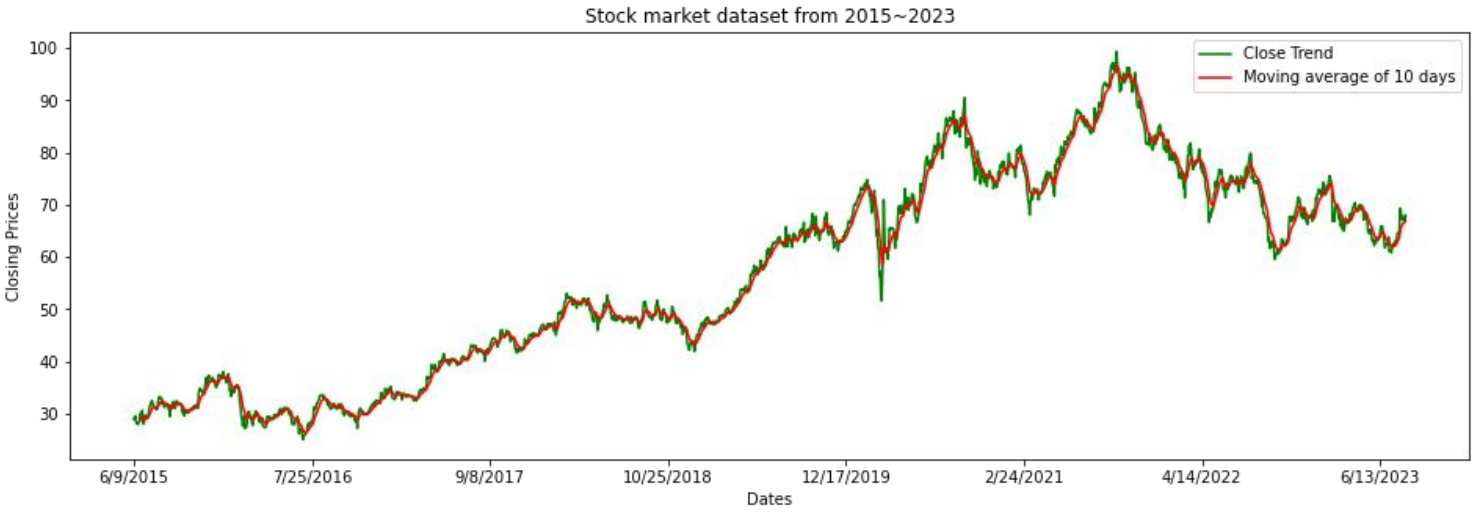

Figure 2 Simple Moving Average trend of 10 days

$$\textit{Simple Moving Average Formula} = \frac{y_1 + y_2 + \ldots\ldots + y_n}{n}$$

3.1.3 Trend and seasonality

The trend is the component of a time series that represents variations of low frequency in a time series, the high and medium frequency fluctuations having been filtered out (Di Pietro, 2022).
The pattern or trend of the series is not entirely visible when it is normally plotted. In order to

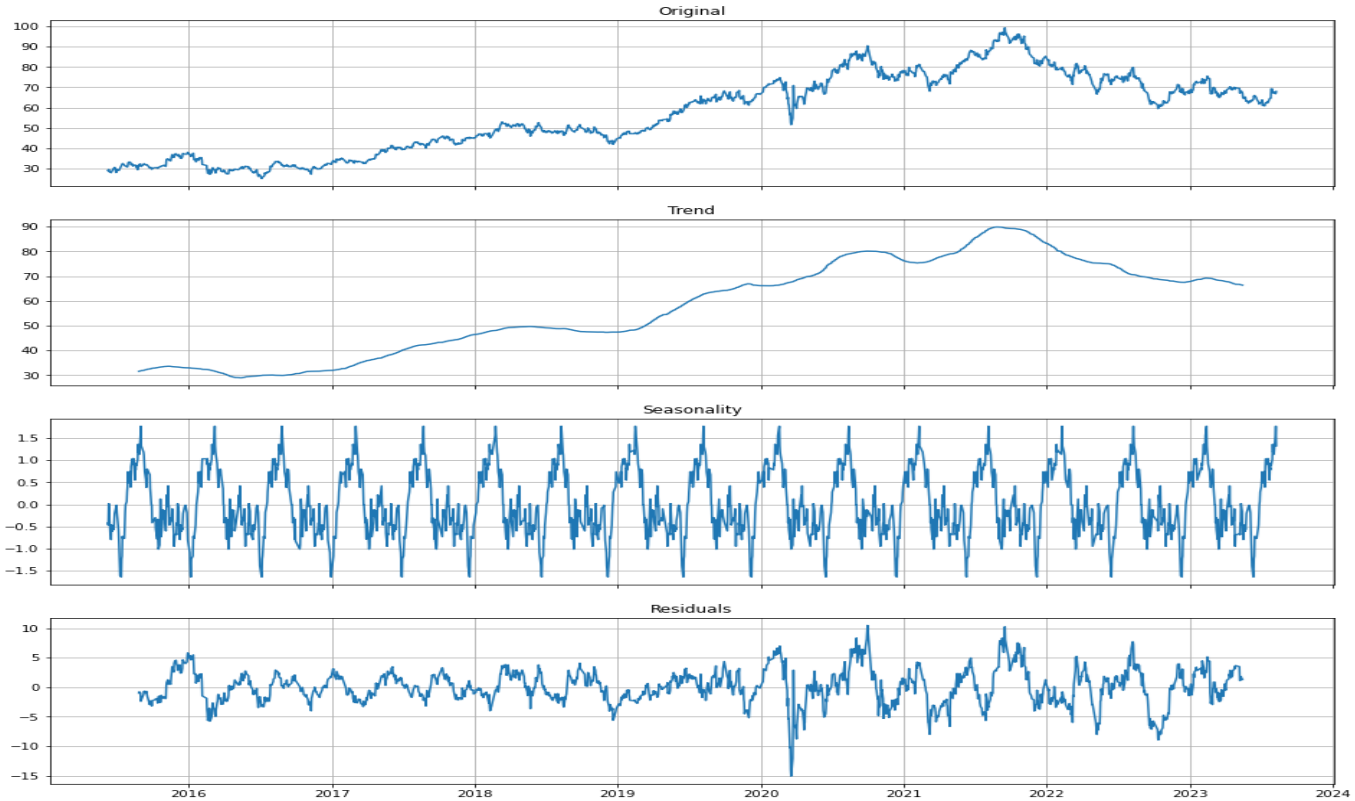

Figure 3 Time series graphical decomposition technique

understand the series trend and seasonality, the decomposition function of Python statistical library 'statsmodels' can help. By using a decomposition technique, you can tell the direction of trend for each cycle and the spikes in the residual component. It also shows if the time series has seasonality. The decomposition algorithm ensures that the seasonal and the residual components vary around 1, and thus the seasonal one oscillates regularly (Kolassa, 2021).

The observation in figure 3 shows that there is a positive linear trend in the data from 2015 to 2021 and a slight drop from 2021 to 2023. The graph also suggests that there is no seasonal variation in the data since the seasonal cycle has a gradual change for every season. The graph also has an evident residual spike near the year 2020. By this, I can assume that the time series is a non-stationary.

3.2 ARIMA model

Autoregressive integrated moving average (ARIMA) models are one of the commonly used for time series analysis and forecasting. Especially in a short-term and linear set of data. ARIMA models are based on determining the autoregressive (AR) component and moving average (MA) component from a statistical regression of a time series data, with a usual differencing for *d* times to established data stationarity. In an ARIMA model, AR and MA component refers to the lag of a time series, with the number of *p* for significant correlation and the number of *q* for regression errors (*Cracan, 2020*).

General form of an ARIMA(*p, d, q*) model with time series X:

$$diff(X, d) = AR(p) + MA(q) + \varepsilon$$

The ARIMA components refers to the approach that involves model selection, time series stationarity, determining the required lag for AR and MA components, and model evaluation which can result on finding the best model that can fit the given time series data. Figure 4 is the structure of the proposed ARIMA model.

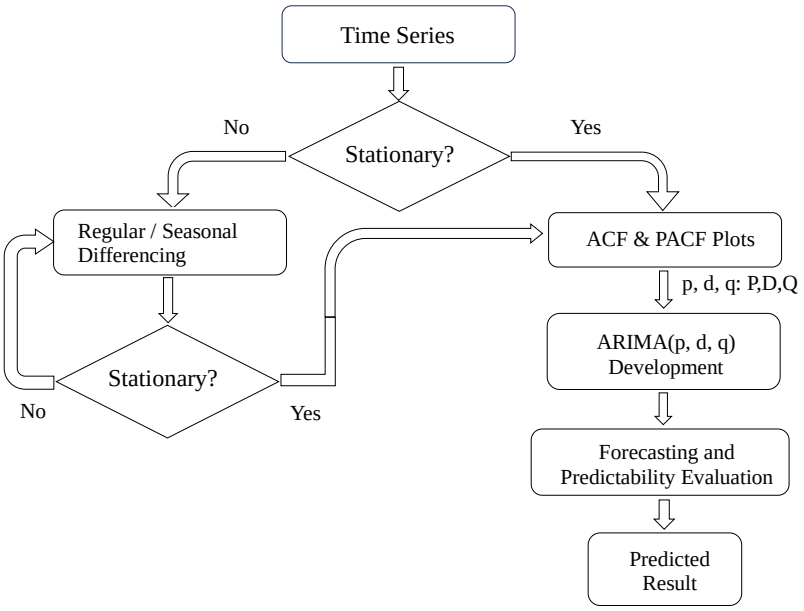

Figure 4 ARIMA Model Architecture

3.2.1 Stationarity

A time series is said to be strictly stationary if its properties are not affected by a change in the time origin. That is, if the joint probability distribution of the observations y_t, y_{t+1},..., y_{t+n} is exactly the same as the joint probability distribution of the observations y_{t+k}, y_{t+k+1},..., y_{t+k+n} then the time series is strictly stationary. Stationary implies a type of statistical **equilibrium** or **stability** in the data. To conclude, a time series is said to be stationary if it has the same statistical properties such as same mean, variance, and covariance no matter at which point it is being measured.

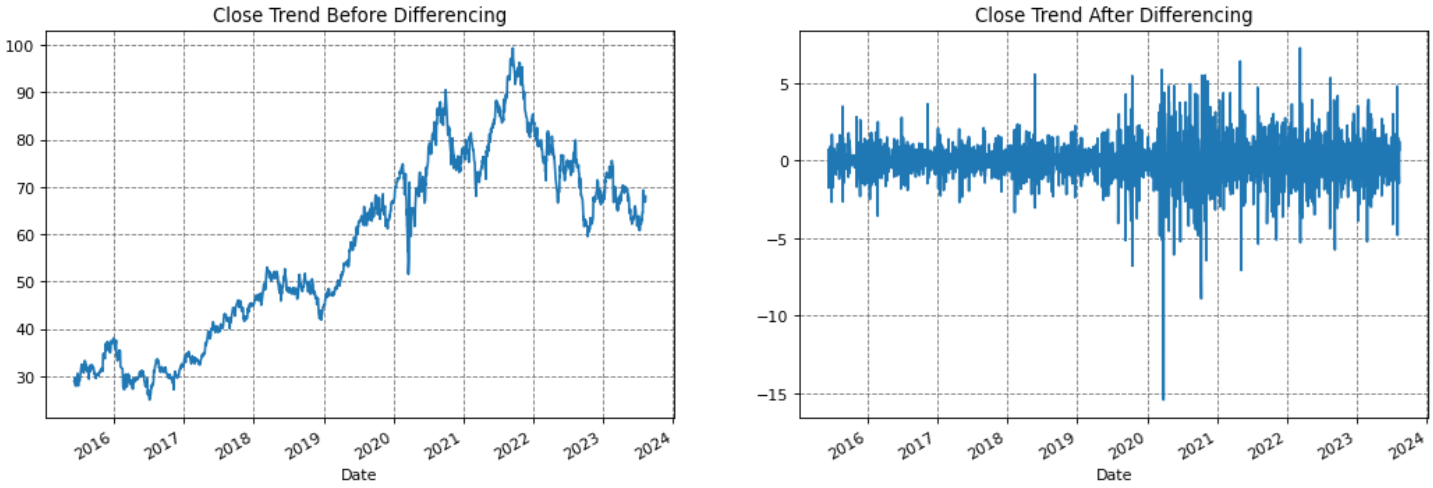

Figure 5 Close trend before and after differencing

The time series must identify its stationarity in order to accurately train an ARIMA model. The decomposition technique and augmented dickey fuller (ADF) test was used to check for series stationarity. The Dickey-Fuller test is a unit root test that tests the null hypothesis of a time series. The Python statistical library 'statsmodel' (specifically the addfuller() function) was used to prove if the null hypothesis of the series should be rejected or not. After the current value of the series was subtracted from the previous one, the null hypothesis of the series was not rejected, and the time series has become stationary.

$$y`_t = y_t - y_{t-1}$$

The function was used to determine if the ADF statistics of the series are higher than any of the critical values and if the p-value obtained is greater than significance level of 0.5.

```
Column Name: Close Variable
ADF Statistic: -1.470788
p-value: 0.547909
Critical Values:
        1%: -3.434
        5%: -2.863
        10%: -2.568
```

Based on result, the critical values are not higher than the ADF statistics, but the p-value is more than 0.5. Therefore, it can indicate that the time series is non-stationary. After the differencing of 1 is applied, the series has yielded a value of stationarity and proves the null hypothesis.

3.2.2 Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF)

Autocorrelation function (ACF) and Partial autocorrelation function (PACF) can yield valuable insights into the behavior of a time series data. Both functions are often used to determine the possible

number of Autoregressive (AR) and Moving Average (MA) for an ARIMA model to select the best model for forecasting (Rehal, 2022).

The ACF displays the correlation between the time series and its own lagged values which represent a correlation coefficient between the series and its past values. For example, Autocorrelation of lag k is the correlation between Y(t) and Y(t-k), measured at different k lags.

The PACF is measured by handling the effects of other lags, generally using linear regression method. The partial correlation for each lag is the unique correlation between the two observations after the intermediate correlations have been removed. For example, PACF of lag k is the correlation between Y(t) and Y(t-k) when the effect of all other intermediate values (Y(t-1), Y(t-2),….Y(t-k+1)) is removed from both Y(t) and Y(t-k).

In summary, the autocorrelation function aids in determining the qualities of a time series. Mainly, the partial autocorrelation function is more beneficial during the definition phase for an autoregressive model. Partial autocorrelation plots can be used to specify regression models with time series data as well as Auto-Regressive Integrated Moving Average (ARIMA) models (*Lendave, V., 2022*).

To determine the order of the model according to the analysis of the ACF and PACF plotted graph, the following characteristics was used:

	AR (*p*)	**MA (*q*)**	**ARMA (*p*, *q*)**
ACF	Tails of (Geometric decay)	Significant at lag *q* / Cuts off after lag *q*	Tails off (Geometric decay)
PACF	Significant at each lag *p* / Cuts off after lag *p*	Tails off (Geometric decay)	Tails off (Geometric decay)

Table 1. Order for AR, MA, and ARMA Model (Iamleonie, 2022)

In figure 6, the Autocorrelation plot shows a slowly diminishing lag to zero, but the data cycle has no seasonality. Hence, ARIMA model with no seasonality will be used to train the dataset. In the PACF plot it shows a significant correlation at lag 1 and the slowly diminishing lags of ACF to zero

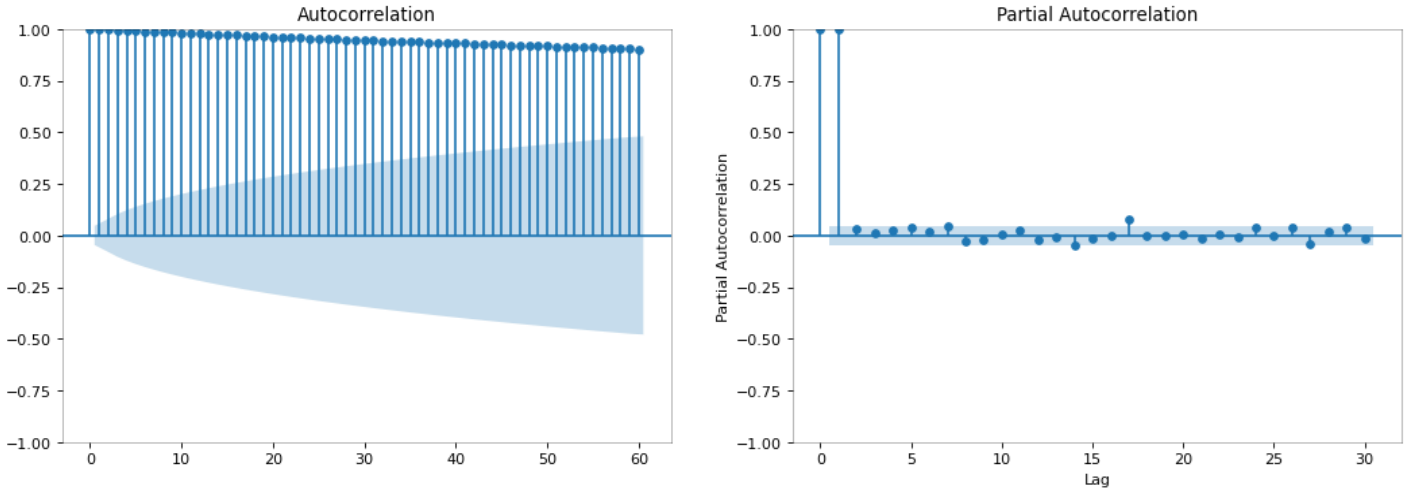

Figure 6 ACF and PACF plot before differencing

or geometric decay. Which can be interpreted as an occurrence of characteristic for AR. Based on these graphs, we can assume that this graph shows an auto-regressive (AR) value of 1 and since there

is no characteristics occurrence for moving average (MA), it will adopt a value of 0. Therefore, the final ARIMA model for this graph can be assumed as ARIMA(1, 0, 0). Note that the dataset used on this plot is based on the original non-stationary series, therefore, this model may not produce a proper prediction.

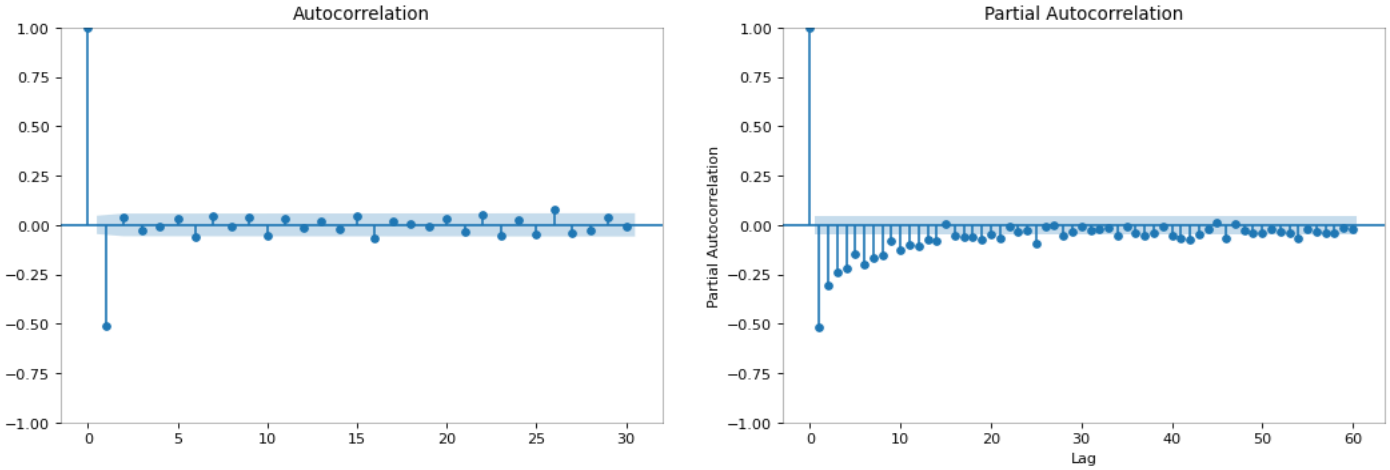

Figure 7 ACF and PACF plot after differencing

On the other hand, figure 7 displays a more visible lag correlation for ACF and PACF graph. It is a one differenced lag of the dataset that resulted to a stationary statistical value based on the null hypothesis of ADF test. The ACF and PACF graph in this figure shows a significant correlation value that can represent a model parameter for an ARMA model. Both ACF and PACF plots show a significant correlation at lag 1 and a geometric decay. This can be considered as an occurrence of characteristics for both AR and MA. This can be conclude that the ARIMA model can now be trained with a value of 1 for AR (*p*), 1 for I (*d*) and 1 for MA (*q*) or ARIMA(1, 1, 1).

3.2.3 Model Result

The dataset was split as 80% for the training set and 20% for the test set. After training the train set into the model the test set produced the following model summary with 358 variables.

358

SARIMAX Results

Dep. Variable:	y	No. Observations:	1786
Model:	ARIMA(1, 1, 1)	Log Likelihood	-2643.688
Date:	Wed, 04 Oct 2023	AIC	5293.375
Time:	12:09:05	BIC	5309.837
Sample:	0	HQIC	5299.455
	- 1786		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.7157	0.086	8.298	0.000	0.547	0.885
ma.L1	-0.7556	0.084	-8.951	0.000	-0.921	-0.590
sigma2	1.1323	0.018	62.560	0.000	1.097	1.168

Ljung-Box (L1) (Q):	0.05	Jarque-Bera (JB):	6385.97
Prob(Q):	0.82	Prob(JB):	0.00
Heteroskedasticity (H):	3.65	Skew:	0.16
Prob(H) (two-sided):	0.00	Kurtosis:	12.26

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Figure 8 Model summary on test set for ARIMA(1,1,1)

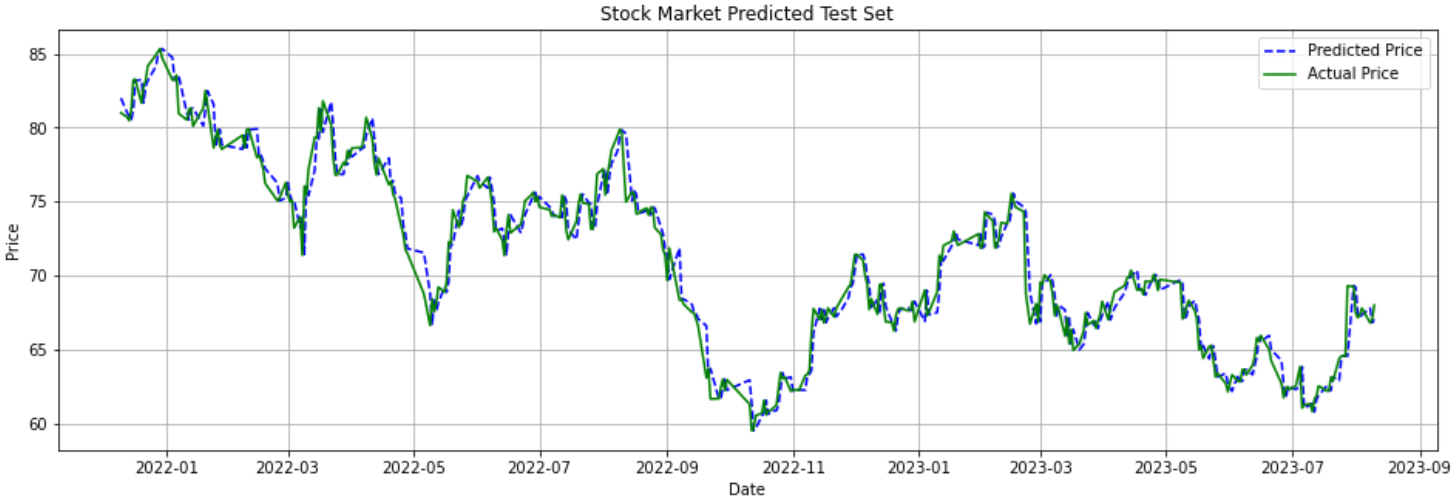

Figure 9 Predicted vs Actual price

3.2.4 ARIMA Model Evaluation

Forecasting is now used to help companies identify future events based on any given data. Thus, organizations can identify trends and patterns or even problems and create actionable solutions for any possible outcome perceived by any forecasting model. Time series forecasting model is a part of a quantitative forecasting technique that uses historical data that summarize patterns in the data and express a statistical relationship between previous and current values of variable (*Montgomery et al., 2015*).

After training the data into the chosen ARIMA model, the forecasting model has been evaluated using the three statistical metrics namely,

- Mean Absolute Error (MAE), where the average absolute difference between the predicted and actual values were computed.

$$MAE = \frac{1}{N} \cdot \sum_{i=1}^{N} |x_i - m_i|$$

- Mean Squared Error (MSE), where the average of the squared differences between the predicted and actual values were computed.

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - m_i)^2$$

- Root Mean Square Error (RMSE). The square root of the MSE which provides same unit as the original data. Both MSE and RMSE comply with the condition that if the derive result is lower, the better a model fits a dataset.

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - m_i)^2}$$

Performance Evaluation

```
# Calculate Mean Absolute Percentage Error using the created dataframes
mape = np.mean(np.abs(np.array(df_predictions) - np.array(new_df_test))/np.abs(new_df_test))
print('MAPE: %f' % mape)
```

```
MAPE: 0.012321
```

```
# Calculate Mean squared error for the list of expected and predicted value
mse = mean_squared_error(new_df_test, df_predictions)
print('MSE: %f' % mse)
```

```
MSE: 1.397698
```

```
# Get root mean squared error
rmse = sqrt(mse)
print('RMSE: %f' % rmse)
```

```
RMSE: 1.182243
```

The ARIMA(1,1,1) resulted on a lower value for the three evaluation metric which emphasized that the chosen model were well suited for the dataset.

```
from sklearn.metrics import accuracy_score, f1_score, mean_absolute_error

print("accuracy:", accuracy_score(ARIMA_trend, target_trend))
print("MAE:", mean_absolute_error(ARIMA_trend, target_trend))
print("F1:", f1_score(ARIMA_trend, target_trend))
```

```
accuracy: 0.4823091247672253
MAE: 0.5176908752327747
F1: 0.47940074906367036
```

For the trend prediction on the test set, the model has an accuracy of 48%. The evaluation metric result for mean absolute error (MAE) and F1-score are 51% and 47% respectively.

3.2.4 ARIMA Residual

The ARIMA residual in a time series are equal to the difference between the observations and the corresponding fitted values.

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

Residual formula

It is said that a good forecasting method will yield two important properties. An uncorrelated residual and a residual with a zero mean. If the residuals have a mean other than zero, then the forecasts are biased (Hyndman & Athanasopoulos, 2018).

The ARIMA residual will be used as an added feature to the stock market dataset to train a CNN model along with other additional features that have been collected on the web. By using the residual as the additional variable, the CNN accuracy rose by about 2% compare when only the collected financial dataset was used.

3.3 Convolutional Neural Network (CNN)

Convolutional Neural Network is a structured recognition technique which has been developed in recent years. This network averts the intricate preprocessing of the image that results for an actual

image to be processed directly. It uses local receptive field, weights sharing and pooling technology that makes the training parameters reduce compared to the neural network (Xia et al, 2017).
CNN is mainly used for object detection in an image commonly referred to as computer vision. The image pixel is extracted to form a matrix dataset where the bias and weights are computed inside the network. CNN learns feature engineering by itself via filters optimization (Wikipedia). In this section, a brief definition of CNN as a part of machine learning technique will be discussed.

3.3.1 The CNN Layer

Convolutional Neural networks (CNN) are a subgroup of machine learning, and they are at the heart of deep learning algorithms. They are made up of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network (IBM). The main layers that precisely describe the building of the CNN model are the following layer:

1. Convolutional layer - is one of the main building blocks of a CNN model. The aim of this layer is to extract all the high-level features of the given matrix and reduced the dimensionality of the matrix to extract only the relevant features (Martinez, 2020). This feature is more convenient for detecting the edges of an image. But for this study, the high-level features will be extracted based on the strong correlation in the data. This is where a kernel or filter is used to extract these relevant features and produce a new function called feature map. A 3x1 kernel size is used for the dataset with a ReLu as the activation function that decides whether a neuron should be activated or not.

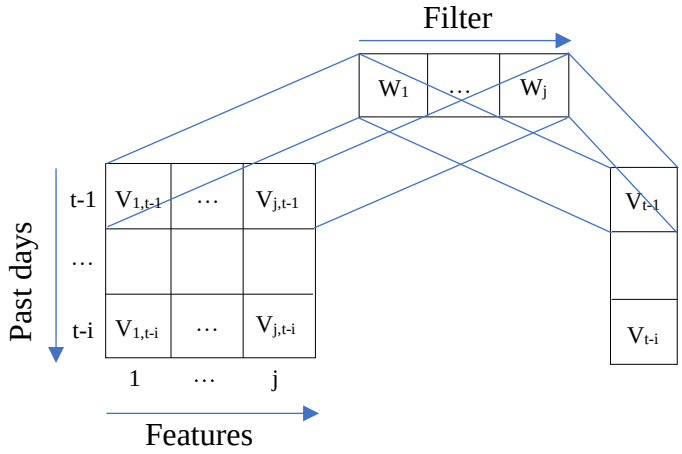

Figure 10 Feature map extraction

2. Pooling Layer – The pooling layer reduces the dimensionality of the feature map to reduce computing time and tackle overfitting. On the process of dimensional reduction, the neurons are being subjected to invariance translation or producing an output layer that emphasizes the important portion of the input layer with minor changes to the principal value (Albawi et al., 2017). There are several pooling functions but the most eminent one is the max pooling function. Which acquires the maximum value from the previous layer which can be occupied

by the window on a specified step of each stride. The size of pooling operation is smaller than the feature map, frequently the size is set to 2x2 with a stride of 2 pixels, extracting the feature map to one quarter a size. In this study the size of the pooling layer is 2x1.

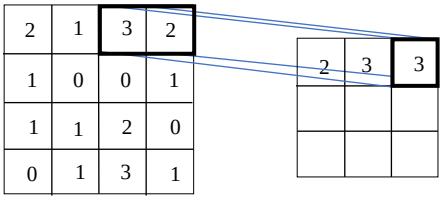

Figure 11 Max Pooling layer with 2x1 stride

3. Fully Connected Layer - In this layer, the dataset undergone flattening from the extracted data of the preceding layer by performing vector matrix multiplication on each neuron to obtain a 1-dimensional column. The sigmoid activation function is used for binary classification for the up and down trends. Thus, produced the output prediction.

3.3.2 Convolutional Neural Network in Time Series Analysis

The only difference between computer vision problems and time series one is that the input we give to the model is in different form. That is, image matrix for computer vision and 1D array for time series forecast. Hence, if we are able to convert the 1D time-series sequence to an input image matrix shape, we could apply a CNN model for the forecasting problem (Pandey, 2021).

The CNN architecture that was used in this project is based on the 2D-CNNpred framework suggested by Hoseinzade & Haratizadeh (2019), that predicts the next day's direction of the market movement. The proposed CNN-based framework can be applied on a collection of data from a variety of resources which mentioned, that can affect the stock market trend such as, exchange rate, world stock market, commodities (gold, silver, oil, wheat and so on), big US companies, future contracts and financial technical analysis, in order to extract features that can increase the stock market trend prediction accuracy. The 60 days is the length of the history that will be fed on the model for each cycle and treated as a basis of information for the last 60 days. The referenced model consists of 82 feature variables for each day. But instead of having 82 variables for features, the proposed CNN model structure only uses 22 feature variables.

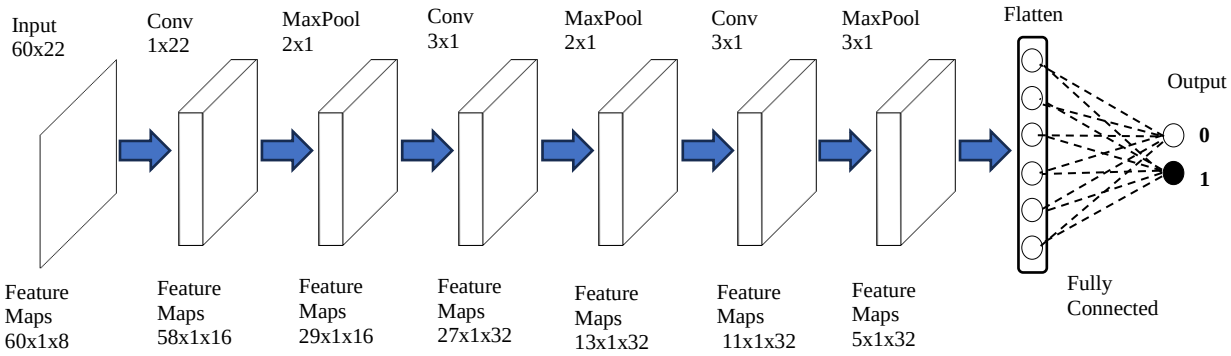

Figure 12 Graphical visualization of Actual CNN model
(Based on 2D-CNNpred proposed by Hoseinzade & Haratizadeh (2019))

3.3.2 Data Preparation

The stock market dataset comprises of 8 years amount of data from 2015 to 2023 which were downloaded from Yahoo finance. Influential world indices, features, exchange rates, and commodities were collected. Some of the dataset dates have mis-aligned trading days. Like for example the Japan Exchange Group has an open trade on September 3, 2018, while the New York Stock Exchange has none. Those mis-aligned dates were removed, and the missing values caused by technical analysis such as moving average for 50 days was filled with zero. The data has also been normalized using the standardscaler of Keras libraries to have a standard scale of the data and ensure data uniformity before training to a CNN model. The target variable was set as an up and down trend of the next day transaction. If the following day has a higher close value than the previous one, the target variable will be set to 1 and vice versa.

3.3.3 Training the CNN model in the stock market dataset

The stock market dataset was split into 60% for training, 20% for data validation and 20% for testing. A data generator function was used to provide data augmentation and help prevent underfitting.

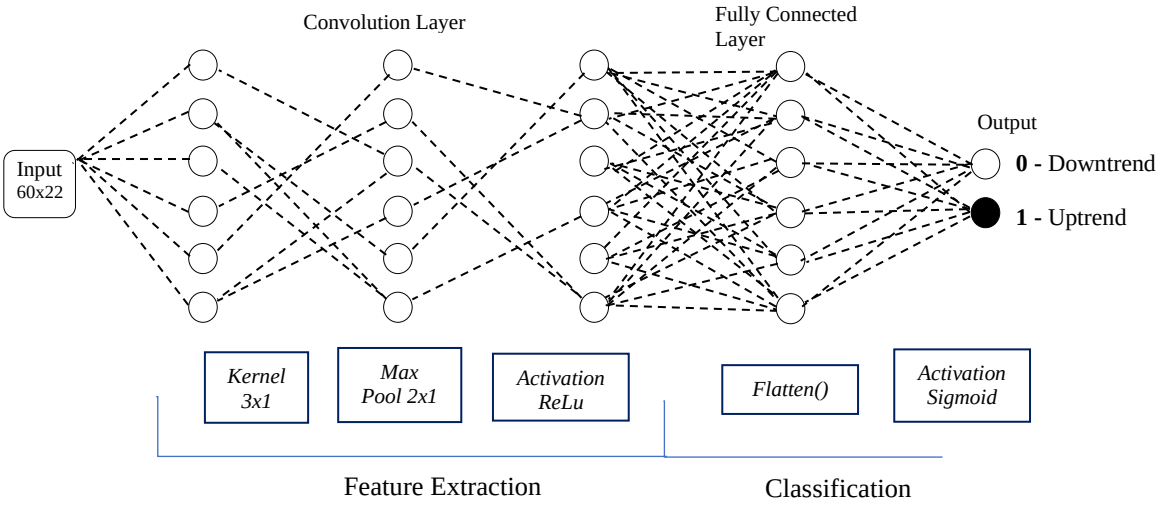

Figure 13 CNN Model Architecture

3.3.4 Results

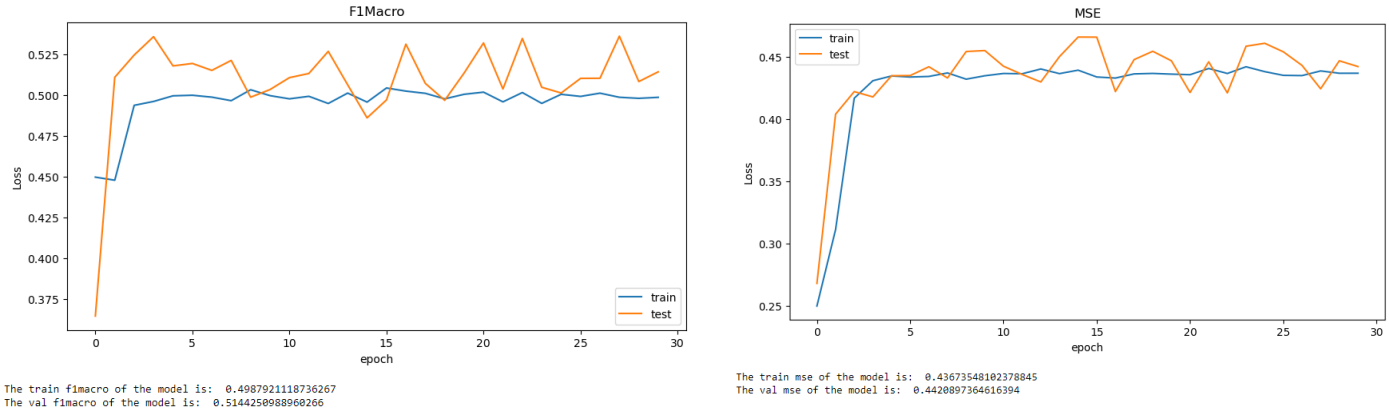

Figure 14 F1-Score vs MSE

The CNN model was trained based on the model structure and hyper-parameter tuning to attain the best model that will fit the diverse dataset. The F1-score and MSE were used to evaluate the model and achieve the best result.

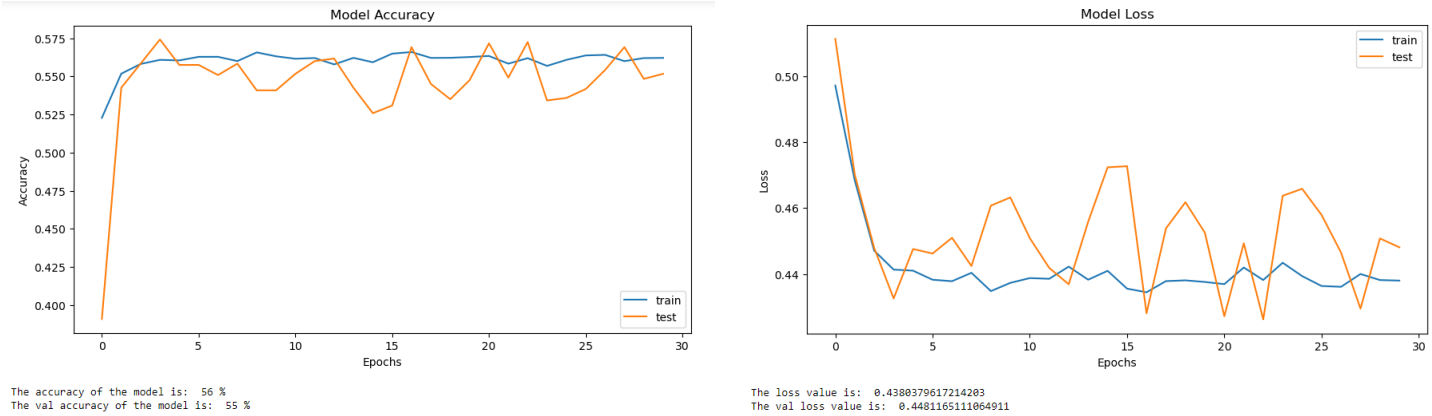

Figure 15 Model Accuracy vs Model Loss

```
print("accuracy:", accuracy_score(CNN_trend, test_target))
print("MAE:", mean_absolute_error(CNN_trend, test_target))
print("F1:", f1_score(CNN_trend, test_target))
```

```
accuracy: 0.547486033519553
MAE: 0.45251396648044695
F1: 0.5803108808290155
```

The CNN model accuracy result for the test set trend prediction is 54%. While the MAE and F1-score is 45% and 58% respectively.

3.4 Hybrid ARIMA-CNN model

This is the last part of the study where the combination of both ARIMA and CNN model prediction are used for the final trend prediction. Support vector machine (SVM) was used as the final binary classifier. The ARIMA model is known to process time series dataset that have a presence of autoregressive or linearity. While CNN model can handle complex form of data that is represented as a matrix.

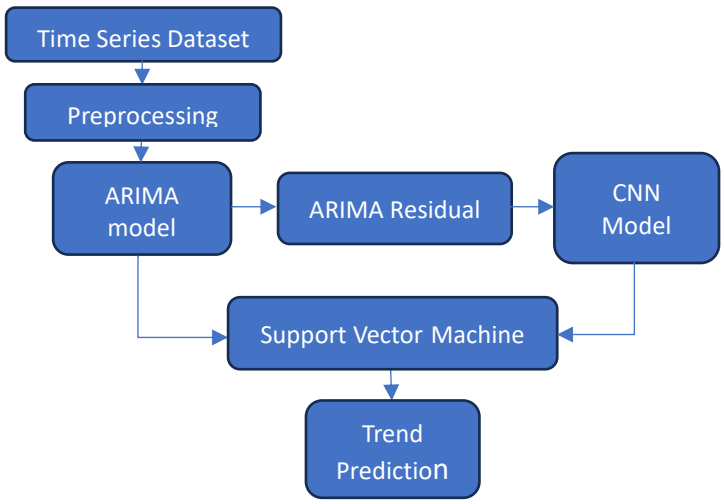

Figure 16 Proposed Hybrid Architecture

3.4.1 Support Vector Machine (SVM)

Support vector machine is a supervise machine learning technique that utilizes classification algorithms for two group classification problems (Stecanella, 2017). It classifies data points using a hyperplane that sets boundaries between two classes. The idea behind this is to transform the input data into a higher-dimensional feature space. The transformation allows it to classify the dataset more effectively into two features and make it easier to find the hyperplane or the linear separation between the two-classified data. This process involves using a kernel function that implicitly computes the dot products between the feature vectors. During the training phase, SVMs apply a mathematical formulation to find the optimal hyperplane in a higher-dimensional space, often called the *kernel space*. This hyperplane is crucial because it maximizes the margin between data points of different classes, while minimizing the classification errors. The kernel function plays a critical role in SVMs, as it makes it possible to map the data from the original feature space to the kernel space. The choice of kernel function can have a significant impact on the performance of the SVM algorithm (Tabsharani, 2023). SVMs are said to perform well even with a limited amount of data and can be applied to both classification and regression problems.

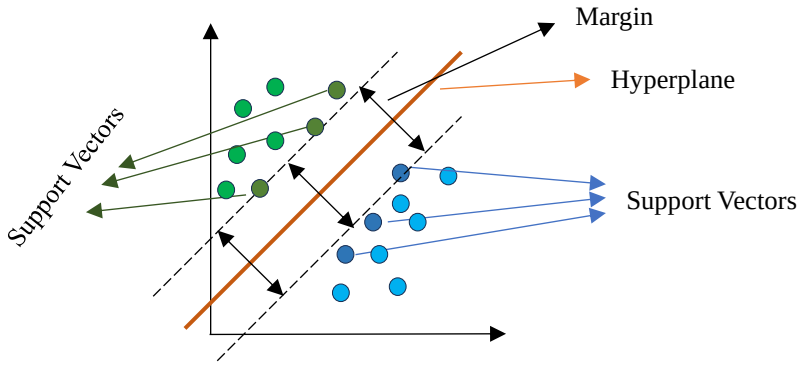

Figure 17 SVM 2D graphical representation

The graphical representation shows how the support vectors became a point for two classifications with the basis of the most optimal hyperplane that separates them.

3.4.2 Training the hybrid model

Both ARIMA and CNN model produced an up and down trend prediction on the 20% of the dataset. For the SVM classifier, these 2D-dataset which compose of 358 rows will be used to train an SVM binary classification model. The dataset is indexed by a dateframe and three columns for ARIMA prediction, CNN prediction and one target variable for the actual up and down trend that is based on the original test set. All the columns are composed of only two variables which are 1 for the up and 0 for the down trend. SVM will give the final prediction for the whole model.

Index	ARIMA Prediction	CNN Prediction	Target Variable
2021-12-10	0	1	0
2021-12-13	0	0	0
…	…	…	…
2023-08-09	1	0	1
2023-08-10	0	0	0

Table 2 Representation of the new dataset for the SVM Model

After merging the datasets, train_test_split() function from scikit-learn library was used to split the dataset for 80% train set and 20% test set. The datasets were then fitted to an SVM classifier.

3.4.3 Result

Below is the confusion matrix graph and model evaluation result after training the dataset. It shows that there are 25 FP (False Positive), 30 TP (True Positive), 1 FP (False Negative), 13 TN (True Negative). The model accuracy has risen to almost 60% and a loss of 40% with the F1-score of 67%.

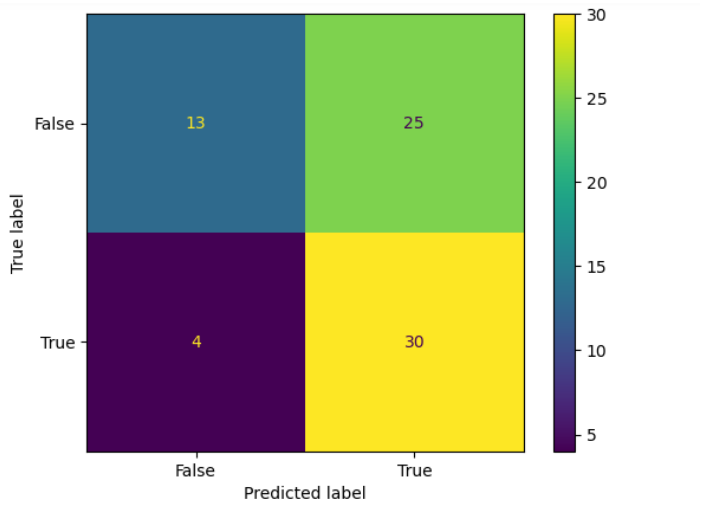

Figure 18 Confusion Matrix Graph / True vs Predicted

```
print("accuracy: %f " % accuracy_score(svc_pred, y_test))
print("MAE: %f" % mean_absolute_error(svc_pred, y_test))
print("F1: %f" % f1_score(svc_pred, y_test))
```

```
accuracy: 0.597222
MAE: 0.402778
F1: 0.674157
```

The accuracy rate for the hybrid model is 59% while the F1-score for trend prediction is 67% and a mean squared error of 67%.

3.4.4 Model Comparison

The table below shows the comparison for the evaluation result of the models.

	Accuracy	MSE	F1-Score
ARIMA Model	48%	51%	47%
CNN Model	54%	45%	58%
Hybrid Model	59%	40%	67%

Table 3 Model Comparison Result

Based on the table, the accuracy and F1-score are increasing as the model is shifted from traditional model to machine learning models. Furthermore, the model's loss is also decreasing by about 5%. This shows that the combination of traditional models and machine learning models can achieve a more accurate result compared to stand alone models.

4. Findings

The ARIMA model has a better way of interpreting a time series or linearly formed data even with a limited amount of data. To find the right model that will fit the dataset, the autocorrelation function (ACF) and partial autocorrelation function (PACF) were used to determine the number of possible values of autoregressive and moving average based on each lag of the data point. The differencing is used when the dataset is found not stationary. One of the example technique that was used to determine whether a data is stationary is by analyzing the dataset using Augmented Dickey Fuller (ADF) test. It is done by examining the statistical value of the (ADF) against the critical values and its p-value. By this analysis, the ARIMA model can then be set to train the data. The auto_arima function from Python statistical library can also be used to automatically detect the best ARIMA model for the dataset.

CNN model is best performed in a complex and large dataset to produce a close to accurate result. It is fed with a matrix form dataset and uses feature extraction to pull out a significant variable from the features. Thus, emphasizing the most relevant features and making predictions. It uses different kernels that perform mathematical computation on each network's node. Depending on the data, model structure and hyperparameter tuning, the outcome of the model can become underfitting or overfitting.

Support vector machine (SVM) is most useful when you have two classes to classify in a data. It uses hyperplanes to categorize and appoint the classified data to its close points. SVM uses kernels that also perform mathematical computation to find the best optimal hyperplanes for each data point. It is also ideal on the final dataset because of the limited amount of data. By utilizing this method, the trend prediction of both models was enhanced and unified, resulting in a higher accuracy prediction.

5. Discussion

This research paper shows that the combination of traditional model and machine learning technique performs better than the application of both as a single model. But because the market has unpredicted movement, predicting the next day trend is still a challenging task for the financial market analyst. The CNN feature extraction method was first attempted to use for the ARIMA model. Where the output from the most significant layer is extracted and arrange to make a dataframe that can be fed in an ARIMA model. However, the output array became complex to point out the exact dates where each data point was derived. The combination of ARIMA and CNN model to produce a much better prediction for a time series dataset is still yet to be achieved. It was said that other machine learning models are more consistent with a time series dataset, whether it's linear or non-linear. The example models are Long-Short Term Memory model (LSTM), Recurrent Neural Network (RNN) and Multi-Layer Perceptron (MLP). Unlike other time series analysis, financial time series analysis is still one of the most difficult tasks for financial experts, due to its inconsistent behavior because of too many affecting factors. This paper still needs further study to improve the model structure and predictions.

References

[1] Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET*)* (pp. 1-6). Ieee.

[2] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In 2014 UKSim-AMSS 16th international conference on computer modelling and simulation (pp. 106-112). IEEE.

[3] Asokan, M. (2022). A study of forecasts in Financial Time Series using Machine Learning methods. Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

[4] Capital One Tech. (2023). ARIMA model tips for time series forecasting in Python. *Capital One*. https://www.capitalone.com/tech/machine-learning/arima-model-time-series-forecasting/

[5] Chan, K. S., & Cryer, J. D. (2008). *Time series analysis with applications in R*. springer publication.

[6] Chen, J. (2023). What is the stock market, what does it do, and how does it work? Investopedia. https://www.investopedia.com/terms/s/stockmarket.asp

[7] Cracan, C. *(2020*). Retail sales forecasting using LSTM and ARIMA-LSTM: A comparison with traditional econometric models and Artificial Neural Networks.

[8] Devi, B. U., Sundar, D., & Alli, P. (2013). An effective time series analysis for stock trend prediction using ARIMA model for nifty midcap-50. International Journal of Data Mining & Knowledge Management Process, 3(1), 65.

[9] Di Pietro, M. (2022, January 3). Time Series Analysis for Machine Learning – towards Data Science. Medium. https://towardsdatascience.com/time-series-analysis-for-machine-learning-with-python-626bee0d0205

[10] Dor. (2021, October 7). Forecasting Time Series with Auto-Arima – Data Science Portfolio. https://www.alldatascience.com/time-series/forecasting-time-series-with-auto-arima/

[11] Durairaj, M., & Mohan, B. K. (2019). A review of two decades of deep learning hybrids for financial time series prediction. International Journal on Emerging Technologies, *10*(3), 324-331.

[12] Dwivedi, S. A., Attry, A., Parekh, D., & Singla, K. (2021, February). Analysis and forecasting of Time-Series data using S-ARIMA, CNN and LSTM. In 2021 international conference on computing, communication, and intelligent systems (icccis) (pp. 131-136). IEEE.

[13] Fiol-Roig, G., Miró-Julià, M., & Isern-Deyà, A. P. (2010). *Applying data mining techniques to stock market analysis.* In Trends in Practical Applications of Agents and Multiagent Systems: 8th

[14] Hayes, A. (2022). *What is a time series and how is it used to analyze data?* Investopedia. https://www.investopedia.com/terms/t/timeseries.asp

[15] Hoover, K. D. (2005). The methodology of econometrics. Available at SSRN 728683.

International Conference on Practical Applications of Agents and Multiagent Systems (pp. 519-527). Springer Berlin Heidelberg.

[16] Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Systems with Applications*,* 129, 273-285.

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

[18] Iamleonie. (2022). Time series: Interpreting ACF and PACF. Kaggle. https://www.kaggle.com/code/iamleonie/time-series-interpreting-acf-and-pacf

[19] Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3), 685-695.

[20] Kirchgässner, G., Wolters, J., & Hassler, U. (2012). *Introduction to modern time series analysis*. Springer Science & Business Media.

[21] Lendave, V. (2022). What are autocorrelation and partial autocorrelation in time series data? Analytics India Magazine. https://analyticsindiamag.com/what-are-autocorrelation-and-partial-autocorrelation-in-time-series-data/

[22] Nielsen, A. (2019). *Practical time series analysis: Prediction with statistics and machine learning*. O'Reilly Media.

[23] Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015*). Introduction to time series analysis and forecasting.* John Wiley & Sons.

[24] Pandey, A. K. (2021). Hands-On Stock Price Time Series Forecasting using Deep Convolutional Networks. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/08/hands-on-stock-price-time-series-forecasting-using-deep-convolutional-networks/

[25] Rehal, V. (2022, December 2). Interpreting ACF and PACF plots - SPUR ECONOMICS. SPUR ECONOMICS - *Learn and Excel*. https://spureconomics.com/interpreting-acf-and-pacf-plots/#:~:text=Autocorrelation%20Function%20(ACF)%20and%20Partial,lags%20for%20the%20ARIMA%20models.

[26] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.

[27] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). *Financial time series forecasting with deep learning: A systematic literature review*: 2005–2019. Applied soft computing, 90, 106181

[28] Sharma, K. (2023). Machine learning in finance: history, technologies and outlook. *Ubuntu.* https://ubuntu.com/blog/machine-learning-in-finance-history-technologies-and-outlook

[29] Shobana, G., & Umamaheswari, K. (2021, January). Forecasting by machine learning techniques and econometrics: *a review.* In 2021 6th international conference on inventive computation technologies (ICICT) (pp. 1010-1016). IEEE.

[30] Shweta. (2022, January 6). Introduction to Time Series Forecasting - towards Data science. Medium. https://towardsdatascience.com/introduction-to-time-series-forecasting-part-1-average-and-smoothing-models-a739d832315

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

[32] Stecanella, B. (2017, June 22). Support Vector Machines (SVM) algorithm explained. MonkeyLearn Blog. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[33] Strader, T. J., Rozycki, J. J., Root, T. H., & Huang, Y. H. J. (2020). Machine learning stock market prediction studies: *review and research directions*. Journal of International Technology and Information Management, 28(4), 63-83.

[34] Tabsharani, F. (2023). support vector machine (SVM). *WhatIs.com*. https://www.techtarget.com/whatis/definition/support-vector-machine-SVM

[35] Tsay, S. R. (2010). *Analysis of Financial Time Series* (third edition). John Wiley & Sons, Inc. https://doi.org/10.1002/9780470644560

[36] Tseng, F. M., Yu, H. C., & Tzeng, G. H. (2002). Combining neural network model with seasonal time series ARIMA model. *Technological forecasting and social change,* 69(1), 71-87.

[37] Wang, S., Tang, Z., & Chai, B. (2016, October). Exchange rate prediction model analysis based on improved artificial neural network algorithm. In *2016 International Conference on Communication and Electronics Systems (ICCES)* (pp. 1-5). IEEE.

[38] Wu, J. M. T., Li, Z., Srivastava, G., Frnda, J., Diaz, V. G., & Lin, J. C. W. (2020, December). A CNN-based stock price trend prediction with futures and historical price. *In 2020 International Conference on Pervasive Artificial Intelligence (ICPAI) (pp. 134-139). IEEE.*

[39] IBM. What are Convolutional Neural Networks? (n.d.). https://www.ibm.com/topics/convolutional-neural-networks

[40] Martinez, J. (2020, December 8). Introduction to Convolutional Neural Networks CNNs. Retrieved September 4, 2023, from https://aigents.co/data-science-blog/publication/introduction-to-convolutional-neural-networks-cnns