Using Hybrid Machine learning Models for Stock Price Forecasting and Trading.

Ahmed Khalil
aawkhalil@aucegypt.edu

THE AMERICAN UNIVERSITY IN CAIRO
الجامعة الأمريكية بالقاهرة
Graduate Studies

Using hybrid Machine learning models for stock price forecasting and trading.

A Thesis Submitted by

Ahmed Khalil

to the

Masters of Science in Finance

Graduate Program

5th of February 2024

In partial fulfillment of the requirements for the degree of

Masters

Table of content

Abstract:

Trading stocks of publicly traded companies in stock markets is a challenging topic since investors are researching what tools can be used to maximize their profits while minimizing risks, which encouraged all researchers to research and test different methods to reach such goal. As a result, the use of both fundamental analysis and technical analysis started to evolve to support traders in buying and selling stocks. Recently, the focus increased on using Machine learning models to predict stock prices and algorithmic trading as currently there is a huge amount of data that can be processed and used to forecast and trade stocks.

The focus of this paper is to use four machine learning models to forecast next day stock prices and trade stocks accordingly. The models used are LSTM model, W-LSTM model in which Wavelet analysis is used to remove the noise of the time series data and provide new coefficients to the LSTM model, LSTM-ARO where ARO as an optimization algorism will select the best hyperparameters for the LSTM model and W-LSTM-ARO model. The model's prediction accuracy is evaluated by MSE, MAE & R2, then the models are tested in terms of profit generation.

The companies studied in this paper are six companies listed on the New York Stock Exchange (NYSE): Apple (APPL), Microsoft (MSFT), Exxon Mobile (XOM), General Electric (GE), AT&T (T) and Procter and Gamble (PG).

The study concludes that there is no correlation between models with high prediction accuracy and the ability of the model to generate profits. There is no best model that will fit in trading all stocks. Finally the profits generated by W-LSTM & W-LSTM-ARO are higher the ones generated by LSTM & LSTM-ARO models. When the profits generated

by W-LSTM & W-LSTM-ARO are compared with the buy & hold, RSI & EMA trading strategies it was concluded that W-LSTM & W-LSTM-ARO models are able to generate profits when the stock is in a downtrend.

Chapter one: Introduction

The stock market is an exchange market for publicly traded stocks. The company's stock price is determined by supply and demand for the company stock; if there is a high demand for buying a specific company, the stock price of that company will rise, and vice versa. Forecasting the stock prices or market direction became a challenge, and various researchers began to look for solutions as in (Nti et al., 2019) and (Thakkar, et al., 2021).

The assumption of the efficient market hypothesis states that there is no way to predict market direction due to the availability of market information. However, researchers later discovered different approaches for predicting stock prices or market direction using fundamental analysis and technical analysis as Bustos, et al. (2020) discussed in his research.

Nti et al.(2019) explained the fundamental analysis focusing on the company's economic indicators rather than the stock price movement. On the other hand, they mentioned that technical analysis focuses on stock price trend.

Technical analysis is based on calculatinga stock's time series data as the daily closing price, with the assumption that real-time stock prices include all market information. They are used to forecast future stock price trends based on historical data patterns. Technical indicators include the Relative Strength Index (RSI), the Simple Moving Average (SMA), and the Exponential Moving Average (EMA). Algorithmic trading can make use of such indicators.

Gülmez (2023) and Milana, et al.(2021) explained Algorithmic trading as the use of computing power to analyze stock market information as technical or fundamental information, and its ability to either generate trading signal (buy, sell or hold) or to execute automated trading. The importance of algorithmic trading has recently increased because it allows investors to trade stocks faster and without emotional intervention while relying on qualitative data. Its efficiency comes from its ability to extract and analyze massive amounts of real-time data, as well as identify trading opportunities. From the data that can be fed to the algorithmic trading system are Stock prices, market trends, news, and technical indicators to be analyzed and as an output market trend will be generated; Accordingly, a buy or sell decision will be made per predefined rules.

Such trading signals can be generated by using technical indicators or Machine learning methods. Keshavarz, et al. (2002) explored trading using technical indicators which rely on using technical indicators as Simple Moving Average (SMA), Exponential Moving Average (EMA) and Relative Strength Index (RSI) which are used to detect a pattern from the previous stock prices and generate either a buy or sell signals.

Thakkar, et al. (2021) explained the importance of Machine learning in the financial sectors due to the increase in the amount of accessible data and its ability to process massive amounts of data effectively.

Machine learning is founded on mathematics, probability and statistics. Milana, et al.(2021) mentioned the three methods to use machine learning supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is a method that uses labelled data to generate a correlation between input and output, similar to regression analysis, and it can be used in prediction. Unsupervised learning, on the other hand, is a method that uses unlabeled data and the model will find a pattern, similar to clustering, and it can be used for customer segmentation. Finally, in reinforcement learning, it learns while interacting with its surroundings and a score is recorded during the learning process.

The scope of this paper is to study the efficiency of both LSTM models and different hybrid LSTM models in stock price prediction and also their ability to generate trading profits. Since there is a gap in research papers where they only study the prediction accuracy of machine learning models without studying such models' ability to generate trading profits.

The models explored in this paper are LSTM model, W-LSTM model in which Wavelet analysis is used to remove the noise of the time series data and provide new coefficients to the LSTM model, LSTM-ARO where ARO is an optimization algorithm selecting the best hyperparameters for the LSTM model and finally the proposed model W-LSTM-ARO. After comparing the forecasting performance of the four models by using three evaluation criteria, buy and sell signals will be generated to execute trades and evaluate the models by their ability to generate profits and to compare it buy and hold strategy and also with technical indicators strategies which are EMA & RSI

Finally, in this paper, we will discuss in chapter two the contribution of other researchers. Then the data set and the models used in our study will be explained in chapter three. followed by discussing the results in chapter four. Finally, chapter five will cover the conclusion of our observations and future work.

Chapter two: Literature review

Keshavarz, et al. (2002) studied 11 technical indicators on stocks listed in the Tehran Stock Exchange and their accuracy in generating buy & sell signals by using different periods (Weekly, Monthly, Quarterly & 6 months). The 11 technical indicators included in this study are the Moving Average (MA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Hull Moving Average (HMA), Relative Strength Index (RSI), The True Strength Index (TSI), Rate of Change (ROC), Average Directional Movement Index (ADX), Average True Range (ATR), Aroon, and Commodity Channel Index (CCI). The conclusion is that the signals generated by MA, EMA & RSI either to buy or sell are more accurate than the one generated by other indicators and more reliable to consider in trading strategies by investors as they generate more profits with smaller risk.

Gandhmal, et al. (2019) explained that price prediction methods include statistical and machine learning methods. Statistical models depending on past stock prices and return include models such as ARMA (Auto-Regressive Moving Average), GARCH (Generalized Auto-Regressive Moving Average) and ARCH (Auto-Regressive Conditional Heteroscedastic). The machine learning method examples are NN (Neural Network), GA (Genetic Algorithm), and Back Propagation algorithm. These models are using time series models to predict the stock returns. Wang, et al. (2019) elaborated that Machine learning algorithms have an edge over the statistical models in dealing with non-stationary and non-linear data.

Yadov, et al., (2021) applied different hybrid machine learning models on 4 stocks in the American stock market. The data used in this study included five companies which are Uber, Apple, Nike Inc and Meta (Facebook). The conclusion of this study is that FastRNN & ASTRNN_CNN_BiLSTM have superiority in predicting the stock pricing over other hybrid and other state-of-the-art models as ARIMA, BiLSTM_Attention_CNN_BiLSTM,CNN_LSTM_Attention_LSTM, LSTM_Attention_CNN_BiLSTM and LSTM_Attention_LSTM.

Akpan, et al., (2021) used Wavelet transform function to remove noise from S&P500 time series data and feed the output to the LSTM model to perform prediction on the return after 15 mins. The W-LSTM model was compared with the logistic regression model (LR) and Wavelet Logistic Regression model (W-RL). As conclusion W-LSTM & LSTM models' accuracy was higher than traditional methods as Logistic regression & W-RL models in predicting the S&P500 price movement.

Gülmez (2023) and Haugland (2022) used Long Short-Term Memory (LSTM) which is a machine learning model considered as one of the Recurrent Neural Network (RNN) models. This model is used in the analysis of time series data such as stock prices and helps in forecasting. It is considered a great tool in dealing with analyzing the stock market due to its ability to process different data types that can influence the change in the stock price. Stock prices can be influenced by different factors such as company news, economic indicators, and market trend.

Haugland (2022) explained how the LSTM model predicts, and described how the data go through three gates in an LSTM model which are the input gate, forget gate and output gate. The input gate controls which information from the previous cell and the current input can enter the current cell. The forget gate specifies which information from the previous cell should be ignored. The output gate controls which information from the current cell should be disregarded, and these gates are controlled by hyperparameters

Haugland (2022) elaborated the importance of Model Optimization for machine learning models, since it works on selecting the best hyperparameters which leads to best performance and accuracy. These hyperparameters, acting as external factors, which influence the behavior and performance of learning algorithms.

Guo.T, et al (2022) introduced the Wavelet analysis to process the data before using it to feed the LSTM mode. Wavelet analysis is a mathematical method which decomposes data as stock market data into signals of both high and low frequencies, and this output is called wavelets. This method allows for understanding the complex patterns and trends inherent in stock market data, which supports the performance of LSTM model in order to perform better in predicting values.

Gavriel (2023) used LSTM model to predict S&P500 index. The model was trained by using two different data sets where the first one was using a single predictor and the other one contained different predictors. The study concluded that LSTM was able to predict the S&P 500 price movement.

Gülmez (2023) introduced the Artificial Rabbits Optimization algorithm (ARO) to select LSTM model hyperparameters for higher prediction accuracy and compared it with different machine learning models. The data set used was stocks in DIJA index, and the LSTM-ARO was compared with one ANN model, three LSTM models (LSTM-1D, LSTM-2D & LSTM-3D) and LSTM-GA in which the hyperparameters are optimized by using Genetic Algorithm. the evaluation criteria used to evaluate the models are MSE, MAE, MAPE and R2. The study concluded that LSTM-ARO was able to perform prediction with higher accuracy.

Gülmez (2023) introduced the use of ARO as one of the Metaheuristic Algorithms, which are considered as a powerful tool for searching and selecting the best hyperparameters for LSTM model. It has an edge over the traditional optimization methods since it can search in a wider search space and find optimum results without getting stock in local minima. Also, it can deal with noisy and complex data as in LSTM model since there are many hyperparameters in LSTM models. Finally, the Artificial Rabbits Optimization algorithm (ARO) is inspired by rabbit survival strategies to reach optimization solutions, and it can be used to optimize machine learning algorithms like LSTM.

Chapter three: Data & Methodology

3.1 Models

The models used in this paper include LSTM, W-LSTM, LSTM-ARO and W-LSTM models. These four models will be evaluated by two different methods. Firstly, the prediction accuracy will be evaluated by MSE, MAE and R2, then the models are evaluated from a profit generation perspective by using a simple trading strategy. Finally, the profitability of the models will be compared against the Buy and hold strategy in addition to traditional trading strategies using EMA & RSI technical indicators.

3.1.1 LSTM

LSTM model is applied on time series data to perform forecasting on the next day price in this study. LSTM has the power to perform prediction that is based on long-term dependencies in the time series data, which is a drawback because it may miss the short-term dependencies that can affect the prediction accuracy.

LSTM cells composition that supports its ability to manage long-term dependencies consists of:

Input gate: controls which information from the previous cell and the current input can pass to the current cell.

Forget gate: specifies which information from the previous cell should be ignored.

Output gate: controls which information from the current cell should be disregarded.

The activation functions manage the information flow within the cell and operate these gates.

The LSTM model training process is managed by using hyperparameters. it can be either configured before training or selected by the optimization algorithm. Below are a few of the key hyperparameters:

Learning rate: determines how frequently is the weights of the model are changed during training. The faster convergence may result from a larger learning rate, but overfitting of the model may become more likely.

Number of Neurons: decides how many neurons in each LSTM layer. The higher the number of the neurons the higher model's ability to learn complex patterns, however, the computing capacity will increase too.

Batch size: determines how many training samples are processed in a single phase. The higher batch size may lead to higher training efficiency. However, it will lead to higher computing capacity.

Number of Epochs: determine how many times the dataset will be used in training the model. the higher the number of Epochs may lead to better performance and also may cause overfitting.

Dropout: it helps in preventing overfitting by dropping some neurons randomly from the network during the training process.

3.1.2 W-LSTM

Wavelet LSTM model is the combination of both the LSTM and wavelet transform. Wavelet transform is used to change the input data as time series to set of frequencies which are low & high frequencies. The generated coefficients help in capturing both long- and short-term data patterns. Such hybrid model solves the limitation of LSTM capturing short term decencies.

3.1.3 LSTM-ARO

LSTM – Artificial Rabbits Optimization algorithm (ARO) model is the combination of both LSTM and ARO, where the ARO is used to optimize the LSTM hyperparameters by selecting the best hyperparameters to perform high accuracy prediction on the used dataset by measuring the error of the prediction. However, this will increase the need for higher computing power.

3.1.4 W-LSTM-ARO

Wavelet -LSTM – Artificial Rabbits Optimization algorithm (ARO) model is the combination of Wavelet decomposition, LSTM and ARO, which will lead to enhancing LSTM model performance since Wavelet decomposition will make LSTM model capture both short- and long-term dependencies and ARO will optimize the hyperparameters for better accuracy in the stock price prediction.

3.2 LSTM Hyperparameters

The hyper parameters for LSTM & W-LSTM are configured manually while ARO is selecting the best hyperparameters for LSTM-ARO & W-LSTM-ARO where the values are:

Hyper Parameter	**LSTM**	**W-LSTM**	**LSTM - ARO**	**W -LSTM- ARO**
Unites	150	150	50 - 200	50 - 200
Dropout-1	0.3	0.3	0.2 – 0.5	0.2 – 0.5
Drop out-2	0.3	0.3	0.2 – 0.5	0.2 – 0.5
Epochs	20	20	10 - 30	10 - 30
Batch Size	32	32	8 - 32	8 - 32

3.3 Data Set

The data were extracted by using Yahoo Finance APIs, it is a free source to extract stock prices and other information such as financial statements and company's news, and all models were developed by Python language. The company studies in this paper are six companies listed on the New York Stock Exchange (NYSE) : Apple (APPL), Microsoft (MSFT), Exxon Mobile (XOM), General Electric (GE), AT&T (T) and Procter and Gamble (PG). The daily closed price & volume for theses six companies are extracted from six years from 1st of Jan 2016 to 31st of Dec 2022. The time window used for the prediction is the previous 20 days. Finally, the Technical Indicators EMA & RSI are calculated by using Pandas_TA library in python.

The predictors used as inputs to the models are three predictors to LSTM and LSTM-ARO which are daily close price, daily trade volume and RSI indicator; And six predictors to W-LSTM & W-LSTM-ARO, which are daily close price, daily trade volume, RSI indicator, daily close price wavelet coefficient, daily trading volume – wavelet coefficient and RSI indicator wavelet coefficient. All the used predictors are used in the four LSTM models are as the following:

Predictors	**LSTM**	**LSTM - ARO**	**W-LSTM**	**W -LSTM- ARO**
Daily close price	Yes	Yes	Yes	Yes
Daily Trading volume	Yes	Yes	Yes	Yes
RSI indicator	Yes	Yes	Yes	Yes
Daily close price – Wavelet Coefficient	No	No	Yes	Yes
Daily Trading volume – Wavelet Coefficient	No	No	Yes	Yes
RSI indicator -Wavelet Coefficient	No	No	Yes	Yes
Total number of Predictors	**3**	**3**	**6**	**6**
Legend	**Yes: the data is used in the model** **No: the data is not used in the model**			

3.4 Evaluation criteria:

(Gülmez, 2023) evaluation criteria can be used to evaluate the accuracy and efficiency of the models in predicting future values.

Mean Absolute Error (MAE), R-squared score (R2) and Mean Squared Error (MSE) are used to evaluate the four models in this study.

3.4.1 Mean Absolute Error (MAE)

MAE measures the quality of the prediction values by taking the difference between predicted values by the model and the real values used as input and taking an average of all absolute values of the calculated differences. The best model has the lower MSE value

$$MAE = \frac{1}{n} \sum_{1}^{n} |y_i - \hat{y}_i| \tag{1}$$

3.4.2 R-Squared score (R2)

R2 describes how the model is fitting the used data. R2 value gets calculated by taking the total value of differences between the predicted value and the real value used as input squared, and dividing them by the sum the difference between mean value of real values (input) and the real values squared. The best model has R2 value nearest to one.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{2}$$

3.4.3 Mean Squared Error (MSE)

MSE measures the quality of the prediction values by taking the difference between predicted values by the model and the real values used as input and squaring the result. The best model has the lower MSE value.

$$MSE = \frac{1}{n} \sum_{i}^{n} \left(y_i - \hat{y}_i\right)^2 \qquad (3)$$

3.5 Trading strategies:

There are the strategies used to either buy or sell stocks and can be influenced by fundamental analysis, technical analysis or machine learning. The data used to test the trading strategies is from 11 October 2021 to 30 Dec 2022.

3.5.1 Machine learning trading signals

After predicting the next day price, if the predicted price is higher than today close price then it is buying signal, if the Predicted price is lower than the close price then it is selling signal. Else it is a hold signal.

3.5.2 Technical Indicator EMA signals

If the EMA-Short (20) is greater than EMA-Long (50) then it is buy signal, else it is a sell signal.

3.5.3 Technical Indicator RSI signals

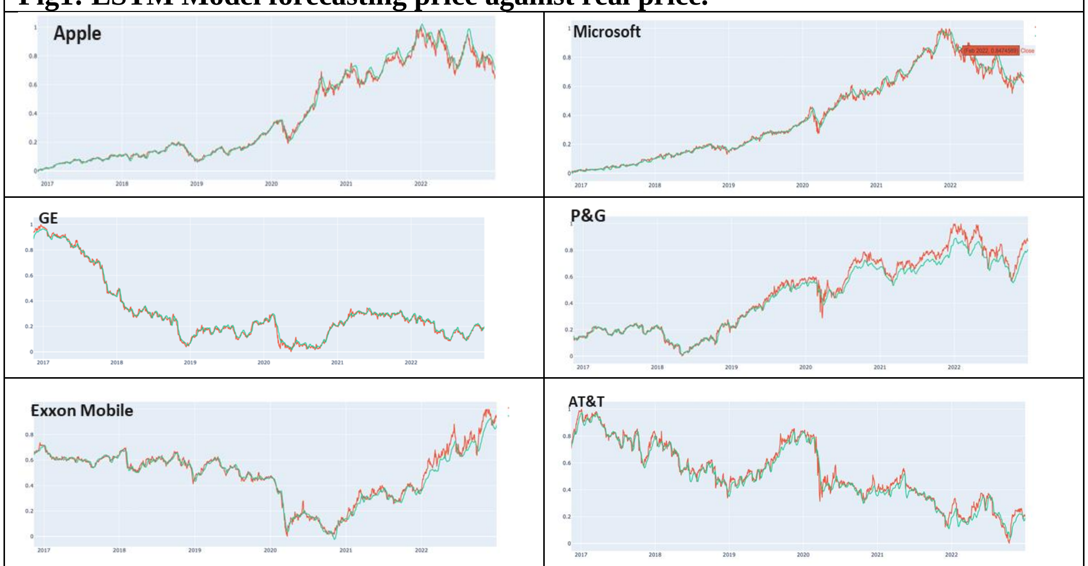
If the price is higher than RSI (70) then it is selling signal, if the price is lower than RSI (30) then it is a buy signal. Else it is hold signal.

Chapter four: Discussion & Results

The scope of the research is to test different machine learning models to evaluate their ability to predict next-day price, and their ability to execute profitable trades. The evaluation criteria used to evaluate the stock price prediction are MAE, MSE & R2. In order to evaluate the models regarding profit generations they are tested against each other and also other trading strategies which are buy and hold and trading strategies using technical indicators which are RSI & EMA.

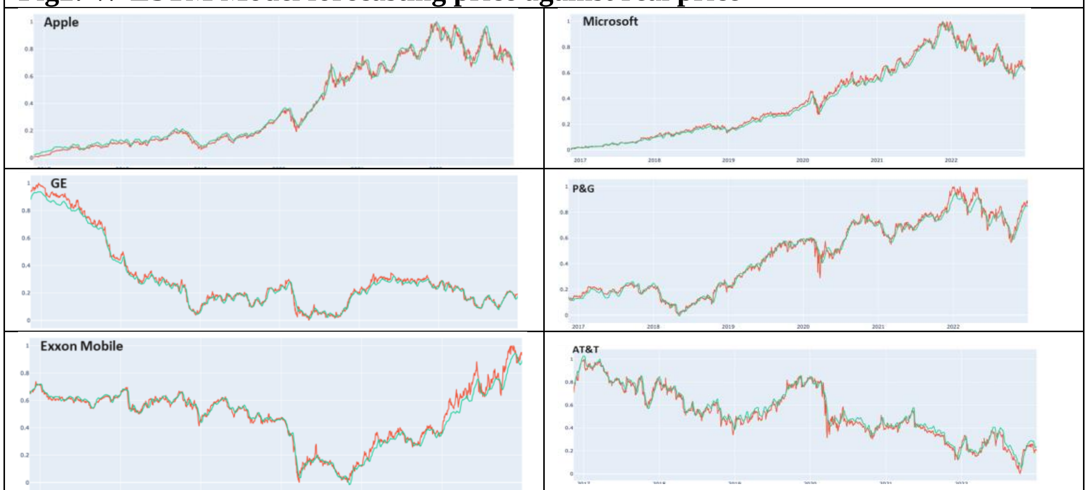
The graphs of the prediction results generated by of LSTM model against real values are displayed in Fig1, the one generated by W-LSTM are listed in Fig2, LSTM-ARO are listed in Fig3 and finally W-LSTM-ARO are listed in Fig4.

Fig1: LSTM Model forecasting price against real price.



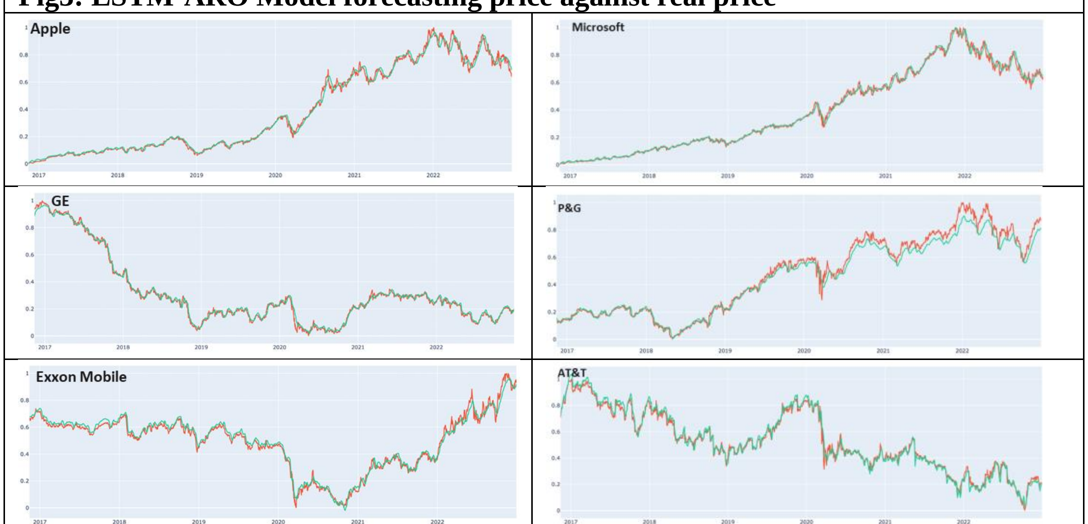
Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig2: W-LSTM Model forecasting price against real price



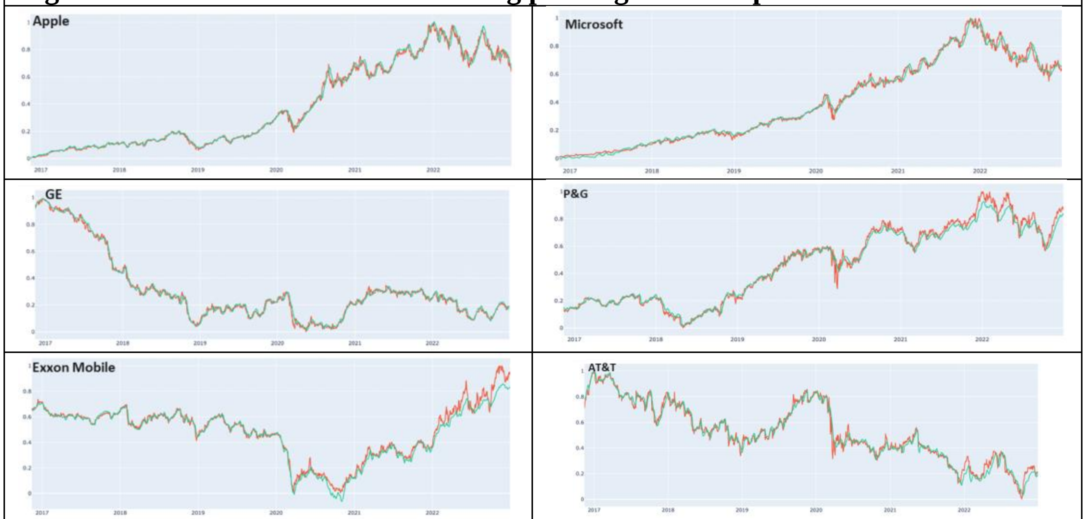
Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig3: LSTM-ARO Model forecasting price against real price



Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig4: W-LSTM-ARO Model forecasting price against real price



Green line: the actual daily close price
Red: the forcasted daily close stock price

The result of stock price predictions shows superiority to LSTM-ARO model since four companies out of six have the lowest MAE as displayed in Fig5, five companies out of six has the lowest MSE as displayed in Fig6 and four companies out of six have the highest R2 as displayed in Fig7.

Fig5: Mean Absolute Error (MAE)

Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	0.018071	0.022323	0.016586	**0.014849**
Microsoft	0.016566	0.020751	**0.013368**	0.017378
Exxon Mobile	0.025721	0.026099	**0.024091**	0.026077
General Electric	0.018260	0.020330	**0.014274**	0.014397
AT&T	0.030563	0.030328	**0.023077**	0.026075
Procter & Gamble	0.034969	**0.023802**	0.025999	0.024216

The number in Bold are the highest number per row

Fig6: Mean Squared Error (MSE)				
Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	0.002309	0.001915	0.001685	**0.001375**
Microsoft	0.001803	0.001594	**0.001174**	0.001503
Exxon Mobile	0.004969	0.004533	**0.001874**	0.005338
General Electric	0.000430	0.000281	**0.000274**	0.000280
AT&T	0.002372	0.002026	**0.000967**	0.002387
Procter & Gamble	0.006473	0.002492	**0.001735**	0.003291
The number in Bold are the highest number per row				

Fig7: R-Squared (R2)				
Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	0.991838	0.991445	0.993649	**0.994705**
Microsoft	0.992812	0.990925	**0.995195**	0.993263
Exxon Mobile	0.966468	0.967070	**0.978954**	0.963754
General Electric	0.991165	0.988933	**0.994328**	0.994242
AT&T	0.967709	0.968592	**0.980714**	0.973881
Procter & Gamble	0.970575	**0.986225**	0.985237	0.984993
The number in Bold are the highest number per row				

After applying simple trading strategy, which generate a buy signal if the forecasted price is higher than the closed price, sell signal if the forecasted price is lower than the close price and hold if there is no change between the forecasted and the close price on the four LSTM models, the result was contradicting with the result from the stock price prediction efficiency, which showed that LSTM-ARO is the best model in terms of forecasting accuracy. Since LSTM-ARO has the worst performance in terms of profit generations when it is compared with LSTM, W-LSTM and W-LSTM-ARO models trading strategies as shown in Fig8.

Fig8: Trading revenues/losses generated by machine learning models.				
Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	23%	13%	1%	**34%**
Microsoft	31%	**49%**	12%	23%
Exxon Mobile	-32%	-21%	-33%	**9%**
General Electric	21%	**43%**	-4%	11%
AT&T	-23%	-30%	-46%	-39%
Procter & Gamble	-18%	**3%**	0%	-9%
The percent in Bold is the highest profit per row The percent in red is the highest loss per row				

The results displayed in Fig8 shows the superiority of both W-LSTM & W-LSTM-ARO in terms of profit generation since W-LSTM model was able to generate 49% from trading on Microsoft stock, 43% from trading on General Electric and 3% from trading on Procter & Gamble. Using W-LSTM-ARO model generated 34% profit from trading on Apple and 9% profit from trading on Exxon Mobile. The profits generated by LSTM model can be considered good relatively to the other models and for sure in comparison to LSTM-ARO which generated the lowest profits and three companies generated losses upon trading using it. Such result is justifying the need of LSTM model to consider both long- & short-term dependencies that is provided by Wavelet decomposition in order to predict the accurate direction of the next day price movement. Since wavelet decomposition solves the limitation of LSTM model in capturing the short-term dependences. After comparing the machine learning models together, the models were compared against other trading strategies which are Buy and Hold strategy and technical indicator strategies which are EMA & RSI as shown in Fig9.

Fig9: Trading revenues/losses generated by machine learning models and other trading strategies.							
Name	**Buy and Hold**	**EMA**	**RSI**	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	-2%	12%	9%	23%	13%	1%	**34%**
Microsoft	-13%	-12%	6%	31%	**49%**	12%	23%
Exxon Mobile	**64%**	-29%	-2%	-32%	-21%	-33%	**9%**
General Electric	-15%	-16%	-3%	21%	**43%**	-4%	11%
AT&T	-5%	-19%	**12%**	-23%	-30%	-46%	-39%
Procter and Gamble	**9%**	-3%	-1%	-18%	3%	0%	-9%
The percent in Bold is the highest profit per row The percent in red is the highest loss per row							

When the seven trading strategies using the four LSTM models, two technical indicators RSI & EMA and buy and hold it was observed that W-LSTM, W-LSTM-ARO and Buy & Hold strategies are the best trading strategies. Since the profit generated by trading Exxon Mobile using buy & hold and Procter and Gamble were more profitable than trading them with the rest of used trading strategies putting in consideration that both assets were in an uptrend. On the other hand, RSI strategy worked well with only one asset which is AT&T which was in a downtrend.

However, W-LSTM & W-LSTM-ARO models generated the highest profits by trading one two assets each. Such a result shows the edge of machine learning models against the traditional trading strategies used in this paper in their ability to generate profits even if the asset stock price is in downtrend. This clarifies that W-LSTM & W-LSTM-ARO were able to predict the right signal to either buy, sell or hold the stock.

Finally, after conducting this study it became very clear that the evaluation criteria as EMA and SMA are not enough to evaluate the prediction model accuracy, and it is very important to apply the models on a trading strategy to test its ability to generate profit. Since there is no correlation between having the best evaluation criteria and the ability to generate profit as what we saw with the LSTM-ARO Model as shown in this paper. In addition to that, no one model can work well with all assets which indicates that traders should use different models for each set of assets as observed in fig9 since there was no one model with the best results against the others.

Chapter five: Conclusion and future work

In this paper, we tested four different machine learning model LSTM, Wavelet LSTM (W-LSTM) and two LSTM models optimized with ARO which are LSTM-ARO and W-LSTM-ARO. The performance of the models was evaluated by two different methods, the first one is evaluation criteria using MAE, MSE & R2 which evaluate the prediction accuracy, and the second one is using trading strategy which generates buy, sell & hold signals to check the ability of the four models to generate profits. Finally, their performance is compared with other transregional trading strategies which are Buy & Hold and technical indicator strategies in which used EMA & RSI.

The results from the evaluation criteria using MAE, MSE and R2 were in favor of LSTM-ARO model, which indicated that LSTM-ARO has the best prediction accuracy against the other three models used. On the other hand, this model was not able to generate any profits since three assets generated losses, one asset generated no profit and the remaining two assets generated small profits in comparison with the other three LSTM models. In addition to that W-LSTM & W-LSTM-ARO models were able to generate good profits while the assets were in a downtrend as shown in Fig9 with companies Apple, Microsoft and General Electric, however, they were not able to generate profits when the assets were in an uptrend as with General Electric & Procter &Gamble.

Finally, this paper proved that there is no correlation between machine learning model high prediction accuracy measured by different evaluation criteria as MAE, MSE and R2 and their ability to generate profits. In addition to that, no model will be suitable to trade all assets as displayed in Fig 9 since each model was able to generate the highest profits with certain stock/assets. This indicated that traders have to train different models and test them on trading strategy to check their ability to generate profits. Furthermore, considering buy & hold strategy and traditional trading strategies using technical indicators may be helpful sometimes.

For future work, adding other machine learning models as XGBoost, CNN_LSTM_Attention_LSTM, LSTM_Attention_CNN_BiLSTM and LSTM_Attention_LSTM and comparing their prediction accuracy using MAE, MSE & R2 with the models used, and test them from profit generation perspective. In addition to that apply the models on a larger set of assets to include not only stocks but also commodities & crypto-currencies to test the model's prediction & profit generation abilities in a highly volatile market.

References

1- Gülmez, B. (2023, October 1). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Systems With Applications. https://doi.org/10.1016/j.eswa.2023.120346

2- Gavriel, S. (2023). Stock Market Prediction using Long Short-Term Memory. Retrieved from http://essay.utwente.nl/85660/1/Research_Project_Stelios_Gavriel.pdf

3- Haugland, J. (2022). A comparative study of machine learning models in stock price prediction. Retrieved from https://biopen.bi.no/bixmlui/bitstream/handle/11250/3099216/MasterThesis.pdf?sequence=1

4- Guo, T., Zhang, T., Lim, E., Lopez-Benitez, M., Ma, F., & Yu, L. (2022). Review of Wavelet Analysis and Its Applications: Challenges and Opportunities. In IEEE Xplore Digital Library. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9785993

5- Keshavarz, S., Sereshk, M. V., Ataabadi, A. A., & Arman, M. H. (2022). Trading strategies based on trading systems: Evidence from the performance of technical indicators. Retrieved from https://sjsm.shiraz.iau.ir/article_688034_eb5711be0c683d69ea9fdc164b67dff0.pdf

6- Gandhmal, D., & Kumar, K. (2019, August 28). Systematic analysis and review of stock market prediction techniques. Retrieved May 05, 2021, from https://www.sciencedirect.com/science/article/abs/pii/S157401371930084X#b23

7- Wang, W., Li, W., Zhang, N., & Liu, K. (2019, October 17). Portfolio formation with preselection using deep learning from long-term financial data. Retrieved May 05, 2021, from https://www.sciencedirect.com/science/article/abs/pii/S0957417419307596

8- Thakkar, A., & Chaudhari, K. (2019, September 1). A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems With Applications. https://doi.org/10.1016/j.eswa.2021.114800

9- Akpan, U. E., & Korda, D. R. (2019). Prediction of Stock Market Returns using LSTM Model and Traditional Statistical Model. In ResearchGate. Retrieved from https://www.researchgate.net/profile/D-Redeemer-Korda/publication/356417853_Prediction_of_Stock_Market_Returns_using_LSTM_Model_and_Traditional_Statistical_Model/links/619ef78600cb7421f1d6d8f5/Prediction-of-Stock-Market-Returns-using-LSTM-Model-and-Traditional-Statistical-Model.pdf

10- Yadav, K., Yadav, M., & Saini, S. (2019). Stock values predictions using deep learning based hybrid models. CAAI Transactions on Intelligence Technology, 7(1), 107–116. https://doi.org/10.1049/cit2.12052

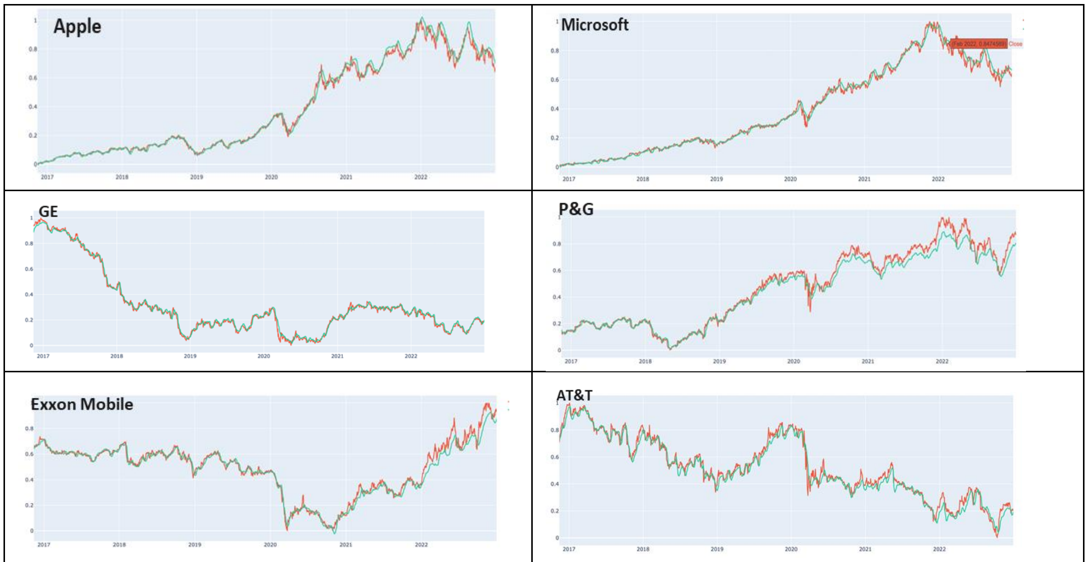
11- Milana, C., & Ashta, A. (2019). Artificial intelligence techniques in finance and financial markets: A survey of the literature. Strategic Change, 30(3), 189-209. https://doi.org/10.1002/jsc.2403

12- Bustos, S., & Pomares-Quimbaya, A. (2019, October 1). Stock market movement forecast: A Systematic review. Expert Systems With Applications. https://doi.org/10.1016/j.eswa.2020.113464

13- Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2019, August 20). A systematic review of fundamental and technical analysis of stock market predictions. Artificial Intelligence Review. https://doi.org/10.1007/s10462-019-09754-z

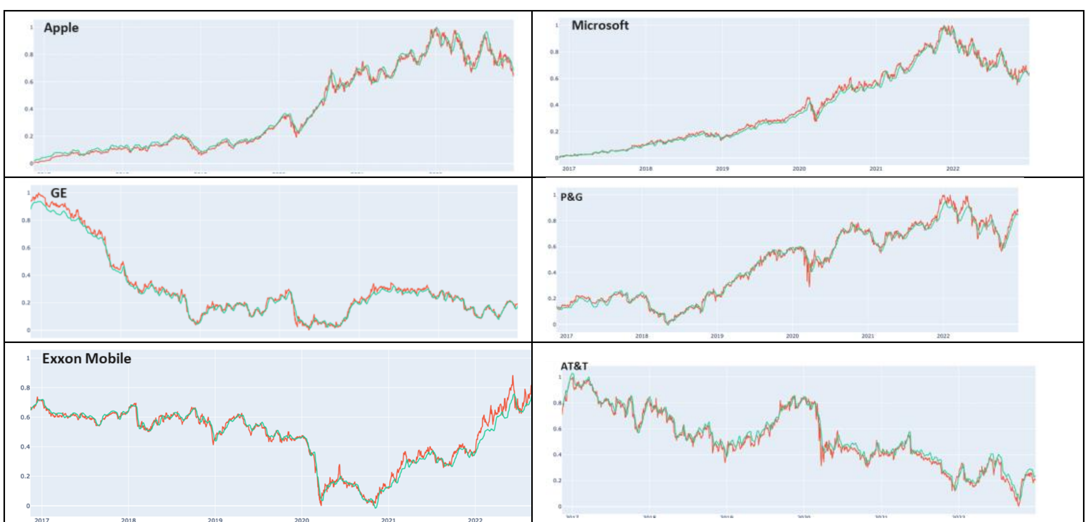
Appendix

Fig1: LSTM Model forecasting price against real price.



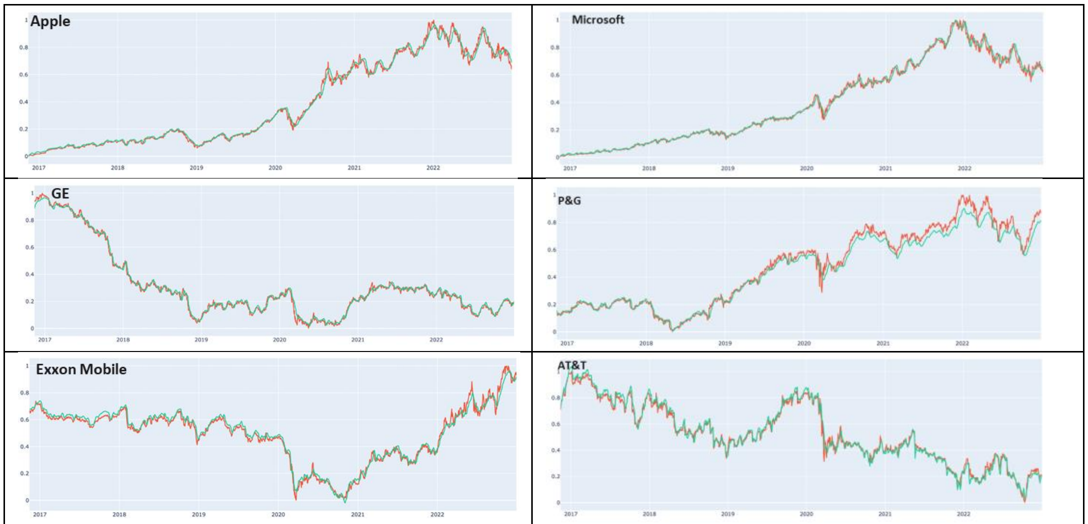
Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig2: W-LSTM Model forecasting price against real price



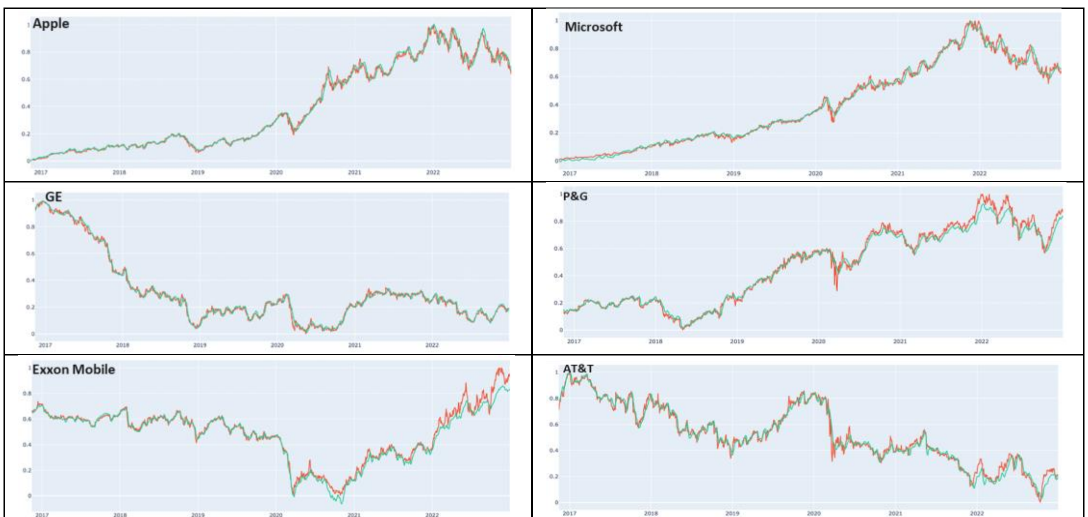
Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig3: LSTM-ARO Model forecasting price against real price



Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig4: W-LSTM-ARO Model forecasting price against real price



Green line: the actual daily close price
Red: the forcasted daily close stock price

Fig5: Mean Absolute Error (MAE)

Mean Absolute Error (MAE)				
Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	0.018071	0.022323	0.016586	**0.014849**
Microsoft	0.016566	0.020751	**0.013368**	0.017378
Exxon Mobile	0.025721	0.026099	**0.024091**	0.026077
General Electric	0.018260	0.020330	**0.014274**	0.014397
AT&T	0.030563	0.030328	**0.023077**	0.026075
Procter & Gamble	0.034969	**0.023802**	0.025999	0.024216
The number in Bold are the highest number per row				

Fig6: Mean Squared Error (MSE)

Mean Squared Error (MSE)				
Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	0.002309	0.001915	0.001685	**0.001375**
Microsoft	0.001803	0.001594	**0.001174**	0.001503
Exxon Mobile	0.004969	0.004533	**0.001874**	0.005338
General Electric	0.000430	0.000281	**0.000274**	0.000280
AT&T	0.002372	0.002026	**0.000967**	0.002387
Procter & Gamble	0.006473	0.002492	**0.001735**	0.003291
The number in Bold are the highest number per row				

Fig7: R-Squared (R2)

R-Squared (R2)				
Name	**LSTM**	**W-LSTM**	**LSTM-ARO**	**W-LSTM-ARO**
Apple	0.991838	0.991445	0.993649	**0.994705**
Microsoft	0.992812	0.990925	**0.995195**	0.993263
Exxon Mobile	0.966468	0.967070	**0.978954**	0.963754
General Electric	0.991165	0.988933	**0.994328**	0.994242
AT&T	0.967709	0.968592	**0.980714**	0.973881
Procter & Gamble	0.970575	**0.986225**	0.985237	0.984993
The number in Bold are the highest number per row				

Fig8: Trading revenues/losses generated by machine learning models.

Name	LSTM	W-LSTM	LSTM-ARO	W-LSTM-ARO
Apple	23%	13%	1%	**34%**
Microsoft	31%	**49%**	12%	23%
Exxon Mobile	-32%	-21%	-33%	**9%**
General Electric	21%	**43%**	-4%	11%
AT&T	-23%	-30%	-46%	-39%
Procter & Gamble	-18%	3%	0%	-9%
The percent in Bold is the highest profit per row The percent in red is the highest loss per row				

Fig9: Trading revenues/losses generated by machine learning models and other trading strategies.

Name	Buy and Hold	EMA	RSI	LSTM	W-LSTM	LSTM-ARO	W-LSTM-ARO
Apple	-2%	12%	9%	23%	13%	1%	**34%**
Microsoft	-13%	-12%	6%	31%	**49%**	12%	23%
Exxon Mobile	**64%**	-29%	-2%	-32%	-21%	-33%	**9%**
General Electric	-15%	-16%	-3%	21%	**43%**	-4%	11%
AT&T	-5%	-19%	**12%**	-23%	-30%	-46%	-39%
Procter and Gamble	**9%**	-3%	-1%	-18%	3%	0%	-9%
The percent in Bold is the highest profit per row The percent in red is the highest loss per row							

Python Libraries used.

```
import pandas as pd
import plotly.express as px
from copy import copy
from scipy import stats
import matplotlib.pyplot as plt
import numpy as np
import plotly.figure_factory as ff
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from tensorflow import keras
from tensorflow.keras.callbacks import EarlyStopping
import yfinance as yf
import pandas_ta as ta
import optuna
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, r2_score,
mean_squared_error
import tensorflow as tf
import pywt
```

Calculating EMA & RSI technical indicators

```
# Exponential Moving Averages
price_volume_df['EMA_S'] = ta.ema(price_volume_df['Close'],
length=20)
price_volume_df['EMA_L'] = ta.ema(price_volume_df['Close'],
length=50)

# Relative Strength Index (RSI)
price_volume_df['RSI'] = ta.rsi(price_volume_df['Close'], length=14)
```

Wavelet decomposition code.

```
# Define the function to apply wavelet analysis
def apply_wavelet_analysis(data, wavelet='haar', level=3):
    coeffs = pywt.wavedec(data, wavelet, level=level)
```

```
    return coeffs
# new code wavelet analysis
def apply_wavelet_analysis(column_data, wavelet='haar', level=3):
    coeffs = pywt.wavedec(column_data, wavelet, level=level)
    flattened_coeffs,_ = pywt.coeffs_to_array(coeffs)
    trimmed_coeffs = flattened_coeffs[:len(column_data)]
    return trimmed_coeffs

# Apply wavelet analysis and extract features for each column
wavelet_columns = price_volume_df.columns
#all_predictors  =  price_volume_df.drop(columns=wavelet_columns)   #
Remove original columns
#wavelet_columns                                                    =
price_volume_df.columns.drop(columns=wavelet_columns)

for column in wavelet_columns:
                   price_volume_df[f'{column}_Wavelet_Features']    =
apply_wavelet_analysis(price_volume_df[column].values)
```

Preparing data for LSTM Model

```
training_data = price_volume_df.iloc[:, 0:6].values

# Normalize the data
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_data)

# Define the window size for the previous days
window_size = 20  # Change this value to adjust the number of previous
days


X = []
y = []

# Creating the training data with the specified window size
for i in range(window_size, len(training_set_scaled)):
    X.append(training_set_scaled [i - window_size:i, :]) # change in
the rest
    y.append(training_set_scaled [i, 0])
```

```
X = np.asarray(X)
y = np.asarray(y)


# Split the data into training, validation, and test sets
split_train = int(0.6 * len(X))
split_val = int(0.2 * len(X)) + split_train

X_train = X[:split_train]
y_train = y[:split_train]
X_val = X[split_train:split_val]
y_val = y[split_train:split_val]
X_test = X[split_val:]
y_test = y[split_val:]
```

Early stop configuration

```
early_stopping   =   EarlyStopping(monitor='val_loss',   patience=5,
restore_best_weights=True)
```

ARO model

```
def create_model(trial):
    # Create the model
          inputs    =    keras.layers.Input(shape=(X_train.shape[1],
X_train.shape[2]))
    x = keras.layers.LSTM(
        units=trial.suggest_int('units', 50, 200),
        return_sequences=True
    )(inputs)
     x  =  keras.layers.Dropout(trial.suggest_float('dropout_1',  0.2,
0.5))(x)
    x = keras.layers.LSTM(
        units=trial.suggest_int('units', 50, 200),
        return_sequences=True
    )(x)
     x  =  keras.layers.Dropout(trial.suggest_float('dropout_2',  0.2,
0.5))(x)
```

```
      x   =  keras.layers.LSTM(units=trial.suggest_int('units',  50,
200))(x)
    outputs = keras.layers.Dense(1, activation='linear')(x)

    model = keras.Model(inputs=inputs, outputs=outputs)
    model.compile(optimizer='adam', loss="mse")

    return model

def objective(trial):
    # Create the model
    model = create_model(trial)

    # Train the model
     model.fit(X_train,  y_train,  epochs=trial.suggest_int('epochs',
10,   30),   batch_size=trial.suggest_int('batch_size',   8,   32),
verbose=0) # can change the epochs

    # Evaluate the model on the test set
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)

    return mse

# Perform hyperparameter optimization using Optuna
study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=5)  ##can change the trials more
than 5 to be any value

# Get the best hyperparameters
best_params = study.best_params
print("Best Hyperparameters:", best_params)
```

```
# Train the final model with the best hyperparameters
best_model = create_model(study.best_trial)

history = best_model.fit(
    X_train, y_train,
    epochs=best_params['epochs'],
    batch_size=best_params['batch_size'],
    validation_data=(X_val, y_val),  # Use the separate validation set
here
    callbacks=[early_stopping]  # Add the EarlyStopping callback here
```

LSTM Model

```
# Create the model
inputs          =          keras.layers.Input(shape=(X_train.shape[1],
X_train.shape[2])) #this may need to be changed in case of increasing
the predictors
x  =  keras.layers.LSTM(units_value,  return_sequences=  True)(inputs)
#changed to get the unites value from ARO it was 150
x = keras.layers.Dropout(dropout_1_value)(x)
x = keras.layers.LSTM(units_value, return_sequences=True)(x) #changed
to get the unites value from ARO it was 150
x = keras.layers.Dropout(dropout_2_value)(x)
x = keras.layers.LSTM(units_value)(x) #changed to get the unites value
from ARO it was 150
outputs = keras.layers.Dense(1, activation='linear')(x)

model = keras.Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss="mse")
model.summary()
```

```
# Train the model with the EarlyStopping callback
history = model.fit(
    X_train, y_train,
     epochs= epochs_value,   # Increase the number of epochs to give
EarlyStopping more chances to trigger #changes to get the parameter
from ARO
    batch_size= batch_size_value, #changes to get the parameter from
ARO
    validation_data=(X_val, y_val),  # Use the separate validation set
here
    callbacks=[early_stopping]  # Add the EarlyStopping callback here
```

```
# Make prediction
predicted = model.predict(X)
```

```
# Append the predicted values to the list
test_predicted = []

for i in predicted:
  test_predicted.append(i[0])
```