A Synergistic Hybrid Architecture with Residual Attention and Mixture-of-Experts for Robust Hour-Ahead Forex Forecasting

Alireza Abbaszadeh[1], Seyyed Abed Hosseini[2], and Mohammad-R. Akbarzadeh-T.[3]
[1]Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad, Iran
Email: alireza.abbaszadeh.research@gmail.com
[2]Department of Electrical Engineering, Ma.C., Islamic Azad University, Mashhad, Iran
Email: sa.hosseini@iau.ac.ir
[3]Department of Electrical Engineering, Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran
Email: akbazar@um.ac.ir

***Abstract*—Forecasting highly volatile exchange rates like EUR/USD is a critical challenge where traditional models fail to capture the associated complex and volatile nonlinear dynamics of these rates. While the modern deep learning architectures have shown considerable promise in the past few years, they also present significant variation, and achieving optimal synergy among their advanced components, such as attention and mixture-of-experts, remains an open research question. This paper introduces a novel, synergistic hybrid architecture engineered to address this gap. Our model (V8) strategically integrates a Residual Block with Multi-Head Attention (RB-MHA) for robust feature extraction, a Bidirectional LSTM for temporal modeling, and a Mixture-of-Experts (MoE) module for adaptive prediction under varying market conditions. Evaluated on over 15 years of hourly EUR/USD data using a rigorous, leak-free methodology, our model sets a new state-of-the-art performance with a Root Mean Squared Error (RMSE) of 0.001863 and an R^2 of 0.985467 on the unseen test set. This result constitutes a 44.1% reduction in RMSE over a strong deep learning baseline (V1), demonstrating the significant impact of our synergistic design and establishing a new benchmark for hour-ahead forex forecasting.**

***Index Terms*—Foreign Exchange Forecasting; Deep Learning; Residual Networks; Attention Mechanism; Mixture of Experts (MoE).**

I. Introduction

Exchange rate forecasting, particularly for highly volatile pairs like EUR/USD, is a critical challenge in finance. The underlying data, often represented by Open, High, Low, and Close (OHLC) prices, exhibits complex, non-stationary dynamics that traditional econometric models like ARIMA and GARCH struggle to capture due to their linearity assumptions [1]–[6]. In response, deep learning has emerged as a powerful alternative. Foundational architectures like CNNs and LSTMs capture local and temporal patterns but are limited by fixed receptive fields and difficulties with long-range dependencies, respectively [7]–[9].

Recent advances have introduced more sophisticated components. Multi-Head Attention (MHA) enables global context modeling but applies a monolithic transformation, failing to adapt to distinct market regimes (e.g., high vs. low volatility) [10]. Concurrently, Mixture-of-Experts (MoE) offers a path to conditional computation by routing inputs to specialized subnetworks [11], [12]. However, the synergistic integration of these advanced components into a unified, hierarchical framework for financial forecasting remains an open research problem.

To address this gap, we propose a novel hybrid architecture (Model V8) that strategically integrates these components. Our model employs a **Residual Block with MHA (RB-MHA)** for robust feature extraction, a **Bidirectional LSTM (Bi-LSTM)** for temporal modeling, and an **MoE module** for adaptive prediction. This design creates a hierarchical pipeline capable of handling the multi-faceted complexities of financial time series. Our main contributions are: (1) A novel, synergistic architecture tailored for financial forecasting; (2) A systematic component analysis demonstrating the significant impact of each module, particularly a 32.5% RMSE reduction from the MoE layer; and (3) A new state-of-the-art result on a large-scale hourly EUR/USD dataset (R^2=0.985467, RMSE=0.001863) achieved via a rigorous, leak-free methodology.

The remainder of this paper details our architecture, experiments, and results, concluding with a discussion of their implications. The remainder of this paper is structured as follows: Section II reviews related literature, Section III details the proposed architecture, Section IV describes the data and preprocessing, Section V presents the experiments, and Section VI concludes the study.

II. Related Work

Financial time series forecasting has transitioned from traditional econometric models, such as ARIMA and GARCH, which struggle with the nonlinear and non-stationary dynamics of financial markets, towards more robust deep learning architectures [4], [7]. Early deep learning approaches successfully

979-8-3315-8973-8/25/$31.00 ©2025 IEEE

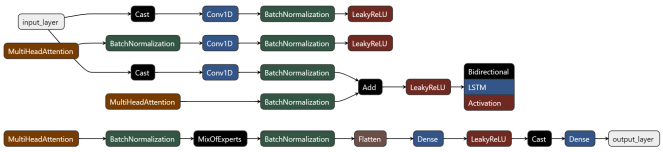

Fig. 1. Conceptual overview of the proposed hybrid architecture, Model V8 (RB-MHA for feature extraction → Bi-LSTM for temporal modeling → a subsequent MHA layer for contextual weighting → and a MoE module for adaptive prediction → Dense).

applied CNNs for local pattern extraction and Recurrent Neural Networks (RNNs), such as LSTMs, for modeling temporal dependencies, often in hybrid structures [9], [13].

A key advancement was the integration of attention mechanisms, particularly the MHA from the Transformer architecture [10], which allows models to dynamically weigh the importance of different time steps and capture complex dependencies more effectively [14], [15]. This led to the development of sophisticated hybrid models integrating attention with CNNs and Bi-LSTMs, sometimes within residual learning frameworks to improve gradient flow and feature extraction [16]–[18].

More recently, to handle the abrupt regime shifts common in financial markets, the MoE paradigm has been employed. MoE architectures use a gating network to route inputs to specialized sub-networks, enabling the model to adapt to different market conditions and improve predictive accuracy [11], [12], [19]. However, despite these advances, the deep integration of multi-scale feature extractors, advanced attention mechanisms, and adaptive MoE layers in a single, synergistic framework remains an under-explored area for currency forecasting. Our work addresses this gap by proposing a novel hybrid architecture engineered to harness the combined strengths of these components.

III. Proposed Hybrid Architecture

This section details the architecture of our proposed forecasting framework, Model V8, which was identified as the best-performing variant in our experiments (Section V). As illustrated in Fig. 1, the model is constructed as a sequential pipeline to capture diverse market dynamics.

The architecture begins with an **RB-MHA** for feature extraction, followed by a **Bi-LSTM** layer to aggregate temporal information. Subsequently, another **MHA** mechanism dynamically weighs time steps. To handle regime shifts, a **MoE** layer adaptively blends outputs from specialized sub-networks. Finally, a **Dense consolidation layer** maps the processed representation to the final price forecast. Each module is described in the following subsections.

A. Residual Feature Extraction Block

The foundational layer of our architecture is a Residual Block, inspired by ResNet [16], which extracts robust, multi-scale features and stabilizes training via a skip connection [5], [15]. The block consists of a main path for feature transformation and a shortcut path to preserve the original signal.

The main path learns complex features by passing the input sequence $\mathbf{X}$ through a one-dimensional (1D) CNN layer, followed by Batch Normalization, a Leaky ReLU activation, and an MHA module to capture global contextual dependencies [10]. The output is then normalized again using a Batch Normalization layer.

$$\mathbf{H}_{\text{conv}} = \text{LeakyReLU}(\text{BN}(\text{Conv1D}(\mathbf{X}))) \tag{1}$$

$$\mathbf{H}_{\text{main}} = \text{BN}(\text{MHA}(\mathbf{H}_{\text{conv}})) \tag{2}$$

Concurrently, a shortcut path projects the input $\mathbf{X}$ using a 1D convolution with a kernel size of 1 to match the dimensions of the main path's output.

$$\mathbf{S} = \text{BN}(\text{Conv1D}_{\text{shortcut}}(\mathbf{X})) \tag{3}$$

Finally, the outputs of the main path ($\mathbf{H}_{\text{main}}$) and the shortcut path ($\mathbf{S}$) are added. The result is passed through a final Leaky ReLU activation to produce the block's output, $\mathbf{F}_{\text{res}}$. This design mitigates gradient degradation and improves generalization [16], [20].

$$\mathbf{F}_{\text{res}} = \text{LeakyReLU}(\mathbf{H}_{\text{main}} + \mathbf{S}) \tag{4}$$

B. Bidirectional Temporal Modeling (Bi-LSTM Layer)

To capture complex temporal dependencies, the model incorporates a Bi-LSTM network. Unlike standard LSTMs, a Bi-LSTM processes information from both past and future contexts at each time step, making it highly effective for modeling financial market dynamics [1], [2], [5], [21].

Given the input sequence from the residual block, $\mathbf{F}_{\text{res}}$, the Bi-LSTM computes forward ($\overrightarrow{\mathbf{h}}_t$) and backward ($\overleftarrow{\mathbf{h}}_t$) hidden states.

$$\overrightarrow{\mathbf{h}}_t = \text{LSTM}_{\text{fw}}(\mathbf{F}_{\text{res}}, t), \quad \overleftarrow{\mathbf{h}}_t = \text{LSTM}_{\text{bw}}(\mathbf{F}_{\text{res}}, T-t+1) \tag{5}$$

These states are then concatenated at each time step t to form the final representation, $\mathbf{H}_t = \overrightarrow{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t$. This dual-context representation enhances the model's ability to capture volatility shifts and cyclical behaviors by leveraging both historical and anticipatory signals, thus improving robustness and reducing information loss in complex sequences [9], [17], [18], [22].

C. Multi-Head Attention Mechanism (Global MHA Layer)

To further enhance temporal modeling, a MHA module is incorporated after the Bi-LSTM layer. MHA enables the network to jointly attend to information from multiple representation subspaces, which is essential for capturing the dynamic nature of financial time series [10], [14], [18], [23].

The core of MHA is the scaled dot-product attention, which computes attention scores between queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$).

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{6}$$

Instead of a single attention function, MHA utilizes h parallel attention "heads", allowing the model to capture dependencies at different temporal and feature levels. The outputs of these heads are then concatenated.

$$\text{head}_i = \text{Attention}(\mathbf{H}\mathbf{W}_i^Q, \mathbf{H}\mathbf{W}_i^K, \mathbf{H}\mathbf{W}_i^V) \tag{7}$$

$$\mathrm{MHA}(\mathbf{H}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)\mathbf{W}^O \tag{8}$$

Integrating MHA after the Bi-LSTM layer allows the model to form a richer contextual representation by highlighting critical segments like price jumps and trend reversals, thereby improving robustness and computational efficiency [10], [12], [14], [18], [19], [24], [25].

D. Mixture of Experts (MoE Module)

To enhance adaptability to dynamic market regimes, the architecture incorporates a MoE module. MoE uses multiple specialized subnetworks ("experts") and a gating network that dynamically assigns weights to their contributions based on the input, making it effective for modeling the complex, multimodal nature of financial time series [11], [12], [26].

Given an input $\mathbf{Z}$, the model includes K expert networks, $\mathcal{E}_k$, each producing a distinct output $\mathbf{Y}_k$. A gating network computes a weight, π_k, for each expert.

$$\mathbf{Y}_k = \mathcal{E}_k(\mathbf{Z}), \quad \forall k \in \{1, \ldots, K\} \tag{9}$$

$$\pi_k = \frac{\exp\left(\mathbf{w}_k^\top \mathbf{Z}\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^\top \mathbf{Z}\right)}, \quad k = 1, \ldots, K \tag{10}$$

The final output, $\widehat{\mathbf{Y}}$, is a weighted aggregation of the expert predictions, allowing the model to capture uncertainty and learn conditional behaviors.

$$\widehat{\mathbf{Y}} = \sum_{k=1}^{K} \pi_k \, \mathbf{Y}_k \tag{11}$$

This approach is particularly advantageous for financial forecasting, as it provides a principled way to model regime-switching behavior and improves generalization in noisy, high-volatility environments [1], [3], [4], [8], [11], [19].

E. Output Projection (Fully Connected Layer)

The final component of the architecture is an FC output layer that maps the latent representation from the MoE module to the final prediction space. This lightweight linear projection converts the deeply processed representations into real-valued forecasts, such as EUR/USD future price positions, ensuring computational efficiency [18].

Given the output $\widehat{\mathbf{Y}}$ from the MoE layer, the FC layer projects each time step independently:

$$\widehat{\mathbf{P}}_t = \mathbf{W}_f \widehat{\mathbf{Y}}_t + \mathbf{b}_f, \quad t = 1, \ldots, T' \tag{12}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are learnable weights and biases.

IV. Data Description and Preprocessing

A. Dataset Description

We obtained the dataset used in this study from the Dukascopy Historical Data Feed, a widely used provider of high-frequency forex data [18]. The data, spanning from January 1, 2010, to February 18, 2025, includes 94,391 hourly records for the EUR/USD pair, providing sufficient granularity to cover a range of market regimes [3], [4], [17], [22].

Each time step is a 4-dimensional vector of (OHLC) prices, which are standard features in financial forecasting [1], [2], [7]. The 'Close' price was used as the prediction target. The data is based on BID prices, and periods of inactivity ("flat" bars) were filtered out to enhance signal quality [5], [19], [27]. All timestamps are aligned to GMT (UTC+0) to ensure temporal consistency [17].

TABLE I
SAMPLE RECORDS FROM THE EUR/USD HOURLY OHLC DATASET.

Open	High	Low	Close
1.43283	1.43303	1.43224	1.43276
1.43287	1.43305	1.43206	1.43249
1.43279	1.43305	1.43218	1.43278

B. Preprocessing Steps

We applied a rigorous preprocessing pipeline to the raw EUR/USD hourly data, which comprises 94,390 records spanning from January 2010 to February 2025. The pipeline began with the removal of any rows containing missing values. Subsequently, the input feature matrix ($\mathbf{X}$) was constructed from the four OHLC price components, while the target variable (y) was defined as the 'Close' price of the subsequent hour.

To preserve temporal integrity, a crucial step for time-series forecasting, the dataset was split chronologically without shuffling [25]. The partitioning was set to **96% for training, 2% for validation, and 2% for testing**. This division allocated 90,614 records to the training set, while the validation and test sets each received 1,888 consecutive records. This partitioning strategy was deliberately chosen to maximize the historical data available for training (96%), which is crucial for a deep learning model to capture complex, long-term temporal patterns, while still reserving a sufficiently large and entirely unseen future period (2% or approximately 78 days) for a robust and realistic evaluation of its generalization capabilities [3], [18].

Following the split, z-score normalization was performed using a `StandardScaler` instance fitted **only on the training data**; this scaler was then consistently applied across all three sets. Finally, the normalized data was transformed into overlapping sequences using a sliding window of $T = 3$ hourly observations, creating the final input-output pairs ($\mathbf{X}_{\text{train, val, test}}$ and $y_{\text{train, val, test}}$) for supervised learning [5], [28].

V. Experiments

This section presents the empirical evaluation of our proposed hybrid forecasting architecture, focusing on the final model (V8). We detail the experimental setup and evaluation protocol, which includes data splitting, normalization, and performance metrics. To rigorously assess the model, we compare it against traditional statistical benchmarks, such as ARIMA and GARCH, as well as established deep learning baselines. Furthermore, we analyze the contribution of different architectural components through A rigorous ablative analysis across ten model variants (V1–V10). The aim is to demonstrate the superiority of our integrated approach (V8) and quantify the utility of its modules for hour-ahead EUR/USD forecasting [7], [29].

A. Experimental Setup

All experiments were conducted in a Python 3.12 environment using TensorFlow 2.18 and Keras 3.8, with GPU acceleration provided by an NVIDIA GeForce GTX 1660 Ti. To enhance computational performance, a mixed-precision policy ('mixed_float16') was enabled. Core model components were implemented using standard Keras APIs, while the Mixture of Experts (MoE) module was a custom layer designed for flexible control over expert behavior [11], [12]. For reproducibility, a global random seed was set to 42 for TensorFlow operations.

The models were trained for a target of 60 epochs with a **batch size of 5000**, adhering strictly to the chronological data partitioning established during preprocessing. We employed the AdamW optimizer [30] with an **initial learning rate of 0.01**. **These key hyperparameters were determined through preliminary experiments by selecting the values that yielded the best performance on the validation set.** To ensure stable convergence and prevent overfitting, two key callbacks were used: a `ReduceLROnPlateau` learning rate scheduler, which reduces the learning rate by a factor of 0.1 if the validation loss does not improve for a patience of 1 epoch, and an `EarlyStopping` callback that terminates training if the validation loss fails to improve for 20 consecutive epochs, restoring the weights from the best-performing epoch. Input sequences were consistently generated using a fixed sliding window of three hourly observations for a one-step-ahead prediction target.

A noteworthy characteristic observed during training was that the validation loss was consistently lower than the training loss. This is a known artifact, primarily attributed to the regularization effect of Batch Normalization layers, which are active only during training. This pattern, combined with the model's strong performance on the final test set, indicates effective generalization and that overfitting was successfully avoided.

B. Evaluation Protocol

We rigorously assessed the model's performance on the held-out test and validation sets, using the chronological partitioning established during preprocessing to ensure a fair and realistic evaluation [18]. The final reported metrics were calculated using the model checkpoint from the epoch that achieved the lowest validation loss during training, which was epoch 51.

Prior to training, z-score normalization was applied using statistics computed **only** from the training set, a crucial step to ensure a reliable evaluation of the model's generalization capabilities [3], [18]. Input sequences were then generated using a sliding window of $T = 3$ consecutive hourly time steps, mapping the four OHLC features to a one-step-ahead 'Close' price target.

1) Performance Metrics: Model performance was assessed using four standard regression metrics to capture prediction precision, robustness to outliers, and explained variance [1].

2) Performance Metrics: Model performance was assessed using four standard regression metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2) [1]. All metrics were computed on the held-out test set to ensure a fair and realistic evaluation [18].

C. Baseline Models

To evaluate our proposed architecture, we implemented two categories of baseline models: traditional statistical models and a series of deep learning variants.

a) Statistical Baselines: We contrast our approach with traditional statistical models, such as ARIMA and GARCH, which are standard benchmarks in financial econometrics [6], [22]. These models often fail to capture the nonlinear and non-stationary nature of financial data, a limitation our deep learning framework is designed to overcome [3], [4], [7], [18].

b) Deep Learning Baselines: We also developed several neural network models (V1–V5) to serve as strong deep learning baselines. The simplest, **Model V1**, combines a 1D CNN with a Bi-LSTM for temporal modeling. To evaluate more sophisticated mechanisms, **Models V2, V3, and V4** progressively integrate attention mechanisms and residual connections into the feature extraction stage. Finally, **Model V5** explores the impact of placing a Multi-Head Attention layer *after* the Bi-LSTM to dynamically weigh time steps [10]. All models were trained under the same protocol to ensure a fair comparison.

D. Model Variants and Comparative Results

1) Goal and Variant Overview: To systematically evaluate the contribution of individual architectural enhancements to predictive performance, we developed and assessed ten model variants (V1–V10). Each variant incrementally integrates or modifies specific advanced components, allowing for a detailed comparison of their impact on forecasting accuracy. The architectures of the variants are summarized as follows:

- **V1**: A baseline model with CNN-1D layers followed by a Bi-LSTM and a final attention layer.
- **V2**: Enhances the CNN blocks with MHA for feature extraction before the Bi-LSTM.
- **V3**: Incorporates the MHA-enhanced CNNs blocks into a residual structure RB-MHA for improved feature learning.
- **V4**: Replaces the final attention mechanism with Additive Attention (Bahdanau-style) after the Bi-LSTM layer.
- **V5**: Uses MHA after the Bi-LSTM layer instead of Additive Attention.
- **V6**: Adds an intermediate dense layer (`Dense(256)`) after the post-Bi-LSTM MHA to consolidate features.
- **V7**: Introduces a MoE module after the post-Bi-LSTM MHA, enhancing the model's ability to adapt to different market regimes.
- **V8**: Our final proposed model, which combines the architecture of V6 with an MoE module, creating a

TABLE II
MODEL COMPLEXITY SUMMARY

Model	Trainable Params	Non-trainable Params	Total Params
V1	336,737	832	337,569
V2	337,625	864	338,489
V3	337,681	880	338,561
V4	338,081	880	338,961
V5	434,261	880	435,141
V6	670,753	1392	672,145
V7	675,629	1008	676,637
V8	795,633	1008	796,641
V9	649,141	1008	650,149
V10	649,981	1008	650,989

TABLE III
COMPARATIVE PERFORMANCE METRICS ON TEST SET

Model	MAE	MSE	RMSE	R^2
V1	0.003 050	0.000 011	0.003 336	**0.986540**
V2	0.003 947	0.000 029	0.005 381	0.878 689
V3	0.004 655	0.000 036	0.006 027	0.847 805
V4	0.001 840	0.000 006	0.002 440	0.975 062
V5	0.009 378	0.000 106	0.010 290	0.556 439
V6	0.002 296	0.000 008	0.002 762	0.968 039
V7	0.005 475	0.000 040	0.006 333	0.831 957
V8	**0.001345**	**0.000003**	**0.001863**	0.985 467
V9	0.002 294	0.000 010	0.003 228	0.956 346
V10	0.001 735	0.000 005	0.002 168	0.980 309

Note: The best result in each column is highlighted in bold.

comprehensive RB-MHA → Bi-LSTM → MHA → MoE → `Dense(256)` pipeline.

- **V9**: Explores an alternative hybrid by combining the RB-MHA block with a post-Bi-LSTM Additive Attention layer and an MoE module.
- **V10**: Tests a variant using Additive Attention within the initial residual block, followed by a post-Bi-LSTM Additive Attention layer and an MoE module.

2) Model Complexity Analysis: The complexity of each model variant, measured by the number of trainable and total parameters, is summarized in Table II. The parameter count generally increases with the integration of more sophisticated components. The initial variants (V1–V4) have a complexity of approximately 340K parameters. The introduction of a post-Bi-LSTM MHA layer (V5) and an intermediate dense layer (V6) moderately increases the count. A significant jump occurs with the addition of the MoE module (V7 and subsequent models), pushing the total parameters to over 650K. Our final proposed model, V8, is the most complex, with approximately 796K parameters, reflecting its rich architecture designed for high performance.

3) Performance Evaluation on Test Set: The predictive performance of all model variants was rigorously evaluated on the held-out test set using MAE, MSE, RMSE, and R^2 metrics. The results, based on the best model checkpoint selected using validation set performance, are presented in Table III.

4) Comparative Analysis of Results:

a) Overall Performance and Final Model Selection: The experimental results in Table III identify **Model V8** as the most robust and well-rounded architecture. On the crucial held-out test set, it delivered the lowest error rates with an **RMSE of 0.001863** and an **MAE of 0.001345**. This selection is made despite minor trade-offs with other variants. For instance, Model V1's marginally higher R^2 score is outweighed by its significantly larger prediction errors, a critical factor in financial forecasting. Similarly, Model V8's comprehensive performance on the final test set confirms its stronger generalization compared to Model V6, which only showed a slight MAE advantage on the validation set. Given the paramount importance of minimizing error magnitude, Model V8 is confirmed as the most effective and reliable architecture.

b) Impact of Architectural Components: The systematic comparison of variants reveals the incremental benefits of each component. While simple models like V1 performed well, the integration of advanced features in V8 led to the best overall result. The introduction of a Mixture-of-Experts (MoE) layer proved to be a critical enhancement. For instance, comparing V8 to its non-MoE equivalent, V6, shows that the addition of the MoE module resulted in a **32.5% reduction in RMSE**. A paired t-test confirmed this observation, yielding a **p-value ¡ 0.001**, which provides strong statistical evidence that the MoE module's inclusion is a critical and statistically significant contributor to the model's accuracy. Similarly, the importance of the intermediate **Dense(256) layer** is evident when comparing V6 to V5. Adding this layer after the attention mechanism resulted in a significant **73.2% reduction in RMSE**, highlighting its vital role in consolidating features before the final prediction stages.

c) Component Impact Summary and Final Model Justification: The comprehensive analysis confirms that the synergistic combination of a Residual Block with MHA, a Bi-LSTM, a post-Bi-LSTM MHA layer, a dense consolidation layer, and an MoE module in Model V8 creates a highly effective framework. It successfully balances complexity and performance, achieving the lowest error rates among all tested variants. While other models, such as V1 and V10, also showed strong R^2 scores, their higher error metrics make them less reliable. **To formally validate this, a paired t-test between the errors of Model V8 and the baseline V1 confirmed that V8's superior predictive accuracy is statistically significant** ($p < 0.001$)**.** Therefore, we select **V8 as our final proposed model** due to its higher, statistically-validated predictive accuracy and robust architectural design.

E. Model Interpretability

While the proposed V8 model demonstrates strong predictive performance, understanding its internal mechanisms is valuable for building trust. The incorporated attention layers (both in the residual block and after the BiLSTM) offer the potential to visualize the model's focus on different input features and time steps [10], [25]. Additionally, the Mixture of Experts MoE module allows for analyzing expert utilization patterns, which can reveal specialization across different market regimes [11], [19].

Although a detailed interpretability analysis using methods like SHAP was beyond the scope of this study, these architectural components provide clear avenues for future work in understanding the model's decision-making process, as detailed further in Section VI-0b.

VI. Conclusion and Future Work

a) Conclusion: This study proposed a novel hybrid deep learning architecture integrating RB-MHA, BiLSTM, MHA, and MoE modules for hour-ahead EUR/USD forecasting. Experimental results identified model **V8** as the best-performing variant, achieving outstanding accuracy (R^2**=0.9855**, **RMSE=0.00186**, **MAE=0.00135**). The proposed model significantly outperformed baseline methods, with RMSE and MAE reductions of **44.1%** and **55.9%**, respectively, validated statistically. This architecture demonstrates superior robustness and accuracy, setting a new benchmark for short-term Forex forecasting.

b) Limitations: Despite strong results, our study has limitations. The reported performance relies on the best epoch without averaging multiple runs across random seeds. Statistical validation was restricted to pairwise t-tests without a comprehensive ablation study on the final model. Additionally, evaluations focused solely on the EUR/USD pair with a one-hour horizon; thus, generalizability to other pairs and forecasting horizons remains to be validated.

c) Future Work: Future research should involve a comprehensive ablation study and rigorous statistical analysis (e.g., ANOVA) to better quantify component contributions. Evaluations across various currency pairs, different forecasting horizons, and higher-frequency data are necessary to confirm generalizability. Additionally, integrating exogenous features (macroeconomic or sentiment data) and developing computationally efficient models, possibly with sparse MoE techniques, could enhance real-time deployment. [11], [12]. Lastly, deeper interpretability analyses using methods like SHAP are recommended to elucidate the model's decision-making process. [31].

Acknowledgments

The authors used generative AI tools for editing and formatting but take full responsibility for the final content.

References

[1] A. Abed, O. Serguieva, and N. Appiah, "Improving currency exchange rate prediction with hybrid models and feature selection," *Financ. Innov.*, vol. 8, no. 1, p. 51, 2022.

[2] B. Alazab, R. Awajan, S. Mesleh, and A. Y. Al-Omari, "Forex market directional trends forecasting with bidirectional-lstm neural network," *J. Forecast.*, vol. 41, no. 6, pp. 1175–1186, 2022.

[3] E. A. Ahmed and L. M. Al-Essa, "A novel hybrid deep learning method for accurate exchange rate prediction," *Risks*, vol. 12, no. 9, p. 139, 2024.

[4] J. Degiannakis and G. Filis, "Forecasting exchange rates with nonlinear models: A review," *J. Econ. Surveys*, vol. 34, no. 4, pp. 757–787, 2020.

[5] A. Baashar, I. Meziane, M. G. Alemany, and H. Nebot, "A novel residual lstm-based stock price predictor exploiting technical indicators," *Expert Syst. Appl.*, vol. 189, p. 116115, 2022.

[6] M. Wang, X. Wang, and M. Gong, "Foreign exchange forecasting models: Arima and lstm comparison," *J. Risk Financ. Manag.*, vol. 14, no. 11, p. 535, 2021.

[7] M. Sezer, B. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Appl. Soft Comput.*, vol. 90, p. 106181, 2020.

[8] P. Chang, C. Fan, and C. Liu, "Hybridizing technical indicator and news sentiment with lstm for stock trend prediction," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 12, pp. 7422–7431, 2021.

[9] F. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018.

[10] A. Vaswani and et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[11] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, and G. Hinton, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," arXiv preprint arXiv:1701.06538, 2017, arXiv:1701.06538.

[12] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, pp. 1–39, 2022.

[13] X. Huang, Z. Yang, and J. Wang, "Densely connected convolutional networks for financial time series forecasting," *Expert Syst. Appl.*, vol. 186, p. 115710, 2021.

[14] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764, 2021.

[15] J. Chen, J. Wu, and H. Huang, "Deep residual network with attention mechanism for exchange rate prediction," *Comput. Econ.*, vol. 58, no. 2, pp. 543–563, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE CVPR*, pp. 770–778, 2016.

[17] M. Jin, Y. Liu, and X. Zhu, "Stock market trend prediction using attention-based cnn-lstm model," *IEEE Access*, vol. 10, pp. 14 849–14 862, 2022.

[18] M. Belletreche and et al., "Hybrid attention-based deep neural networks for short-term wind power forecasting using meteorological data in desert regions," *Sci. Rep.*, vol. 14, no. 1, p. 21842, 2024.

[19] Z. Wei and et al., "Cross-market mixture of experts model for global stock prediction," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 2, p. 15, 2023.

[20] Z. Wang, J. He, and K. Xu, "Attention-based deep residual learning for stock market prediction," *Complexity*, vol. 2021, p. 5539568, 2021.

[21] D. Shao, B. Chen, Y. Hu, and Z. Yin, "A novel cnn-bilstm-am method for time series forecasting," *Appl. Intell.*, vol. 53, pp. 10 733–10 749, 2023.

[22] L. Zhao and W. Q. Yan, "Prediction of currency exchange rate based on transformers," *J. Risk Financ. Manag.*, vol. 17, no. 8, p. 332, 2024.

[23] A. Vasudevan and R. S. Rani, "Multi-head attention and gated cnn-lstm for foreign exchange rate forecasting," *Int. J. Comput. Appl.*, vol. 44, no. 4, pp. 359–370, 2022.

[24] G. Kang, W. Chen, and J. Yang, "Attention-based neural network for time series prediction of financial indices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 1680–1692, 2023.

[25] M. Yang, K. Zhou, L. Zhou, and L. Yang, "Explaining deep learning for time-series forecasting: analysis of model attention and prediction uncertainty," *Knowl.-Based Syst.*, vol. 260, p. 110021, 2023.

[26] Z. Chen, Y. Li, B. Sun, and L. Chen, "Towards understanding the mixture-of-experts layer in deep learning," *Proc. NeurIPS*, pp. 25 679–25 692, 2022.

[27] J. Huang, X. Zhang, and C. Li, "Exchange rate forecasting using improved ceemdan and lstm with a denoising framework," *Expert Syst. Appl.*, vol. 207, p. 117834, 2022.

[28] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data," *PLoS One*, vol. 14, no. 2, p. e0212320, 2019.

[29] C. Zhang, N. N. A. Sjarif, R. Ibrahim, and M. K. Khan, "Deep learning models for price forecasting of financial time series: a review of recent advancements (2020–2022)," *WIREs Data Min. Knowl. Discov.*, vol. 13, no. 1, p. e1519, 2023.

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[31] X. Kong and et al., "Deep learning for time series forecasting: a survey," *Int. J. Mach. Learn. Cybern.*, vol. 16, no. 2, pp. 481–506, 2025.