TẠP CHÍ KHOA HỌC VÀ CÔNG NGHỆ, Trường Đại học Khoa học, ĐH Huế
Tập 33, Số 1 (2026) ISSN: 2354-0842

DOI: 10.64302/joshusc.v33n1.1352

A SYTEMATIC LSTM FRAMEWORK FOR HIGH-FREQUENCY GOLD PRICE FORECASTING

Nguyen Hoang Ha[1], Truong An Binh, Nguyen Dinh Hoa Cuong[2*]

[1] University of Sciences, Hue University

[2] University of Finance - Marketing

*Email: cuongndh@ufm.edu.vn

Received: 22/12/2025; Received in revised form: 30/01/2025; Accepted: 4/02/2026

ABSTRACT

Forecasting intraday high-frequency gold prices presents a significant challenge due to the data's complex and non-linear dynamics. While deep learning models are widely applied, rigorous comparisons under standardized conditions remain limited. This study addresses this point by conducting a rigorous comparative analysis of five key deep learning architectures: one-dimensional convolutional neural networks (1D-CNN), simple recurrent neural networks (RNN), gated recurrent units (GRU), long short-term memory (LSTM), and bidirectional LSTM (BiLSTM). Utilizing a sliding window preprocessing technique, all models are systematically benchmarked under an identical training pipeline and evaluated across three distinct temporal window configurations to assess performance robustness. Experimental results demonstrated that the LSTM architecture consistently achieves superior forecasting accuracy, recording the lowest error metrics (MAE, MSE, RMSE) and highest R^2 score across all tested scenarios. These findings establish a rigorous performance benchmark, identifying LSTM as an effective and robust architecture for this high-frequency forecasting task.

Keywords: Gold Price Forecasting, High-Frequency Data, 1D-CNN, RNN, GRU, LSTM, BiLSTM, Deep Learning.

1. INTRODUCTION

Financial market forecasting constitutes a field of inquiry deemed pivotal for enabling financial analysts, investors, and policymakers to execute risk management and optimize investment strategies, thereby significantly enhancing the efficiency and effectiveness of economic decision-making [1], [2]. The fundamental task of predicting future values of financial time series remains exceptionally challenging, primarily attributable to the intricate nature and inherent uncertainty characterizing the data, which commonly exhibit nonlinearity, volatility, heteroscedasticity, and autocorrelation [2]. Consequently, advancements in deep learning methodologies, particularly data-driven neural network technologies, have propelled these techniques into the mainstream for predicting financial outcomes, gradually supplanting reliance upon conventional statistical and machine learning paradigms [1], [2]. Among these innovative deep learning architectures, the LSTM network, a specialized variant of the RNN designed specifically to process sequential data, has garnered extensive application, demonstrating a robust capacity for capturing temporal dependencies within financial sequences and establishing itself as a preferred methodology for price forecasting tasks [2]. Furthermore, investigations into commodity assets, notably including gold, have incorporated these advanced deep learning techniques, resulting in the implementation of models such as the CNN-LSTM structure specifically for gold price time-series forecasting [1], [2].

Notwithstanding the established efficacy of LSTM networks in sequence modeling, a demonstrable deficiency persists within the extant literature concerning the systematic and comparative evaluation for high-frequency data of competing deep learning architectures when applied to high-frequency financial time series data. Although research endeavors have addressed various financial assets and their predictions, the majority of reviewed scholarship utilized daily historical trading data, indicating a prioritizing on lower-frequency observation despite the documented availability of minute-level or tick-level data [2]. Previous scholarship, while applying the CNN-LSTM combination model to gold price forecasting, has not provided rigorous systematic benchmarking that spans across leading deep learning methodologies in this specific high-frequency domain [1], [2]. Furthermore, while recent scholarships have begun to explore complex architectures such as Transformers and hybrid models for financial forecasting, a definitive performance baseline for fundamental deep learning architectures specifically within the high-frequency domain remains under-established. The focus on architectural novelty often leads to less attention being paid to the necessity of rigorously benchmarking foundational models like LSTM against competing paradigms (e.g., CNN, GRU) under standardized conditions. Without such a benchmark, assessing the true incremental value of increased model complexity becomes challenging.

The contributions of this study are threefold. First, we propose a standardized comparative framework to address the specific challenge of high-frequency gold price forecasting by evaluating five distinct deep learning models: 1D-CNN, RNN, GRU, LSTM, and BiLSTM. This framework ensures a valid comparison by mandating a consistent architectural pattern, including three sequential layers and a common prediction head, for all models. Second, we detail a systematic model training pipeline, outlined in Algorithm 2, which utilizes a sliding window technique and a strict data scaling process to prevent data leakage. Finally, our empirical evaluation provides a comprehensive performance benchmark specifically for this high-frequency task, tested across three distinct temporal forecasting configurations. This benchmark demonstrates that the LSTM architecture consistently achieves superior accuracy, recording the lowest error metrics and highest R^2 score in all tested scenarios.

The organization of this paper is structured as follows: Section 2 provides a review of the existing literature concerning financial time series forecasting utilizing deep learning. Subsequently, Section 3 details the methodology, encompassing the dataset, data preprocessing procedures, and the architecture of the five models under evaluation. Section 4 then outlines the experimental setup and training process, which is followed by an analysis of the results. The paper concludes in Section 5 with a summary of the findings and a discussion of their implications.

2. RELATED WORKS

Financial time series forecasting is historically difficult due to the stochastic nature of market data, where volatility and non-stationarity often violate standard statistical assumptions [1], [2]. Consequently, literature showed a pervasive shift toward deep learning methodologies. Within this domain, LSTM networks have emerged as a dominant architecture, widely favored over standard RNNs for their specific design that mitigates the vanishing gradient problem and effectively captures long-term temporal dependencies [1], [2]. The exploration associated with price movements in the gold market constitutes an essential area of inquiry, especially because fluctuations affect the positioning of investments and the stability of global financial structures [3], [4]. Research efforts have employed various methodologies attempting to capture the intricate dynamics associated with this specific time series, progressing from classical econometric techniques to sophisticated deep-learning models which possess capabilities surpassing prior approaches.

The initial methodological attempts at gold price forecasting relied upon established statistical and econometric models, including the Auto-Regressive Integrated Moving Average (ARIMA) and Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) frameworks. For example, to predict daily gold prices characterized by high volatility, Setyowibowo et al. [5] applied an ARIMA-GARCH hybrid model to market data collected between 2016 and 2020. They evaluated the model's performance using error metrics such as Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), identifying the (1,1,1)-GARCH (2,1) configuration as the optimal fit. The results revealed a long-term upward trend in prices, suggesting that short-term forecasting required frequent model updates to account for immediate price influences. Boongasame et al. [4] addressed the limitations of linear relationship methods in explaining gold price patterns by utilizing a non-linear approach that integrated association rules with LSTM networks. The authors employed association rules to select relevant input features, determining that the US Dollar Index (DXY) significantly influenced the gold spot price. Their simulations showed that the LSTM model using these features achieved a lower Mean Absolute Percentage Error (MAPE) than Simple Moving Average (SMA), Weight Moving Average (WMA), Exponential Moving Average (EMA), and Auto-Regressive Integrated Moving Average (ARIMA) baselines. While these traditional econometric models struggled to capture the non-linear dynamics and heteroscedasticity inherent in volatile markets, our research pivots to deep learning architectures designed specifically for such complexity. Our work establishes a comprehensive deep learning framework to rigorously compare five distinct neural architectures under a standardized high-frequency training pipeline..

The limitations observed in statistical methods spurred the development of approaches rooted in artificial intelligence, marking a transition toward machine learning techniques. Artificial Neural Networks (ANN), encompassing the Multi-Layer Perceptron (MLP) architecture, demonstrated capabilities superior to statistical models in achieving higher prediction quality. Cohen and Aiche [21] evaluated regression trees, random forests, and gradient boosting, finding that the Extreme Gradient Boosting (XGBoost) model achieves the highest accuracy when incorporating diverse market indicators. Providing a comparative analysis, Gong et al. [11] examined LSTM networks versus Linear Regression and observe that the latter yields better directional accuracy, while the former results in lower error metrics. In related work on commodity price forecasting, Foroutan and Lahmiri [15] investigated the application of sixteen deep learning and machine learning architectures, reporting that the Temporal Convolutional Network (TCN) architecture yields the lowest Mean Absolute Error (MAE) for WTI, Brent, and silver, while the Bidirectional Gated Recurrent Unit (BiGRU) model achieves the highest accuracy for gold prices. Similarly, Jabeur et al. [10] evaluated the predictive performance of six machine learning algorithms, including XGBoost, CatBoost, and LightGBM, to address the challenges of forecasting gold price fluctuations. Focusing on model optimization, Weng et al. [26] introduced a framework that integrates a Genetic Algorithm with a Regularized Online Sequential Extreme Learning Machine (ROSELM) to enhance parameter stability and predictive performance, while Abu-Doush et al. [3] addressed the limitations of gradient descent in MLP training for gold price forecasting by integrating an archive-based mechanism into the Harris Hawks Optimizer, with comparative experiments validating the efficacy of the selected input features and accurate predictions. While these machine learning approaches effectively capture nonlinear patterns, existing literature often lacks a unified evaluation methodology specifically tailored for high-frequency financial data. To address this, we propose a standardized comparative framework that systematically evaluates five distinct deep learning models—including 1D-CNN, RNN, and LSTM—using a rigorous training pipeline designed to prevent data leakage. Our empirical benchmarking extends these prior findings by demonstrating that the LSTM architecture consistently yields superior accuracy compared to other deep learning variants in high-frequency gold price forecasting scenarios.

More recently, research efforts have focused increasingly upon deep learning methodologies, which inherently overcome limitations associated with shallow architectures and traditional statistical models. The LSTM network, standing as a variant of the RNN, was specifically engineered to address the persistent problem associated with long-term dependencies in sequential data, mitigating the vanishing gradient issue observed in standard RNNs. Elsaraiti and Merabet et al. [17] conducted a comparative analysis of the Autoregressive Integrated Moving Average (ARIMA) and LSTM models for wind speed forecasting, finding that the LSTM model, with its ability to capture long-term dependencies, achieves lower Root Mean Square Error (RMSE) rates than the ARIMA model. Focusing on financial markets, Shahi et al. [25] performed a comparative study on deep learning architectures for stock market price forecasting, indicating that the LSTM model achieves lower error rates, such as Mean Squared Error (MSE) and RMSE, compared to the GRU model, suggesting superior capability in capturing long-term dependencies. Furthermore, Chi et al. [16] proposed a multivariate LSTM-based forecasting model to predict real-time exchange rates in the Forex market, demonstrating that the feature-augmented model, utilizing engineered features like the daily percentage change in closing price (Close Change), achieves lower error rates compared to baseline models like GRU. To address data complexity, Zhang et al. [18] developed an enhanced LSTM architecture, termed LSTM-P, to forecast the prices of Bitcoin and gold, employing a wavelet transform-based noise reduction method and reporting that the optimized LSTM-P model outperforms conventional LSTM models and other time series forecasting techniques in terms of accuracy and precision. Unlike prior studies focusing on specific architectural modifications, this work establishes a standardized framework comparing five distinct deep learning models. We utilize a strict leakage-prevention pipeline to isolate LSTM's superior performance in high-frequency gold price forecasting.

The recognition of limitations inherent in single model performance has driven the development of hybrid deep learning models, which combine architectural strengths to enhance prediction accuracy. Numerous studies utilize hybrid deep learning frameworks, particularly combinations of CNN and LSTM units, to enhance gold price forecasting. Livieris et al. [9] constructed a foundational hybrid model where CNNs extract features and LSTM units model temporal dependencies, reporting reduced error rates compared to single-architecture benchmarks. Architectural enhancements include Amini and Kalantari [14] integrating CNNs with Bidirectional LSTM (Bi-LSTM) units and using grid search for hyperparameter tuning, and Santika et al. [22] tuning a CNN-LSTM model by adjusting batch sizes and unit counts to minimize predictive errors. Other studies focus on complex mechanism modifications: Wang et al. [23] developed the SGRU-AM model by modifying the GRU reset gate and adding an attention mechanism, while Memon et al. [7] investigated the MA-GRUS architecture combining multi-layer GRUs with multi-headed attention and skip connections. Addressing non-linearity, Djunaidy [12] employed the Gramian Angular Field (GAF) technique to transform time-series into image representations for visual detection of temporal structures. Finally, a parallel approach utilizes signal decomposition techniques, such as VMD and CEEMDAN [19] or ICEEMDAN [8], to break down the time series into frequency sublayers that are then individually forecasted using models like GRU, BiGRU, or LSTM to enhance accuracy. In contrast to these studies, which often focused on hybrid architectures or external data integration, our work provides a standardized comparative framework that systematically evaluates five distinct, non-hybrid deep learning models under consistent architectural and data processing constraints specifically for high-frequency gold price forecasting.

Existing research frequently contrasted statistical methods with deep learning models [1], [2]; results showed that LSTM and its variants generally outperformed ARIMA, SVR, and shallow networks for gold price prediction [4], [15]. For example, studies confirmed that LSTM networks surpassed ARIMA models in forecast accuracy for time series problems [17]. Furthermore, within the domain of deep learning, comparisons existed between architectures, often revealing that hybrid models surpassed single models [9], [22]. The CNN-LSTM model consistently demonstrated a performance advantage over standalone CNN and LSTM models in predicting daily gold price [9]. However, inconsistent findings existed; for instance, the hybrid CNN-LSTM model did not exhibit superior performance compared with individual CNN or LSTM models in one gold price study [15]. The challenge associated with synthesizing these diverse results arose from the variability across experimental setups, which included different datasets, feature engineering strategies, hyperparameter configurations, and evaluation metrics employed. For instance, optimal model parameters for forecasting gold price were often found to differ from those required for predicting gold price fluctuation [10], [26]. Different studies employed distinct input features; examples included lagged gold prices, stock indices [4], crude oil prices [15], and uncertainty measures [13]. The widespread application of various deep learning and machine learning methods necessitated a rigorous, systematic comparative assessment to determine the performance advantage of specific architectures for specialized forecasting tasks. Although the literature contained numerous comparative studies, differences in methodology often obscured a definitive conclusion regarding model superiority [1], [2].

The current literature exhibits a lack of normalized comparisons conducted under identical conditions, particularly concerning the relative performance of fundamental deep learning architectures (LSTM, GRU, CNN, BiLSTM) for high-frequency financial time series. Consequently, a definitive statement regarding which deep learning model represents the optimal choice for achieving the prediction of better quality remains elusive. Therefore, this study is aimed because it conducts a systematic benchmarking exercise, utilizing a standardized training pipeline and identical evaluation metrics across key deep learning architectures, thereby establishing a robust performance metric for the difficult task associated with high-frequency gold price forecasting. Therefore, establishing a definitive performance benchmark requires a rigorous comparative analysis of fundamental architectures under standardized training conditions. Such an analysis is critical to resolve existing contradictions and identify the most robust architecture for high-frequency gold price forecasting.

3. METHODOLOGY

This section details the complete methodology used in this study. It begins by introducing the sliding window technique used for time series data processing (3.1). Next, it details the architectures of all five deep learning models evaluated in the comparison (3.2). Following this, the paper presents the algorithm for the model training pipeline (3.3) , the process for predicting future prices (3.4), and finally, the comparative evaluation process used to assess model performance (3.5).

3.1. Sliding window – based time series data preprocessing

The dataset is a multivariate time series $X \in R^{m \times n}$, where m is the number of timesteps and n is the number of features. It can be represented as a sequence of feature vectors $X = (x_1, x_2, \dots, x_m)$, where each $x_t \in R^n$. To prepare this sequential data for the supervised learning models, a sliding window technique is applied.

To prepare this sequential data for supervised learning, a sliding window technique is applied. Let w be the input window size and f be the forecast horizon. The time series is transformed into a set of input-output pairs. An input sequence $X^{(i)}$ is defined as the sequence of w full feature vectors starting at time i: $X^{(i)} = (x_i, x_{i+1}, \dots, x_{i+w-1})$. Let c_t represent the 'Close' price at timestep t. The corresponding target sequence $Y^{(i)}$ consists only of the 'Close' price for the f steps following the input window: $Y^{(i)} = \left(c_{i+w}, c_{i+w+1}, \dots, c_{i+w+f-1}\right)$

This sliding window transformation maps the original 2D tensor $X \in R^{m \times n}$ into two new tensors: a 3D tensor of input samples $X \in R^{k \times w \times n}$ and a 2D tensor of target samples $Y \in R^{k \times f}$, where $k = m - w - f + 1$ is the total number of generated samples. This resulting dataset facilitates model training. Algorithm 1 presents the pseudocode of the sliding window operation.

Algorithm 1: Sliding window technique

function slidingWindow(data, window, forecast)

data_size = |data|

X = []

y = []

for i from 0 to data_size - window - forecast:

X = X U data[i : i + window, :]

y = y U data[i + window : i + window + forecast, -1]

return X, y

end function

3.2. Architecture of Evaluated Deep Learning Models

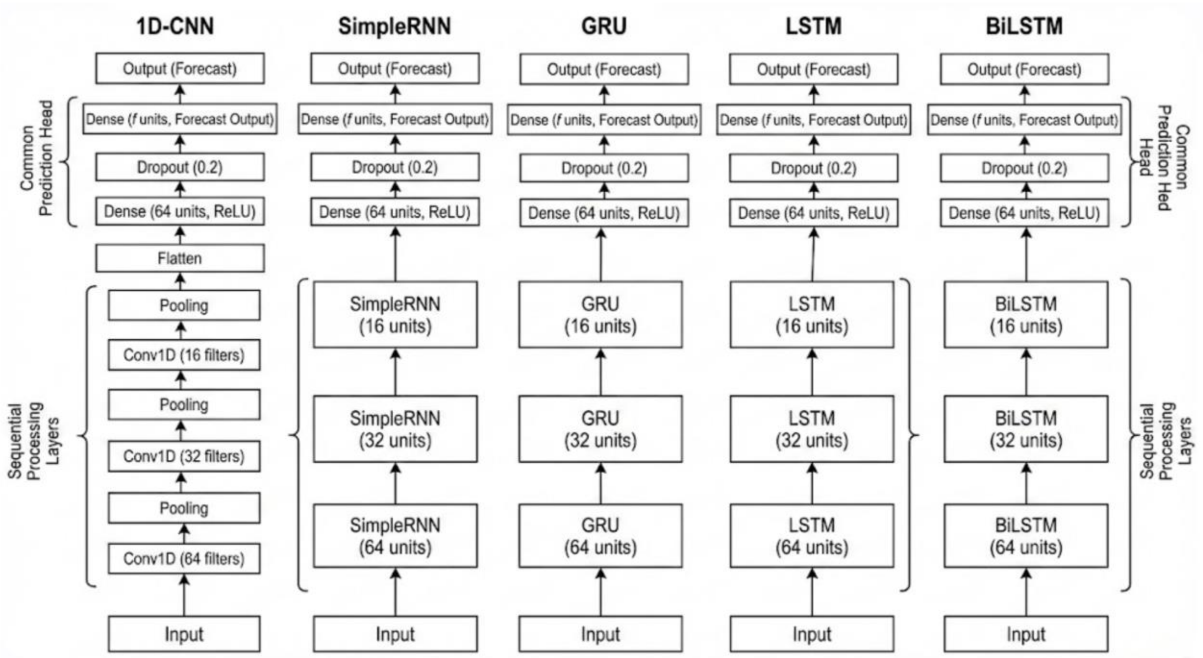

Fig 1. Architectural configurations of the five standalone deep learning models under evaluation.

This study evaluates five standalone deep learning models which share a consistent design principle, each constructed with three sequential processing layers of decreasing complexity (64, 32, and 16 units or filters). The 1D-CNN model utilized

Conv1D layers with ReLU activation, each followed by a MaxPooling1D layer. The recurrent models were composed of stacked SimpleRNN, GRU, LSTM, or Bidirectional LSTM layers, all employing the tanh activation function.

All architectures terminated in a common head structure, comprising a Dense layer of 64 units with ReLU activation, a Dropout layer with a rate of 0.2, and a final Dense output layer with *f* units (matching the forecast horizon, forecast_size). For training, the Adam optimizer was used in conjunction with the Mean Squared Error (MSE) loss function. Models were trained for a maximum of 100 epochs with a batch size of 64. An early stopping protocol was implemented to monitor the validation loss, configured with a patience of 20 epochs and a minimum delta of 10^{-5}. A visual representation of these architectural configurations is illustrated in Fig 1.

3.3. Model training process

The systematic methodology for data preparation and model training is outlined in Algorithm 2. The process is initiated by loading the historical dataset. To prevent data leakage, the data is immediately partitioned into training (70%), validation (15%), and test (15%) sets. A scaler is then instantiated and fitted exclusively on the training data, after which it is used to transform all three data partitions. The slidingWindow function (from Algorithm 1) is subsequently applied to each normalized set to generate the input (X) and target (y) tensors. Finally, the selected model architecture (model_type) is instantiated and trained using the X_train and y_train tensors, returning the trained model and testing artifacts for the prediction phase.

Algorithm 2: Model Training Pipeline

function train_model(file, w, f, model_type, patience)

data = read_data(file)

train, val, test = temporal_split(data, 70, 15)

scaler = fit_scaler(train)

train_s, val_s, test_s = scaler.transform(train, val, test)

X_train, y_train = create_sliding_window(train_s, w, f)

X_val, y_val = create_sliding_window(val_s, w, f)

model = initialize_model(model_type).compile(optimizer='Adam', loss='MSE')

early_stop_cb = EarlyStopping(monitor='val_loss', patience=patience)

model.fit(X_train,y_train,validation_data=(X_val,y_val), callbacks=[early_stop_cb])

return model, scaler, X_test, y_test // X_test/y_test creation assumed

end function

3.4. Predict next n-minute gold prices

The inference procedure for forecasting future gold prices is formally described in Algorithm 3. This process is initiated by loading the trained model and the scaler previously fitted on the training set. Input data is first normalized and reshaped to match the tensor requirements of the deep learning model. Once the predict method is executed, an array of predicted values is generated within the normalized range. To convert these values back into meaningful market prices, an inverse transformation is applied. Since the multivariate scaler anticipates a full feature set, a reconstruction step is performed using a temporary array to correctly isolate and restore the predicted 'Close' price to its original scale.

Algorithm 3: Predict next n-minute Gold prices

function predict_next_m_minutes(raw_data, trained_model, scaler):

input_features = raw_data['open', ..., 'close']

scaled_input = scaler.transform(input_features)

reshaped_input = reshape(scaled_input)

scaled_prediction = trained_model.predict(reshaped_input)

dummy_array = initialize_array(shape=(m, n_features))

dummy_array[:, 4] = scaled_prediction[:, 0]

inverse_transformed_output = scaler.inverse_transform(dummy_array)

final_price = inverse_transformed_output[:, 4]

return final_price

end function

4. EXPERIMENT

The experiments were designed to forecast the closing price of gold using several standalone deep learning models: 1D-CNN, RNN, GRU, LSTM, and BiLSTM for the task of forecasting gold closing prices. The dataset, sourced from Yahoo Finance[1], comprises 1-minute Open, High, Low, Close, and Volume (OHLCV) data points spanning from August 1, 2025, to August 29, 2025. To assess model performance under different conditions, three distinct sequence-to-sequence configurations were implemented. These configurations are defined by (input sequence length, output sequence length)

[1] https://finance.yahoo.com/

pairs: (i) a 5-minute historical window to predict the subsequent 1-minute value, (ii) a 10-minute window to predict the subsequent 2 minutes, and (iii) a 20-minute window to predict the subsequent 5 minutes. The dataset was partitioned into training, validation, and testing sets with proportions of 70%, 15%, and 15%, respectively.

The implementation and training of all models were conducted within the Google Colab[2] environment, leveraging its GPU support to accelerate computational processes. The TensorFlow[3] framework served as the primary deep learning library for constructing and training the neural network architectures. This setup provided a robust and efficient platform for executing the extensive series of experiments required for this research.

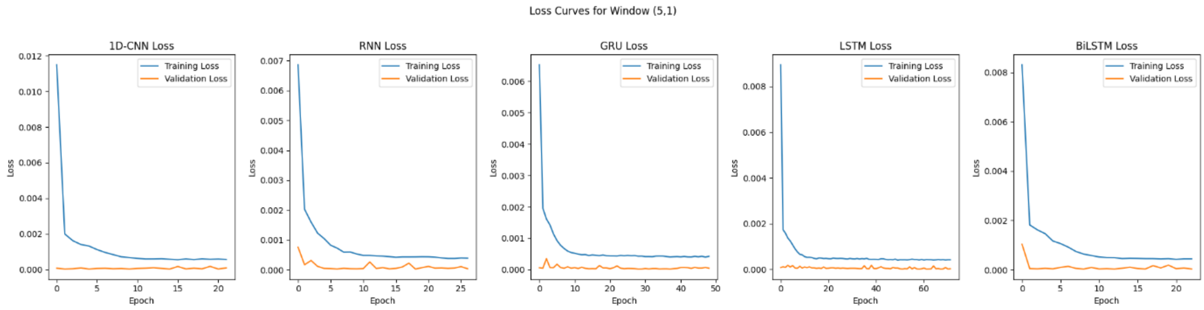

Fig 2. Loss Curves for Window Size 5 and Step Size 1

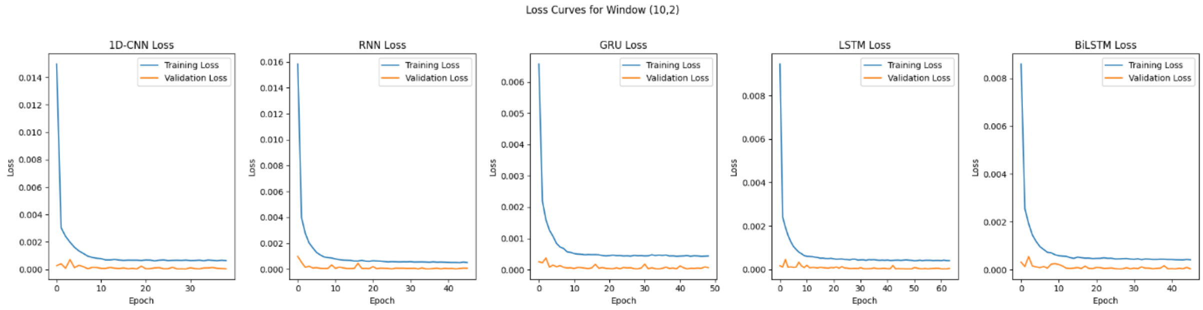

Fig 3. Loss Curves for Window Size 10 and Step Size 2

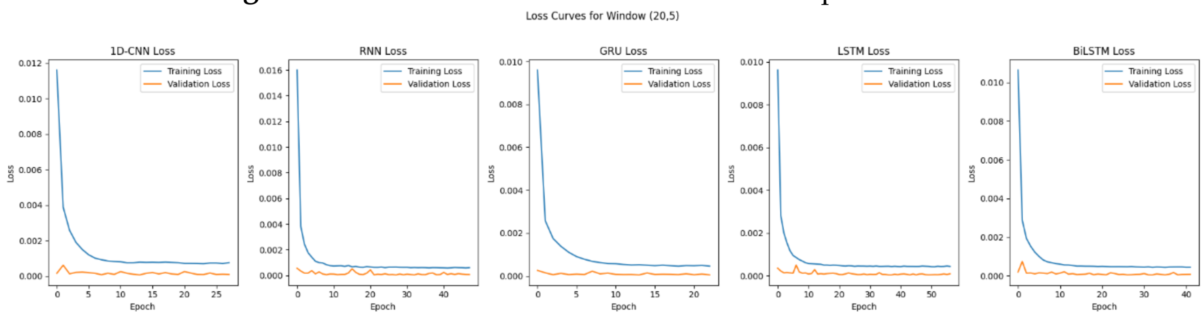

Fig 4. Loss Curves for Window Size 20 and Step Size 5

[2] https://colab.google/

[3] https://www.tensorflow.org/

The three images illustrate the training and validation loss curves for the 1D-CNN model, LSTM model, GRU model, BiLSTM model and RNN model, each trained with different sliding window configurations. The learning curves for all experimental configurations illustrate consistent training behavior. During the training phase for each model, the training loss demonstrated a sharp initial decrease, followed by a convergence to a stable, low value. The validation loss closely followed the trajectory of the training loss, also settling at a low value without significant divergence as the number of epochs increased. This parallel behavior between the two loss curves across all experiments suggests that the models generalized well from the training data to the unseen validation data, and no significant overfitting was observed.

In order to evaluate the trained models, the MSE, MAE, RMSE, and R^2 metrics, with formulas listed in Eqs (1), (2), (3), and (4), respectively. The predicted values of the trained models over three test sets are visualized in Fig 5, Fig 6 and Fig 7, while the evaluation results are presented in Table 1.

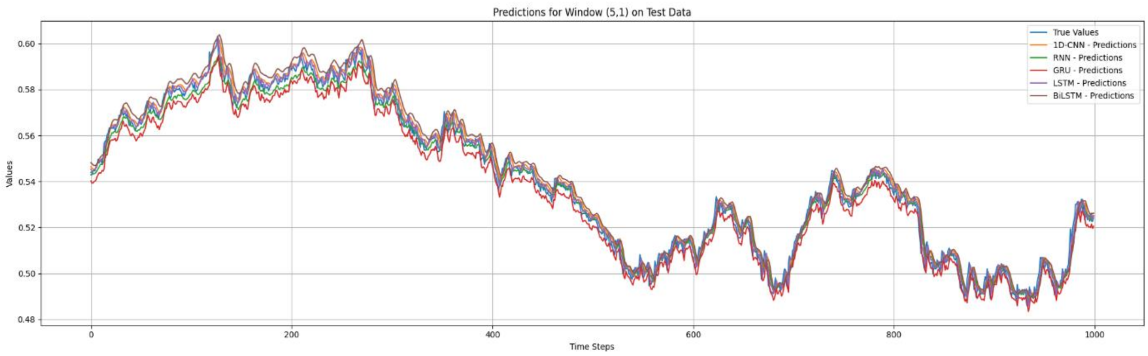

Fig 5. Model Predictions vs. true values for 5-minutes window size and 1-minutes step size configuration

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2 \tag{1}$$

$$MAE(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N-1}|y_i - \hat{y}_i| \tag{2}$$

$$RMSE(y, \hat{y}) = \sqrt{MSE(y, \hat{y})} \tag{3}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1}(y_i - \bar{y})^2} \tag{4}$$

where N is the total samples; y and $\hat{y}$ are true and predicted values, respectively; and $\bar{y} = \frac{1}{N}\sum_{i=0}^{N-1} y_i$.

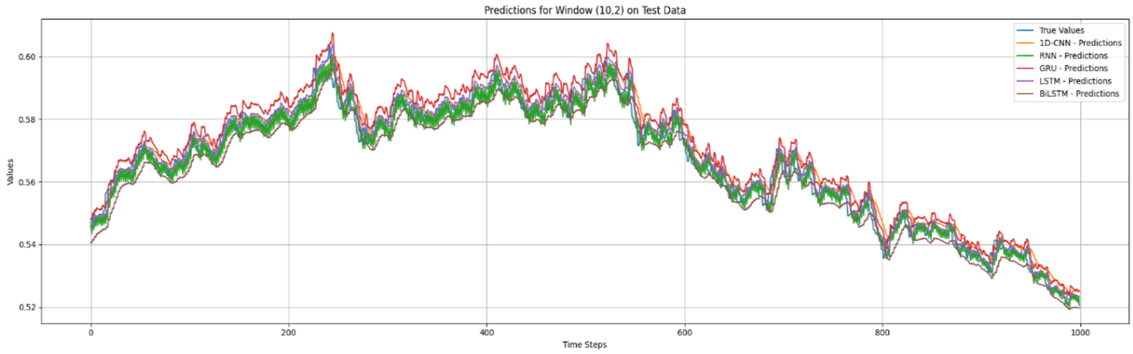

Fig 6. Model Predictions vs. true values for 10-minutes window size and 2-minutes step size configuration

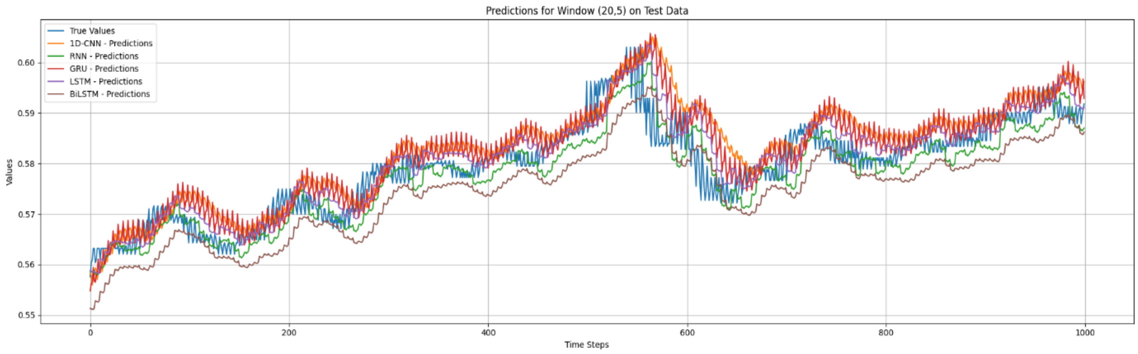

Fig 7. Model Predictions vs. true values for 20-minutes window size and 5-minutes step size configuration

Table 1. Evaluation of model performances

Window (n, m)	Model	MAE	MSE	RMSE	R2
Configuration 1: Window size (5 minutes) Step size (1 minute)	1D-CNN	0.003947	0.000026	0.005083	0.998551
	RNN	0.006753	0.000146	0.012063	0.991839
	GRU	0.010642	0.000175	0.013222	0.990196
	LSTM	0.002901	0.000014	0.003760	0.999207
	BiLSTM	0.007021	0.000069	0.008278	0.996157
Configuration 2: Window size (10 minutes) Step size (2 minutes)	1D-CNN	0.005083	0.000060	0.007737	0.996641
	RNN	0.007611	0.000186	0.013635	0.989568
	GRU	0.008210	0.000091	0.009533	0.994899
	LSTM	0.004402	0.000030	0.005490	0.998309

	BiLSTM	0.005961	0.000064	0.008029	0.996382
Configuration 3: Window size (20 minutes) Step size (5 minutes)	1D-CNN	0.005941	0.000062	0.007853	0.996534
	RNN	0.005324	0.000064	0.007988	0.996415
	GRU	0.005328	0.000048	0.006952	0.997285
	LSTM	0.004302	0.000033	0.005770	0.998130
	BiLSTM	0.010891	0.000215	0.014647	0.987941

Table 1 presents the experimental results evaluated by Mean Absolute Error—MAE, Mean Squared Error—MSE, Root Mean Squared Error—RMSE, and the R^2 score. These metrics facilitate a performance comparison of five models—1D-CNN, RNN, GRU, LSTM, and BiLSTM—across three distinct sliding window configurations. For the first configuration, utilizing a 5-minute window with a 1-minute step, the LSTM model demonstrated superior performance. It achieved the lowest error metrics, with an MAE of 0.002901, an MSE of 0.000014, and an RMSE of 0.003760. Correspondingly, it obtained the highest R^2 score of 0.999207, indicating the strongest predictive fit among the tested models for this short-term scenario.

In the second configuration, which employed a 10-minute window and a 2-minute step, the LSTM model again emerged as the most effective. It registered the lowest MAE of 0.004402, MSE of 0.000030, and RMSE of 0.005490. Furthermore, it produced the highest R^2 score of 0.998309. This outcome indicates that the LSTM architecture maintained its superior performance for the intermediate window size. Finally, for the third configuration, characterized by a 20-minute window and a 5-minute step, the LSTM model continued to deliver the most robust performance. It recorded the lowest error values with an MAE of 0.004302, MSE of 0.000033, and RMSE of 0.005770. The model also achieved the highest R^2 score of 0.998130. In contrast, the BiLSTM model yielded the highest error metrics in this configuration, establishing it as the least effective model for this longer-term dependency task.

Synthesizing these findings, the empirical evaluation demonstrates that the LSTM model consistently outperformed all other architectures across the three tested temporal configurations. Its superior accuracy in short, medium, and long-term window scenarios establishes it as the most effective and robust model for the forecasting task examined in this study. The results do not indicate that the optimal model is contingent on the specific temporal parameters; rather, they identify the LSTM architecture as the unambiguously superior choice.

The LSTM’s superiority in this high-frequency domain stems from its distinct structural mechanisms. Unlike the GRU’s simplified merged-gate design, the LSTM’s tripartite gating system—specifically the decoupled output gate—provides granular control over the cell state, effectively filtering stochastic noise inherent in 1-minute intervals. Furthermore, while BiLSTM incorporates future context, its bidirectional processing introduces unnecessary complexity and look-ahead bias that misaligns with the strictly causal nature of real-time forecasting. Consequently, the unidirectional LSTM achieves an optimal balance between capacity and generalization, avoiding BiLSTM's overfitting tendencies while preserving long-term dependencies through robust gradient flow.

Despite the framework's efficacy, this study entails specific constraints. First, the reliance on univariate historical data excludes exogenous determinants—such as geopolitical events or monetary policies—which frequently precipitate structural market breaks. Second, the finite dataset availability may not fully encapsulate diverse long-term economic cycles or extreme "black swan" events. Finally, as validation is restricted to the gold market, the framework's transferability to asset classes with distinct volatility profiles, such as cryptocurrencies or energy commodities, requires future verification.

5. CONCLUSION

This study established a comparative framework to benchmark deep learning models for high-frequency gold price prediction. We evaluated five architectures (1D-CNN, RNN, GRU, LSTM, and BiLSTM) using a leakage-prevention pipeline across varying temporal horizons. Results demonstrate that the LSTM architecture achieved the lowest error metrics (MAE, MSE, RMSE) and highest R^2 scores in all configurations. Findings reveal that architectural complexity, specifically in the BiLSTM model, did not result in higher accuracy; LSTM outperformed both RNN and BiLSTM variants. Consequently, this research identifies LSTM as a baseline architecture for high-frequency gold price forecasting. Future work will integrate multivariate data sources and hybrid mechanisms to refine precision.

ACKNOWLEDGEMENT

This paper is sponsored by Hue University under Grant No. DHH2025-01-225.

REFERENCES

[1] W. Bao, Y. Cao, Y. Yang, H. Che, J. Huang, and S. Wen, "Data-driven stock forecasting models based on neural networks: A review," Information Fusion, vol. 113, p. 102616, Jan. 2025, doi: 10.1016/j.inffus.2024.102616.

[2] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022," WIREs Data Min & Knowl, vol. 14, no. 1, p. e1519, Jan. 2024, doi: 10.1002/widm.1519.

[3] I. Abu-Doush, B. Ahmed, M. A. Awadallah, M. A. Al-Betar, and A. R. Rababaah, "Enhancing multilayer perceptron neural network using archive-based harris hawks optimizer to predict gold prices," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 5, p. 101557, May 2023, doi: 10.1016/j.jksuci.2023.101557.

[4] L. Boongasame, P. Viriyaphol, K. Tassanavipas, and P. Temdee, "Gold-Price Forecasting Method Using Long Short-Term Memory and the Association Rule," JMM, Sept. 2022, doi: 10.13052/jmm1550-4646.1919.

[5] S. Setyowibowo, M. As'ad, S. Sujito, and E. Farida, "Forecasting of Daily Gold Price using ARIMA-GARCH Hybrid Model," JEP, vol. 19, no. 2, pp. 257–270, Feb. 2022, doi: 10.29259/jep.v19i2.13903.

[6] S. Nallamothu, K. Rajyalakshmi, and P. Arumugam, "Gold Price Prediction Using Skewness and Kurtosis Based Generalized Auto-regressive Conditional Heteroskedasticity Approach with Long Short Term Memory Network," J. Inst. Eng. India Ser. B, vol. 105, no. 6, pp. 1715–1727, Dec. 2024, doi: 10.1007/s40031-024-01070-7.

[7] B. A. Memon, R. Tahir, H. M. Naveed, and K. Cheng, "Forecasting Gold and Platinum prices with an enhanced GRU model using multi-headed attention and skip connection," Miner Econ, May 2025, doi: 10.1007/s13563-025-00520-y.

[8] Y. Liang, Y. Lin, and Q. Lu, "Forecasting gold price using a novel hybrid model with ICEEMDAN and LSTM-CNN-CBAM," Expert Systems with Applications, vol. 206, p. 117847, Nov. 2022, doi: 10.1016/j.eswa.2022.117847.

[9] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Comput & Applic, vol. 32, no. 23, pp. 17351–17360, Dec. 2020, doi: 10.1007/s00521-020-04867-x.

[10] S. B. Jabeur, S. Mefteh-Wali, and J.-L. Viviani, "Forecasting gold price with the XGBoost algorithm and SHAP interaction values," Ann Oper Res, vol. 334, no. 1–3, pp. 679–699, Mar. 2024, doi: 10.1007/s10479-021-04187-w.

[11] W. Gong, "Research on gold price forecasting based on lstm and linear regression," SHS Web Conf., vol. 181, p. 02005, 2024, doi: 10.1051/shsconf/202418102005.

[12] M. Salim and A. Djunaidy, "Development of a CNN-LSTM Approach with Images as Time-Series Data Representation for Predicting Gold Prices," Procedia Computer Science, vol. 234, pp. 333–340, 2024, doi: 10.1016/j.procs.2024.03.007.

[13] W. You, J. Chen, H. Xie, and Y. Ren, "Which uncertainty measure better predicts gold prices? New evidence from a CNN-LSTM approach," The North American Journal of Economics and Finance, vol. 76, p. 102375, Jan. 2025, doi: 10.1016/j.najef.2025.102375.

[14] A. Amini and R. Kalantari, "Gold price prediction by a CNN-Bi-LSTM model along with automatic parameter tuning," PLoS ONE, vol. 19, no. 3, p. e0298426, Mar. 2024, doi: 10.1371/journal.pone.0298426.

[15] P. Foroutan and S. Lahmiri, "Deep learning systems for forecasting the prices of crude oil and precious metals," Financ Innov, vol. 10, no. 1, p. 111, July 2024, doi: 10.1186/s40854-024-00637-z.

[16] D. T. K. Chi, H. N. T. Kien, and T. Q. Nguyen, "Enhancing forex market forecasting with feature-augmented multivariate LSTM models using real-time data," Knowledge-Based Systems, vol. 330, p. 114500, Nov. 2025, doi: 10.1016/j.knosys.2025.114500.

[17] M. Elsaraiti and A. Merabet, "A Comparative Analysis of the ARIMA and LSTM Predictive Models and Their Effectiveness for Predicting Wind Speed," Energies, vol. 14, no. 20, p. 6782, Oct. 2021, doi: 10.3390/en14206782.

[18] X. Zhang, L. Zhang, Q. Zhou, and X. Jin, "A Novel Bitcoin and Gold Prices Prediction Method Using an LSTM-P Neural Network Model," Computational Intelligence and Neuroscience, vol. 2022, pp. 1–12, May 2022, doi: 10.1155/2022/1643413.

[19] Y. Zhang, Y. Peng, and Y. Song, "Metal commodity futures price forecasting based on a hybrid secondary decomposition error-corrected model," J Big Data, vol. 12, no. 1, p. 166, July 2025, doi: 10.1186/s40537-025-01240-4.

[20] Y. Li, S. Wang, Y. Wei, and Q. Zhu, "A New Hybrid VMD-ICSS-BiGRU Approach for Gold Futures Price Forecasting and Algorithmic Trading," IEEE Trans. Comput. Soc. Syst., vol. 8, no. 6, pp. 1357–1368, Dec. 2021, doi: 10.1109/TCSS.2021.3084847.

[21] G. Cohen and A. Aiche, "Forecasting gold price using machine learning methodologies," Chaos, Solitons & Fractals, vol. 175, p. 114079, Oct. 2023, doi: 10.1016/j.chaos.2023.114079.

[22] I. W. Krisna Gita Santika, S. Sa'adah, and P. E. Yunanto, "Gold price prediction using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM)," KINETIK, Aug. 2021, doi: 10.22219/kinetik.v6i3.1253.

[23] J. Wang, Y. Li, T. Wang, J. Li, H. Wang, and P. Liu, "A Gold Futures Price Forecast Model Based on SGRU-AM," IEEE Access, vol. 9, pp. 146745–146754, 2021, doi: 10.1109/ACCESS.2021.3122140.

[24] Y. Zhao, Y. Guo, and X. Wang, "Hybrid LSTM–Transformer Architecture with Multi-Scale Feature Fusion for High-Accuracy Gold Futures Price Forecasting," Mathematics, vol. 13, no. 10, p. 1551, May 2025, doi: 10.3390/math13101551.

[25] T. B. Shahi, A. Shrestha, A. Neupane, and W. Guo, "Stock Price Forecasting with Deep Learning: A Comparative Study," Mathematics, vol. 8, no. 9, p. 1441, Aug. 2020, doi: 10.3390/math8091441.

[26] F. Weng, Y. Chen, Z. Wang, M. Hou, J. Luo, and Z. Tian, "Gold price forecasting research based on an improved online extreme learning machine algorithm," J Ambient Intell Human Comput, vol. 11, no. 10, pp. 4101–4111, Oct. 2020, doi: 10.1007/s12652-020-01682-z.

ĐỀ XUẤT KHUNG MÔ HÌNH LSTM CHO DỰ BÁO GIÁ VÀNG TẦN SUẤT CAO

Nguyễn Hoàng Hà[1], Trương An Bình, Nguyễn Đình Hoa Cương[2*]

[1] Trường Đại học Khoa học, Đại học Huế

[2] Trường Đại học Tài chính - Marketing

*Email: cuongndh@ufm.edu.vn

TÓM TẮT

Dự báo giá vàng tần suất cao trong ngày đang đặt ra một thách thức đáng kể do tính chất phức tạp và phi tuyến của dữ liệu. Mặc dù các mô hình học sâu hiện đang được ứng dụng rộng rãi, tuy nhiên các nghiên cứu so sánh chuyên sâu dưới những điều kiện chuẩn hóa vẫn còn hạn chế. Nghiên cứu này giải quyết vấn đề trên thông qua việc thực hiện phân tích, so sánh giữa năm kiến trúc học sâu chủ chốt: mạng nơ-ron tích chập một chiều (1D-CNN), mạng nơ-ron hồi quy đơn giản (RNN), đơn vị hồi quy có cổng (GRU), bộ nhớ ngắn-dài hạn (LSTM) và LSTM hai chiều (BiLSTM). Nhóm tác giả đã sử dụng kỹ thuật tiền xử lý cửa sổ trượt, tất cả các mô hình được thực hiện trong cùng một quy trình huấn luyện và được đánh giá trên ba cấu hình cửa sổ thời gian khác nhau nhằm xác định tính ổn định của hiệu năng. Kết quả thực nghiệm chứng minh rằng kiến trúc LSTM đạt độ chính xác một cách nhất quán trong việc dự báo giá, ghi nhận các chỉ số sai số (MAE, MSE, RMSE) thấp nhất và điểm R^2 cao nhất trên mọi kịch bản thử nghiệm. Những phát hiện này thiết lập một thước đo hiệu năng cơ sở, xác định LSTM là kiến trúc hiệu quả và ổn định nhất cho tác vụ dự báo giá vàng tần suất cao.

Từ khóa: Gold Price Forecasting, High-Frequency Data, 1D-CNN, RNN, GRU, LSTM, BiLSTM, Deep Learning.