NATIONAL BOARD FOR TECHNICAL EDUCATION
NIGERIAN JOURNAL OF TECHNICAL EDUCATION
Volume 22 Nos. 1 2024
ISSN No. 2992-3522

ENSEMBLE METHOD FOR THE PREDICTION OF STOCK PRICE USING MACHINE LEARNING TECHNIQUES

B.B. Alhaji[1] and I. C.Okorie[1]

[1]Department of Mathematical Sciences, Nigerian Defence Academy,

Kaduna - Nigeria

ABSTRACT

An ensemble model, referred to as SVR-LSTM-CNN (Support Vector Regressor - Long Short-Term Memory - Convolutional Neural Network), was proposed for stock price forecasting. This model combines the SVR for capturing linear features and a deep neural network structure that incorporates LSTM and CNN layers to capture both linear and nonlinear data features. In the SVR-LSTM-CNN model, SVR is utilized to capture linear features, LSTM captures long-term dependencies in the data, and CNN captures hierarchical data structures. The study focused on four different stocks from the Nigerian Stock Exchange historical database. A comprehensive performance evaluation, using daily stock prices, revealed that the SVR-LSTM-CNN model outperformed benchmark models in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) performance measure.

Keywords: Machine Learning, Ensemble Method, Support Vector Regressor, Convolutional Neural Network, Long Short-Term Memory, Stock Price Prediction

1.0 INTRODUCTION

A stock, in finance, represents a share in a company's ownership, and those who own stock (also called stock holders or shareholders) can claim a percentage of the company's assets and earnings. Units of stock are called 'shares.' Whenever a company issues shares to the public for the first time to raise money for expansion, they often do so through an initial public offer (IPO), and these shares are sold in the primary market. After the first sale, the shares can be resold by stockholders in a place called the secondary market, which is mostly referred to as the stock market, also called the stock exchange market. An example of the stock exchange market is the Nigerian Exchange Group (NGX). A share sold in the market at a particular price is called a stock price.

A person interested in investing in the stock market needs to create time for proper monitoring of the stock market due to price fluctuations, which can either rise or fall depending on certain factors like news, inflation, and politics. These factors have effects on the market, thereby making prediction complicated.

Li-Pang (2020) mentioned that stock price has a time-dependent feature, and time series analysis is extensively employed to describe data that varies over time. Machine learning (ML) is an emerging technology that empowers machines to glean insights from historical data and automatically perform a given task. ML allows computers to learn from the experience independently while using statistical methods to improve their efficiency and the ability to forecast results without explicit programming ML is a subdivision of artificial intelligence (AI) that helps one to build AI-driven applications. Some popular machine learning applications are email spam filtering, product recommendations, face recognition, and self-driving cars. One of the major tasks of machine learning techniques is to use large data to build a mathematical model that aids in predicting the future accurately. Some machine learning techniques like random forests and support vector machine (SVM) have been used by Xuetal. (2020) for financial stock forecasting. Likewise, Parray et al. (2020) applied logistic regression and artificial neural networks (ANN) machine learning techniques to predict the trend of stocks for the next day.

Aside from the various approaches based on ML, several deep learning techniques have emerged for research in time series forecasting. Deep learning (DL) is a member of ML that can extract high-level abstraction features for data representation like convolutional neural networks, which havealso been used for image processing. It is also used for analyzing financial time series data.

The goal of investors in stock markets is to make a profit in future, and the expectation for developing the best methods suitable for predicting stock prices to avoid loss has increased. As a result, it is crucial to develop proper models to obtain reliable results with the best accuracy for making precise and consistently correct predictions as many underlying patterns, repetition, factors and self-organization are involved.

Over the years, researchers predicted stock market prices using a support vector regressor (SVR) combined with different kinds of windowing functions as a data preprocessing step to feed the input into the machine learning algorithm for pattern recognition. Hastie et al. (2008) discussed on the SVMs and flexible machine learning kernel methods and others such as Li-Pang (2020) compared the performance of SVR and LSTM to the hybrid model of CNN-LSTM while Lu et al. (2020), Long et al. (2020) and Hao et al. (2020) used the hybrid model of CNN-LSTM and observed the accuracy and performance. Wen et al. (2019) first exploited the sequences reconstruction method to reduce the noise of the financial time series data, after which they employed the CNN model to extract spatial structure from the time series data for stock prediction. Sezer et al. (2018) first transformed stock technical indicators into 2D (2- dimensional) images and designed a novel method based on CNN for the stock price prediction task while Yu et al. (2021) proposed that the model performs better if the hybridization comprises more than two components, implying that a model's complexity can improve its prediction accuracy.

Three precarious issues were raised by Nti et al. (2020) on constructing ensemble classifiers and regressors, which are - the choice of base regressor or classifier technique adopted, the combination techniques used to ensemble multiple regressors or classifiers and the quantum regressors or classifiers to been sembled. The result of the research suggested that an innovative study in the domain of stock market prediction ought to include ensemble techniques in their algorithms.

However, in these researches, studies have been limited to the hybrid of CNN-LSTM based on comparisons of prediction accuracy. Thus, in this research, to enhance prediction accuracy and efficiently extract features, SVR will be combined to the hybrid of LSTM-CNN integrating into a cohesive framework and suggesting a novel time series prediction ensemble model named SVR-LSTM-CNN. The new ensemble model will be tested for better accuracy and performance. It will also be compared to the CNN-LSTM hybrid model based on accuracy and performance.

The major parties benefiting from this research are the industry, academia and shareholders. The industry will benefit because variables and machine learning techniques used in this research can further provide better insights on underlying trends predicting the stock price in the industry. In academia, this research will bring a better understanding of the stock market to the limelight, and the insights provided can be used for future research. Finally, for the shareholders, this will be a boost in making better decisions to get higher returns from the stock market.

2.0 MATERIALS AND METHODOLOGY

The stock market remains a favoured investment choice owing to the potential for high returns, and the daily behaviour of the market prices revealed that the stock prices are volatile, which makes prediction difficult, according to Akwuegbo et al. (2010). A tutorial on SVR was done by Alex et al. (2004), exposing the basic ideas under lying SVR for function estimation and how this algorithm is currently used for training support vector machines, covering both the quadratic (or convex) programming part and advanced methods for dealing with large datasets. With the advancement in technology, the stock market price prediction problems can be solved using various techniques. Li-Pang (2020) considered three machine learning techniques: LSTM, CNN and SVR, for stock price prediction and it was further shown in the research that a machine learning model could be trained on one stock dataset, which can also be used to predict various stock prices with a modest compromise in terms of precision. However, in this research, to avoid feature loss by extracting deep features efficiently, a hybrid model was structured to LSTM-CNN by changing the processing order and testing for accuracy to determine if it would perform better than CNN-LSTM. Also, the new hybrid model (LSTM-CNN) was used to develop an ensemble model (SVR-LSTM-CNN). Thus, the study was conducted in the following stages:

2.1 Data Collection

The data was sourced from NGX, which was formerly known as the Nigerian Stock Exchange from January 4, 2010, to April 8, 2019, for four banks which are Guaranty Trust Bank (GTB), Zenith Bank Plc (ZBN), First Bank (FBN) and United Bank for Africa (UBA). The data under went four general processes: data collection, pre-processing, data analysis and getting the result.

2.2 Model Architecture

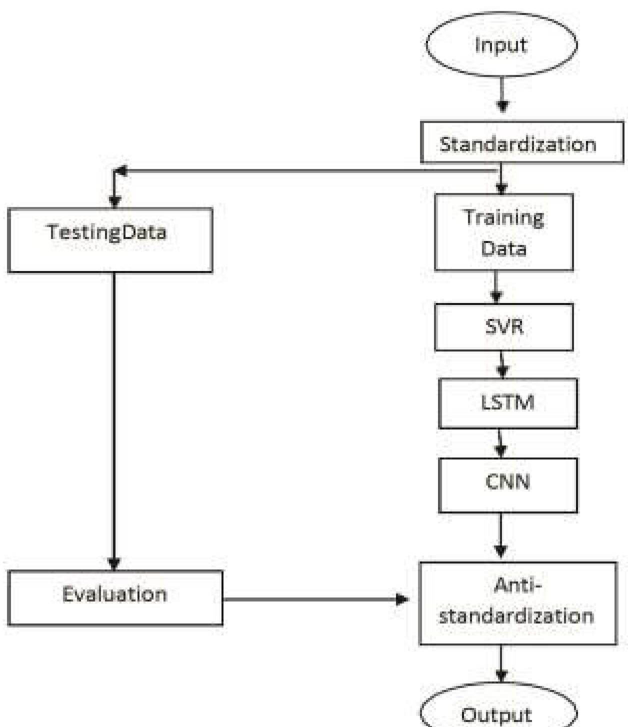
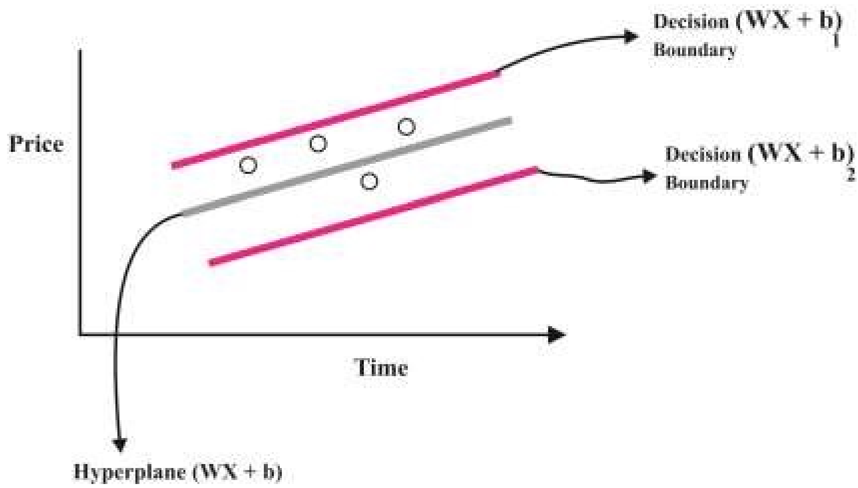


Figure 2.1: Architecture of SVR - LSTM - CNN model

2.3 Algorithms

The following algorithms were used in this research work to create the ensemble model:

2.3.1 Support Vector Regressor (SVR): The SVR is a machine learning technique for pattern recognition. The objective of SVR is to minimize errors by tailoring the hyper plane, maximizing the margin while allowing for a tolerance in certain error instances.



Source: www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/

Figure 2.2: Image on the idea behind SVR

Visualize the two lines as the decision boundary and the central third line as the hyperplane in Figure 2.2. The objective is to concentrate on the points within the decision boundary line. The optimal fit line aligns with the hyperplane that encompasses the maximum number of points (vectors).

Assuming that the decision boundary lines are at a distance, say 'a', from the hyperplane, then, the lines that we draw at a distance from the hyperplane will be a^+ and a^-.

Let us assume that the equation of the hyperplane is as follows:

$$(x)=\vec{w}\vec{x}+b \qquad 2.1$$

Then the equations of decision boundary become:

$$\vec{w}\vec{x}+b=a^+ \qquad 2.2$$

$$\vec{w}\vec{x}+b=a^- \qquad 2.3$$

Thus, any hyperplane that satisfies our SVR should satisfy:

$$a^-<(x)-\vec{w}\,\vec{x}+b<a^+ \qquad 2.4$$

where

$\vec{x}$ is the vector of the predictor variables

$d(x)$ is the sample classification

$\bar{w}$ is the weight

vector b is a constant

To describe the SVR classifier explicitly, let

$(x)=\bar{w}\,\dot{x}+b>0; x\epsilon a^{+}$ 2.5

$(x)=\bar{w}\,\dot{x}+b<0; x\in a^{-}$ 2.6

$(x)=\bar{w}\,\dot{x}+b=0; x\in$ Hyperplane 2.7

To maximize the margin of the hyperplane to the nearest points from the two classes $a^{+}$and a^{-}, let D be the width of the margin. For the best accuracy, the decision line has to be in the middle of the decision boundary, so we have $\frac{D^{-}}{2}$ on the left and $\frac{D^{+}}{2}$ on the right of the decision line which must be equal.

Assume that w is a unit vector, that is$\|w\|=1$

If$\bar{w}\,\dot{x}_i+b\geq\frac{D^{+}}{2}; {}_i\in a^{+},=1,\ldots n$ 2.8

and

$\bar{w}\,\dot{x}_i+b\leq\frac{D^{-}}{2}; x_i\in a^{-},=1,\ldots,n$ 2.9

Then,

$\frac{T\bar{w}\,\dot{x}_i+b}{\|w\|}\geq\frac{D^{+}}{2}; {}_i\in a^{+},=1,\ldots,n$ 2.10

and

$\frac{T\bar{w}\,\dot{x}_i+b}{\|w\|}\leq\frac{D^{-}}{2}; {}_i\in a^{-},=1,\ldots,n$ 2.11

This implies that

$\bar{w}\,\dot{x}_i+b\geq\|w\|\frac{D^{+}}{2}; x_i\in a^{+}$ 2.12

and

$\bar{w}\,\dot{x}_i+b\leq-\|w\|\frac{D^{-}}{2}; x_i\in a^{-}$ 2.13

Using$\|w\|=\frac{2}{D}$ $i(3.12)$which is an arbitrary choice because the magnitude of$\|w\|$does not matter, then we will get

$\bar{w}\,\dot{x}_i+b\geq1$ 2.14

Since$\|w\|=\frac{2}{D}$,then$D=\frac{2}{\|w\|}$ 2.15

The goal is to maximize the margin D such that there is no training sample inside the margin, and this has translated mathematically into 2.12 and 2.13.

Therefore, the optimization problem can be written as

$Max\ \frac{2}{\|w\|}$

While taking into account the constraints in 2.12. and 2.13 2.16

Since we have two equations in the constraints, we will simplify into one equation by introducing y_i such that the i^{th} training sample will be

$$y(\bar{w}\grave{x}_i+b)\geq 1 \; for \; i=1,\ldots,n \tag{2.18}$$

So, we $Maximize \; \frac{2}{\|w\|}$

Subject to

$$y(\bar{w}\grave{x}_i+b)\geq 1 \qquad for \; i=1,\ldots,n \tag{2.19}$$

Or equivalently,

$$Minimize \; \frac{1}{2}\|w\|^2$$

Subjectto

$$y(\bar{w}\grave{x}_i+b)\geq 1 \qquad for \; i=1,\ldots,n \tag{2.20}$$

Note that maximizing the margin (that is,$aximize$ $\frac{2}{\|w\|}$ which is the key principle of a support vector classifier problem) is equivalent to $Minimize \frac{1}{2}\|w\|^2$as stated by Winston (2010).

By the Lagrange multiplier method, we create a primal Lagrangian to solve the optimization problem

$$L_p(w,b,\alpha_i) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\bar{w}\grave{x}_i+b)-1))$$

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i \bar{w}\grave{x}_i - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i \tag{2.21}$$

$$\frac{\partial L_p}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i \bar{\grave{x}}_i = 0 \tag{2.22}$$

Therefore,

$$w = \sum_{i=1}^{n} \alpha_i y_i \bar{x}_i \tag{2.23}$$

$$\frac{\partial L_p}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.24}$$

Replacing (3.24) and (3.23) in (3.21), we have a dual Lagrangian

$$L(\alpha_i) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \bar{x}_i \grave{x}_y - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \bar{x}_i \grave{x}_y + \sum_{i=1}^{n}\alpha_i$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \bar{x}_i \grave{x}_y \tag{2.25}$$

In essence,

$$MaxL_d(\alpha_i)$$

Subject to

$\alpha_i \geq 0$ 2.26

Implies that when

$\alpha_i = 0,_i$ is not a support vector

$\alpha_i \neq 0,_i$ is a support vector

$\alpha_i \neq 0, nd\alpha_i$ *is very high*, x_i is a support vector and an outlier training sample.

If the training data is non-linearly separable, there will be a transformation into a higher dimensional space say $\emptyset(x)$.

Since from 3.1,

$$d(x) = \bar{w}\hat{x} + b$$

This implies that putting 3.23 in 3.1, we have

$(x) = \sum_{i=1}^{n} \alpha_i y_i \bar{x}_i \bar{x}_y + b \qquad for\, i, j = 1, \ldots, n$ 2.27

We observed that the maximization depends on the dot product of pairs of samples, that is $\bar{x}_i \bar{x}_y$ in 2.25 and 2.27.

To maximize in the transformed feature space $(\emptyset(x)$, we have

$L_d(\alpha) = \sum_{i=1} \alpha_i - \frac{1}{2}\sum_{i=1} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \emptyset(x)_i . \emptyset(x)_j$ 2.28

and

$(x) = \sum_{i=1}^{n} \alpha_i y_i \emptyset(x_i) . \emptyset(x_j) + b$ 2.29

To avoid large number of operations, we define K to be a kernel function such that

$$(x_i, x_j) = \emptyset(x_i) . \emptyset(x_j)$$

This implies that to have the classifier, 2.28 and 2.29 become

$L_d(\alpha) = \sum_{i=1} \alpha_i - \frac{1}{2}\sum_{i=1} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i, x_j)$ 2.30

$(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x_j) + b$ 2.31

For the regression problem, our aim is to predict actual stock prices rather than merely forecasting upward or downward trends. Thus, y_i represents a continuous real value. The optimization problem can be modified to

$maximize \frac{1}{2}\sum_{i=1} (\alpha_i - \alpha_i^*)(y_i - d(x_i)) - C\sum_{i=1} (\alpha_i + \alpha_i^*)$ 2.32

subject to $\sum_{i=1} (\alpha_i - \alpha_i^*) = 0$ and $0 \leq \alpha_i \leq c$ 2.33

$$\text{where}(x)=\sum_{i=1}^{n}(\alpha_i-\alpha^*_i)K(x_i,x)+b$$

Errors with an absolute value smaller than Ɛ areignored. Theregularization constant $c>0$ plays a crucial role in balancing the non-linearity of d and the extent to which deviations larger than Ɛ are taken into account. Commonly preferred kernel functions include the radial basis functionand the polynomial kernel according to Hastie *et al*. (2008).

2.3.2 Convolutional Neural Network (CNN): A Convolutional Neural Network (CNN) belongs to the category of deep neural networks primarily employed for analyzing visual imagery and recognizing patterns. CNN utilizes a specialized technique called convolution, a mathematical operation on two functions that results in a third function depicting how one function is altered by the other. Essentially, CNN's purpose is to diminish redundant characteristics or features inthe data, transforming it into a more manageable form without sacrificing crucial features essential for accurate predictions. In essence, CNN is employed to extract profound features by capturing the inherent characteristics of time series data. The CNN structure consists of the following:

Input Layer: The input layer is solelyused to read the data, and there are no parameters to learn in this layer.

Convolutional Layer: This layer is good at extraction and reorganization.

Pooling Layer: This layer is used for reduction of redundant characteristics of the data, and there are no parameters to learn in this layer.

2.3.3 Fully-connected Layer: Each input unit in this layer has an independent weight associated with every output unit. Also, each output node in this layer is accompanied by a bias contributing to the final prediction result..

Output Layer: In this layer, we have the final result.

In a straight forward scenario, the size of the output CNN layer is determined by the formula 'input_size - (filter_size - 1)'. For instance, if the initial prediction is (50,50), and the filter size is (3,3), the calculation would be (50 - (3 - 1)) = 48.

2.3.4 Long Short-Term Memory (LSTM): The LSTM network model incorporates a memory function designed for an extended period. Employing a gate control mechanism, the model regulates information flow and systematically determines the quantity of incoming information retained at each time step. The LSTM unit consists of a storage unit and three control gates, namely the input gate, output gate, and forgetting gate. For instance, if x_t and h_t represent the input and hidden states at time t, respectively, f_t, i_t, and o_t denote the forgetting gate, input gate, and output gate, and C_t is defined as the candidate information for input storage. However, the storage amount is then governed by the input gate. Each memory cell involves three layers of sigmoid transformation and one layer of than transformation. The computational processes for each gate, input candidate, cell state, and hidden state are elucidated in the subsequent formula:

:

$f_t=(W_f[h_{t-1},x_t]+b_f)$ 2.35

This will yield a value ranging from0 to 1, with 0 indicatingthe exclusion ofall information and 1 signifying the retention of all information from the previous period.

The input gate, governed by the sigmoid function, decides the state of the cell that requires updating..

$i_t=(W_i{}^*[h_{t-1},x_t]+b_i)$ 2.36

It also updates the information that needs to be updated to the cell at period t, using a tanh function.

$\tilde{C}_t=\tanh(W_c{}^*[h_{t-1},x_t]+b_c)$ 2.37

Here, the memory cell state gets updated through

$C_t=f_t{}^*C_{t-1}+i_t{}^*\tilde{C}_t$

The output information is computed through a sigmoid layer and subsequently treated as a tanh function as

$O_t=(W_o{}^*[h_{t-1},x_t]+b_o)$ 2.38

and the final output generated by the cell is

$h_t=O_t{}^*\tanh(C_t)$, 2.39

where $W_f, W_i, W_o,$ *and* W_c denote the weight matrices for the forgetting gate, input gate, output gate, and update state, respectively. Correspondingly, b_f, b_i, b_o, and b_c represent the bias vectors for the forgetting gate, input gate, output gate, and update state. Here, x_t represents the time series data for the current time interval $t$and h_{t-1}represents the output of memory unit from the previous time interval $t-1$.

2.4 Accuracy Measures

In order to compare the techniques, we will be using two (2) measures: root mean squared error (RMSE) and mean absolute error (MAE) to evaluate the result from the SVR-LSTM-CNN model.

2.4.1. Root Mean Squared Error (RMSE)

RMSE, or root mean square error, denotes the standard deviation of residuals or prediction errors. Residuals measure the distance of data points from the regression line, and RMSE measures the spread of these residuals. Essentially, it offers insight into how closely the data forms a cluster around the line of best fit. RMSE was used to compare forecasting errors of the three different algorithms and the ensemble model for the particular NGX data set due to its sensitivity to outliers. It significantly compares a predicted value and an actual value. To achieve more accurate predictions, the smallest or lowest RMSE obtained as the result was chosen.

RMSE is always non-negative and a value closest to 0 (or 0, which rarely achieved in practice) would indicate the best fit to the data.

The formula is:

$$RMSE_{fo}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_{f_i}-z_{o_i})^2} \quad 2.40$$

where:

$(z_{f_i} - z_{o})_i^2$= differences between forecasts and observed values, squared n

= sample size.

There is a direct relationship with the correlation coefficient when standardized observations and forecasts are used as inputs for RMSE. An example is provided wherein a correlation coefficient of 1 signifies that the RMSE will be 0, indicating that all points align perfectly on the regression line resulting in no errors.

2.4.2 Mean Absolute Error

The Mean Absolute Error (MAE) represents the average of all absolute errors, disregarding their direction. An absolute error signifies the extent of bias or in accuracy in measurements, indicating the difference between the measured value and the 'true' value. The formula is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \quad 2.41$$

where:

n = the number of errors

$|x_i - x|$= the absolute errors

3.0 EMPIRICAL STUDIES

In this research, the daily stock prices of four Nigerian banks (GTB, ZBN, FBN and UBA) from 4 January 2010, to 8 April 2019, were obtained from NGX. A computer program was developed using Python and R for data analysis, preparing the dataset through pre-processing, and implementing the machine learning algorithms. The data variables used were daily open price, close price, high price, low price, average price, deals, the volume of trades and value of trades. The dataset contained a total of 9100 daily observations.

The dataset was processed to examine the distribution of prices and daily log-returns, as well as the correlations among various stocks and autocorrelations in the prices. The processed dataset was divided into a training set (80%) and a testing set (20%). The training set contained 7,280 observations of the raw data set and the testingset contained 1,820 observations. The benchmark models used to forecast the stock price were SVR, CNN, LSTM, CNN-LSTM and LSTM-CNN.

SVR-LSTM-CNN model is an ensemble of SVR developed by Vapnik*et al.* (1964), CNN developed by Yann*et al.* (1998) and LSTM developed by Hochre iter*et al.* (1997). Firstly, the SVR model was used to predict the stock price and determine the residual of the SVR model, an assumption was made regarding the presence of spatial and temporal dependencies among neighboring observations of the SVR model's residual. To address this, LSTM layers and a CNN model were employed to extract spatio-temporal features from the SVR model's residual. The LSTM model captured long-term temporal dependencies, while the CNN model focused on extracting spatial relationships among adjacent residuals.

In the SVR model, the radial basis function (RBF) was selected as the kernel function, and the optimal combination of hyperparameters across all datasets included C=2000, Epsilon=0.001, and Gamma=0.001. The architecture of the LSTM-CNN model comprises an input layer, LSTM layer, convolutional layer, max pooling layer, and fully connected layer. The LSTM model consists of a single hidden layer with 54 hidden neurons, while the CNN model includes one convolutional layer, one pooling layer, and one fully connected layer (with 128 filters). The convolutional layer employs the Rectified Linear Unit (ReLU) activation function, and the pooling size is set to 1.

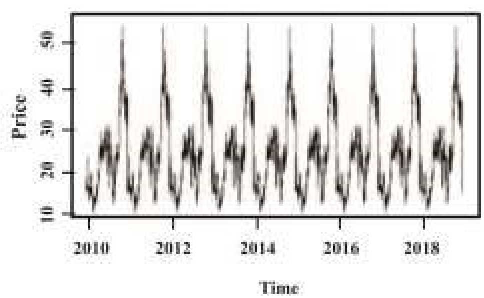


Figure 3.1: Time series plot for GTB daily stock prices

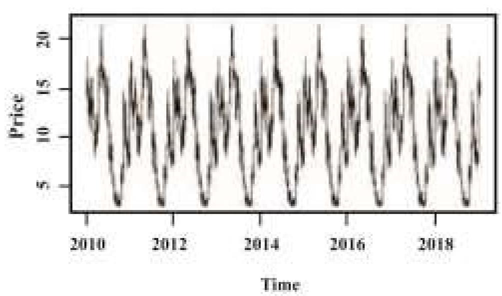


Figure 3.3: Time series plot for FBN daily stockprices

4.0 RESULTS AND DISCUSSION

To verify the efficacy of the proposed ensemble model, different models, including SVR, LSTM, CNN, CNN-LSTM, LSTM-CNN and SVR-LSTM-CNN, are compared on the four stock datasets. The prediction results are shown in the corresponding tables and figures, where the predicted closing price of the stock is represented by the blue curve, while the actual closingprice is depicted by the red curve. The time is displayed on the x-axis, and the y-axis reflects the normalized value

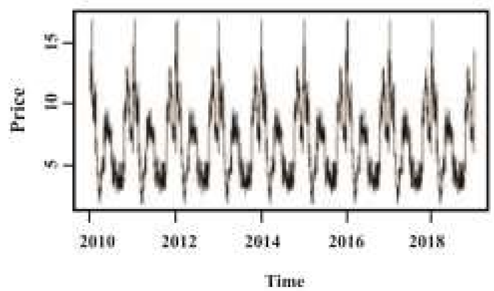


Figure 3.2: Time series plot for UBA daily stockprices

of the stock price.

As shown in Table 1 below for ZBN, the RMSE and MAE of the hybrid SVR-LSTM-CNN model have the smallest values of 59.123 and 42.521 respectively for ZBN among all methods. The SVR model has the highest RMSE value of 76.725 and MAE value of 62.821.CNN-LSTM

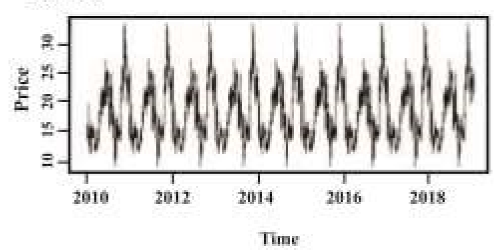


Figure 3.4: Time series plot for ZBN daily stockprices

model has the value RMSE value of 67.210 and MAE value of 50.075 while LSTM-CNN hasthe RMSE value of 64.906 and MAE value of 45.293.

Table1: RMSE and MAE of different forecasting models for ZBN

	SVR	CNN	LSTM	CNN-LSTM	LSTM-CNN	SVR-LSTM-CNN
RMSE	76.725	76.012	70.842	67.210	64.906	59.123
MAE	62.821	59.562	51.987	50.075	45.293	42.521

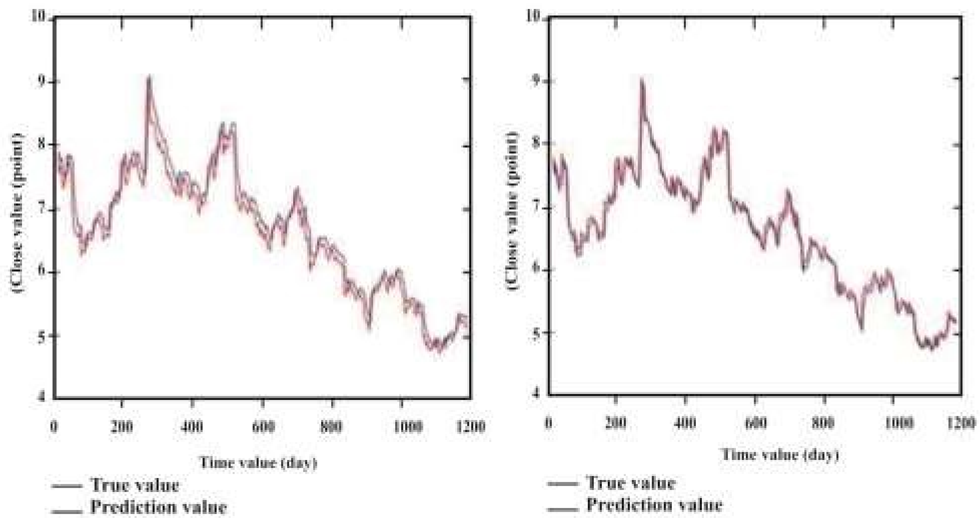


Figure 4.1: Prediction result of LSTM-CNN and SVR-LSTM-CNN respectively for ZBN

In Table 2 for GTB, the RMSE and MAE for GTB show that the SVR-LSTM-CNN model has the least value with 44.227 and 28.349 respectively. The CNN-LSTM model has RMSE of 48.179 and MAE of 32.289 while LSTM-CNN has RMSE of 46.736 and MAE of 32.237. Also, the SVR model has the highest values of RMSE and MAE as 91.908 and 79.914 respectively.

Table2: RMSE and MAE of different forecasting models for GTB

	SVR	CNN	LSTM	CNN-LSTM	LSTM-CNN	SVR-LSTM-CNN
RMSE	91.908	64.978	51.027	48.179	46.736	44.227
MAE	79.914	50.585	35.419	32.289	32.237	28.349

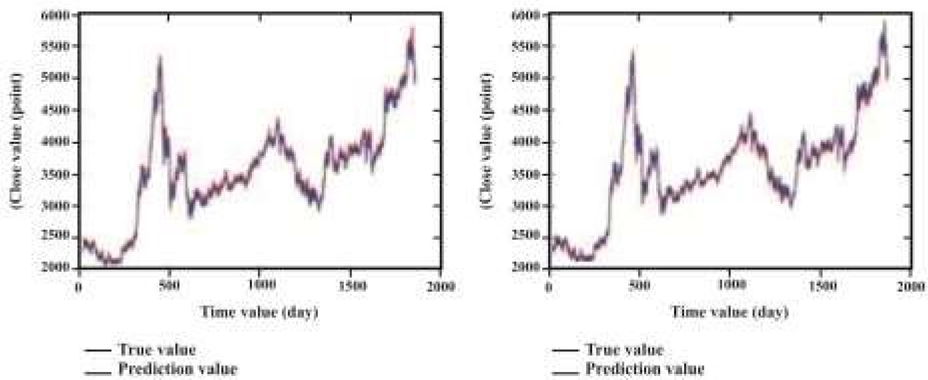


Figure 4.2: Prediction results of LSTM-CNN and SVR-LSTM-CNN respectively for GTB

According to Table 3 for FBN, the SVR-LSTM-CNN model has the smallest values for RSME and MAE with 58.604 and 41.604 respectively. The CNN-LSTM model has the value of 68.144 for RMSE and 51.143 for MAE while LSTM-CNN has the value of 64.507 for RMSE and 46.453 for MAE. The SVR model has the highest values of RMSE as 78.855 and MAE as 63.981.

Table3: RMSE and MAE of different forecasting models for FBN

	SVR	CNN	LSTM	CNN-LSTM	LSTM-CNN	SVR-LSTM-CNN
RMSE	78.855	76.218	71.356	68.144	64.507	58.604
MAE	63.981	58.679	52.119	51.143	46.453	41.604

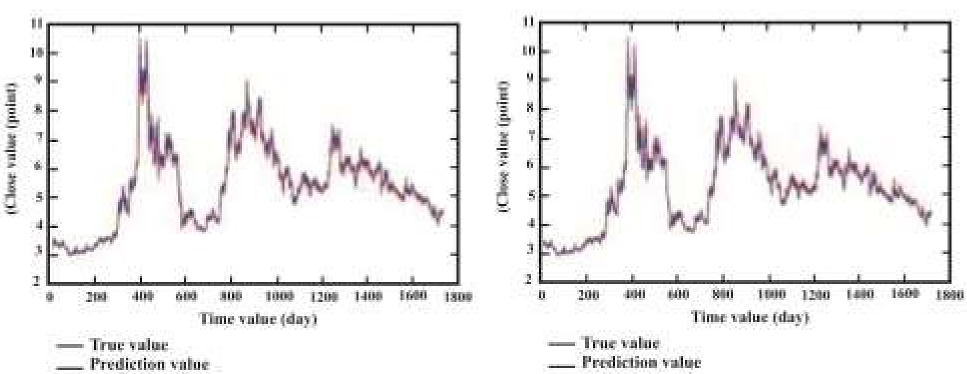
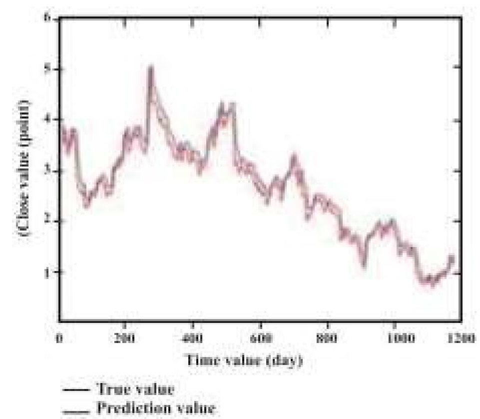


Figure 4.3: Prediction results for LSTM-CNN and SVR-LSTM-CNN respectively for FBN

In Table 4 for UBA, the CNN-LSTM model has 49.675 as RMSE and 31.308 as MAE while the LSTM-CNN model has 47.394 as RMSE and 30.927 as MAE. The SVR-LSTM-CNN model has the least RMSE of 45.856 and MAE of 29.034. Also, the SVR model has the highest value of RMSE as 90.657 and MAE as 78.053.

Table 4: RMSE and MAE of different forecasting models for UBA

	SVR	CNN	LSTM	CNN-LSTM	LSTM-CNN	SVR-LSTM-CNN
RMSE	90.657	63.893	50.234	49.675	47.394	45.856
MAE	78.053	50.464	34.591	31.308	30.927	29.034



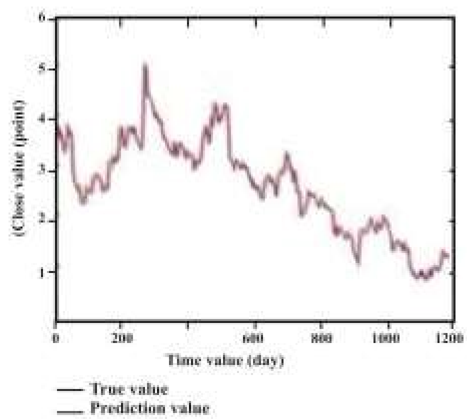


Figure 4.4: Prediction results for LSTM-CNN and SVR-LSTM-CNN respectively for UBA

According to the results shown in Tables 1-4, the RMSE and MAE of the hybrid SVR-LSTM- CNN models are minimal in comparison to other models. It was also discovered that the prediction results of theLSTM-CNN models are better than the CNN-LSTM models because the RMSE and MAE of LSTM-CNN models are smaller than CNN-LSTM models in the different data sets. The results indicate that the LSTM-CNN takes the trend of the stock price as time series into consideration, which is a crucial factor for enhancing forecast accuracy. It was also observed that the SVR-LSTM-CNN performance is better than LSTM-CNN and the othermodels because SVR-LSTM-CNN can exploit the subsequent information of stock price time series.

The results indicate that the SVR-LSTM-CNN can better capture the spatio-temporal features of future stock prices. Furthermore, from the prediction results in the tables, SVR-LSTM-CNN models are in closer proximity to the actual stock price, resulting in smaller errors. In summary, the experimental results confirm the viability and efficacy of the proposed network model.

4.1 SUMMARY AND CONCLUSION

4.2 Summary of Results

This research focused on predicting stock prices using a hybrid model called SVR-LSTM-CNN. The model combined three different models to capture distinct features of the stock price data.

Firstly, SVR model was used as the base to captureboth the linear and nonlinear data features of the stock price. Subsequently, the LSTM model was integrated to capture prolonged temporal dependencies within the data. Finally, the CNN model was employed to extract distinct spatial relationship features from the residual obtained from the SVR model.

The performance of the SVR-LSTM-CNN model was evaluated by comparing it with the following benchmark models SVR, LSTM, CNN, CNN-LSTM, and LSTM-CNN. The experimental results show that the SVR-LSTM-CNN model had the best prediction accuracy on the test data compared to the other models.

This research showed that combining the strengths of each model can significantly improve the accuracy of stock price predictions. The SVR model is good at capturing both linear and nonlinear features, while the LSTM model is suitable for modeling long-term temporal dependencies. The CNN model excels at extracting spatial features, making it a useful additionto the SVR-LSTM hybrid model.

The efficacy of using a hybrid model approach to improve the accuracy of stock pricepredictions was demonstrated in this research. Also, it was discovered in this research that multiple types of features can be captured in the data to achieve better predictive performance by combining the strengths of different models.

4.3 Conclusion

The research findings in this work indicate that the ensemble model of SVR-LSTM-CNN outperforms the benchmark models in fitting stock price data. The SVR-LSTM-CNN model combines the strengths of each individual model, resulting in more accurate predictions of stock prices.

Furthermore, it was discovered in this research that the LSTM-CNN model is particularly effective in predicting daily stockprices when compared to the CNN-LSTM model. This is likely due to the ability of the LSTM model to capture long-term temporal dependencies, combined with the CNN's ability to extract spatial features. These results suggest that the LSTM-CNN model may be a suitable choice for daily stock price prediction applications.

Overall, this work demonstrated the superior performance of the SVR-LSTM-CNN model in fitting stock price data, and highlights the effectiveness of the LSTM-CNN model in daily stock price prediction. These findings have practical implications for financial analysts and traders who rely on accurate predictions of stock prices for investment and trading decisions.

References

Agwuegbo, S.O., Adewole, A.P. & Maduegbuna, A.N. (2010). A Random Walk Model for Stock Market Prices. *Journal of Mathematics and Statistics*, 6(3), 342-346.

Alex, M., Adil, S.H., Raza, K. and Ali, S.S.A.. (2016). Stockmarket prediction using machine learning techniques. Int Conf Comp Inf Sci. (ICCOINS) (pp. 322-327). ICCOINS.

Hao,Y. and Gao, Q.(2020). Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning. *Applied Sciences*, 10(11), 3961.

Hastie, T, Tibshirani, R, Friedman, J. (2008). The elements of statistical learning:data mining, inference and prediction. New York: Springer.

Li- Pang, C.. (2020). Using Machine Learning Algorithms on Prediction of Stock Price. *Journal of Modeling and Optimization*, 84-99.

Long,J., Chen, Z., He,W., Wu,T. and Ren,J.(2020). An integrated frame work of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. *Applied Soft Computing*.

Lu, W., Li,J., Li,Y., Sun, A. and Wang, J.(2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, 10.

Nti,I.K., Adekoya, A.F. & Weyori, B.A. A comprehensive evaluation of ensemble learning for stock- market prediction. J Big Data 7, 20 (2020). https://doi.org/10.1186/s40537-020-00299-5

Parray, I. R., Khurana, S. S.,Kumar, M. and Altalbe, A. A. (2020). Time series data analysis of stock price movement using machine learning techniques. *Soft Computing*, 16509-16517.

Sezer, O. B. and Ozbayoglu, A. M. . (2018). Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach. *AppliedSoft Computing*, 525-538.

Wen, M., Li, P., Zhang, L. and Chen, Y. (2019).Stock market trend prediction using high-order information of time series. IEEE Access, 28299-28308.

Winston, P.H.(2010). *Artificial Intelligence*. MIT6.0342010http://ocw.mit.edu/6-034F10

Xu,Y., Yang, C., Peng, S. and Nojima, Y. (2020). A hybrid two-stage financial stock forecasting algorithm based on clustering and ensemble learning. *Applied Intelligence*, 3852-3867.

Yu, Chen, Ruixin, F, Ting, L., Zongyu, S., Shicheng, L., Yugen, Y.,Wei, Z. and Huilin, S.. (2021). Stock Price Forecast on CNN-BiLSTM-ECA Model. *Hindawi*,p.20.