International Journal of Data Science and Analysis
2022; 8(2): 38-46
http://www.sciencepublishinggroup.com/j/ijdsa
doi: 10.11648/j.ijdsa.20220802.14
ISSN: 2575-1883 (Print); ISSN: 2575-1891 (Online)

Deep Convolution Neural Network with Dropout in Modeling Exchange Rate Volatility

Samuel Wanjiru, Anthony Waititu, Anthony Wanjoya

Department of Statistics and Actuarial Sciences, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya

Email address:
Samuelwanjiru69@gmail.com (S. Wanjiru), agwaititu@gmail.com (A. Waititu), awanjoya@gmail.com (A. Wanjoya)

To cite this article:
Samuel Wanjiru, Anthony Waititu, Anthony Wanjoya. Deep Convolution Neural Network with Dropout in Modeling Exchange Rate Volatility. *International Journal of Data Science and Analysis*. Vol. 8, No. 2, 2022, pp. 38-46. doi: 10.11648/j.ijdsa.20220802.14

Received: March 5, 2022; **Accepted**: March 24, 2022; **Published**: March 31, 2022

Abstract: Exchange rate data possesses time-series features such as a trend. Based on a convolutional neural network (CNN) deep learning algorithm, which has the advantages of detecting patterns, extracting effective features, finding interdependence of time series data, and its computational efficiency, this paper proposes a convolutional neural network with dropout model-based approach to model and forecast exchange rates. In the meantime, this paper uses the CNN to first model and predict exchange rates and the corresponding results of this model are compared with those of the CNN-WD. The experimental results showed that the CNN-WD is superior to the CNN model in terms of the error value, fitting degree and training time. The dataset used for this research is that of daily exchange rates for the period between December 1, 2003, and October 15, 2021, which is comprised of 6528 daily trading observations. Adjusted closing rates are chosen. First, this paper adopts a CNN to effectively identify patterns and extract relevant data features of the exchange rate dataset, making use of the past 21 days. Dropout regularization is then adopted to help prevent the CNN model from overfitting data by temporarily removing a neuron from the network along with all its incoming and outgoing connections during training if its generated random value is below the set dropout rate. This paper further evaluates the reducibility and identifiability of the CNN-WD. As an application, this paper uses the CNN-WD to forecast the next month's average tea price in Kenya.

Keywords: Convolutional Neural Network (CNN), Dropout Regularization, Convolutional Neural Network with Dropout (CNN-WD)

1. Introduction

The role of exchange rates in finance was recognized a few decades ago. Being the rate at which a particular currency will be exchanged for another currency, exchange rates play a vital role in the relative level of a country's economic health. Globally many countries have different currencies while others share the same currency. The US dollar, for example, is the most preferred currency due to its level of acceptance and stability.

Exchange rate volatility and its effects on the volume of international trade has been studied both on theoretical and empirical grounds since a floating exchange-rate regime was adopted in 1973. Exchange rate volatility in this case is the risk associated with the unexpected movement in exchange rates. Because of this uncertainty, researchers and policymakers have looked into the nature and scope of their impact on trade volume. As a result, understanding volatility is critical, as currency rate risk can increase transaction costs while reducing returns from foreign trade and this understanding can boost activities such as asset pricing and circumventing risk [1].

The analysis of financial data has received considerable attention in the literature over the last few decades. Different models have been suggested for capturing special features of financial data. Some of the models are the parametric econometric models and machine learning techniques. However, the parametric models have faced a model misspecification problem whereby the models post serious and inefficient inference if the model such as GARCH is misspecified [2].

Yann LeCun, initially proposed CNN in the 1980s, building on the work of Kanihiko Fukushima, who had invented the neocognitron, a primitive image recognition neural network [3]. CNN has been shown to outperform time series models in forecasting exchange rates since their

inception in time series.

The current trend in financial time series utilizing deep learning is to use the advantages of multiple methodologies to better machine-learning methods [4]. This paper proposes an exchange rate price forecasting method based on a Convolutional Neural Network with dropout (CNN-WD) regularization for predicting the next day's exchange rate so as to make good use of time series characteristics of the exchange rate data, extract features, and enhance exchange price prediction accuracy by combining the advantages of CNN, which can detect patterns, extract effective features, find interdependence of time series data and its computational efficiency, and that of dropout regularization, which prevents the CNN neurons from converging towards the same goal hence preventing it from overfitting the data. This paper uses daily exchange rates from December 1, 2003, to October 15, 2021, to demonstrate the efficacy of the proposed model, with 80 per cent being the train set and 20 per cent the test set.

2. Literature Review

At the moment, the financial market is noisy and nonparametric, and forecasting methods are classified into two main categories: conditional heteroskedastic techniques and machine learning models [4]. Engle introduced the autoregressive conditional heteroscedastic (ARCH) processes in 1982 and in 1986, Bollerslev improved the model and proposed the GARCH model, which is a generalized form of the ARCH model [5, 6]. The GARCH models are however not suitable for modelling exchange rate volatility as they can be misspecified leading to spurious and inefficient inferences [7].

Kristjanpoller et al. used a hybrid Neural Network-GARCH model for predicting future volatility for three stock exchange indices in Brazil, Chile, and Mexico in 2014 and the results showed that an ANN model improves the forecast accuracy of the GARCH model, but the accuracy still needs improving [8]. In 2016, Galeshchuk showed that a multilayered perceptron with a single hidden layer could offer accurate point estimates for exchange rates for practical usage, but the accuracy was poor, making the technology less viable as a framework for constructing profitable trading strategies [9].

Borovykh et al experimental results in 2017 demonstrated that a convolutional neural network is very well suited for regression problem instances as it can effectively learn the dependencies in and between the series, is time-efficient and simple to implement, and outclasses linear and recurrent models [10]. Yamashita et al 2018 experiment showed that a CNN is designed to learn automatically and is able to adapt on its own, but it experiences a data-overfitting problem, giving rise to less accurate results. In 2021, the experimental results of Livieris et al. demonstrated that the use of dropout regularization to try to prevent the recurrent neural network from overfitting data positively affects the forecasting performance [11].

This paper's key contributions are as listed below:

1. A deep learning method (CNN-WD) is proposed for modeling and predicting exchange rates. CNN detects and extracts data features as well as performing data forecasting in this method and dropout regularization prevents the CNN method from overfitting.
2. Comparing the valuation metrics of CNN-WD and CNN, it was discovered that CNN-WD has relatively high prediction accuracy and is best suited for exchange rate forecasting.

3. Deep Convolution Neural Network with Dropout

This study considers a convolutional neural network with (t+ 1) input nodes, L layers of H hidden nodes such that $l \in (1, \dots L)$, a single output layer and a Relu activation function $\psi(x)$. Dropout(Υ) is implemented after the flatten layer of the convolutional neural network. The weights W_h for $h \in (1, \dots, H)$ and α_h for $h \in (0, \dots, H)$ connect the input, hidden and output layers. The hidden layer consists of layers that perform the convolutions, which is the layer that performs a dot product of the convolutional kernel and with the input.

Dropout regularization: To help prevent overfitting and enhance model generalization, dropout regularization is adopted. A parameter is set to a value 0 and 1 such that:

$$\Upsilon_i \sim Bernoulli(p)$$

that is, $\Upsilon = 1$ with probability p and 0 otherwise, Υ_i being the dropout rate. A random value is generated for every neuron that is about to be used. If the value is below the parameter's value, that neuron is not activated.

3.1. CNN-WD

The CNN-WD calculation steps are as follows:

Step 1

Convolution layer: Having the input vector $X = (x_1, \dots, x_n) \in \Re^t$, then, the convolutional layer output is obtained through the convolution of each filter w_h^1 for $h \in (1, \dots, H)$ with the input, as shown in the following formula:

$$u^1(X; \theta, n) = \sum_{n=1}^{N} conv1D(w_{nh}^1, x_n) + b_h^1 \quad (1)$$

b_h^1 being the bias of the h^{th} neuron at layer 1. To reduce the dimensions of the extracted features, average pooling is applied.

Step 2

The output $u^1(X; \theta, n)$ is then passed through an activation function as shown in the equation below:

$$\phi^1(X; \theta, n) = \psi(u^1(X; \theta, n)) \quad (2)$$

In each successive layer $l = 2, \dots, L$ the input feature map is as shown in the equation below:

$$\phi^1(X; \theta, n)^{\{l-1\}} \in \Re^{1 \times N_{\{L-1\}} \times H_{\{l-1\}}} \quad (3)$$

where $1 \times N_{\{L-1\}} \times H_{\{l-1\}}$ is the input filter size from the previous convolution with $N_{\{L-1\}} = N_{\{L-2\}-k+1}$ convolved with a set of H_l filters $w_h^1 \in \Re^{1 \times k \times H_{\{l-1\}}}$ to formulate a feature map $O^l \in \Re^{1 \times N_l \times H_l}$. The receptive field of each output node is controlled by k which is the size of the filter. Therefore, the net input into the output node from the l^{th} hidden layer is the value shown in the following equation:

$$O^l(X;\theta,n) = \sum_{n=1}^{N} \sum_{h=1}^{H\{l-1\}} \alpha_h^l \, \phi^1(X;\theta,n)_{n,h}{}^{l-1} \quad (4)$$

Step 3

The output $O^l(y;\theta,n)$ is then passed through a flatten layer and dropout regularization applied as shown in the equation below:

$$\hat{O}^1(X;\theta,n) = O^l(X;\theta,n) \times \Upsilon_i \quad (5)$$

Finally, the net's output is shown in the following equation:

$$\tilde{Y}\,(X{:}\,\theta,n) = \psi(\hat{O}\,(X;\,\theta,n)) \quad (6)$$

as used above, θ refers to all parameters α0,..., αH and w_h for h = (1,..., H) of the net. $\psi(.)$ is the rectified linear unit activation function. This activation function is as follows:

$$\psi(X) = \max(0,x) \quad (7)$$

Relu outputs the exchange rate inputs directly if positive else it outputs a zero. Relu solves the problem of vanishing gradients, allowing the model to learn quicker and function effectively. Its nonlinearity enables the model to learn complex relationships in data.

Step 4

To reduce convolution layer output dimension and network computation, average pooling is used and its function is as shown in the equation below:

$$\frac{X-z+2v}{S} + 1 \quad (8)$$

X is the input vector, z the filter size, v padding and S the stride.

Step 5

To train the network, the error is monitored and the weights updated until the error is minimized. The error function is as shown in the following equation:

$$\mathrm{Q(X;\theta)} = \frac{1}{2} \sum \left(x_i - \tilde{Y}\,(X{:}\,\theta,n) \right)^2 \quad (9)$$

The statistical model relating x_i to $\tilde{Y}\,(X{:}\,\theta,n)$ is:

$$x_i = \tilde{Y}\,(X{:}\,\theta,n) + \xi. \quad (10)$$

To minimize the model error, this study adopts the backpropagation approach where the network weights are updated until the error $\mathrm{Q(X;\theta)}$ is minimized as follows:

Taking the activation function $\psi(x)$, the model weights are adjusted as follows:

$$W^{l+1} = W^l + \Delta W \quad (11)$$

$$\alpha^{l+1} = \alpha^l + \Delta\alpha \quad (12)$$

Taking the individual weights, the l[th] iteration weights is:

$$W_h^{l+1} = W_h^l - \zeta_1 \left(\frac{\partial Q(X:\theta)}{\partial W_h} \right), h = (1,\dots,H) \quad (13)$$

$$\alpha_h^{l+1} = \alpha_h^l - \zeta_2 \left(\frac{\partial Q(X:\theta)}{\partial \alpha_h} \right), h = (1,\dots,H) \quad (14)$$

From equations (11) and (12), ζ_1 and ζ_2 are step gains. These weights are then adjusted and an adaptive moment estimation used to optimize the gradients of the loss function with respect to the weights. Adam, as the optimizer adapted by this paper, uses the estimates of the first and second moments of gradients to adapt to the learning rate of each weight of the CNN-WD. The number of oscillations are minimum during gradient descent and are controlled to ensure that global minimum is achieved while taking a big step to pass the local minima. This is implemented as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\Delta W \quad (15)$$

$$u_t = \beta_2 u_{t-1} + (1-\beta_2)\Delta\alpha^2 \quad (16)$$

Where m_t and u_t are moving averages, ΔW and $\Delta\alpha^2$ are gradients and β_1 and β_2 the decay rates. The moving averages have a tendency of being biased towards 0 and to correct the bias and control the weights while achieving global minimum, the following is implemented:

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t}, u_t = \frac{u_t}{1-\beta_2^t} \quad (17)$$

To update the weights, the bias-corrected weights are taken and implemented as in the equation below:

$$w_t = w_{t-1} - \zeta \left(\frac{\widehat{m}_t}{\sqrt{u_t}+p} \right) \quad (18)$$

Where w is the model weights and p a small positive constant, which helps avoid a zero division error.

3.2. CNN-WD Statistical Properties

This paper evaluated the following statistical properties:

3.2.1. Model Irreducibility

This paper considers the input-output map of the CNN-WD as in equation (11) below.

$$X \to \tilde{Y}\,(X{:}\,\theta,n) \quad (19)$$

Based on the above input-output map, this paper makes the following assumption:

Assumption: Assume that there exists an input-output $O^l(X;\theta,n)$ such that $\alpha_0,\dots,\alpha_H$ and W_h are the same for both parameter vector θ and $\hat{\theta}$. Therefore, identifiability of the parameter vector θ from the map holds for $O^l(X;\theta,n)$ if and only if it holds for $\hat{O}^l(X;\theta,n)$. To therefore deal with irreducibility, this paper considered the redundancy of θ. The following proposition is therefore made:

Proposition: If there another net with less number of

neurons that delivers the same output, then a CNN with θ is superfluous.

This paper therefore considered the following conditions that lead to the network being redundant as listed below:

1. $\alpha_0 = 0$ for a few $i = 1, \dots, H$
2. Any of the outputs $\upsilon(X; \theta, n)$ is a constant.
3. There are two distinct indexes $i_1, i_2 \in (1, \dots, H)$ for which the functions υ_{i1} and υ_{i2} are equal, that is $\upsilon_{i1}(X; \theta, n) = \pm\upsilon_{i2}(X; \theta, n)$

If the CNN-WD meets any of the above listed conditions, then the neural network is redundant. This paper noted that the presence of irrelevant neurons brings about conditions 1 and 2. Schwarz Information Criterion (SIC) as proposed by [12] was thus adopted as the model selection criterion. This criterion was implemented as shown in the following equation:

$$SIC(h) = \log(\hat{\sigma}) + (h(2 + t) + 1)\frac{\log(n)}{n} \quad (20)$$

The quality of fit measure is the first part, while the complexity penalty is the second part. A single neuron is added and SIC (1) determined. This method is repeated until no further improvements to SIC (h) can be made. To identify a model with h neurons, h+1 models were estimated. To deal with condition 3, this paper adopted the following assumption as stated by [12].

Assumption: Assume that there exists no two distinct indexes $i_1, i_2 \in (1, \dots, H)$ for which the function $\upsilon_{i1}(X; \theta, n)$ and $\upsilon_{i2}(X; \theta, n)$ are of equal sign.

However, regardless of θ being now irreducible, it is still unidentifiable. The unidentifiability of θ is hence discussed below.

3.2.2. Model Identifiability

Unidentifiability of parameters is an elementary problem of the neural network. This paper considered different sets of parameters with their corresponding distributions being identical. These parameters are therefore not unique. To clearly explain this, the CNN-WD weights were represented as below:

α_0 and $\Theta = (\alpha_i, W_i)$ for $i = (1, \dots, H)$ where $W_i = (W_{i0}, W_{i1}, \dots, W_{il})$.

To evaluate model identifiability, this paper considered the following two transformations that leave the input-output map of the CNN-WD unchanged.

a) The permutation of $\Theta_i's$.

Consider two hidden neurons h_r and h_p where r and p indicate the position of the nodes. Relabel these two nodes as h_p and h_r. Also, relabel the corresponding weights α_p and α_r , W_p and W_r then, the corresponding output, $\tilde{Y}\ (X: \theta, n)$ remains unchanged.

b) The symmetry of $\psi(x)$.

This implies that:

$$\psi(x) = -\psi(-x) \quad (21)$$

Select a neuron h_p and negate W_p as well as α_p. Then, the network map does not change.

This further indicates that $(\alpha_0, \Theta_1, \dots, \Theta_i, \dots, \Theta_H)$ and $(\alpha_0, \Theta_1, \dots, -\Theta_i, \dots, \Theta_H)$ have a similar input-output map. From these transformations, 2^H distinct models with similar outputs are yielded [13, 14]. This paper further adopted the following proposition.

Proposition: Let all the transformations be λ. Then, similar to [13], the transformations are characterized as being a composite function of $(\lambda_1, \dots, \lambda_H)$ where:

$$\lambda_1(\alpha_0, \Theta_1, \dots, \Theta_H) = (\alpha_0, -\Theta_1, \dots, \Theta_i, \dots, \Theta_H)$$

and

$$\lambda_i(\alpha_0, \Theta_1, \dots, \Theta_i, \dots, \Theta_H) = (\alpha_0, \Theta_i, \Theta_2, \dots, \Theta_{\{i-1\}}, \Theta_1, \Theta_{\{i+1\}}, \dots, \Theta_H) \quad (22)$$

This paper further considered the following theorem:

Theorem 1: Assume that the activation function, the model's overall output and model (10), and further that the activation function is continuous satisfying condition A of [13]. Then, θ is identifiable up to the family of transformations generated by equation (22).

3.3. Model Comparison

The forecasting performance of CNN-WD is compared to that of the CNN model and the root mean square error (RMSE) and mean absolute error (MAE) is used as the model evaluation metrics. The RMSE calculation formula is as shown below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)} \quad (23)$$

Where $\hat{y}_i$ is the predicted value and y_i is the actual value.

The MAE is given by equation (24) below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (24)$$

Then, the closer the RMSE and MAE values are to zero, the greater the model forecasting performance [15].

3.4. Forecasting Application

This paper takes tea prices in Kenya and forecasts the next month's average price using the CNN-WD. The dataset was obtained from indexmundi website. A one-plus ahead forecast was carried out using the formula shown below:

$$\tilde{Y}_{n+1} = \psi(\sum_{n=1}^{N}\sum_{h=1}^{H_{L-1}}(\alpha_h^L \times \tilde{Y}\ (\boldsymbol{X}: \theta, n)^{L-1}) + b^L) \quad (25)$$

From the above formula, $\tilde{Y}_{n+1}$ is the one-step ahead forecast, $\tilde{Y}\ (X: \theta, n)^{L-1}$ the last neural network output value, α_h^L the network weights, ψ the Relu activation function and b^L the bias.

CNN-WD training and forecasting: This process is as highlighted in figure 1 below.

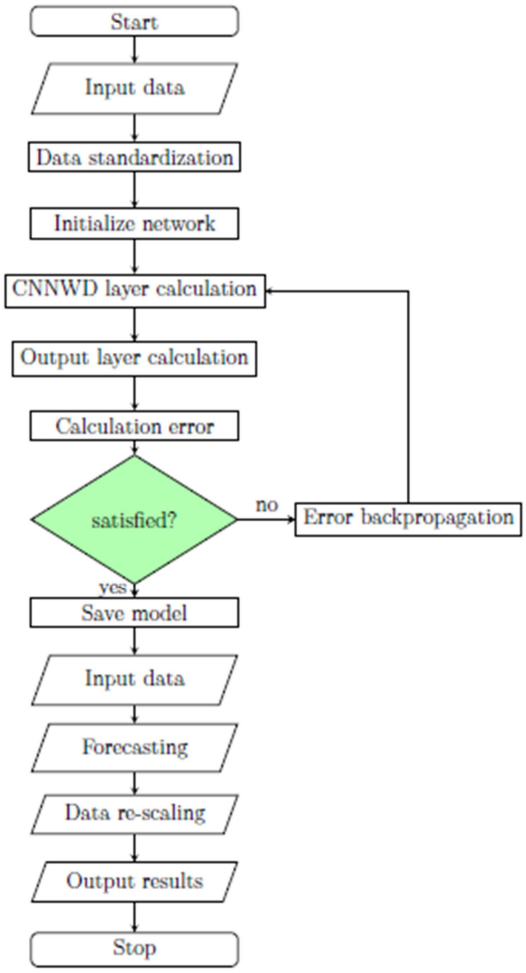

Figure 1. *CNN-WD training and forecasting process.*

The following are the major steps:

1) Input the exchange rate dataset.
2) Scaling: Scale the data to a scale of 0-1 using the min-max scaler, using the following formula:

$$y_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{26}$$

3) The weights and biases of the network are then initialized.
4) The input data set is then passed through the following layers: Convolution, average pooling, flatten and dropout layers. In these layers, effective data features are extracted and layer outputs are obtained.
5) From the dropout layer, the flattened output enters the fully connected layer and a single output is obtained.
6) The error is then computed by comparing the actual and predicted values.
7) If the error value is below the required threshold, the model is saved else the process enters process eight.
8) The weights and biases are updated using the backpropagation process until the error falls below the set threshold.
9) The updated model is then saved and used to forecast the next day's closing exchange rate.
10) The data needed for forecasting is then fed into the model.
11) The forecast value is finally obtained.
12) The output value is rescaled back to the input data original scale.
13) The result is recorded.

4. Results

To demonstrate the effectiveness of dropout regularization, we compared CNN-WD with CNN under the same operating environment using the same test and training set data. The running environment under which the experiments were carried out is of intel core i3-6100U, CPU 2.30 GHz, 8.00 GBs of RAM, 500GBs of hard disk and windows 10 pro. According to the influence factor, the adjusted closing price, the next day's closing price is predicted. The data used is as described below:

Table 1. *Data representation.*

Date	Opening Rate	Highest Rate	Lowest Rate	Closing Rate	Adjusted Closing Rate
2003-12-01	75.501999	75.980003	75.487000	75.699997	75.699997
2003-12-02	75.164001	76.433998	75.164001	75.580002	75.580002
2003-12-03	76.135002	76.397003	75.839996	76.050003	76.050003
2003-12-04	76.037003	76.300003	75.577003	76.300003	76.300003
2003-12-05	75888000	76.039001	75.778000	75.900003	75.900002

In this analysis, the Kenya shilling versus US dollar exchange rates is selected as the experiment data. Daily prices of 6528 trading days from December 1, 2003, to October 15, 2021, are extracted from Yahoo Finance website. The data contains six items, namely, Opening price, highest price, lowest price, closing price and adjusted closing price. The first five rows of the data are as shown in table 1.

4.1. Implementation of the CNN-WD

The CNN-WD is implemented using the parameters shown in table 2. Based on these parameters, the procedure is as follows:

Table 2. *Parameter setting of CNN-WD.*

Parameter	Value
Convolution layer filters	32
Convolution layer kernel size	1
Convolution layer Stride	1
Padding	Same
Pooling layer pool size	1
Pooling layer stride	1
Activation function	Rectified linear unit (Relu)
Time step	21
Dropout rate	0.5
Optimizer	Adaptive moment estimation (Adam)
Loss function	Mean square error (MSE)
Epoch	100

The train set data, which is the first input, is converted to a three-dimensional vector, which is (None, 21, 1) from which 21 is data from the past 21 days and 1 the data feature utilized in the input phase. The data first enters the convolution layer where key features are extracted and the output obtained from this layer is (None, 21, 32). From this output, 32 is the size of filters in the convolutional layer. This output then serves as the input for the next layer, which is the average pooling layer, and (None, 21, 32) is obtained as this layer's output. The output then proceeds to the flatten layer where it is flattened to obtain (None, 672). This flattened output then enters the fully connected layer and (None, 1) is obtained as the net's final output. The CNN-WD structure is shown in figure 2.

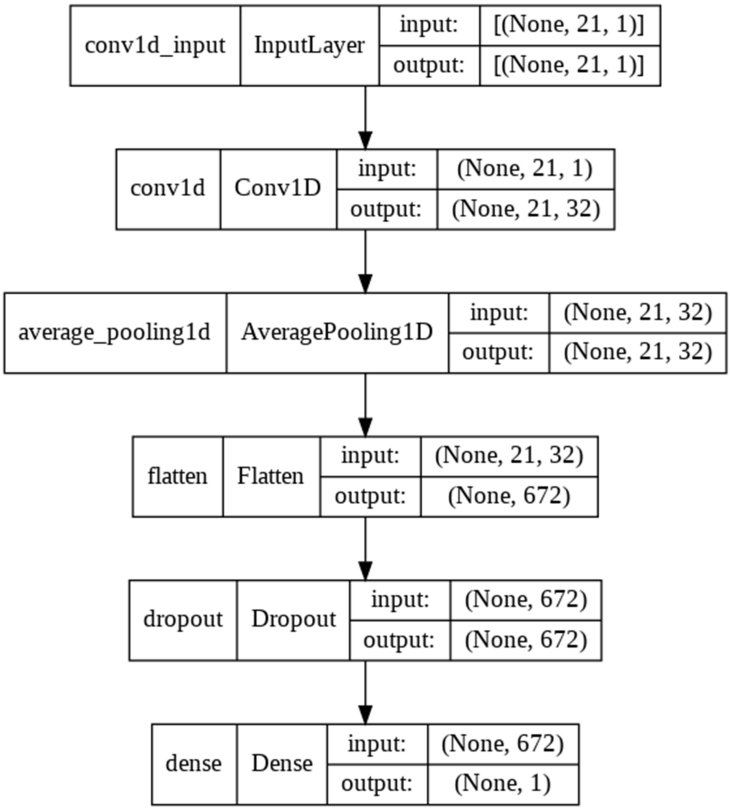

Figure 2. *The model structure of CNN-WD.*

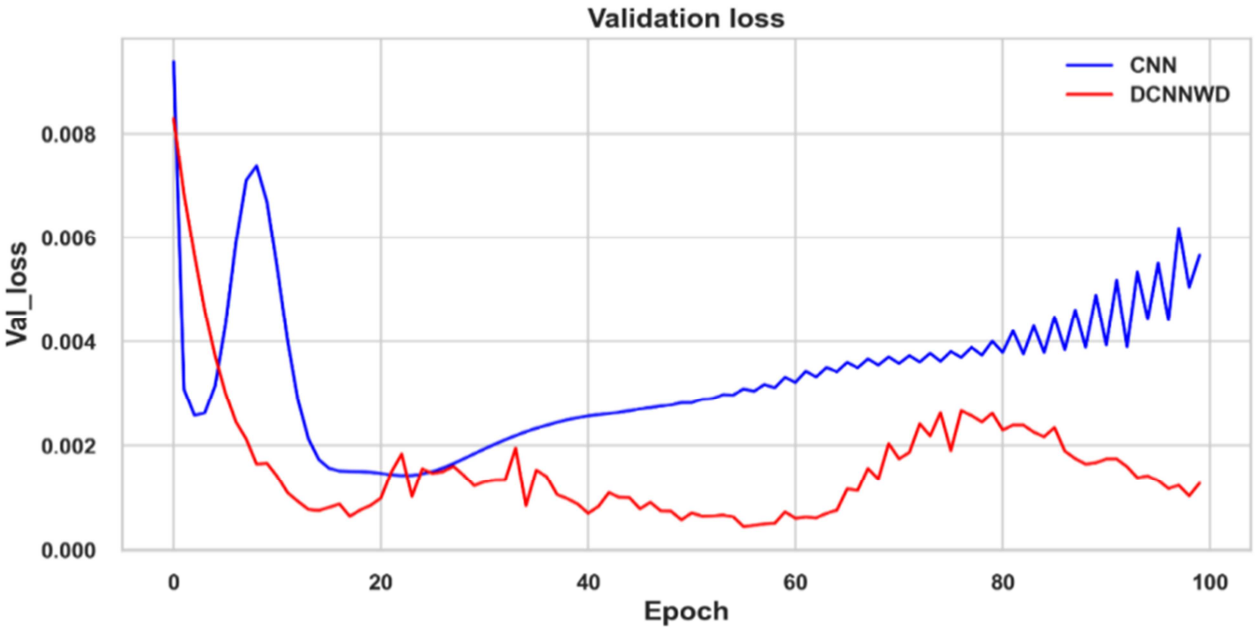

Figure 3. *Comparison of validation loss for CNN and CNN-WD.*

4.2. Model Data Overfitting

After the two models are trained, they are used to predict the unseen train dataset. This paper then investigates model overfitting by making use of the validation loss as shown in figure 3. From this, it is evident that the CNN model overfits the training data as its loss decreases and increases between epoch 0−20 and then increases steadily from epoch 20 to epoch 100. The loss of the CNNWD on the other hand decreases steadily from epoch 80 towards zero. It is evident that CNN is able to memorize the training data and thus overfits the data.

4.3. Model Performance Comparison

As demonstrated in figures 4 and 5, CNNWD and CNN are used to predict the unseen testing data, and the predicted values are compared to the actual values, CNN being the reference-predicting model.

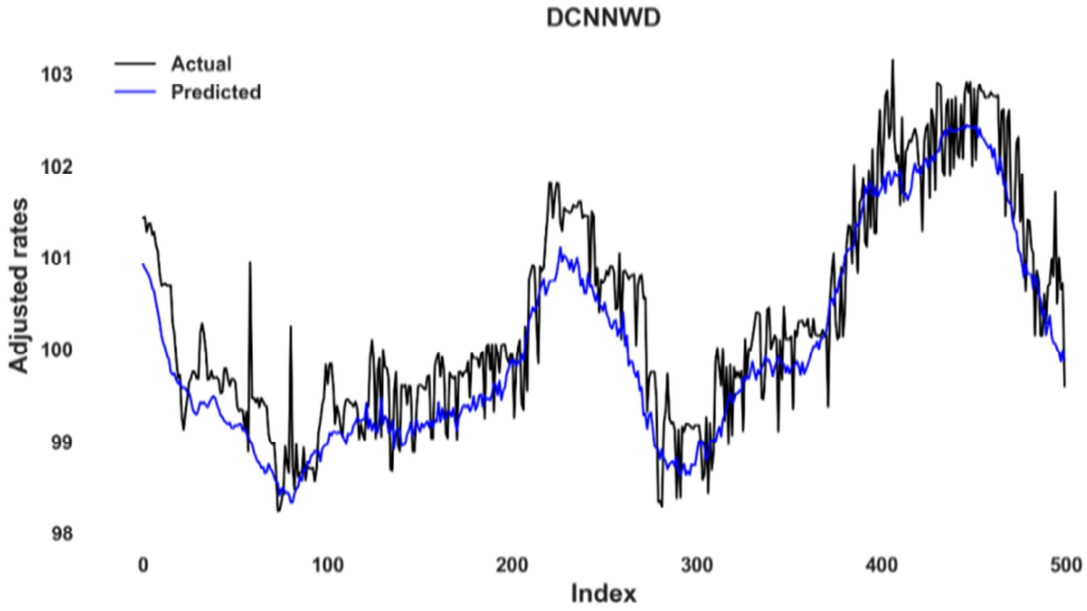

Figure 4. *Comparison of actual and predicted values of CNN-WD.*

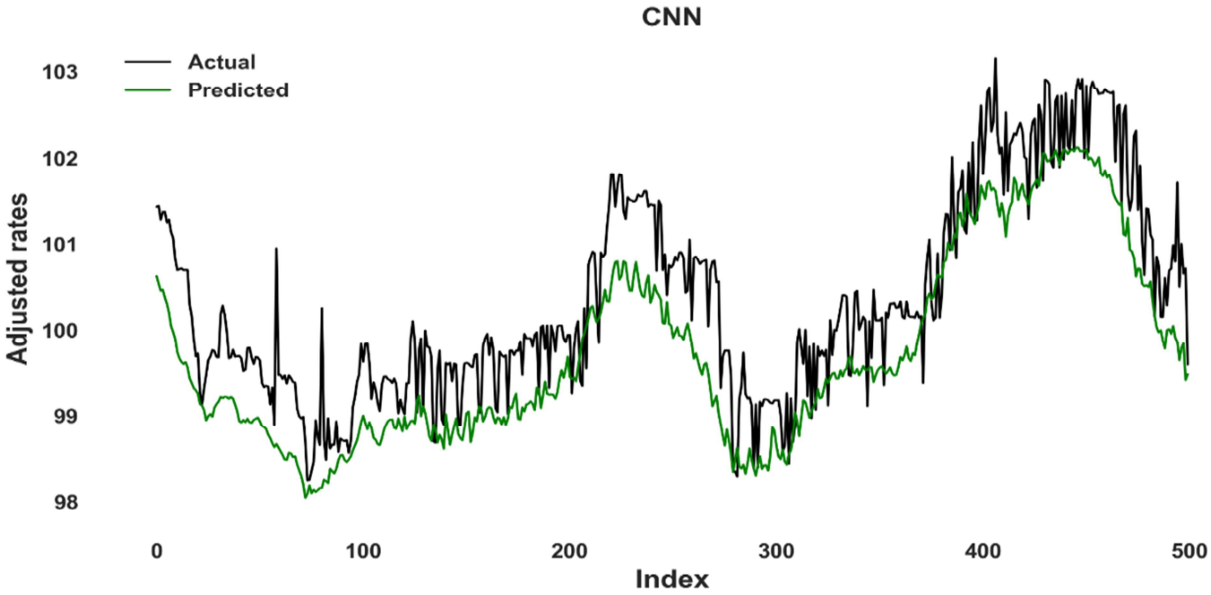

Figure 5. *CNN's actual and predicted values comparison.*

The predicted and actual values in figure 4 are close to each other. However, those in figure 5 are considerably apart, as shown in the above figures. This indicates that the CNN-WD has a better prediction performance when compared to the CNN model.

The performance of the two methods is further shown in table 3 below:

Table 3. *Evaluation metrics.*

Model	RMSE	MAE
CNN	1.11	0.93
CNN-WD	0.89	0.7

From table 3 above and figure 6 below, the RMSE and MAE of CNN are the largest at 1.11 and 0.93 respectively. On the other hand, the RMSE and MAE of CNNWD are the smallest at 0.89 and 0.7.

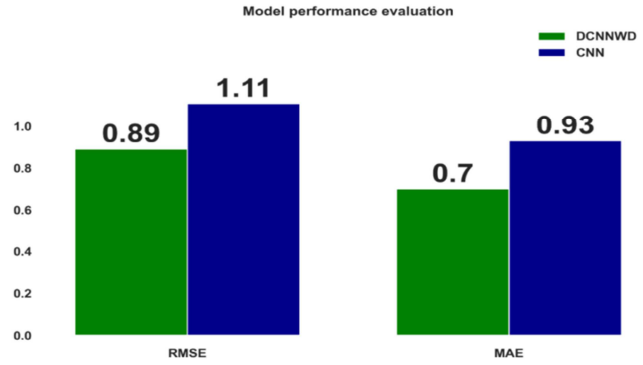

Figure 6. *Comparison of RMSE and MAE.*

Putting training waiting time into consideration, the CNN method trains for a longer period than CNN-WD. From figure 7, it was observed that the CNN method takes 260 seconds to train which is 40 seconds greater than the time taken by CNN-WD to train.

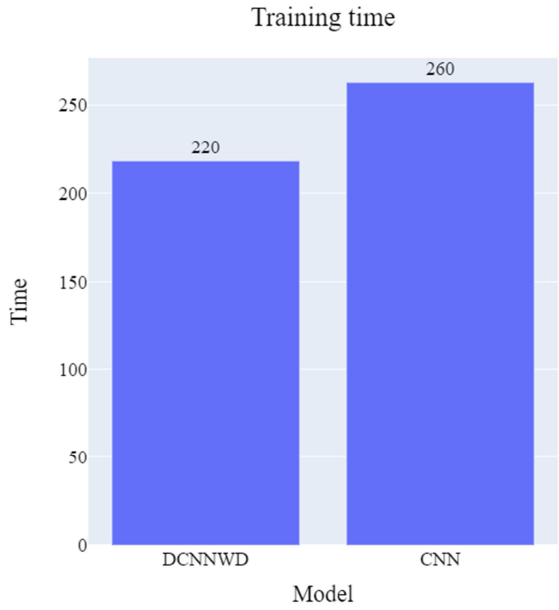

Figure 7. *Comparison of training time.*

4.4. Forecasting

As an application, this study takes the relevant Kenyan tea price data obtained from indexmundi.com. The dataset was partitioned in the ratio of 80 per cent to 20 percent for training and test set. The next month's average tea price was forecasted using the CNN-WD as shown in figure 8 below:

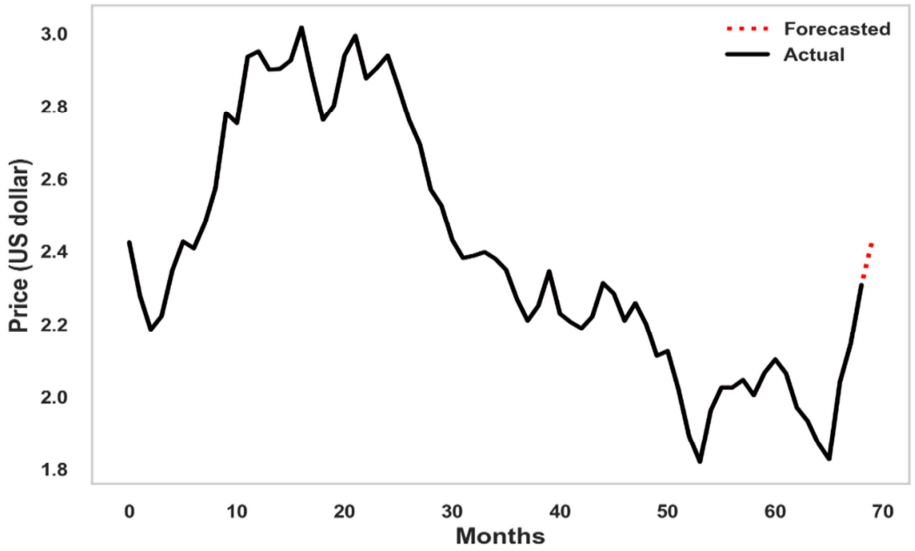

Figure 8. *Actual and forecasted tea prices.*

5. Discussion of Results

After using the processed training set data to train the CNN-WD and CNN methods, the validation loss for both methods was compared as shown in figure 3. The CNN validation loss varies between epoch 0 and 20 and finally increases from epoch 20 to 100. This indicates that the CNN method does overfit the training data. Adding the dropout regularization layer to the method, the validation loss was seen to decrease steadily towards 0 indicating that the model now does not overfit the data.

The methods were then used to predict the test data and the actual values compared with the predicted values as shown in figures 4 and 5. Between the two methods, CNN-WD had the highest broken line fitting which almost coincided with each other as compared to that of CNN.

According to the actual and predicted values, the evaluation

metrics were calculated and as shown in table 3 and figure 6, the RMSE and MAE of CNN-WD were the smallest when compared to that of the CNN method indicating that the CNN-WD had the better forecasting performance [15]. Comparing these evaluation metrics showed that adding a dropout regularization layer to the CNN method improves its results significantly. This was evidenced by the decrease of the RMSE by 0.22 and MAE by 0.23.

Putting waiting time into consideration, this study compared the training time for both the CNN and CNN-WD methods as shown in figure 7. The results indicated that CNN trains for a longer period than the CNN-WD method making the CNN-WD the better method [16]. This further proved that dropout regularization has a positive significant effect on the CNN method's overall performance.

Therefore, the CNN-WD method proposed in this study is superior to CNN as it is evident that adding a dropout regularization layer improves its performance in terms of the error value, fitting degree and training time significantly.

6. Conclusions and Recommendations

According to the chronological characteristics of exchange rate data, this study proposes a CNN-WD to predict the exchange rate of the next day. This method uses the adjusted closing exchange rate values, making use of the time sequence characteristics of the exchange rate data. CNN is used to detect the data pattern, extract features of the input data and predict the adjusted closing exchange rates. Dropout regularization is used to help prevent the CNN model from overfitting. This study takes the Kenya Vs US dollar exchange rate data from Yahoo finance as its experimental base. The results show that CNN-WD has the highest forecasting accuracy and performance as compared to CNN. The MAE and RMSE of CNN-WD are the smallest of the two methods.

CNN-WD is, therefore, suitable for the forecasting of exchange rates and can provide a relevant reference for investors to maximize investment returns and minimize risk for traders.

This study also takes the relevant Kenyan tea price data and forecasts the next month's average price as its application. The CNN-WD can therefore be used to predict tea prices and this can form a good planning base for traders and tea industry stakeholders.

However, this paper only considers one model regularization technique and one model for the comparative study. To progress with this research, it is of vital importance that other model regularization techniques as well as different model combinations be employed to further improve the performance of the proposed method. Also, other neuron selecting techniques other than the Schwartz Information criterion (SIC) adopted by this paper should be used to help optimize the number of hidden neurons so as to circumvent the chances of having a redundant model. The proposed method should also be used to investigate diverse domains such as pattern and speech recognition, as well as image classification.

References

[1] D. Erdemlioglu, S. Laurent and C. J. Neely, "Econometric modeling of exchange rate volatility and jumps," in Handbook of Research Methods and Applications in Empirical Finance, Edward Elgar Publishing, 2013.

[2] A. G. Halunga and C. D. Orme, "First-order asymptotic theory for parametric misspecification tests of GARCH models," Econometric Theory, vol. 25, p. 364–410, 2009.

[3] B. Dickson, "What are convolutional neural networks (CNN)," URL: https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets, 2020.

[4] W. Lu, J. Li, Y. Li, A. Sun and J. Wang, "A CNN-LSTM-based model to forecast stock prices," Complexity, vol. 2020, 2020.

[5] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," Econometrica: Journal of the econometric society, p. 987–1007, 1982.

[6] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," Journal of econometrics, vol. 31, p. 307–327, 1986.

[7] R. Uppal and T. Wang, "Model misspecification and underdiversification," The Journal of Finance, vol. 58, p. 2465–2486, 2003.

[8] W. Kristjanpoller, A. Fadic and M. C. Minutolo, "Volatility forecast using hybrid neural network models," Expert Systems with Applications, vol. 41, p. 2437–2442, 2014.

[9] S. Galeshchuk, "Neural networks performance in exchange rate prediction," Neurocomputing, vol. 172, p. 446–452, 2016.

[10] A. Borovykh, S. Bohte and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," arXiv preprint arXiv: 1703.04691, 2017.

[11] I. E. Livieris, S. Stavroyiannis, E. Pintelas, T. Kotsilieris and P. Pintelas, "A dropout weight-constrained recurrent neural network model for forecasting the price of major cryptocurrencies and CCi30 index," Evolving Systems, p. 1–16, 2021.

[12] A. W. Gichuhi, "Nonparametric changepoint analysis for bernoulli random variables based on neural networks," 2008.

[13] J. G. Hwang and A. A. Ding, "Prediction intervals for artificial neural networks," Journal of the American Statistical Association, vol. 92, p. 748–757, 1997.

[14] H. J. Sussmann, "Uniqueness of the weights for minimal feedforward nets with a given input-output map," Neural networks, vol. 5, p. 589–593, 1992.

[15] G. Brassington, "Mean absolute error and root mean square error: which is the better metric for assessing model performance?," in *EGU General Assembly Conference Abstracts*, 2017.

[16] K. Das and R. N. Behera, "A survey on machine learning: concept, algorithms and applications," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 5, p. 1301–1309, 2017.