EISSN: 2707-0425

East African Journal of Science, Technology and Innovation, Vol. 6 (Special issue 1): December 2024

Stock price prediction: a comparative analysis of classical and quantum neural networks

[1]NGUMA B., [1]KANAKA P V R., [1]PAMAIN A*

[1]Department of Physics, The University of Dodoma, Dodoma, Tanzania

***Corresponding Author**: tumainibrian8@gmail.com

Abstract

Stock price prediction is one of the fast-growing fields, in developing countries like Tanzania it has a significant impact on investments in the public sector, private, and individual investors. The traditional prediction methods under statistical and econometric techniques cannot handle non-stationary financial time series stock data and produce accurate results. Hence, machine learning techniques are used for stock price prediction for this complex problem relying on past stock prices. This study utilizes a Long Short Term Memory neural network designed with a dropout layer, which reduces data's overfitting and diffusion gradient by Rectifier Linear Unit activation. Further, a Multiple Linear Regression network with more hidden layers for additional tuning of weights for better prediction and a Variational Quantum Neural Network utilizing PennyLane and Quantum Dense libraries to run quantum circuits for prediction are considered. The results of performance predictors like the coefficient of determination, the mean square error, the mean absolute error, and the mean absolute percentage error are compared to identify the best neural network for low stock price prediction on the National Microfinance Bank and the Cooperation Rural Development Bank data of Tanzania.

Keywords: *Classical neural network; Financial time series data; Machine learning; Stock price prediction; Variational quantum neural network*

Received: 18/06/24
Accepted: 05/12/24
Published: 20/12/24

Cite as, *Nguma et al., (2024)*. Stock price prediction: a comparative analysis of classical and quantum neural networks. *East African Journal of Science, Technology and Innovation 6 (Special issue 1).*

Introduction

A neural network (NN) is a machine learning (ML) approach that works similarly to the human brain in decision-making by employing copying in the action process, the way biological neurons cooperate to recognize phenomena, weigh options, and come to conclusions. Neurons represent a non-linear function by cooperating with one or more inputs to a single real number, NN assists in decision-making by learning faster the relationships between input and output data that are non-linear and complex, (Oh *et al.*, 2020). NN can be used for medical image classification, chemical compound identification, and financial predictions by processing historical data of financial instruments.

A stock market is a public market in which shares are bought and sold by publicly listed companies. Ownership of stock means you are part of the asset and profit of the company in terms of the number of shares held, (Quadir *et al.*, 2023). Stock price prediction is more significant to capital gains and cash flows to investors which signifies the economic health of the country. Stock price prediction is a difficult task for maximum profit and minimum risk for investments since it is featured by dynamic, volatility, and geo-political war which makes the prediction process complex (Ribeiro *et al.*, 2021; Rai and Grag, 2021; Doong and Doan, 2021, Avramov *et al.*, 2021; Singh and Khushi; 2021).

The advancement of artificial intelligence (AI) has led to the application of ML and deep learning (DL) methods to predict stock prices, they employ historical data and different algorithms to identify patterns rather than relying on traditional statistical methods (Ezugwu *et al.*, 2022). Classical NNs (CNNs) are those that incorporate neurons with input layers, one or more hidden layers, and output layers. The neurons are connected with an associated weight, and threshold value, by backward, and forward propagation of input patterns that can be recognized (Neilsen, 2015). The CNN models considered in this study are long short-term memory (LSTM), and multiple layer regression (MLR). The LSTM is the recurrent neural network (RNN) with the importance of pattern recognition of stock data with long-term dependencies and solves the diffusion gradient problem in conventional RNNs by incorporating a better learning mechanism from sequential stock data.

Alkhatib *et al.* (2022), employed an LSTM model to predict the stock price of Tesla stock data and assessed the mean absolute percentage error (MAPE) values for different numbers of inputs. The MAPE values obtained are 1.89 and 2.533 for four and six inputs respectively. Further, the multiple layer sequential long short-term memory (MLS-LSTM) methods were used to predict Samsung's low stock price data the results of the coefficient of determination (R^2) is 98.1% and with a MAPE value of 2.18, the presence of more hidden layers in the MLS-LSTM was advantageous as compared to the previous research, (Quadir *et al.*, 2023). This makes the LSTM network applicable to financial time series prediction due to its powerful memory capacity.

The MLR model utilizes the linear relationship between independent and dependent variables, the MLR model was used in predicting house prices in Boston to assess the accuracy of the Spearman correlation coefficient which increases the constructive robustness of the model (Zhang, 2021). Further, the R^2 value of the close stock price of NVDA, AMD, and INTC was evaluated at 0.5955, 0.7520, and 0.8369 respectively under the MLR technique which shows more moderate values (Wang *et al.*, 2023). The MLR technique has shown more simple relations between variables, flexibility, adaptability, and minimizing the cost of computation.

Quantum machine learning (QML) is a field that uses the computational advantages of quantum systems to utilize neural networks trained and operated. QNN network known as quantum circuit learning is built through ML algorithm for quantum computers. The variational quantum algorithm relies on a quantum-classical hybrid algorithm that operates with noisy intermediate-scale quantum (NISQ) devices (Preskill, 2018). Quantum Neural network (QNN) uses high dimensional quantum states as trial functions which are difficult to generate on classical computers and utilizes unitarity of quantum circuits (QC) which reduces overfitting by regularization of the stock dataset (Mitarai *et al.*, 2018).

A QNN network is used to minimize the differences between the output of the QC and actual stock price by tunning the QC parameters to their maximum values Quantum operations are encoded and processed data into the quantum neural networks (QNNs), by employing quantum superposition, and entanglement which accelerate the learning process (Ciliberto *et al.*, 2018). QNN network has supervised more interest due to resistance to overfitting by utilizing a small number of training datasets for stock price prediction. Apart, from QNN networks, quantum kernel techniques use quantum operations to generate kernels from the inner products of quantum states, and classical data are transposed into the quantum domain. Quantum computing involves qubits which are the basic units of quantum data that act as a computer memory by taking quantum states of 0, 1, or both. This is mostly faster and higher powered than classical computers.

Quantum mechanics was used in the Association of Southeast Asian Nations (ASEAN) stock exchange for value at risk (VaR), and expected shortfall (ES) predictions by using Bayesian inference led to wave function development for risk management, (Chaiboonsari and Satawat, 2021). Quantum computers were used to predict option prices under gated- based on the utilization of amplitude estimation (AE) algorithm which is faster than classical Monte Carlo methods. QC was built for input states, and operators for AE for different option prices (vanilla, multi-asset, and path-dependent). IBM Q Tokyo quantum device was used to assess QC for option pricing, and error mitigation for error reduction in noisy two-qubit gates, (Stamatopoulos *et al.,* 2020).

Despite many developments in predictive neural networks, especially when stock data is considered the inclusion of dropout in the LSTM neural network avoids overfitting and decreases sensitivity hence in the present study these factors are considered. Further, the inclusion of more than four inputs in the MLR neural network provides flexibility and adaptability of data, hence in the study this factor is considered. Comparison of the CNN, and QNN models for stock price prediction, helps in understanding the performance capabilities and robustness in the stock prediction in developing markets like Tanzania. Hence, in the present study, the LSTM, MLR, and QNN model's efficacy in the stock prediction of, Cooperative Rural Development Bank (CRDB) and National Microfinance Bank (NMB) data were considered. To assess the predictive capabilities, performance metrics such as R^2, mean absolute percentage error (MAPE), mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE) were used.

Materials and Methods

Stock market data of CRDB and NMB were obtained from the Dar Es Salaam Stock Exchange (DSE) with 2525 stock data points for each company with 80% for training, and 20% for testing. Errors in the data, where trading from one of the stocks was not available on a particular day were cleaned for consistency in the data. Further, the data was used to evaluate the performance predictors using LSTM, MLR, and VQNN.

Long Short-Term Memory (LSTM) network

LSTM is a type of RNN and is mostly powerful in maintaining the memory of the input, which makes it suitable for application in the financial time series stock market (Jia, 2016). The LSTM involves two steps which are data preparation and dropout as shown in Figure 1**.** The stock data set takes into consideration stock price indexes such as volume, turnover, deals, close, high, and low prices. The scaling feature is used to normalize all the stock data as defined in equation 1 which is given at different ranges of conversion from 0 to 1. Unnormalized stock datasets may cause massive gradient error values and result in massive and unstable values for the LSTM model. Normalization of the stock dataset is done by MinMaxScaler which is present in the sklearn preprocessing library of Python. This helps to keep the stock dataset in columns within the given ranges of 0 and 1 and preserves the shape of the stock dataset without any distortion with 100 as a batch size, (Quadir *et al.,* 2023). The ratio for training and testing is set to 80:20, (Al Bashabsheh and Alsal, 2021).

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where, X^* represents the predicted stock price, X represents the recorded stock price. min represents the minimum stock price, max represents the maximum stock price.

Activation functions are an output of the networks whereby the strength is determined between the connection of weights and neurons. Rectifier Linear Unit (ReLU) was applied as an activation function, when input values are less than or equal to zero, the function gives a zero output, while the input values are greater than zero, the output values will be the same as a linear function as represented in equation 2 since it simplifies backpropagation that makes learning faster, improves efficiency, and makes features sparse, (Houssein *et al.,* 2021).

$$f(x) = \begin{Bmatrix} x, x > 0 \\ 0, \ x \leq 0 \end{Bmatrix} = Max(0; x) \quad (2)$$

The LSTM network involves a sequential way of processing by incorporating LSTM layers in the network. The LSTM incorporates eight layers that are the maximum number of layers for higher accuracy predictions by avoiding overfitting and employing dropout which reduces the sensitivity of the model. The first layer of the network is taken from the Keras library. The input layer shape is (105, 1) where the time step used is 105, it specifies an input size of 105, and one feature column is used as input. The first layer's sequential layer, is the LSTM layer (50 units), with a dropout layer (0.2), and the same is continued for another layer. The last layer is a dense layer that acts as output by fully connecting the previous layer. Adam optimizer utilizes a large capacity of financial time series stock dataset and a different number of parameters.

The LSTM network employs a batch size of 100 with an epoch size of 105 by neutralizing stock data together into 100 batch sizes in the LSTM network. The 105 epoch means that stock data is being applied to the network 105 times, to accomplish training and testing for the stock data set. The inverse transform function is being used to normalize low predicted stock price back into normal scale, this facilitates comparison between recorded and predicted low stock price for accuracy measurement by performance indices.

The dropout technique is a method of avoiding overfitting by short-term removal of the unit structure of neural networks from a framework at a certain probability. The dropout technique increases the efficiency of supervised learning activities such as stock predictions, speech recognition, classification, and computational biology, (Srivastava *et al.,* 2014). The LSTM networks employ a large number of parameters as shown in Table 1 that are more useful to decrease overfitting and increase performances over regularization methods.

Multiple Layer Regression (MLR) network

The MLR network falls under the category of time series regression which involves the prediction of future stock prices which are time-dependent stock data relying on past information to make stock price predictions. It involves one dependent variable and several different independent variables as shown in Equation 3, (Makridakis *et al.,* 2008). The conditions for the MLR model are a linear relationship between independent and dependent variables, normal distribution of residuals, multicollinearity should be avoided, and a constant variance of errors must be maintained, (Deb, 2020).

$$y = X_1\beta_1 + X_2\beta_2 + X_n\beta_n + \mathcal{E} \quad (3)$$

where, y represent predicted stock price, and x, ε, and β are the augmented vectors.

The MLR involves three parts of the neuron layer which are the input neuron layer (Input layers 1 through 4, Input layers 2 through 3, and Input layer 4), the hidden neuron layer (Hidden layer-1, Hidden layer-2, Hidden layer-3, and Hidden layer-4), and output neuron layer with a greater number of hidden neuron layer. Both neurons are connected to the remaining last layer in the network which is known as the dense layer by which neurons from the same layer are not connected. The weights keep changing in a neuron after error evaluation in the stock data when comparing the predicted low stock price with the expected low stock price by applying the learning rate (0.001). The learning in an MLR network is visualized in Figure 2 when each input (X) to neuron is multiplied by a weight (w) and then added to bias (b) in equation 4, (Jia, 2016).

$$m = b + \sum_{i=1}^{n} X_i W_i \quad (4)$$

This sum is then processed by the activation function $f(m)$ which turns the signal on or off in a nonlinear way as an output in equation 5**.** Bias is a measurement of how simple it is to cause perceptron to produce a 1, (Mari *et al.,* 2020).

$$Output = f(m) \quad (5)$$

The MLR network utilizes four stock datasets with three layers which are the input (sequential) layer, the hidden (100 dense neurons) layer as shown in Table 1, and the output (single dense neuron) layer. It employs the ReLU activation function in equation 2 with auto batch size, random state (1), and maximum iteration (2000), the ratio for training and testing is 80:20. The evaluation of the MLR model can be done by using $sklearn.\ metric$.

Variational Quantum Neural Network (VQNN)

A variational quantum circuit (VQC) are quantum algorithm that relies on free parameters (rotational angles) such as basic QC with three parts which are the preparation of a fixed initial state like zero or vacuum state, x is the classical data points into an n quantum bit in a quantum state |Ψ> as given by equation 6.

$$|\Psi(x)> = F(x)|0>^{\otimes} n \quad (6)$$

A quantum circuit (ansatz) generates qubit states with entanglement and rotation gates, with variational with free parameters sets of ansatzes (θ), as shown by equation 7. (Sebastianelli *et al.,* 2021)

$$|\phi(x, 0> = A(\theta) \ \ |\Psi(x)> \quad (7)$$

Observable measurement $\hat{\beta}$ and the eigenvalue appears similar to the recorded resultant quantum state. The repetition run of VQC with an input x and free parameters vector θ produces an expectation value (f) as given in equation 8**,** (Killoran *et al.,* 2019)

$$f(x, \theta) =< 0\left|U^{+}(x, \theta)\widehat{\beta}\, U(x, \theta)\right|0> \quad (8)$$

The expectation value is used as an output in ML after the employment of VQC. The classical optimization algorithm is used to train VQC to understand the uncertainty of a quantum device. This guides the algorithm to come up with the best parameters for each step taken.

VQNN model executes data, firstly the input parameters are encoded into the required qubits state with exact qubit numbers, then measures the qubits. Qubit is added to the PQC of Hadmard gate and CNOT gate. The first step is through initialization of the states in the VQNN algorithm in which Hadmard gate applied qubits into a superposition state by accepting qubits as input of the model. The stock data are added to the kernel for the VQNN algorithm with the stock data as shown in equation 9.

$$D = \{(X_i, \ Y_i)\} \quad (9)$$

Whereby,

$D - represent\ a\ stock\ data,$

$X_i - represent\ the\ input\ vector.$

$Y_i - represent\ the\ output\ vector.$

The classical bit is changed into qubits by employing PQC for the performance of complex computation of Hilbert space in high dimensional. Superposition and entanglement are obtained by manipulation of qubit states, considering inner product over superposition states. Figure 3 shows the architecture of a Quantum neural network (QNN). The VQNN model through the measurement layer produces an output that is classical output by applying distribution function F(D) and is updated by classical post-processing in equation 10.

$$|K_* > = \frac{1}{\sqrt{F(D)}} \sum_{i=1}^{n} K(X_* \ \ .X_i)\ |i> \quad (10)$$

Whereby,

$K - represent\ Kernel\ function,$

$n - represent\ number\ of\ stock\ data.$

$|K_* > - represent\ the\ classical\ output.$

The VQNN algorithm employs a learning rate (0.001) with an epoch (1) batch size (1) and random state as shown in Table 1 with an Adam optimizer that can use different numbers of hyperparameters to assess the efficiency of adaptable weight applied in the model.

Performance metrics are employed to assess the effectiveness of the networks designed.

MSE is a measurement of the average squared difference between the actual (recorded) and predicted stock prices within a dataset as indicated in Equation 11. It is applicable in ML, and regression evaluation; it is useful when is smaller which shows that stock data points are aligned around its mean central.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_{actual}^{i} - Y_{predict}^{i}\right)^2 \qquad (11)$$

where,

n represent number of stock data points. $y_{predict}^{(i)}$ represent the prediction value of the stock. $Y_{actual}^{(i)}$ represents the actual value of the stock.

MAE is the absolute difference between an expected value and a forecast result (Moon and Yao,2011). MAE is given by equation 12.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{actual}^{(i)} - y_{predict}^{(i)}\right| \qquad (12)$$

MAPE is the indicator that gives the result of how accurate a prediction network is and the amount of prediction error about the actual value (De Myttenaere *et al.,* 2016). MAPE is given by equation 13.

$$MAPE = \frac{\sum \frac{y_{actual}^{(i)} - y_{predict}^{(i)}}{y_{actual}^{(i)}}}{n} \times 100\% \qquad (13)$$

R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variable. The R^2 value is given by equation 14.

$$R^2 = \frac{\left(1 - \sum_{i=1}^{n} y_{actual}^{i} - y_{predict}^{i}\right)^2}{\left(\sum_{i=1}^{n} \hat{Y} - y_{predict}^{i}\right)^2} \qquad (14)$$

where,

$\hat{Y}$ represent mean square of the stocks value.

RMSE is the value between 0 and 1 with 0 being the most favorable value as shown in equation 15.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{actual}^{(i)} - y_{predict}^{(i)}\right)^2} \qquad (15)$$

Results

The study involves three DL networks LSTM, MLR, and VQNN to predict the low stock price of (CRDB, and NMB). Python software (3.11.0) was used in this study under Jupiter Notebook server (anaconda3) with different libraries such as Keras, PennyLane, pandas, numpy, matplotlib, sklearn, and quantum dense.

The RMSE values for different epochs for selected neural networks for CRDB and NMB stocks were calculated and indicated in Table 2. It is observed that the predictions in the VQNN network outperformed other networks by indicating major improvement between actual and predicted low stock prices due to strong resistance to overfitting.

The MAE values for different epochs were calculated and indicated in Table 3. It can be observed from Table 3, that the VQNN network had the lowest value of MAE values which are due to less overfitting of stock data which makes the network applicable to small training datasets. However, the MAE has no regular trend with increasing epoch value in VQNN.

Table 4 shows MAPE values for selected neural networks, it is observed that MLR and LSTM models outperformed the VQNN model due to the presence of more hidden layers in the MLR model, and a dropout layer in the LSTM layer which avoids overfitting of the stock data during the prediction of the low stock price.

From, Table 5 it is observed that the prediction of the stock price was more accurate for the LSTM and MLR networks compared to the VQNN network due to the addition of more layers in the networks which is evident from the higher values of correlation coefficients.

Figure 4, illustrates the actual and predicted CRDB low stock price using the MLR, LSTM, and VQNN network from the 3rd of March 2021 to the 2nd of September 2023. An overlap between the recorded and predicted low stock price indicated that the selected neural networks could forecast the trend in the stock price. However, the trend in MLR outperformed the other neural networks which is evident from the overlapping of actual stock prices as shown.

Figure 5, demonstrates the actual and predicted NMB low stock price using the MLR, LSTM, and VQNN network from the 1st of October 2020 to the 1st of April 2023. An overlap between the recorded and predicted low stock price indicated that the selected neural networks could forecast the trend in the stock price. However, the trend in MLR outperformed the other neural networks which is evident from the overlapping of actual stock prices as shown. The lower prediction values of VQNN are attributed to the large volatility in the NMB data sets as compared to CRDB data sets.

Table 1

RMSE comparison for the neural networks

Epoch	CRDB			NMB		
	LSTM	MLR	VQNN	LSTM	MLR	VQNN
15	21.3099	6.2204	0.1793	136.1077	53.7286	0.1231
30	22.0850	6.3890	0.6522	118.0746	52.1690	0.1228
45	18.8869	6.3147	0.2305	107.9708	52.5774	0.5901
60	20.2611	6.1933	0.1768	103.2190	53.2441	0.6023
75	18.0351	6.1793	0.1784	92.7938	54.9446	0.2347
90	14.4779	6.2683	0.2187	111.0922	53.3167	0.3629
105	14.5792	7.9462	0.1742	92.2959	52.6985	0.1694

Table 2

MAPE values for neural network

Epoch	CRDB			NMB		
	LSTM	MLR	VQNN	LSTM	MLR	VQNN
15	0.0484	0.0133	0.1574	0.04361	0.01532	0.2147
30	0.0456	0.0128	0.6291	0.03719	0.01484	0.2014
45	0.0390	0.0132	0.1947	0.02939	0.01478	0.1752
60	0.0435	0.0136	0.1538	0.02851	0.01547	0.3577
75	0.0383	0.0129	0.1502	0.02448	0.01563	0.6838
90	0.0298	0.0130	0.1885	0.03131	0.01516	0.4454
105	0.0300	0.0136	0.1463	0.02583	0.01494	0.3545

Table 3

R^2 comparisons for the neural network

Epoch	CRDB			NMB		
	LSTM	MLR	VQNN	LSTM	MLR	VQNN
15	0.9292	0.9939	0.9477	0.9260	0.9884	0.9127
30	0.9239	0.9836	0.9901	0.9443	0.9891	0.9834
45	0.9441	0.9837	0.8232	0.9534	0.9889	0.9127
60	0.9360	0.9940	0.8560	0.9574	0.9886	0.9945
75	0.9443	0.9940	0.9893	0.9656	0.9879	0.9200
90	0.9673	0.9938	0.8801	0.9507	0.9886	0.9834
105	0.9668	0.9900	0.8406	0.9660	0.9889	0.9875

Table 4

RMSE value comparisons with different DL models

Epochs	Hybrid model Srivinay et al. (2022)	DNN	LSTM (Present study)	MLR	VQNN
15	6.50	11.50	21.30	6.22	0.17
30	5.60	13.60	22.08	6.38	0.65
45	7.10	12.60	18.88	6.31	0.23
60	6.30	11.20	20.26	6.19	0.17
75	7.20	13.11	18.03	6.17	0.17
90	6.75	12.40	14.47	6.26	0.12

Table 5

R^2 values comparisons with different DL models

Epoch	MLS-LSTM Quadir et al. (2023)	LSTM	MLR (Present study)	VQNN
15	0.38	0.90	0.99	0.92
30	0.77	0.94	0.98	0.94
45	0.84	0.95	0.98	0.95
60	0.93	0.96	0.99	0.95
75	0.98	0.95	0.99	0.96

Figure 1

Neural Network architecture of the LSTM model with a dropout layer

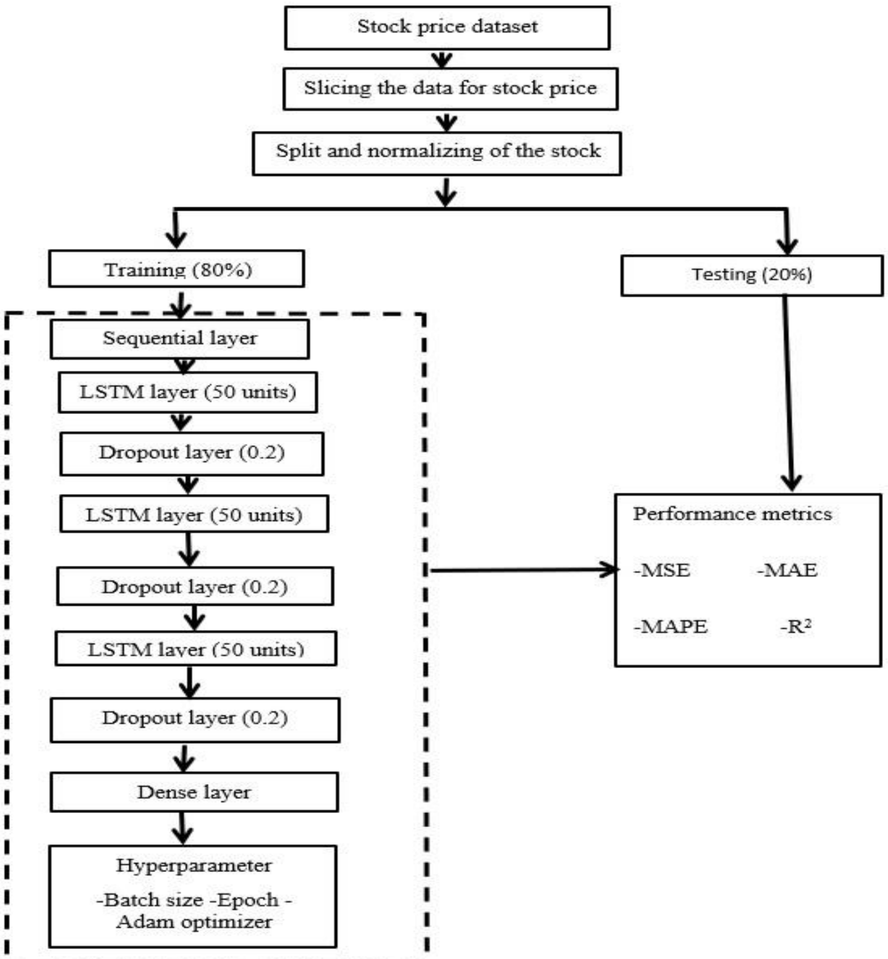


Figure 2

Neural Network architecture of the MLR model

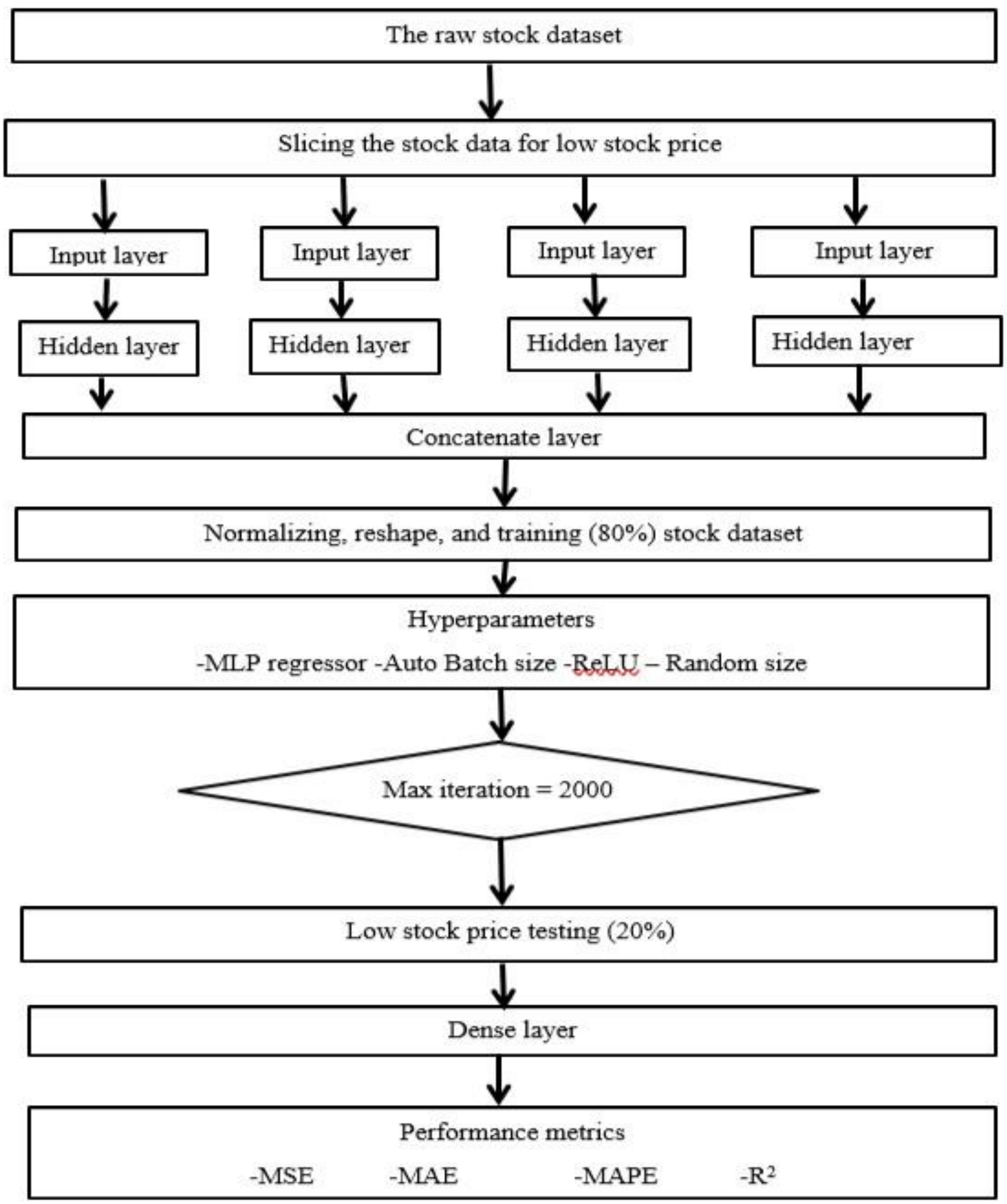


Figure 3

Neural network architecture of the VQNN model

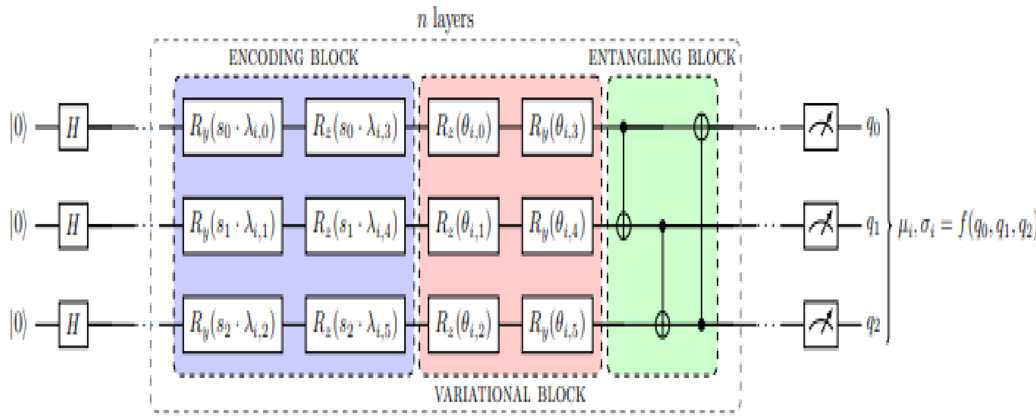


Figure 4

Comparison of the recorded, and predicted CRDB low stock price

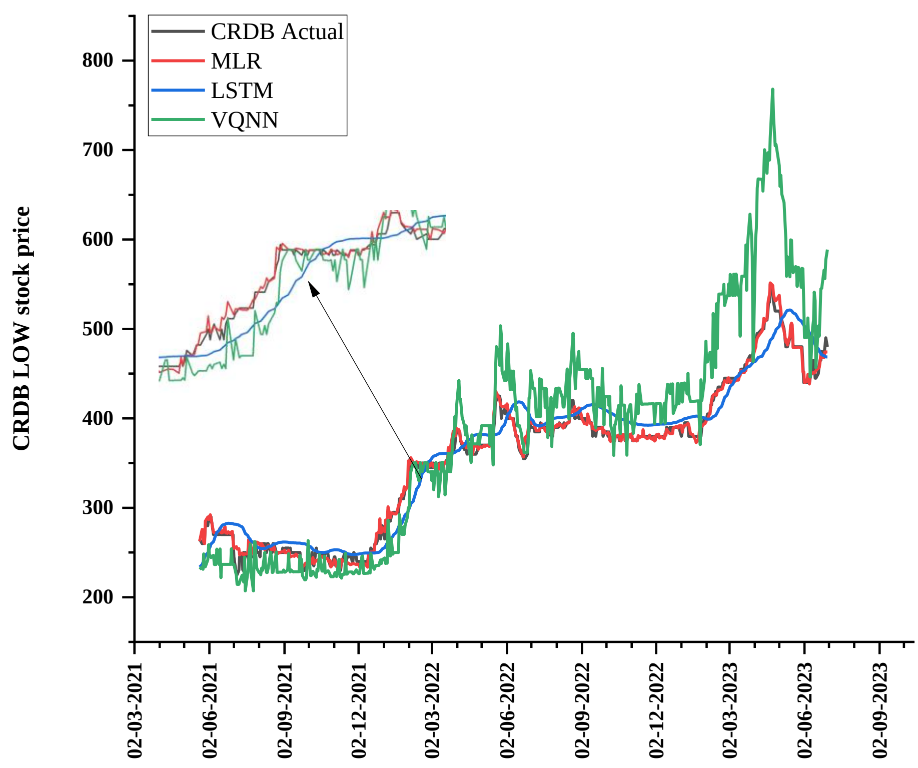
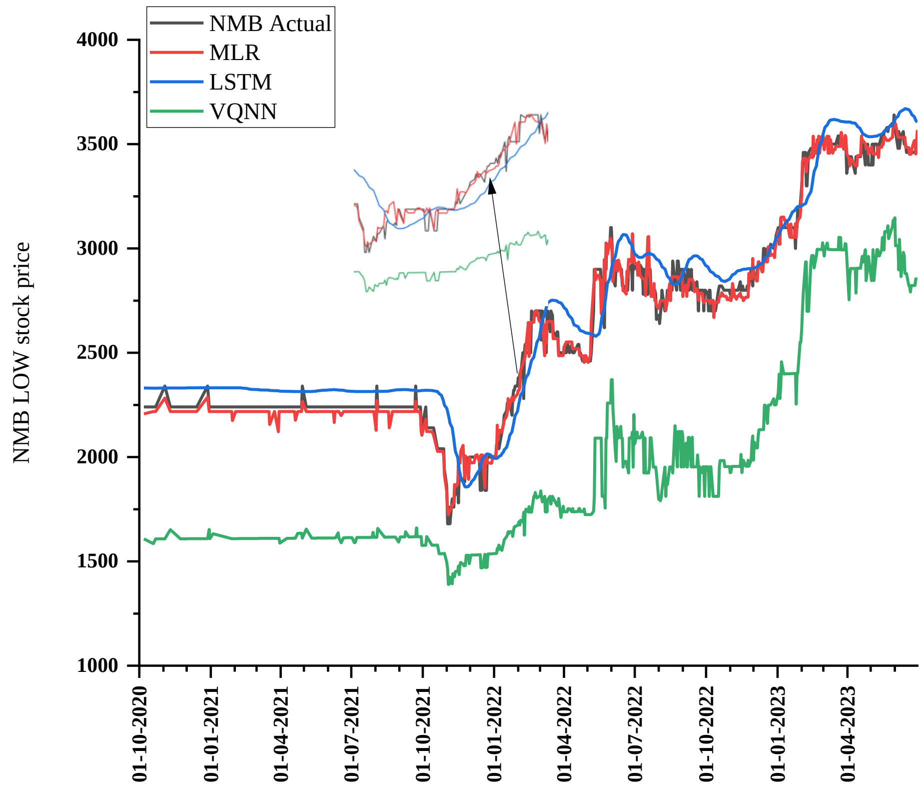


Figure 5

Comparison of the recorded, and predicted NMB low stock price



Discussion

The results from this study for CRDB's low stock price are compared with those existing in the literature and are tabulated for the predictor variables as indicated in Tables 6, 7, 8 and 9. Srivinay et al. (2022) used a hybrid model and deep neural network (DNN) in the prediction of the stock price of Kotak, ICICI, Axis, and Yes Bank which are of high capital and with more volatility, the result of RMSE, and MAE values obtained are shown in Table 6 and 7 respectively and compared to the proposed models (LSTM, MLR, VQNN). It can be observed that the dropout LSTM and DNN models had the largest RMSE and MAE values which increase incapability to long-term dependencies but VQNN, hybrid model, and MLR models had the lowest value due to proper tunning of hyperparameter to overcome the instability in the prediction model, further, it is observed that addition of hidden layers increases the accuracy of the models.

Quadir *et al.,* (2023) used the MLS-LSTM model in the prediction of the low stock price of Samsung, the result of MAPE values obtained, and compared with DL networks (LSTM, MLP, RNN, CNN) in the prediction stock price of Maruti, HCL, and Axis Bank within 400 days, (Rahman *et al.,* 2019), but the MLS-LSTM model outperformed the DL networks due additional layers used. Further, the MAPE result of the MLS-LSTM model was compared to the proposed models (LSTM, MLR, VQNN) as shown in Table 8. It can be observed that the LSTM and MLR models outperformed the MLS-LSTM, and VQNN models due to the presence of more hidden layers, and dropout layers which reduce overfitting and increase the capability of long-term dependencies.

MLS-LSTM for Samsung's low stock price prediction (Quadir *et al.*, 2023), the R^2 value was obtained, and compared to support vector machine (SVM), and linear regression in stock price prediction (Bonab *et al.*, 2013). The MLS-LSTM model outperformed both the SVM and linear regression model in the R^2 value which indicates more accuracy for stock price prediction of the MLS-LSTM. The R^2 values of the MLS-LSTM model were compared to the proposed models (LSTM, MLR, VQNN) as shown in Table 9, proposed models were more accurate for the prediction of stock price.

The observation of the result produced by proposed models (LSTM, MLR, and VQNN) on projecting the low stock price, shows that the LSTM model which is a single input, and single output had better result due to the dropout layer which reduce the overfitting of the stock data, and can be observe in figure 4, and 5 had more convergence which indicate the accuracy of the model in prediction, (Yadav *et al.*, 2020). The MLR model from the comparison of different studies (Wang *et al.*, 2023; Alkhatib *et al.*, 2022) has shown that with many inputs more than four in traditional models has more advantage to allow flexibility, and adaptability which increases the accuracy performance of the model, and the overlap, and convergence between recorded, and predicted low stock price is good as shown in figure 4, and 5.

The VQNN model show the performance to predict the trend of the low stock price despite of being faster in learning, a smaller number of hidden neurons, and high speed in processing stock data, (Jeswal and Chakraverty, 2018). The model is encountered by overfitting of the stock data due to large data used during training and testing process, (Hirai, 2024). The trend of the low stock price was predicted but the convergence between recorded, and predicted low stock price was very less which indicate incapability to predict low stock price as shown in figure 4, and 5. The models which will be the best to predict the stock price will maximize the profit to the investment, the projection of the stock price has been using different approaches, (Kiran, 2021).

Conclusion

The results are promising for the three networks, which are LSTM with dropout, MLR, and VQNN networks for predicting low stock prices of CRDB and NMB datasets. The result from the LSTM network showed that the application of the layer has increased the efficiency of the network by reducing the occurrence of overfitting, and sensitivity of the stock dataset. Further, the utilization of more hidden layers in the MLR network has triggered updates on the weights and biases of the network for a more accurate prediction. The VQNN network had less efficiency and showed inconsistency in terms of performance indices due to the overfitting of stock datasets. The average correlation coefficients in terms of the predictor R^2 of the three networks were 0.9673, 0.9940, and 0.8647 for the LSTM, MLR, and VQNN networks respectively. Hence the MLR and LSTM networks were accurate in predicting the low stock price for CRDB and NMB datasets considered due to the presence of more additional layers and dropout layer which reduce generalization of the results.

Recommendations

Though LSTM performed better, MLR has also produced good results in terms of the correlation coefficient and it is observed that the speed of computation is better with VQNN. It is recommended that further analysis of different datasets considering various input and output parameters and considering evolutionary algorithms and hyper-tuned VQC for speed and accuracy should be explored to generalize the performance comparison. Fundamental and sentiment analysis must be taken into action since public opinions affect the performance of the company and allow a change in the prediction model to fit sentiment for more precise stock price prediction.

Acknowledgment

The author would like to communicate his deepest gratitude to Prosper Mtui, Adela Nzao Fredrick Nguma, and Teddy Mtui for support during the study.

References

Al Bashabsheh. E., & Alasal. S.A. (2021). *Detecting and Rating Humor and Offensive Text Using Deep Learning.* In Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pp. 1102–1107.

Alkhatib. K, Khazaleh. H, Alkhazaleh. H.A, Alsoud. A.R, & Abualigah. L, (2022). *A New Stock Price Forecasting Method Using Active Deep Learning Approach.* J. Open Innov. Technol. Mark. Complex. 8, 96. https://doi.org/10.3390/ joitmc8020096.

Avramov. D, Kaplanski. G, & Subrahmanyam. A, (2021). *Moving average distance as a predictor of equity return.* Rev. Financ. Econ. 39, 127–145.

Bonab. H. T, Gharehchopogh.F. S, & Khaze. R. S, (2013). *A linear regression approach to prediction of stock market trading volume: a case study,* Int. J. Manag. Value Supply Chains 4 (3) 25.

Borodin. A, Mityushina. I, Streltsova. E, Kulikov. A, Yakovenko. I, & Namitulina. A, (2021). *Mathematical Modeling for Financial Analysis of an Enterprise: Motivating of Not Open Innovation.* J. Open Innov. Technol. Mark. Complex. 7, 79.

Chaiboonsri. C., & Satawat. W. (2021). *Multi-process analysis and portfolio optimization based on quantum mechanics (QM) under risk management in ASEAN Exchanges: A Case Study of Answering to the E-Commerce and E-Business Direction.* In Handbook of Research on Innovation and Development of E-Commerce and E-Business in ASEAN. 2 vols. Hershey: IGI Global, 16p. DOI: 10.4018/978-1-7998-4984-1.ch003.

Ciliberto. C, Herbster. M, Ialongo. D. A, Pontil. M, Rocchetto. A, Severini. S, & Wossnig. L, (2018). *Quantum machine learning: a classical perspective,* Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474, 20170551.

De Myttenaere. A., Golden. B., Le Grand. B., & Rossi. F. (2016). *Mean Absolute Percentage Error for regression models.* Neurocomputing, vol. 192, pp. 38–48.

Deb. D, (2020). *Multiple Linear Regression with Python.* Dibyendu Deb, 4 May 2020, dibyendudeb.com/multiple-linear-regression/#Assumptions_for_multiple_linear_regression.

Doong, S. C., & Doan, A. T. (2021). *The influence of political uncertainty on commercial banks in emerging market countries.* International Journal of Public Administration, 104, 1–17.

Ezugwu. A.E., Ikotun. A.M., Oyelade, O.O., Abualigah, L., Agushaka. J.O., Eke. C.I., Akinyelu. A.A. (2022). *A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects.* Eng. Appl. Artif. Intell. 110, 104743. https://doi.org/10.1016/j.engappai.2022.104743.

Hirai. H, (2024). *Practical application of quantum neural network to materials informatics.* Natureportfolio. Scientifc Reports 14:8583 https://doi.org/10.1038/s41598-024-59276-0.

Houssein. H. E., Oliva.D., & Hinojosa.S. (2021). *Metaheuristics in Machine Learning: Theory and Applications,* Springer International Publishing, Cham, Switzerland.

Jeswal. K. S, & Chakraverty. S, (2018). *Recent Developments and Applications in Quantum Neural Network: A Review.* Archives of Computational Methods in Engineering https://doi.org/10.1007/s11831-018-9269-0

Jia. H. (2016). *Investigation into the effectiveness of long short-term memory networks for stock* *price prediction.* arXiv 2016, arXiv:1603.07893.

Killoran. N, Bromley. T.R, Arrazola. J.M, Schuld. M, Quesada. N, & Lloyd. S, (2019). *Continuous-variable quantum neural networks.* Physical Review Research 1(3), 033063.

Kiran. G.M, (2021). *Stock Price prediction with LSTM Based Deep Learning Techniques.* Int. J. Adv. Sci. Innov. 2, 18–21

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (2008). *Forecasting Methods and Applications (3rd ed.).* Wiley Singapore.Istanbul Stock Exchange. Expert Syst. Appl, 38, 5311–5319.

Mari. A., Thomas. R. B., Izaac. J., Schuld. M., & Killoran. N. (2020). Transfer learning in hybrid classical-quantum neural networks. Quantum 4, 340. https://doi.org/10.22331/q-2020-10-09-340.

Mitarai. K., Negoro. M., Kitagawa. M., & Fujii. K. (2018). *Quantum circuit learning*. Phys. Rev. A 98, 032309.

Moon. Y., & Yao. T., (2011). *A robust mean absolute deviation model for portfolio optimization*. Comput. Oper. Res., vol. 38, no. 9, pp. 1251–1258.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning* Vol. 25 Determination Press.

Oh. S., Choi. J., & Kim. J, (2020). *A tutorial on quantum convolutional neural networks (QCNN).* In: 2020 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, pp. 236–239.

Preskill. J, (2018). *Quantum computing in the NISQ era and beyond.* Quantum 2, 79.

Quadir, Md. A., Kapoor, S., Junni, A. V. C., Sivaraman, K. A., Kong, F. T., Sabireen, H., & Janakiraman, N. (2023). *A novel optimization approach for stock price forecasting using multi-layered sequential LSTM.* Applied Soft Computing, 134, 109830.

Rahman.O.M, Hossain.S.M, Junaid.S.T, Forhad.A.S.M, & Hossen.K.M, (2019).*Predicting prices of the stock market using gated recurrent units (GRUs) neural networks,* Int. J. Comput. Sci. Netw. Secure. 19 (1) 213–222.

Rai, K., & Garg, B. (2021). *Dynamic correlations and volatility spillovers between stock price and exchange rate in BRIICS economies: Evidence from the COVID-19 outbreak period.* Applied Economics Letters, 229, 1-8.

Ribeiro, G. T., Santos, A. A. P., Mariani, V. C., & dos Santos Coelho, L. (2021). *Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility.* Expert Systems with Applications, 184, 115490.

Sebastianelli. A, Zaidenberg. D.A., Spiller. D, Le Saux. B, & Ullo. S.L., (2021). *On circuit-based hybrid quantum neural networks for remote sensing imagery classification.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 15, 565–580.

Singh. J, & Khushi. M, (2021). *Feature Learning for Stock Price Prediction Shows a Significant Role of Analyst Rating.* Appl. Syst. Innov. 4, 17.

Srivastava.N, Hinton. G, Krizhevsky. A, Sutskever. I, & Salakhutdinov.R (2014). *Dropout. A Simple Way to Prevent Neural Networks from Overfitting.* Journal of Machine Learning Research 15, 1929-1958.

Srivinay, Manujakshi. B.C, Kabadi, M.G, & Naik. N (2022). *A Hybrid Stock Price Prediction Model Based on PRE and Deep Neural Network.* Data, 7, 51. https://doi.org/ 10.3390/data7050051.

Stamatopoulos. N., Egger. J. D., Sun. Y., Zoufal. C., Iten. R., Shen.N., & Woerner. S. (2020). *Option Pricing using Quantum Computers.* Quantum 4, 2. *https://doi.org/10.48550/arXiv.1905.02666*

Wang. Q, Chang .X, & Tieyan. Z (2023). *Stock price prediction based on multiple linear regression.* BCP Business & Management, Volume 36.

Yadav. A., Jha. C. K., & Sharan. A. (2020). *Optimizing LSTM for time series prediction in the Indian stock market.* Procedia. Comput. 167, 2091-2100. https://doi.org/10.1016/j.procs.2020.03.257.

Zhong. Y & Yuan. C (2012) *Quantum competition network model based on quantum entanglement.* J Comput 7(9):2312–231