INTERNATIONAL JOURNAL OF FINANCIAL STUDIES, ECONOMICS AND MANAGEMENT, 2023, Vol.2, No 2, 1-16, https://doi.org/10.2022/ijfsem.v2i2.111

Stock Market Prediction Using a Hybrid of Deep Learning Models

Chika YINKA-BANJO
Department of Computer Science, University of Lagos, Lagos, Nigeria

Mary I. AKINYEMI
Department of Statistics, University of Lagos, Lagos, Nigeria

Bouchra ER-RABBANY
Baum Tenpers Research Institute, United States

Abstract: Financial markets play an essential role in developing modern society and enabling the deployment of economic resources. This study focuses on predicting stock prices using deep learning models. In particular, the daily closing prices of two different stocks from the Casablanca Stock Market Viz Bank of Africa and Itissalat Al-Maghrib (IAM) are considered. The datasets were pre-processed and passed through the Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), and Convolutional Neural Networks (CNN) models. The models' performances were compared based on the performance evaluation metrics, viz: mean squared error (MSE) and root mean squared error (RMSE) and Mean Absolute Error (MAE). The paper proposes a novel hybrid model. The hybrid design of the model improves its predictive power as the results of the Hybrid network performance surpassed all the other models.

Keywords: *Deep learning, Stock Market Prediction, Financial Markets, Financial Times Series, Hybrid Models*

Article Info

- *Received December 5, 2022*
- *Available online May 22, 2023*

1. Introduction

Financial markets are crucial in building modern society. To be precise, Stocks, also called shares or fractions of a company, are the most well-known assets since they have huge potential. (Hiransha, Gopalakrishnan, Vijay, & Soman, 2018) Describe two types of stock markets. The primary market, where securities are directly issued by companies to investors through Initial Public Offerings (IPOs), however, for secondary markets investors exchange shares they already own (Liu, Chao, Yu-Chen, & Chih-Min, 2019) mentioned that the Stock market is characterized by its uncertainty, volatility, and randomness. Such characteristics make it a chaotic place with a continuously changing stream of data which makes it quite difficult to predict and act on those predictions for profit motives, especially for time series forecasting.

The stock market price depends on several factors in the current financial world. This includes basic determinants such as supply and demand. Similar to all commodities, an imbalance between supply and demand generates rises and lows in the stock price. (Mishra, 2012) showed two ways of describing the terms and conditions of the exchange, such as a bear market or bull market, that point to the condition of the stock market and reflect the economy's state. A bear market is in decline and is seen as a true bear market when it dropped by 20% compared to recent highs where shares are in continuous decline, this is leading to a downward trend, which investors expect to continue. During a bear market, the economy is slowing down and unemployment augment. In contrast, a bull market can be classified as a sustained increase in the price within the stock market which can be noted in the rising price of a company's shares. Moreover, Investors expect that the upward trend will continue in the future. In a bear market, people want to sell their shares than buy them, the demand is remarkably lower than supply and, consequently, the share price falls. While in a bull market, the reverse is true, the demand is high, but the supply is low for stocks, and few are ready to sell them. Thus, the stock price will increase as investors compete for available stocks.

Given the highly volatile nature of financial markets and their numerous influencing factors, it is no wonder that stock price forecasting has been at the forefront of concern for years as it can lead to significant growth by making various investments. Therefore, both academic researchers and industry practitioners focus on predicting stock market prices, with the main objective which is increasing revenues.

Traditional methods of analysis such as assessments based on the company's dashboard, stock exchange indexes, and newspaper information may not always perform well in the stock market price prediction, due to its complexity, which poses a challenge to investors. To make good decisions in the exchange market, researchers believe that changes in stock markets can be predicted using statistical methods and historical data. Many studies have been conducted to determine the historical financial time series model, and due to the evolution of storage and tracking systems, extensive historical data is available for analytical purposes. As a result, machine learning algorithms, especially Deep Learning Models (DLM) became the main focus for new works, to deal with stock market challenges.

To make good decisions, investors must make a forecast contingent on technical assessments, like the company's dashboard, stock exchange indexes, and information from newspapers. The complexity of the exchange market makes it challenging for investors to analyze and predict the price through traditional methods. Therefore, researchers believe that changes in stock markets can be predicted using statistical methods and historical data. Many studies have been conducted to determine the historical financial time series model. With the advent of more sophisticated storage and tracking systems, extensive historical data is more available for analytical purposes, making machine learning algorithms especially Deep Learning Models (DLM) more popular choices to deal with stock market challenges.

2. Literature Review

In their 2021 paper, Patel *et al.* (2021) introduced and reviewed more possible ways to predict stock movements with high accuracy. They identified that the best method for stock market prediction is to use multiple algorithms for instance Artificial Neural Network (ANN), Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Random Forest. In addition, several features affect stock prices, including news sentiment, currency, commodity prices etc., which needs to be taken into consideration for the AI models. Therefore, it is necessary to combine the results of all algorithms. Hiransha *et al.* (2018) used several Deep Learning (DL) architectures to predict India's National Stock Exchange (NSE) and New York Stock Exchange (NYSE) stock market prices. They developed MLP, RNN, LSTM, and

Convolutional Neural Network (CNN) for TATA MOTORS' NSE stock exchange price. The resulting models served to predict the share prices of MARUTI, HCL and AXIS BANK on the NSE equity market and to forecast the share price of BANK OF AMERICA (BAC), and CHESAPEAKE ENERGY (CHK) from the NYSE. The results showed that the models were able to identify the existing features of both stock markets. In addition, Hiransha *et al*. (2018) shows that there is a common underlying dynamic to the stock markets and that the deep learning models outperformed the ARIMA model. Moreover, CNN performed the best of the three models considered because it was able to capture sudden price movements when a particular window is used to forecast the next instance. However, they did not explore the advantage of combining several networks to get a better result. In order to forecast the stock market through deep learning methods, Liu *et al*. (2019) tackle the issue of noise in share prices, which is difficult because there is a lot of noise and uncertainty in the information associated with exchange prices. Consequently, they use scattered autoencoders with 1-D (one dimension) residual convolutional networks, considered as a deep learning model for clearing data. Long-short-term memory (LSTM) is subsequently utilized for predicting share prices. The results show that it is better to forecast share prices based on the rate of price change than to directly forecast absolute prices. In addition, the method used provides a better forecast than a model that combines wavelet transforms (WT) and stacked autoencoders (SAEs). Hamoudi and Elseifi (2020) Introduced a new method for training a deep neural network for automated training. Instead of trying to predict the exact return at every future time step, the stock market price prediction is considered as a binary classification problem, with the positive class chosen as the trades resulting in returns within the first ten percentile of all returns in the training options. Additionally, the objective is to formulate high-frequency automated algorithmic trading in unidimensional CNN and LSTM models. Furthermore, the feature matrix has been expanded with the addition of a logical array at each time step to preserve the information on missing characteristics. However, a key challenge for machine learning in predicting stock time series is the time component. Although time adds supplementary information, it also makes time series problems harder to deal with than other forecasting tasks. For one, forecasting of share prices by using the single linear time series forecast model or the neural network model has certain limits. At present, the benefits of the combination of multiple methods and the use of various best algorithms to enhance the hybrid method is the trend for developing financial deep learning. To make the most of the features of the time series data, enhance the performance of share price prediction. This paper suggests a method for predicting share prices based on the exploitation of hybrid methods. With regards to variables' parameters and given the escalating competition in stock markets on daily basis, it is necessary to combine the parameters: price, quality, and time for the purpose of raising capital and profit. In this project, we will highlight the parameters price and time excluding quality due to measurement difficulties. The main contribution to the existing literature is comparing three top-notch DL algorithms which are Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN) for capturing the dynamic nature of stock prices within the financial market.

The rest of the paper is arranged as follows : the next Section summarizes past literature on stock market predictions, Section 4 exposes the adopted methodology. Section 5 consists of a discussion of data analysis and deep learning models including MLP, LSTM and CNN, how neural networks learn, and the problems in training a neural network model, Section 6 concludes.

3. Methods Review

3.1. Overview of Deep Learning Algorithms

Deep learning utilizes a neural network (NN) of processing units to extract features from data and make predictions about emerging data. The method allows machines to take accurate decisions. Ivakhnenko *et al*. (1967) issued the initial general work learning algorithm for supervised perceptron, deep, feedforward, and multilayer. Artificial neural networks are capable of managing complex data sets and process them according to the training received. Like biological neurons where the input signals are received and then processed according to a set of calculations and generate output signals. Artificial neural networks work similarly. After the correct training, ANN can self-learn and continuously update to provide increasingly accurate output data.

3.1.1. Artificial Neural Network (ANN)

An ANN is composed of different types of layers, each layer is an ensemble of artificial neurons. The number of artificial neurons in every layer is a hyperparameter. It may vary based on your individual choice. Also, the communication between neurons essentially transits from the first layer through the middle layers to the last layer. These layers are called the input layer, hidden layer, and output layer successively. Figure 1 is a schematic representation of a neural network.

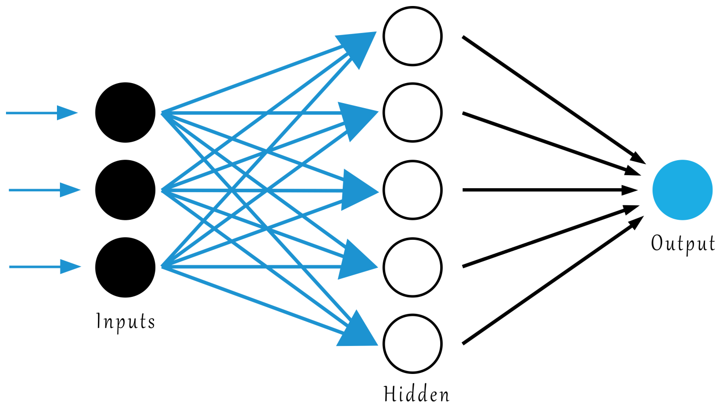

Figure 1: Illustration of ANN Technique

Numerous types of neural networks are mentioned in the literature, dedicated to many applications. The simple neural network has only one layer of unit linked to the input values. The complex network comprises other units between the entrance and the final exit, which are "hidden" layers, as indicated in Figure 2. The feed-forward network has several layers that may connect from one layer to the next in one direction, or there may be feedback connections between superior and inferior levels. Neural networks learn through different routes, including supervised, reinforcement, and unsupervised learning.

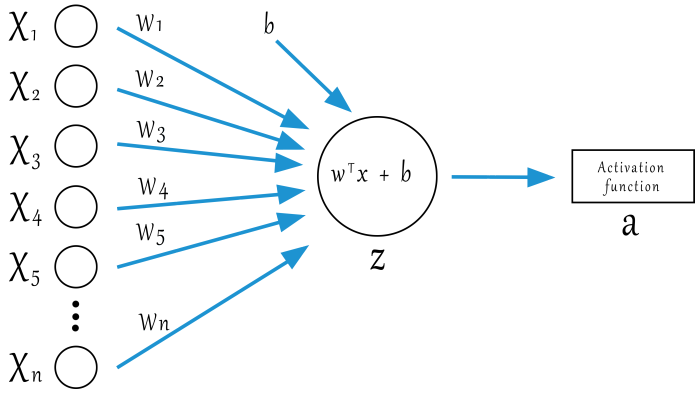

Figure 2: Neuron mathematical representation

Supervised learning is when labeled datasets are used for training algorithms in data classification or to forecast an outcome precisely. The reinforcement learning method is based on reward for desirable behaviors and/or punishment for undesirable behaviors. Also, this method aids to maximize some portion of the cumulative reward. Unsupervised learning is when only input data are given to the network, essentially analyzing and clustering unlabeled data sets, through finding patterns in the data without human intervention (for example, recognizing within an unlabeled data set that there are three types of coins).

In Artificial neural networks, The neuron has 'n' inputs $X(x_1, ..., x_n)$, with each input linked to the neuron through a weighted linkage $W(w_1, ..., w_n)$. Where the neuron adds up the inputs and takes the products by the weights using the following equation:

$$Z = \sum_{i=1}^{n} x_i w_i + b \qquad (4.1.1)$$

The net sum is the output z which is applied to the activation function f(Z). A neuron activation function defines the output of this neuron based on a set of inputs. It is also biologically motivated by the workings of the human brain where different neurons are fired or activated by a different stimulus. For example, when one hears drumbeats, certain neurons in the brain will fire up and become activated such that a human can distinguish the sound from any other sound. The activation function is a major component of an artificial neuron network that helps to decide as to whether to activate a neuron and this is a nonlinear transformation that can be performed on the input before sending it to the next layer of neurons or completing the output.

Activation functions can be classified into two types, linear and non-linear. The linear activation function is an identity function while non-linear activation functions are sigmoid, softmax, rectified linear unit (ReLu), and hyperbolic tangent function (tanh). Sigmoid functions are differentiable, defined for all real input values and have a positive derivative.

At each point and are bounded because the sigmoid function is a logistic function, and the output is ranging between 0 and 1.

$$\sigma(z) = \frac{1}{1+\exp(-z)} \qquad (4.1.2)$$

$$\sigma(z)' = \sigma(z)(1 - \sigma(z)) \tag{4.1.3}$$

The output of the activation function will always be within the range [0,1] compared to the co-domain $(-\infty, \infty)$ of the linear function. It is nonlinear, continuously differentiable, monotonous, and has a fixed output range. But it is not zero-centered. Meanwhile, Rectified Linear (ReLu) function is very broadly used and mathematically expressed as:

$$f(x) = \max(0, x) \tag{4.1.4}$$

Although it does appear to be a linear function, it is not. ReLU has a derivative function and permits back-propagation. There is one problem with ReLU. If we assume that most of the input values are negative or 0, that the ReLU output is 0, and that the neural network cannot carry out backpropagation. This is called the Dying ReLU problem. In addition, ReLU is an open-ended function which means that it does not have a maximum value.

The hyperbolic tangent function (tanh) generates an output within the [-1, 1] range and is a continuous function. This means that the function generates an output for each x-value. If the input x is a negative number, then tanh transforms it into a number much closer to 1. If x is a positive number, then the image by tanh will be a number close to 1. Lastly, if x is a number much closer to 0, tanh transforms it into a number between -1 and 1.

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{4.1.5}$$

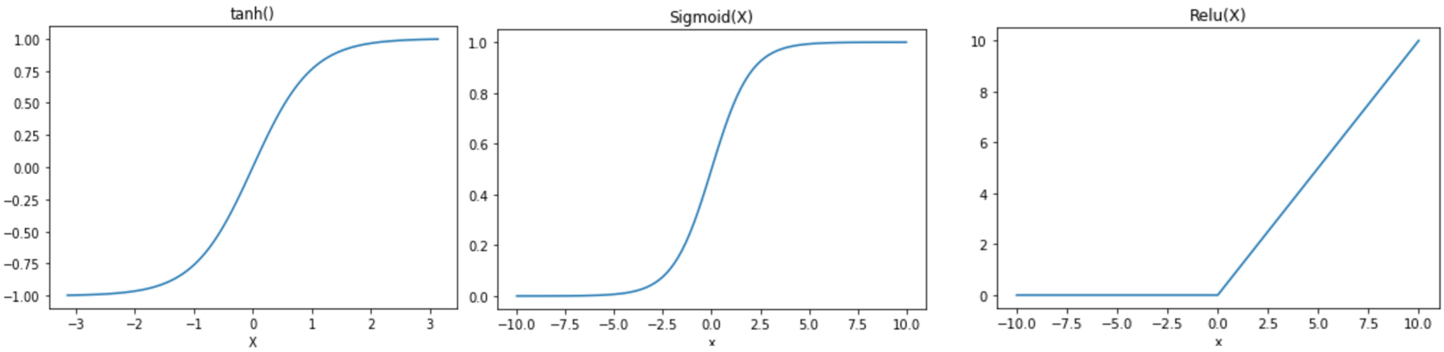

Figure 3: Non-Linear Activation Function

The ANN training process involves solving an optimization problem that minimizes a function called the cost or loss function. The cost function is minimized using an optimization algorithm or simply an optimizer. Its objective is to identify weights and biases which reduce the value of the cost function to as close as zero. As to how the optimizer attains its aim, it employs a method called backward propagation just backpropagation (Nielsen, 2015).

Backpropagation is an algorithm used to compute gradients of the cost function during the tuning of the adjustable parameters of the model. During training, all parameters will be changing continuously to reach optimal values. This is called tuning. Back-propagation is said to be computationally fast and provides an overview of how modifying weights and biases affects the general behavior of the network (Nielsen, 2015). Additionally, with the main goal being the computation of partial derivatives of the cost function, backpropagation adopts two assumptions about the form of the cost function (Nielsen, 2015): The cost function may be written as a cost function average for individual training examples. It can also be written depending on the outputs from the neural network.

3.1.2. LSTM

The RNN is a class of artificial neural networks in which the connectivity across the nodes forms a directed graph rather than a directed graph along a time sequence, it relies on two sources, one from the past to exploit the anterior information and the other from the present, to determine how they respond to the new set of data. The output of each occurrence is an input of the following moment, and this is done with the use of a feedback loop where this process presents a memory for RNN. Each entry sequence contains a lot of information, which is stored in the hidden state of recurring networks. RNNs face the problem of vanishing gradients, which slow down the learning process for large data sequences. The gradient transmits the information used in updating the RNN parameters and when the gradient disappears, the parameter updates become trivial, meaning no actual learning is done (Dupond & Samuel, 2019) .

LSTM is one type of RNN. It was introduced by Hochreiter & Schmidhuber (1997). They have been developed to address the vanishing gradient issue. A joint LSTM unit consists of a cell, an entrance gate, an exit gate and a forget gate. The cell stores values at specific time intervals and the three gates control the flow of information inside and outside of the cell.

3.1.3. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) were introduced first by Hubel (1968) who was inspired by a biological process. Used to address classification and regression issues, CNNs are a regularized version of fully connected networks Multilayer perceptron. CNNs include two main types of processing layers – convolutional layers and pooling or subsampling layers: convolutional layer is the basic component of a CNN, and most of the calculations are done there. It needs some components, which are a filter, input data, and a feature map. Pooling layers help to rationalize the underlying computation and merge the output of groups of neurons from one layer into one neuron from the next layer. Hence, reduce the dimension of the input (data).

3.2. Performance Evaluation

Time series are generally based on the forecast of real values, referred to as regression problems which belong to supervised learning. Validation and evaluation of a data science model give more color to the hypothesis and help evaluate various models that would provide better results against the data. As a result, project performance measures will focus on methodologies for evaluating predictions of real value. The performance of a regression model can be evaluated using different methods by making use of the residual values $(y_i - \hat{y}_i)$. Some of these performance measures are:

3.2.1. Mean Square Error (MSE)

Statistically, MSE is the Mean or Average of the square of the difference between the real values and the estimated values, in other words, is the average of the squares of the residuals. This is like the variance of the residual values with a slight bias. MSE is always positive and formulated mathematically by the following Equation below.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (4.1.2)$$

3.2.2. Root Mean Square Error (RMSE)

RMSE is the standard deviation of Residuals or prediction errors. Residuals are a measurement of the distance between the data points of the regression curve. RMSE is a measurement of how these residuals are distributed, it shows how much data is distributed around the line of best fit. Root Mean Square Error is often used in forecasting, climatology, and regression. This is based on the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.1.2}$$

3.2.3. Mean Absolute Error (MAE)

MAE is the arithmetic average of the magnitude of the residual values. It is easy to interpret and is given by the Equation below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.1.3}$$

4. Methodology

4.1. Preparing data

Stock price forecasting is challenging because of noise and volatile features. Therefore, the extraction of useful information will be needed. Feature Selection is one of the core concepts in machine learning, which hugely impacts the performance of the model. It is necessary to identify the related features from the dataset to remove the less relevant and less important features, which contribute less to our target for achieving better accuracy in the training of the model. Preparing data is a critical step in developing deep learning models. The main data preparation phases of deep learning are presented in Figure 4 below:

Figure 4: Data Preparation Process

Collecting data is the process of gathering data from different sources, the official website of Casablanca Stock market is used for this purpose. Followed by data cleaning step, which involves the identification and processing of missing, inaccurate data. It is important to remove any data errors that may affect the performance of the deep learning models. Data pre-processing consists of transforming the data to fit the deep learning algorithms, which includes feature scaling, normalization, and dealing with outlier. Then, splitting data refers to the process of dividing a dataset into separate subsets, training, testing and validating sets. The aim of this phase is to properly train and evaluate a deep learning model.

4.2. Formulating deep learning model

The Pre-processed and cleaned data is used to create, build, and train various machine learning algorithms which can be used in predictions, including Long Short-Term Memory (LSTM) as an artificial Recurrent Neural Network (RNN) architecture used in deep learning. (LSTMs) are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. Also, Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN) algorithm will be used. Then the results of different algorithms will be compared. To improve the performance of our models, a hybrid approach will be proposed. Furthermore, we focus on the realization of two different approaches. The first approach consists of working directly with deep learning models, more precisely MLP, LSTM, and CNN as the well-performed model seen in the literature review, then giving a brief comparison of these models to come out with the best model. The second approach uses a novel method that aims to combine the results of the three models as input of another model to improve the performance. Approaches overview is given by the following figure:

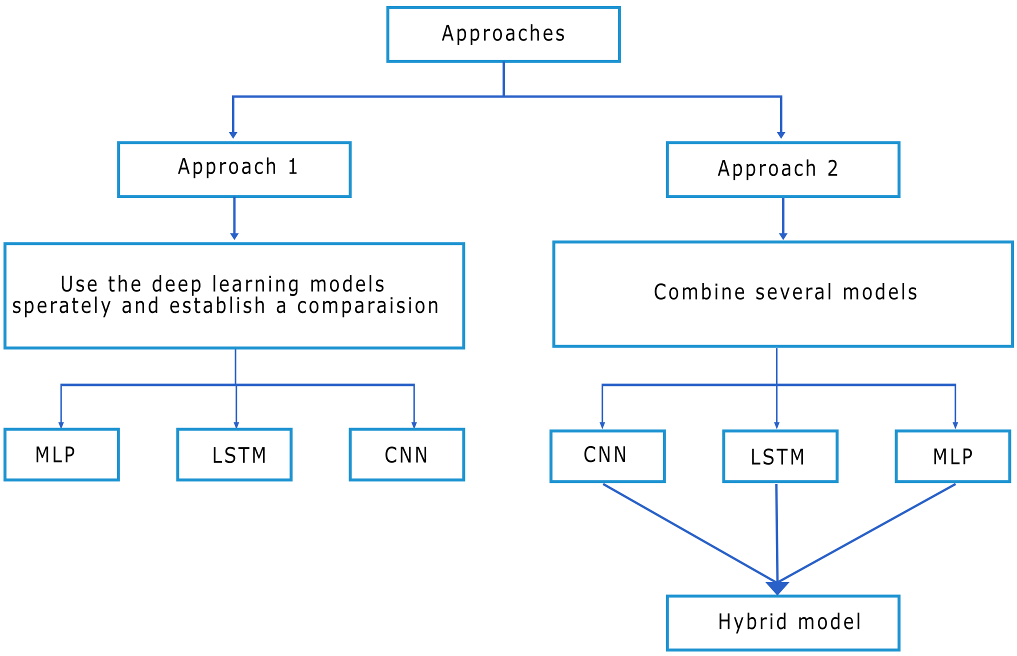

Figure 5: Adopted approaches

4.3. Hybrid Method

The hybrid method consists of combining different results obtained from previous models including LSTM, MLP, and CNN, using them as input data for the new model formulated below:

$$Z = w_1x_1 + w_2x_2 + w_3x_3 + b \qquad (4.3.1)$$

Where z presents vector price, model output that approximates the real vector price. $x_1(x_{11}, x_{12}, ..., x_{1n})$ is the vector output the LSTM model which present the predicted price, $x_2(x_{21}, x_{22}, ..., x_{2n})$ present the vector price output the MLP model and $x_3(x_{31}, x_{32}, ..., x_{3n})$ is the forecasted price using CNN model. Moreover, the vector weight $W(w_1, w_2, w_3)$ initially takes the following values:

$$w_1 = \frac{Total - MSE_1}{Total} \tag{4.3.2}$$

$$w_2 = \frac{Total - MSE_3}{Total} \tag{4.3.3}$$

$$w_3 = \frac{Total - MSE_3}{Total} \tag{4.3.4}$$

Where,

$$\text{Total}=\sum_{i=1}^{3} MSE_i \tag{4.3.5}$$

MSE_1 is the mean square error obtained in the LSTM model, MSE_2 present mean square error of the MLP model, and MSE_3 consists of the mean square error of the CNN model.

Generally, MSE and RMSE are more commonly used as evaluative measures in regression problems and RMSE is beneficial when the data has a significant scale. While MAE is commonly used in problems where the ability to interpret the results and robustness to outliers are critical. The initialization of weights is an essential aspect of the training of neural networks. The weights of a neural network are the parameters which are learned during training. Proper initialisation of these weights can greatly affect model performance. Some advantages of proper weight initialization of a machine learning model include faster convergence, because weights are initialized near their optimum values, which reduces the time required for the model to learn the correct values. Avoiding vanishing or exploding gradients and better performance.

The initialization of coefficients w_1, w_2, w_3 in Equations 3.2.2-3.2,4 is based on MSE metric, because it consists of the error measures of the previous models including MLP, CNN and LSTM. The maximum coefficient of coefficients w_1, w_2, w_3 should be associated to the input x_i where i ∈ 1, 2, 3 that had the minimum MSE, this justifies the presence of minus in the formula −MSE. Also, the Total in Equation 3.2.5 appears in the denominator to avoid the problem of saturation which can lead to the vanishing gradient issue, the coefficients must not be initialized with large values, especially when using sigmoid activation functions such as logistic and tanh. For this reason, it is advisable to standardise coefficients on a small scale.

The main objective of constructing this new model is to obtain a better approximation of the vector price Y, to achieve this goal, the mean square error will be considered as a performance evaluation function. Thus, the below optimization model will be considered:

$$\min_{z} \mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(z_i - y_i)^2 \tag{4.3.6}$$

Gradient descent is a commonly used optimization algorithm to train machine learning models and neural networks to update the parameters for all the layers. Training data helps these models learn over time. It is utilized to iteratively converge towards the global minimum of the cost function. The idea is to make repetitive steps in the opposite direction of the gradient of the function at the actual point because this is the direction of the steepest descent. reciprocally, a step in the direction of the gradient will result in a local maximum of that function (Pattanayak & John, S. , 2017).

Input: w_0 initial parameters of the model, α learning rate

Output: w_f regularized parameters of the model

while index ¡ iteration's number **do**

$w_{t+1} \leftarrow w_t + \alpha \frac{\partial L}{\partial \omega}$;

$b_{t+1} \leftarrow b_t + \alpha \frac{\partial L}{\partial b}$;

end

return *optimal vector weight.*

Algorithm 1: Gradient Descent Algorithm

The gradient Descent Algorithm assists us in making decisions in an effective and efficient manner through the use of derivatives in Equation 3.2.7.

$$w_{t+1} - w_t - \alpha \frac{\partial \mathcal{L}}{\partial \omega_t} \tag{4.3.7}$$

Where α is the size of steps taken to reach the minimum called Learning Rate. More areas can be covered with larger steps/higher learning rates but there is a risk of exceeding minima. In addition, smaller steps/learning rates will take a long time to reach the lowest point. Gradient descent as an optimization algorithm uses derivates to decide if weight should be increased or decreased to increase or decrease an objective function, because the derivative of a function determines which direction to proceed to minimize it.

$$\frac{\partial \mathcal{L}}{\partial \omega_1} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \omega_1} = \frac{2}{n}x_1\sum_{i=1}^{n}(z_i - y_i) \tag{4.3.8}$$

$$\frac{\partial \mathcal{L}}{\partial \omega_2} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \omega_2} = \frac{2}{n} x_2 \sum_{i=1}^{n}(z_i - y_i) \tag{4.3.9}$$

$$\frac{\partial \mathcal{L}}{\partial \omega_3} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \omega_3} = \frac{2}{n} x_3 \sum_{i=1}^{n}(z_i - y_i) \tag{4.3.10}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial b} = \frac{2}{n} \sum_{i=1}^{n}(z_i - y_i) \tag{4.3.11}$$

5. Results

In this paper two datasets from different sectors were used; the first dataset refers to the historical data of Itissalat Al-Maghrib (IAM) stock from investing source, it covers 11 years and 6 months including 3000 trading days. Six features were extracted including Open, High, Low, Volume, and Change % while the target is close price. The second dataset is that of the Bank of Africa from the Casablanca stock market, it covers the period from 15 August 2019 to 11 August 2022 with 748 instances.

Table 1 presents descriptive statistics of the variables. Mean of close price was about. 130.72, was observed while the average adjusted close price was 170.80 for the Bank of Africa dataset. The distributions of some variables were platykurtic (negative kurtosis) i.e., it has a flatter peak and thinner tails when compared to a normal distribution. The probability values of Jarque-Bera show that the price does not follow a normal distribution. The adjusted close price of the second dataset is however normally distributed.

Variables	IAM	Bank of Africa
Count	3000.00	748.00
Mean	130.72	170.80
Std	17.42	23.69
Min	86.91	121.55
25%	116.00	152.58
50%	137.00	175.00
75%	143.05	192.00
Max	159.90	223.00
Skewness	-0.63	-0.27
Kurtosis	-0.72	-1.09
Jarque-Bera	263.58	46.39
Probability	0.00	8.43

Table 1: Summary Statistics

Table 2 presents the Augmented Dickey-Fuller (ADF) test for stationarity results. The results of the ADF test indicates that p-value for both datasets is greater than 0.05 which points at non-stationarity in both series. In addition, the KPSS test results shows that p-value for both datasets are the same 0.010 which is less than 0.05. Also, KPSS test statistics are more extreme than the critical value for both datasets, further confirming the non-Stationarity of both datasets.

IAM historical data					
ADF test			KPSS test		
ADF statistic	P-value	Critical value 5%	KPSS statistic	P-value	Critical value5%
-1.81	0.38	-2.86	1.17	0.01	0.15
Bank of Africa					
ADF test			KPSS test		
ADF statistic	P-value	Critical value 5%	KPSS statistic	P-value	Critical value5%
-1.34	0.61	-2.87	0.65	0.01	0.15

Table 2: Stationarity tests

The autocorrelation and partial autocorrelation plots for both data-sets is presented in Figure 6 and 7. Notice in Figure 6 how the coefficient is high at the first 500 lag. From the partial autocorrelation, it can be observed that for a 0.05 level of significance there is some partial autocorrelation at different lags, with lag 0 and 1 recording, very high partial autocorrelation.

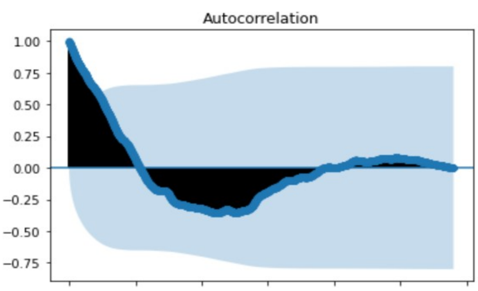

Figure 6: Autocorrelation for IAM data

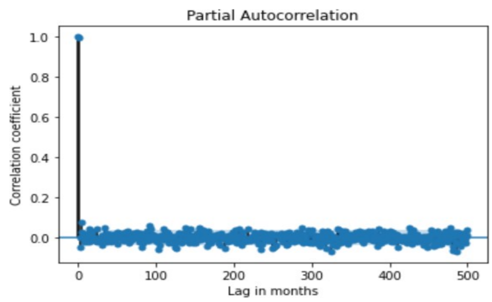

Figure 7: Partial Autocorrelation for IAM data

The Breusch-Pagan test for homoscedasticity is reported in Table 3. Here, for IAM historical data the Lagrange multiplier statistic for the test comes out to be equal to 622.335 with a corresponding p-value of 3.016e-132. The Bank of Africa the Lagrange multiplier statistic value for the test is equal to 648.795 with a corresponding p-value of 2.657e-140. Since the p¡0.05 for both datasets, we cannot accept the null hypothesis of homoscedasticity, confirming the presence of heteroskedasticity.

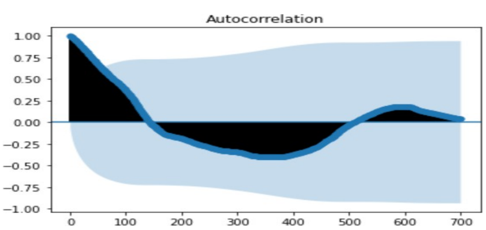

Figure 8: Autocorrelation for Bank of Africa

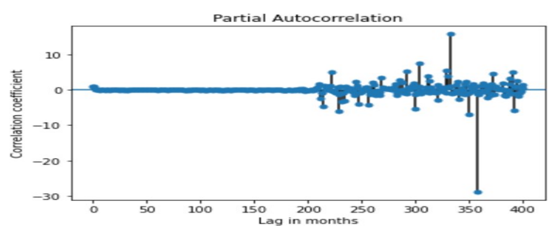

Figure 9: Partial Autocorrelation for Bank of Africa

The IAM historical data split for the neural networks is as follows. 70% of the dataset was used for training (50%) and validation (20%) of the models, the remaining 30% was used for evaluating the models as the testing dataset. The validation part was added to allow us to check for model performance, more precisely overfitting of the models. While the size of Bank of Africa is not enough to be split into three parts, therefore, the splitting outputs training and testing.

Data	Modelp-value Lagrange multiplier statistic(p-value)	f-Value(p-value)
IAM data	622.34(3.02e-132)	156.73(2.91e-148)
Bank of Africa	648.80 (2.66e-140)	1621.92(0.00)

Table 3: Breusch-Pagan test results

Dataset values may be scaled to lie within a certain range before inputting them into the models. Scaling data may be realized through standardizing the values such that mean=0 and a standard deviation=1. This method improves the speed of the model train and diminishes the likelihood of getting caught in a local minimum cost function, the reason why this method was used in this project.

In the design part of a deep learning model, it is recommended to begin with the simplest model and then build it into more complex models, the model's performance depends on the hyper-parameters used. For the MLP experiments the hyper-parameters were tuned manually and in a systematic way.

Therefore, the current architecture is built from the previous models by changing the hyperparameters based on the performance of the previous. Also, make sure not to add too many hyperparameters in order not to overfit the model and use a simple architecture as much as possible. Hence, the MLP model is constructed with one hidden layer using Relu, and Linear activation functions with 30 epochs gave the best model efficiently. For CNN model it is designed by using several layers including 1D-convolution layer MaxPooling-1D, it utilizes Relu for the activation function. Regarding the LSTM model, linear and tanh activation functions were used 30 epochs were used to train the model and model performance was evaluated using Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). Each LSTM layer is accompanied by a dropout layer. This layer helps prevent overflowing into the training by passing randomly selected neurons thus reducing sensitivity to specific weights of the individual neurons.

IAM				
Models	LSTM	MLP	CNN	Hybrid
MSE	5.2910^{-4}	8.0710^{-5}	6.1210^{-5}	2.8110^{-5}
RMSE	2.3010^{-2}	8.9810^{-3}	7.8210^{-3}	5.3010^{-3}
MAE	2.3010^{-2}	8.9810^{-3}	7.8210^{-3}	5.3010^{-3}
Bank of Africa				
Models	LSTM	MLP	CNN	Hybrid
MSE	1.7110^{-2}	5.5810^{-7}	1.7010^{-7}	1.3910^{-7}
RMSE	1.3110^{-1}	7.4710^{-4}	4.1310^{-4}	3.7310^{-4}
MAE	1.3110^{-1}	7.4710^{-4}	4.1310^{-4}	3.7310^{-4}

Table 4: Performance Evaluation

By analyzing the results obtained from the above table, the coming principal conclusions can be drawn comparing the LSTM and the MLP models for both datasets and by considering the different metrics used including MSE, RMSE, and MAE, the MLP's errors are less than the errors for the LSTM in the sense that MLP performs better than LSTM, while CNN works slightly better than MLP. The hybrid model achieves the best performance of the three previous models since it has the minimum errors for all metrics considered,

6. Conclusion

With the aim of predicting stock market prices, this paper examined three deep learning models viz: LSTM, MLP, and CNN which are considered among the well-performed models in literature. After the design and implementation using Python programming, the results show that CNN performs better than other models based on the mean square error metric nevertheless, the experimental studies demonstrated that the results can be improved by exploring the advantages of combining those results in a hybrid model which is designed using a novel approach in this paper. In future works, we can focus on other features that can impact the stock market price including financial news and sentimental analysis.

7. References

- Dupond, & Samuel. (2019). *A thorough review on the current advance of neural network structures*.
- Hamoudi, & Elseifi. (2020). Stock Market Prediction using CNN and LSTM. *CS230 , 8*.
- Hiransha, M., Gopalakrishnan, E., Vijay, K. M., & Soman, K. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science , 132*, 1351–1362.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
- Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, *195*(1), 215-243.

- Ivakhnenko, A. G., Lapa, V. G., & Lapa, V. G. (1967). *Cybernetics and forecasting techniques* (Vol. 8). American Elsevier Publishing Company.
- Lachiheb, O., & Gouider, M. S. (2018). A hierarchical deep neural network design for stock returns prediction. *Procedia Computer Science*, *126*, 264-272.
- Liu, J., Chao, F., Lin, Y. C., & Lin, C. M. (2019). Stock prices prediction using deep learning models. *arXiv preprint arXiv:1909.12227*.
- Mishra, D., & Kanti, S. (2012). Stock Market: Race between Bull vs. Bear. *Stock Market: Race between Bull vs. Bear (June 14, 2012)*.
- Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25, pp. 15-24). San Francisco, CA, USA: Determination press.
- Patel, A., Patel, D., & Yadav, S. (2021, May). Prediction of stock market using artificial intelligence. In *Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021)*.
- Pattanayak , & John, S. . (2017). Deep Learning with TensorFlow . *Pro , 10*, 1730-1760.