International Journal of Data Science and Analysis
2026; 12(5): 129-138
http://www.sciencepublishinggroup.com/j/ijdsa
doi: 10.11648/j.ijdsa.20261205.13

The Impact of Varying Window Sizes on the Performance of a Hybrid CNN-LSTM Model A Case of Forecasting USD VS KES

Crispus Waweru Mwangi[1, *], Martin Mutwiri Kithinji[1, 2], Mutua Kilai[1]

[1]Department of Mathematics,Kirinyaga University, Kerugoya, Kenya
[2]Department of Mathematics, Pan African Insitute of Basic Science, Technology and Innovation, Nairobi, Kenya

Email address:

wawerucrispus77@gmail.com (Crispus Waweru Mwangi), mkithinji@kyu.ac.ke (Martin Mutwiri Kithinji), kmutua@kyc.ac.ke (Mutua Kilai)

To cite this article:

Crispus Waweru Mwangi, Martin Mutwiri Kithinji, Mutua Kilai. (2026). The Impact of Varying Window Sizes on the Performance of a Hybrid CNN-LSTM Model A Case of Forecasting USD VS KES. *International Journal of Data Science and Analysis, 12*(5), 129-138. https://doi.org/10.11648/j.ijdsa.20261205.13

Received: 18 August 2026; **Accepted:** 18 August 2026 **Published:** 28 September 2026

Abstract: Financial time series are characterized by non-linearity, inherent volatility, and changing temporal patterns making accurate predictions very challenging. This study deploys a CNN-LSTM hybrid model to predict the USD/KES exchange rate giving particular emphasis to the effect the size of the historical input window has on the hybrid model forecasting performance. The study relied on Daily USD/KES exchange-rate data obtained from the Central Bank of Kenya to train, evaluate and validate the hybrid model. This data was prepossessed and transformed into sequential inputs using different window configurations, allowing the models to learn from varying amounts of historical information. The study combined a CNN-based local pattern extraction with an LSTM-based temporal dependency learning model to develop a hybrid CNN-LSTM model. The hybrid's model performance was then subjected to a comparison with standalone CNN and LSTM model and evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) The hybrid CNN-LSTM prediction results indicated that the model was more effective in prediction among the evaluated architecture. The choice of the input window sizes affected the model's ability to learn and capture trends in the time series data. The results from the rolling-window experiments demonstrated that shorter and moderately sized historical windows can provide a more responsive representation of recent exchange-rate movements than an expanding window, particularly during periods of pronounced market variation. These findings highlight the importance of considering both model architecture and input-window design when developing deep-learning approaches for USD/KES exchange-rate forecasting. The study provides an empirical basis for selecting appropriate historical window sizes when applying hybrid deep-learning models to volatile financial time series in the Kenyan foreign-exchange market.

Keywords: USD/KES Exchange Rate, Exchange-rate Forecasting, CNN-LSTM, Deep Learning, Time-series Forecasting, Rolling Window, Window Size, Financial Time Series

1. Introduction

Financial markets across the world have faced critical problems and complexities due to their volatility, non-linearity and dynamism therefore making accurate predictions a challenge[1, 2]. The Kenyan currency market, in particular, has faced a growing need for greater accuracy in predicting exchange rate movements. This need has largely been as a result of currency fluctuations, which causes significant impact to trade and economic stability. The resultant effect is felt by financial institutions, policymakers, and investors. The US/KES is the main exchange currency pair in Kenya today. Large volumes trade daily because the country relies heavily on imports and exports which largely involve the two currencies. There is therefore a critical need for accurate forecasting of USD/KES exchange rates[3, 4]. Researchers have in the past deployed various methods in an attempt to make dependable predictions. However, very little success has been achieved because traditional

methods have not guaranteed dependable predictions.Zang (2003) describes ARIMA models as linear forecasting models and proposes neural networks which have the capability to model complex non-linear relationships which are often exhibited in time-series data[5]. Deep learning models have however, offered promising alternatives. CNN models have the ability detect local patterns in sequences [6], while LSTM models excel in capturing long-term dependencies in time series data[7]. When used as standalone models, however, they have limitations: CNN models tend to ignore sequential dependencies while LSTM models tend to overlook important local variations in the financial time series data. Combining CNN and LSTM models has attracted attention in financial prediction because local temporal features in data are identified by convolutional layers while sequential dependencies are extracted by LSTM networks [8]. Even though advancements in financial predictions have been witnessed, the performance of deep learning models is highly dependent on the structure of input data, particularly the choice of window size used to construct input sequences. The size of the window plays a crucial role in determining to what extent of the data the model will interact with and this afterwards influences its ability to identify relevant time-based patterns[9].

2. Literature Review

2.1. Related Work

Yu and Wang conducted research on advancements in financial machine learning and discovered that it presented a range of advanced techniques designed to boost predictive accuracy in financial markets.The study highlighted the importance of feature engineering, model selection, and validation methods in improving the reliability and precision of machine learning models.A significant contribution was an introduction of fractional differentiation, a technique that renders financial time series stationary while maintaining memory that is crucial for effective model training. The research also took a closer look at ensemble methods and advanced techniques like meta-labeling to boost the predictive power of trading strategies.However, machine learning models often struggle with identifying long-term dependencies and the non-linear relationships that are typical in financial time series data. Additionally, these ML models usually require extensive feature engineering and can be prone to overfitting, especially in highly volatile markets.To tackle these issues, deep learning techniques offer a range of advantages.Yu and Wang (2005)[11] evaluated the Feed Forward Neural Networks (FFNN) trained using the back propagation technique in a bid to forecast exchange rates and realized that they outperformed methods-that had been used earlier such as ARIMA-in predicting short-term currency movements.Despite providing early evidence of the potential of neural networks in capturing non-linear patterns in foreign exchange markets, the study was limited to short-term horizons and lacked macroeconomic input features for better long-term forecasting.

Machine learning models, including support vector machines (SVMs), multi-layer perceptrons (MLPs), and ensemble techniques were compared and ensemble methods were found to produce better stability and accuracy predicting trends in stock and foreign exchange rates highlighting the importance of hybrid models. However, these hybrid models were limited to the exclusion of deep learning techniques such as CNN and LTSM [12]. GARCH models are frequently used to predict volatility in financial markets. They account for time-varying volatility (conditional heteroscedasticity) which is characteristic of Foreign exchange markets [13]. However, GARCH models assume a constant mean, and they tend to fall short in modelling the complex, non-linear dependencies in financial time series. A survey conducted to predict foreign exchange and stock price using DL models, explores various deep learning models applied to foreign exchange and stock price prediction. The study analyses deep learning techniques, including CNN, LSTM networks, Deep Neural Networks (DNNs),RNNs, and Reinforcement Learning. The survey classifies these models based on their methodologies and performance metrics such as RMSE, MAPE, MAE, MSE, accuracy, Sharpe ratio, and return rate.The findings indicate that combining LSTM with other models, such as DNN, shows promising results [14].

2.2. Research Gap

Research history is rich in prior studies that have demonstrated the effectiveness of hybrid CNN-LSTM models in financial time-series forecasting.Majority of these studies however, have focused primarily on the architecture of the model and their ability to predict accurately leaving limited attention to the role of input representation, particularly window size selection.

Researchers have in the past adopted fixed or chosen window sizes arbitrarily failing to evaluate systematically how the model performance is affected by changing the window size.

Most of the research studies have also focused their attention to major currency pairs such as USD/EUR or USD/GBP, with very little attention given to growing market currencies like the US/KES currency market.

3. Model Development and Architecture

3.1. CNN Model

The data used for this research was the historical foreign exchange rates between the Kenyan Shilling (KES) and the US Dollar (USD).The data was gathered from a trustworthy financial database-the official website of the Central Bank of Kenya. The dataset spans 11 years and included the daily foreign exchange rate of KES/USD from January 2014 to June 2025 which had the mean, buy, and sell rates. Preprocessing included:

1) Handling missing values using forward-fill imputation.

2) Normalizing data to the range [0,1] with MinMax scaling.
3) Constructing 30-day, 60 day and expanding window sequences to capture temporal dependencies for model input.

A CNN is a powerful algorithm known for its ability to distinguish between different types of inputs. It's made up of layers that include both convolutional and pooling layers, all organized in a stacked format.

3.1.1. Input Layer

The input layer is set up to take in historical foreign exchange data. This data will be organized as a 1-Dimensional Matrix, where each row represents a time step and each column shows the foreign exchange rates, capturing how these rates change over time.

3.1.2. Convolution Layer

The convolutional layer in a CNN plays a crucial role in picking up local patterns from the input data. It does this by using learnable filters, or kernels, that slide over the input, performing element-wise multiplications and summations to create a feature map.

The feature map is passed through the Rectified Linear Unit (ReLU) activation function, which adds a bit of non-linearity to the model.

ReLU works by turning negative values into zero while keeping the positive ones intact, leading to a rectified feature map.

In mathematical terms, the ReLU function is defined as:

$$\text{ReLU}\left(\sum_{i=1}^{n} w_i x_i + b\right) = \max\left(0, \sum_{i=1}^{n} w_i x_i + b\right) \quad (1)$$

where:

1) w_i are the kernel weights,
2) x_i are the input values,
3) b is the bias term, and
4) $\text{ReLU}(z) = \max(0, z)$ is the element-wise activation function.

Equation (1) will define a linear intergration of input values and kernel weights while assigning negative results to zero thus introducing non-linearity.

3.1.3. Pooling Layers

These layers play a crucial role in simplifying data and trimming down historical information while keeping the essential features intact, which helps reduce the chances of overfitting. For example, max pooling can be used to grab the highest value from a specific area, which might indicate a peak in exchange rates. Additionally, the weight sharing in the convolutional layer, combined with well-chosen pooling methods, gives the CNN some important invariance traits.

3.1.4. Fully Connected Layers

In this part, the study brings together all the traits gathered from earlier layers to make the predictions. CNNs are particularly good at forecasting foreign exchange trends because they can spot local patterns, like sudden spikes or drops. They also help reduce dimensionality and lighten the computational load through pooling, all while skillfully managing high-frequency fluctuations. After that, the feature maps are flattened into a single, long, continuous vector.

3.1.5. CNN Mathematical Formulations

(i).Convolution Operation

$$z_t^{(k)} = (x * w^{(k)})(t) + b^{(k)} = \sum_{i=0}^{m-1} w_i^{(k)} \cdot x_{t+i} + b^{(k)} \quad (2)$$

where:

1) $x \in \mathbb{R}^T$ is the input time series of length T
2) $w^{(k)} \in \mathbb{R}^m$ is the k^{th} convolutional filter
3) $b^{(k)}$ is the bias term for the k^{th} filter
4) $z_t^{(k)}$ is the output (feature map) of the k^{th} filter at position t
5) $*$ denotes the convolution operation
6) t denotes the current time index (position) where the convolution is computed, $t = 0, 1, 2, \ldots, T - m$

(ii). Activation Function (ReLU)

$$a_t^{(k)} = \text{ReLU}(z_t^{(k)}) = \max(0, z_t^{(k)}) \quad (3)$$

where:

1) $a_t^{(k)}$ is the activated feature map value after applying the ReLU function;
2) $z_t^{(k)}$ is the output of the convolution operation before activation.

(iii). Max Pooling

$$p_t = \max\left(a_t, a_{t+1}, ..., a_{t+P-1}\right) \quad (4)$$

where:

1) p_t is the pooled output at position t
2) P is the pooling size

Max pooling helps to down-sample the data and retain the most important features

(iv). Output

The resulting feature maps are passed on to either:

1) A fully connected (Dense) layer for classification which is the LSTM layer.]

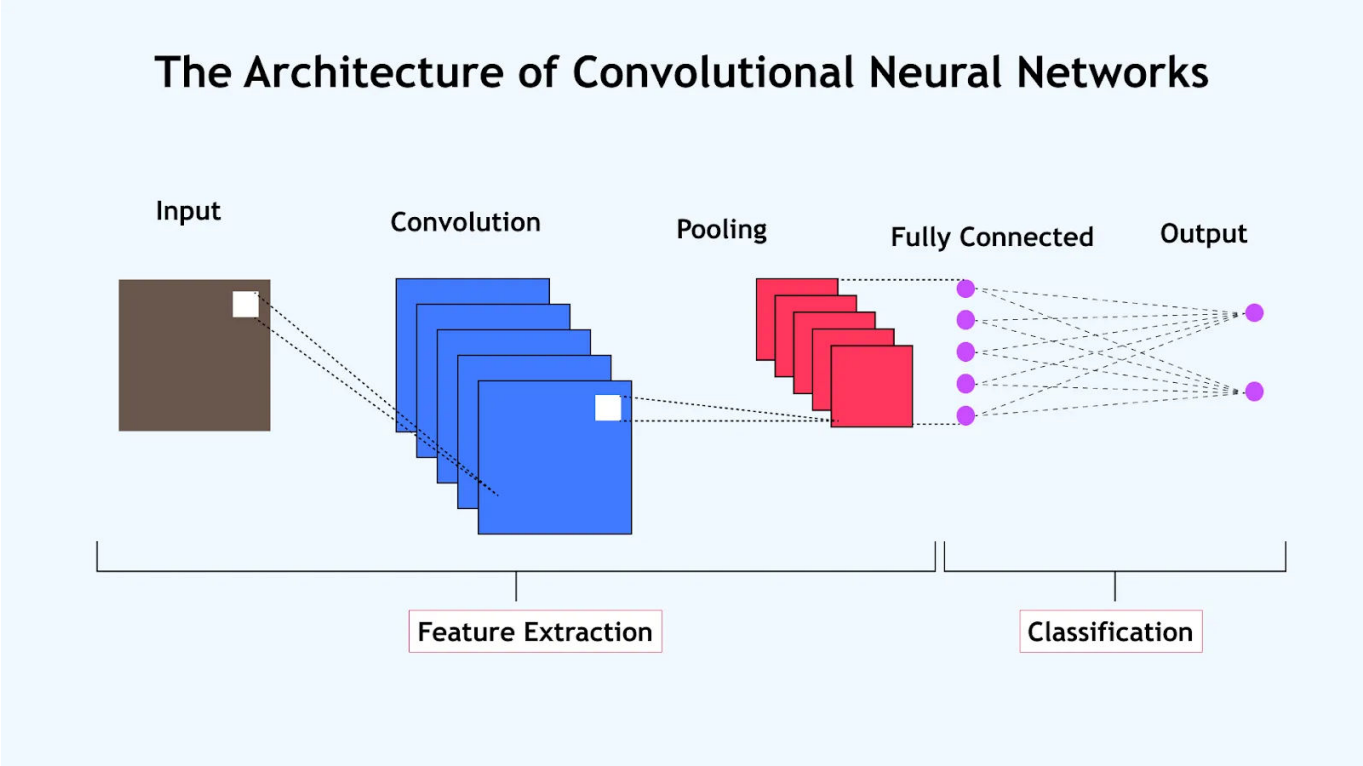

Figure 1. *A Diagrammatic representation of CNN Architecture.*

Figure 1 represents the architecture of the CNN layer:

3.2. LSTM Model

LSTM models tackle the problem of vanishing gradient descent by introducing a memory cell. This memory cell is designed to hold onto information for longer periods, making it a powerful tool in deep learning[15].

3.2.1. LSTM Cells

LSTM layers have three gates which control information movement in and out of the cells: the forget gate, the input gate and the output gate;

1) *Forget Gate (f_t)*: This gate chooses the information from the previous cell that ought to be forgotten or dropped.
2) *Input Gate (i_t)*: This gate chooses information to be added to the cell state.
3) *Output Gate (o_t)*: Controls what part of the cell state should be output as the hidden state.

Mathematical Equations

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{(Forget gate)} \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && \text{(Input gate)} \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) && \text{(Candidate cell state)} \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t && \text{(Updated cell state)} \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{(Output gate)} \\ h_t &= o_t \odot \tanh(C_t) && \text{(Updated hidden state)} \end{aligned} \tag{5}$$

where:

1) x_t is the input at time step t
2) h_{t-1} is the hidden state from the previous time step
3) f_t is the forget gate output at time step t
4) i_t is the input gate output at time step t
5) $\tilde{C}_t$ is the candidate cell state at time step t
6) C_t is the updated cell state at time step t
7) o_t is the output gate at time step t
8) h_t is the updated hidden state at time step t
9) W_f, W_i, W_C, W_o are weight matrices for the forget, input, candidate, and output gates respectively
10) b_f, b_i, b_C, b_o are the bias vectors for each gate
11) σ is the sigmoid activation function
12) $\tanh$ is the hyperbolic tangent activation function
13) $\odot$ denotes element-wise multiplication
14) $t = 1, 2, ..., T$ is the time index in the sequence of length T

3.2.2. CNN-LSTM Model Equations

(i). Input

Let:

$$X = [x_1, x_2, ..., x_T] \tag{6}$$

where x_t is the exchange rate at time t, and T is the total time steps in the input window.

(ii). CNN Feature Extraction (1D Convolution)

$$z_t^{(k)} = \sum_{i=0}^{m-1} w_i^{(k)} \cdot x_{t+i} + b^{(k)} \quad \text{for } t = 1, ..., T - m + 1 \tag{7}$$

$$a_t^{(k)} = \text{ReLU}(z_t^{(k)}) \tag{8}$$

$$p_t^{(k)} = \max(a_t^{(k)}, a_{t+1}^{(k)}, ..., a_{t+P-1}^{(k)}) \quad (9)$$

where:

1) $w^{(k)}$ is the k^{th} filter, $b^{(k)}$ is the bias
2) $z_t^{(k)}$ is the convolution output (feature map)
3) $a_t^{(k)}$ is the ReLU-activated output
4) $p_t^{(k)}$ is the result after max pooling (optional)

(iii). Sequence Reshaping for LSTM

The pooled feature maps (or activated maps) are reshaped into sequences:

$$\tilde{X}_t = [p_t^{(1)}, p_t^{(2)}, ..., p_t^{(K)}] \quad \text{for } t = 1, ..., T' \quad (10)$$

where K is the number of filters and T' is the reduced time dimension after convolution and pooling.

(iv). LSTM Processing

The reshaped sequence $\tilde{X}_t$ is passed to the LSTM:

$$f_t = \sigma(W_f \cdot [h_{t-1}, \tilde{x}_t] + b_f) \quad (11)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, \tilde{x}_t] + b_i) \quad (12)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, \tilde{x}_t] + b_C) \quad (13)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (14)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, \tilde{x}_t] + b_o) \quad (15)$$

(v). Output Layer

The final hidden state is passed through a dense layer for prediction:

$$\hat{y}_t = W_y \cdot h_t + b_y \quad (16)$$

where:

1) $\hat{y}_t$ is the predicted USD/KES exchange rate
2) W_y, b_y are weights and bias of the output (dense) layer

3.3. Training Procedure

1) Data split: 70% training, 15% validation, 15% test.
2) Loss function: Mean Squared Error (MSE) optimized using Adam.
3) Learning rate: tuned from 0.001 using grid search.
4) Early stopping: monitors validation loss to avoid overfitting.
5) Batch size: 32 sequences per batch.

3.4. Evaluation Metrics

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (17)$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (18)$$

where y_i is the actual exchange rate and $\hat{y}_i$ is the predicted value.

4. Results

4.1. Model Implementation and Evaluation

The CNN block that considers local patterns and the LSTM block that detects the local effects are combined in this section.Having split the data into training, validation and testing sets there was needed to reshape the input data (X train) in to a format that the CNN-LSTM model understands. LSTM accepted input in certain batch size and each sequence contained several time steps with one feature per time step.

4.1.1. Expanding Window Size

The model here was subjected to the entire available data up to a time just before the prediction date for July 1st 2025.This meant that the model was trained without a fixed length window. The data kept increasing as the training continued with each new day and thus this suggested that the training data size kept increasing. The prediction curve over time compared to the actual trend line indicated a very smooth, flat and failed to resonate with the sudden fluctuations in the data which was an indication that the model was not able to cope with the sudden shocks in the short term thus discrediting the choice of window configuration. This shape also exhibited RSME and MAE errors of 13.4455 and 11.6859 respectively which were very high errors an indication that prediction were most likely to be inaccurate.

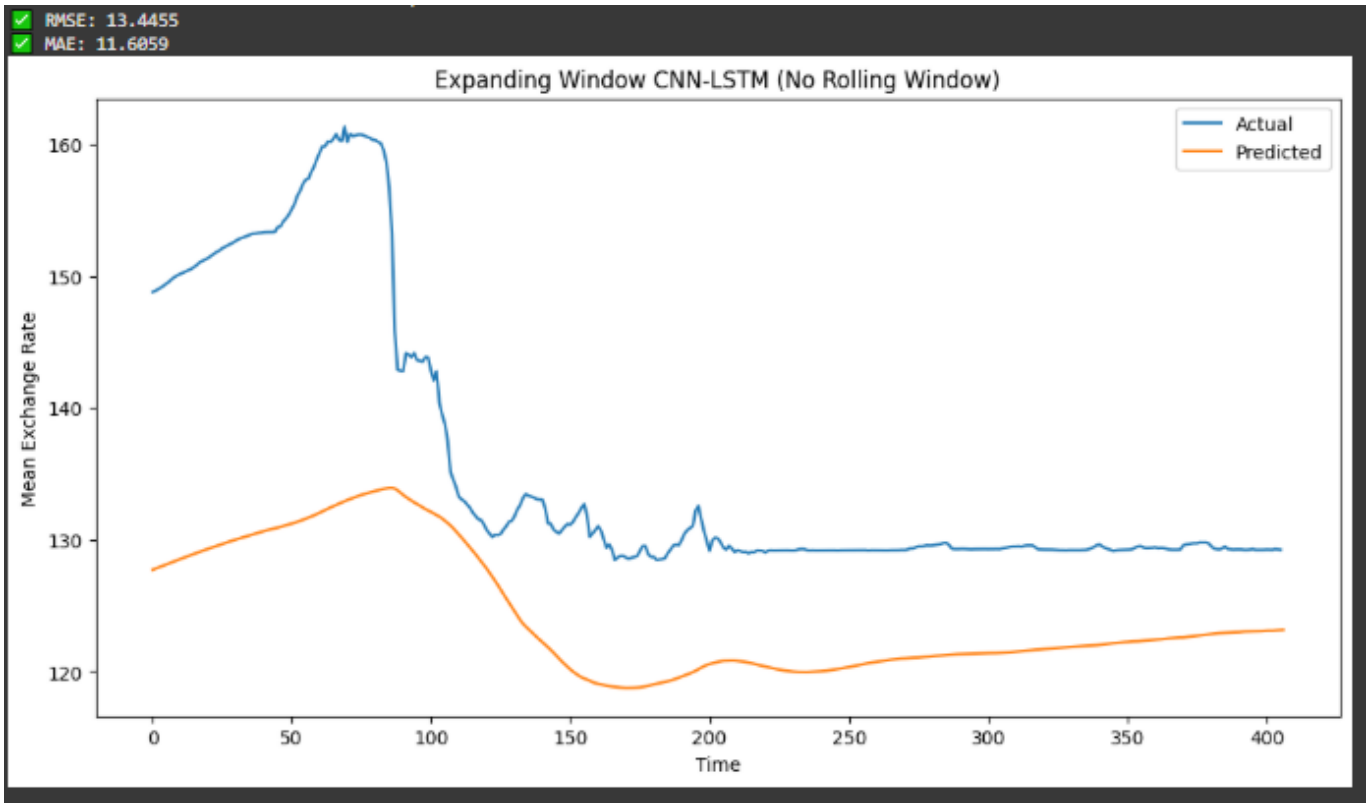

Figure 2. *Actual and predicted USD/KES exchange rates using the expanding window.*

4.1.2. 60-Day's Rolling Window

Historical data used the past 60 days was used in model training to predict the next day's exchange rate.The predicted curve was seen to follow the actual trend fairly well. The predicted curve also witnessed incidences when sudden movements- spikes or drops- were missed by the curve.This suggested that the window's shape was long enough to create a smoothing effect due to the longer window and hence the model became very less reactive to recent fluctuations. Evaluation errors were relatively sensible as well with a RSME of 2.3096 and an MAE of 1.4675 which meant that the model performed a bit better over all in reducing the error but a rather relatively higher average error.

4.1.3. 30-Day's Rolling Window

The study now conducted the third experiment using the most recent 30-day window to predict the next day's mean exchange rate.The prediction curve was seen to follow the actual trend the closest even during sharp movements insinuating that the model responded quickly and accurately. The input shape output RSME and MAE errors of 2.7248 and 1.3532 respectively.These errors were low compared to those of the expanding window input shape.

The table and graphs below visualizes the output of the evaluation metrics and prediction curves:

Table 1. *Model Performance Comparison and Window Size Tuning.*

Model Configuration	RMSE	MAE
Expanding Rolling Window	13.4455	11.6859
60-Day Rolling Window	2.3096	1.4675
30-Day Rolling Window	2.7248	1.3532

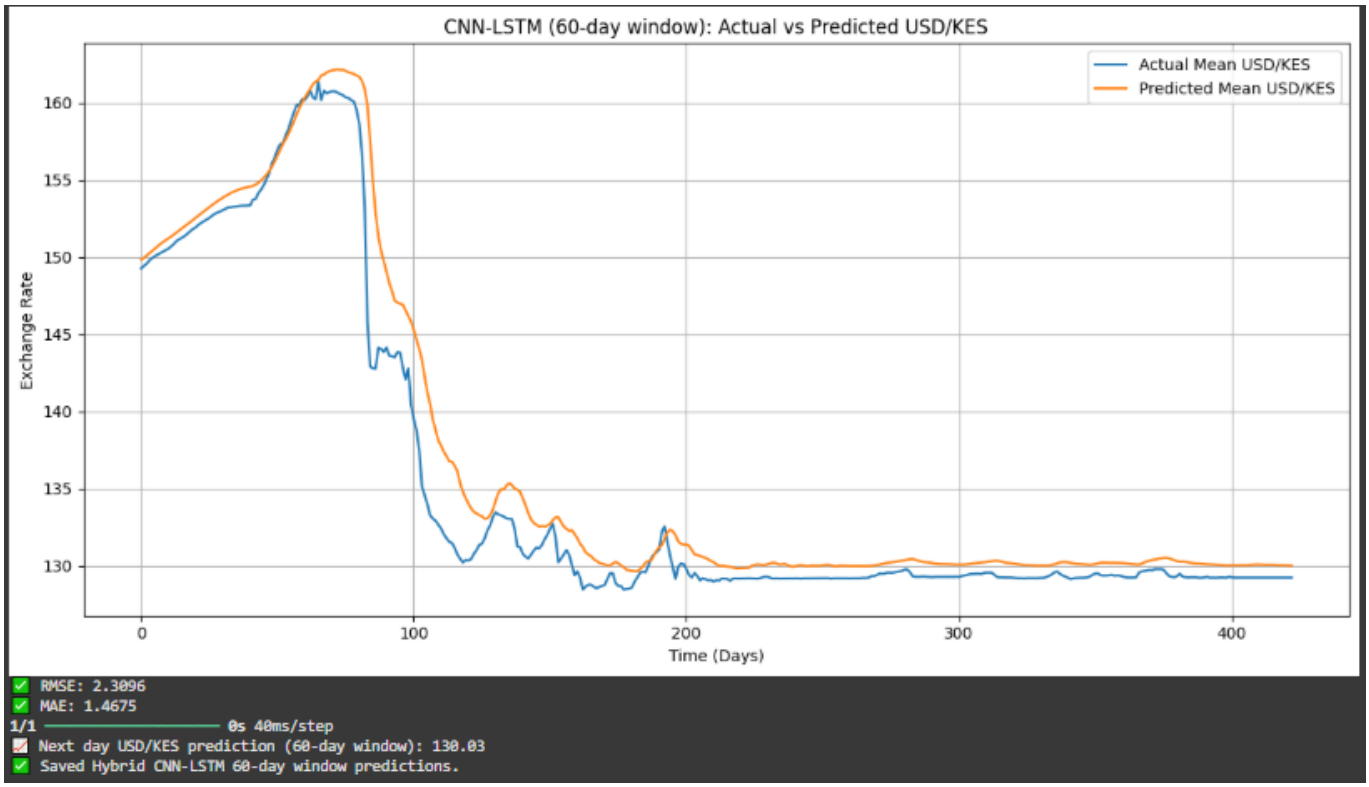

Figure 3. *CNN–LSTM forecasting performance using (a) 30-day and (b) 60-day historical windows.*

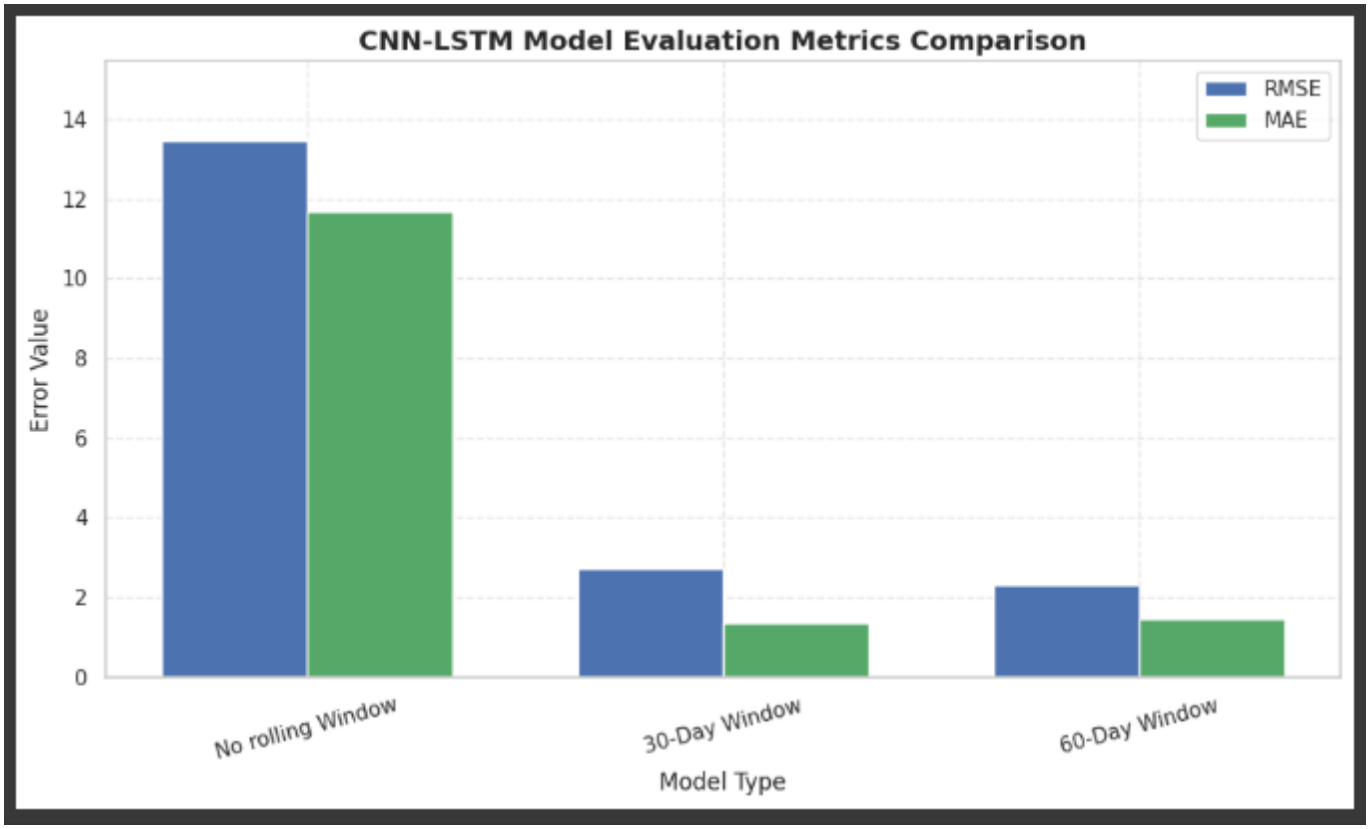

Figure 4. *Evaluation Metrics Visualization for window sizes.*

4.2. Forecasted Exchange Rate

The predicted USD/KES exchange rate for the next 30 day was:
30-Day Forecasts
Hybrid CNN-LSTM 30-Day Forecast:

Table 2. *Hybrid CNN–LSTM Forecast for the Next 30 Days of USD/KES Exchange Rate.*

No.	Date	Predicted	No.	Date	Predicted
1	2025-05-31	129.6165	16	2025-06-15	130.1831
2	2025-06-01	129.6565	17	2025-06-16	130.2246
3	2025-06-02	129.6875	18	2025-06-17	130.2720
4	2025-06-03	129.7402	19	2025-06-18	130.3019
5	2025-06-04	129.7693	20	2025-06-19	130.3501
6	2025-06-05	129.8059	21	2025-06-20	130.3843
7	2025-06-06	129.8382	22	2025-06-21	130.4104
8	2025-06-07	129.8925	23	2025-06-22	130.4411
9	2025-06-08	129.9289	24	2025-06-23	130.4823
10	2025-06-09	129.9762	25	2025-06-24	130.5285
11	2025-06-10	130.0126	26	2025-06-25	130.5645
12	2025-06-11	130.0515	27	2025-06-26	130.6043
13	2025-06-12	130.0630	28	2025-06-27	130.6482
14	2025-06-13	130.1056	29	2025-06-28	130.6911
15	2025-06-14	130.1370	30	2025-06-29	130.7353

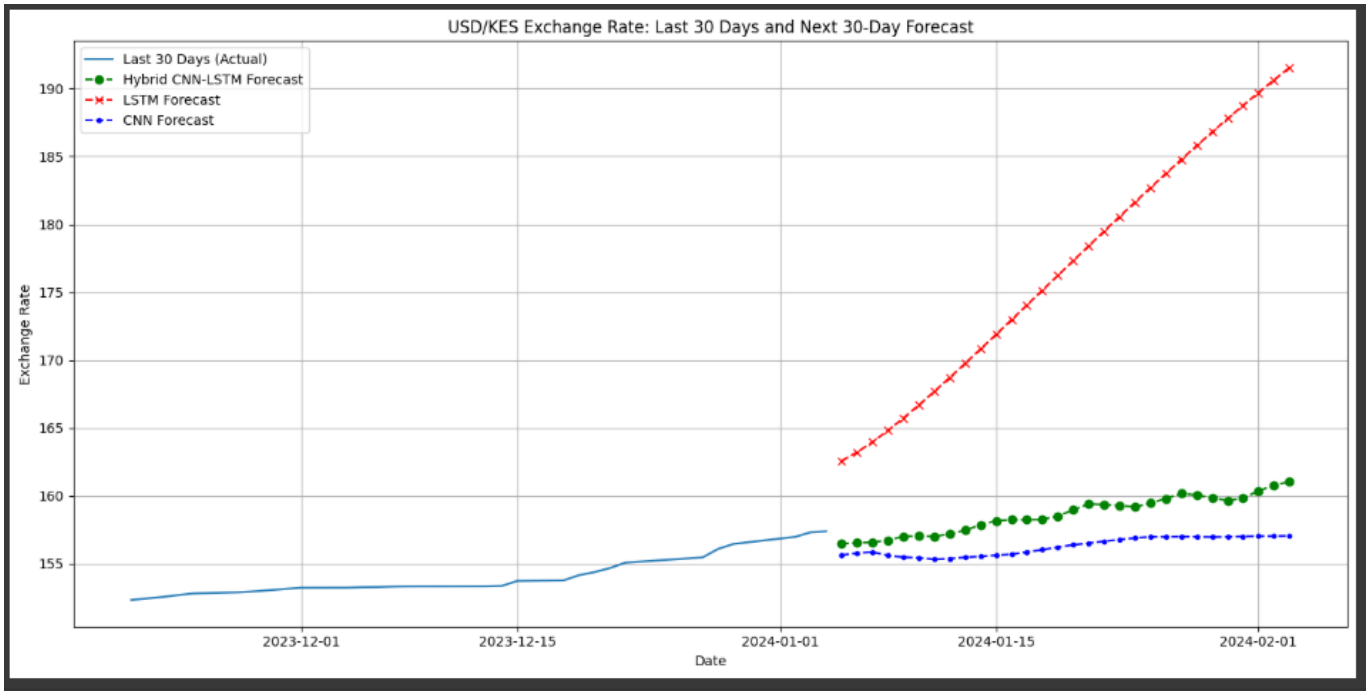

Figure 5. *Actual versus predicted CNN-LSTM Trend line.*

5. Discussion

The evalation metrics demonstrate that varying the window size in the input data is impacted the accuracy of predictions significantly using a hybrid CNN-LSTM model. RMSE and MAE values for the expanding window experiment indicated the poorest performance of 13.4455 and 11.6859 respectively. The prediction curve appeared to be overly smooth which was an indication that it struggled to capture short-term fluctuations.This was as a result of model interacting with excessive historical data which introduced noise and reduced the model's sensitivity to recent patterns. The hybrid model

is seen to prioritize long term averages at the expense of short term fluctuations which led to underfiting. The 60-day window reduced RMSE to 2.3096 and MAE to 1.4675 indicating the model was able to capture the general trend of the exchange rate.However, model still missed sudden spikes and drops suggesting that the longer window size introduced smoothing effects that limited the model's adaptability to volatile market conditions. The 30-day window size further reduced the MAE to 1.3532 although its RMSE of 2.7248 was slightly higher than that of the 60-day window. The prediction curve illustrated a superior alignment with the actual trend, particularly during sharp movements. This is an indication that the shorter window ensured the model captured recent patterns better and reacted more effectively to short-term volatility. This generally is an indication that recent observations carry more predictive value in the USD/KES exchange rate than long term historical data.

6. Conclusions

The study established that the selection of an optimal window size for the input data is as important as the model architecture in deep learning-based time-series forecasting.

ORCID

0009-0005-0275-4257 (Crispus Waweru Mwangi)
0000-0002-0449-3776 (Martin Mutwiri Kithinji)
0000-0001-6290-9728 (Mutua Kilai)

Abbreviations

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DL	Deep Learning
FFNN	Feed Forward Neural Network
KES	Kenyan Shilling
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
SMA	Simple Moving Average
USD	United States Dollar

Author Contributions

Crispus Waweru Mwangi: Conceptualization, Data curation, Methodology, Writing – original draft

Martin Mutwiri Kithinj: Formal Analysis, Software, Validation, Writing – review & editing

Mutua Kilai: Funding acquisition, Resources, Supervision, Writing – review & editing

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. Applied soft computing, 90, 106181.

[2] Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., ... & Li, J. (2022). A survey on machine learning models for financial time series forecasting. Neurocomputing, 512, 363-380.

[3] Musyoki, D., Pokhariyal, G. P., & Pundo, M. (2012). The impact of real exchange rate volatility on economic growth: Kenyan evidence. Business and Economic Horizons, 7(1), 59-75.

[4] Nyambariga, M. D. (2017). Effects of exchange rate volatility on imports and exports in Kenya. International Journal of Economics, 2(3), 71-84.

[5] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.

[6] Aryal, S., Nadarajah, D., Kasthurirathna, D., Rupasinghe, L., & Jayawardena, C. (2019, December). Comparative analysis of the application of Deep Learning techniques for Forex Rate prediction. In 2019 international conference on advancements in computing (ICAC) (pp. 329-333). IEEE.

[7] Dautel, A. J., Härdle, W. K., Lessmann, S., & Seow, H. V. (2020). Forex exchange rate forecasting using deep recurrent neural networks. Digital Finance, 2(1), 69-96.

[8] Biswas, A., Uday, I. A., Rahat, K. M., Akter, M. S., & Mahdy, M. R. C. (2023). Forecasting the United State Dollar (USD)/Bangladeshi Taka (BDT) exchange rate with deep learning models: Inclusion of macroeconomic factors influencing the currency exchange rates. PloS one, 18(2), e0279602.

[9] Azlan, A., Yusof, Y., & Mohsin, M. F. M. (2019). Determining the impact of window length on time series forecasting using deep learning. International Journal of Advanced Computer Research, 9(44), 260-267.

[10] Yu, L., Wang, S., & Lai, K. K. (2005). Adaptive smoothing neural networks in foreign exchange rate forecasting. In International conference on computational science (pp. 523530).

[11] de Prado, M. L. (2018). Advances in nancial machine learning. Hoboken, NJ: John Wiley & Sons.

[12] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42(1), 259268.

[13] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307327.

[14] Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of Forex and stock price prediction using deep learning. Applied System Innovation, 4(1), 9.

[15] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.