

Received: xxx Accepted: xxx Published: xxx.

DOI: [10.30473/coam.2026.77493.1402](https://doi.org/10.30473/coam.2026.77493.1402)

Volume xxx, Issue xxx, p.p. 1-55: xxx

Research Article



Open Access

Control and Optimization in Applied Mathematics - COAM

Multi-Timeframe Forex Trend Prediction via Ichimoku-Based Feature Engineering and Starfish-Optimized CNN-LSTM/CNN-GRU Networks

Seyedeh Yasamin Hashemi[✉], Omid Solaymani Fard[✉]

Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

✉ Correspondence:

Omid Solaymani Fard

E-mail:

soleimani@ferdowsi.um.ac.ir

How to Cite

Hashemi, S. Y., Solaymani Fard, O. (2027). "Multi-timeframe Forex trend prediction via Ichimoku-based feature engineering and starfish-optimized CNN-LSTM/CNN-GRU networks". *Control and Optimization in Applied Mathematics*, X(x), 1-55. <https://doi.org/10.30473/coam.2026.77493.1402>

Abstract. As the Forex market becomes increasingly complex, accurate directional forecasting is essential for effective trading decisions. This study proposes a multi-timeframe framework to evaluate directional trend prediction for the EUR/USD exchange rate using Ichimoku Cloud-inspired feature engineering and CNN-based hybrid architectures, including CNN, CNN-LSTM, and CNN-GRU. The study optimizes model hyperparameters using the Starfish Optimization Algorithm (SFOA). Specifically, the study investigates multi-step trend prediction for the H1 and H4 timeframes, and one-step-ahead prediction for the daily (D1) timeframe. Historical EUR/USD data are used to train continuous forecasting models, whose outputs are transformed into discrete trend classification labels. The framework ultimately employs a voting mechanism to determine the final trend decision at the daily (D1) timeframe. To assess the model's zero-shot classification capability and generalization, eight additional currency pairs are evaluated using the same evaluation criteria at the relevant stages to assess both individual model performance and overall framework effectiveness. Regression performance is assessed using MSE, MAE, and MAPE, while classification performance is evaluated using Accuracy and F1-score. Statistical robustness is further examined using the Diebold-Mariano test and bootstrap analysis. Experimental results demonstrate that incorporating GRU layers into CNN architectures consistently improves forecasting performance across multiple timeframes and currency pairs, with the proposed IICGS architecture achieving the most competitive overall performance. Selected results and implementation code are publicly available on GitHub.

Keywords. Forex trend prediction, Deep learning (CNN, LSTM, GRU), Ichimoku-based feature engineering, Starfish Optimization Algorithm, Diebold-Mariano test.

MSC. 68T07; 91G15; 90C59.

<https://mathco.journals.pnu.ac.ir>

©2026 by the authors. Licensee PNU, Tehran, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY4.0) (<http://creativecommons.org/licenses/by/4.0>)

1 Introduction

The foreign exchange (Forex) market is widely recognized for its high liquidity, substantial volatility, decentralization, and rapid transaction execution. These characteristics have attracted considerable attention from both researchers and market practitioners, particularly in algorithmic trading, wherein automated systems use historical market data and computational models to optimize trading strategies. In this context, technical analysis serves as a foundational methodology, utilizing historical price data—such as open, high, low, and close prices—along with mathematical indicators to identify market trends, price behavior patterns, and potential reversals. Among these indicators, the Ichimoku Kinko Hyo (Ichimoku Cloud) stands out for its comprehensive nature, offering insights into support and resistance levels, trend direction, momentum, and trading signals. Developed in pre-World War II Japan by journalist Goichi Hosoda under the pseudonym “Ichimoku Sanjin” (meaning “at a glance”), this tool was designed to provide a holistic “mountain view” of market dynamics by integrating multiple moving averages and forward-looking components. Despite its historical origins and initially complex manual calculations, the Ichimoku Cloud is still widely used in modern trading applications for both short-term and long-term market analysis [8].

Recent advances in deep learning have substantially increased interest in financial forecasting and trend prediction, particularly in the foreign exchange (Forex) market. Models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Networks (CNNs) have demonstrated strong capabilities in capturing nonlinear and complex temporal patterns. Their effectiveness in modeling challenging dynamical systems has also been demonstrated beyond financial applications. For instance, Mirzajani et al. [27] proposed feedback-based LSTM architectures for multivariate prediction of the chaotic Lorenz system, achieving higher predictive accuracy than conventional RNN and LSTM models. These findings highlight the potential of recurrent architectures to capture complex temporal dependencies in nonlinear and chaotic systems, which are also characteristic of financial markets. In the Forex domain, a systematic literature review by Ayitey Junior et al. [1] identified a growing shift toward hybrid machine learning and deep learning approaches, emphasizing their potential to improve forecasting accuracy in noisy financial time series. Consistent with this trend, Islam et al. [15] proposed a hybrid GRU-LSTM architecture that outperformed standalone GRU and LSTM models in forecasting several major currency pairs. Similarly, Mabrouk et al. [25] developed a CNN-GRU-based trading framework that integrated technical indicators with deep learning for daily trading-signal prediction, achieving superior forecasting accuracy and trading performance compared with conventional machine learning methods. Collectively, these studies demonstrate the growing potential of deep and hybrid learning architectures for capturing the nonlinear and dynamic behavior of financial markets. However, the predictive performance of these models depends heavily on appropriate hyperparameter selection. Consequently, meta-

heuristic optimization algorithms have become increasingly popular for automatically tuning hyperparameters and improving forecasting performance. For example, Gülmez [11] proposed a metaheuristic-based LSTM model for stock price prediction and reported significant improvements in MAE, MSE, and MAPE through optimized hyperparameter selection. Owing to their ability to efficiently explore large and complex search spaces, metaheuristic algorithms are particularly well suited for financial prediction problems. Motivated by these advantages, this study employs the Starfish Optimization Algorithm (SFOA) [31] to optimize the hyperparameters of the proposed deep learning models. Inspired by the adaptive and regenerative behavior of starfish, SFOA performs population-based exploration to identify suitable hyperparameter configurations, including network architecture, thereby improving convergence stability and reducing the risk of premature convergence to local optima.

Beyond model architecture and hyperparameter optimization, recent studies have also emphasized the importance of feature engineering in enhancing forecasting performance. Deng et al. [5], for instance, proposed a hybrid trading system that combined Ichimoku-based feature extraction with fuzzy rough sets and NSGA-II optimization for crude oil futures forecasting, demonstrating improvements in both profitability and risk management. Likewise, Sevastjanov et al. [29] showed that integrating multiple forecasting models with novel technical indicators can reduce overfitting and improve adaptability under changing market conditions. Özorhan et al. [28] improved EUR/USD trend prediction using ZigZag-based motif discovery and SVM classification, while Khattak et al. [18] integrated Fibonacci indicators with a hybrid CNN-LSTM architecture for cryptocurrency trend prediction, highlighting the importance of profitability-oriented evaluation in practical trading environments. Collectively, these studies indicate that combining advanced deep learning architectures with effective optimization techniques and carefully designed technical indicators provides a promising direction for developing more accurate and robust financial forecasting systems.

Although these studies have reported encouraging results, several limitations remain. Many existing approaches focus mainly on price forecasting accuracy while paying less attention to directional trend prediction, which is often more relevant in trading applications. In addition, most studies rely heavily on regression metrics such as MAE, MSE, and MAPE. While these measures evaluate numerical forecasting error, they do not necessarily reflect whether the predicted market direction is correct. In practical trading, even a small prediction error may still lead to an incorrect trading signal and financial loss. Another limitation is the lack of rigorous statistical validation in many forecasting frameworks. Comparisons between models are often reported without statistical tests capable of determining whether performance differences are truly significant.

To address these limitations, this study proposes a hybrid forecasting framework that combines Ichimoku Cloud feature engineering with CNN-based deep learning architectures for multi-timeframe trend prediction. The proposed framework considers H1, H4, and D1 time-

frames to capture short-, medium-, and long-term market behavior. Regression models are trained to perform one-step-ahead price forecasting at each timeframe. At H1 and H4, successive one-step-ahead forecasts are generated, and the predicted prices corresponding to the daily forecasting horizon are compared to determine the associated daily directional trend. At D1, the daily trend is obtained by directly comparing two consecutive one-step-ahead predicted prices. The directional signals from H1, H4, and D1 are then integrated through a voting mechanism to produce a single final daily trend decision. Thus, regression constitutes the training and optimization objective, whereas directional classification represents the final decision-making objective of the framework. Three deep learning architectures—CNN, CNN-LSTM, and CNN-GRU—are investigated. The Starfish Optimization Algorithm (SFOA) is employed for hyperparameter optimization and is compared with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO). Regression performance is evaluated using MSE, MAE, and MAPE, with validation MAE serving as the fitness function for SFOA. At the final decision stage, the directional outputs are assessed using Accuracy and F1-score. The statistical reliability of the regression results is examined using the Diebold–Mariano (DM) test with Holm–Bonferroni correction, together with a moving block bootstrap procedure to assess the stability of the final classification performance. In addition to EUR/USD, eight previously unseen currency pairs are considered to evaluate the zero-shot classification capability of the trained framework. The main contributions of this study are as follows:

- A multi-timeframe hybrid framework is developed that explicitly separates the regression training objective from the final directional decision-making objective. One-step-ahead price forecasts at H1, H4, and D1 are converted into directional signals (by comparing prices aligned with the daily horizon at H1/H4 and consecutive predictions at D1) and aggregated via a voting mechanism to obtain the final daily trend.
- A unified evaluation strategy is established in which MSE, MAE, and MAPE guide model training and optimization, while Accuracy and F1-score evaluate the ultimate trend-prediction performance of the overall framework.
- The modeling and optimization strategy is systematically compared across CNN, CNN-LSTM, and CNN-GRU architectures and against established metaheuristic methods (GA, PSO, GWO), with statistical significance assessed by the DM test (Holm–Bonferroni adjusted) and performance stability examined via moving block bootstrap.
- The zero-shot cross-market robustness of the trained framework is evaluated on eight currency pairs that were not used during training, assessing both regression forecasting quality and the final daily trend-classification performance obtained through the voting mechanism.

The remainder of this paper is organized as follows. Section 2 describes the proposed methodology, data collection process, and model development procedure. Section 3 presents the experimental results and statistical analyses. Section 4 discusses the findings and practical implications of the study, and Section 5 concludes the paper and outlines directions for future research.

2 Material and methods

The way in which the creator of Ichimoku views and analyzes the market forms the basis, motivation, and goal of designing our proposed neural networks. Before exploring the main topic, a brief overview of the concepts behind LSTM, GRU, and CNN is essential. In addition, a detailed explanation of the SFOA metaheuristic algorithm is provided in Appendix.

2.1 Long Short-Term Memory

Hochreiter et al. [12] introduced long short-term memory (LSTM) networks, a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data such as time series, natural language, and audio. Unlike traditional RNNs, which suffer from vanishing or exploding gradients, LSTMs overcome this limitation through a gating mechanism consisting of a forget gate, input gate, and output gate. These gates enable the model to selectively retain, update, or discard information, making LSTMs particularly suitable for financial forecasting tasks such as stock prediction, Forex trend analysis, and cryptocurrency modeling.

Let x_t represent the input vector at time step t and h_{t-1} the hidden state of the previous step. The forget gate, f_t , controls the proportion of the previous cell state, C_{t-1} , to retain:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (1)$$

The input gate, i_t , determines the extent to which new information is stored in the cell state. The candidate cell state, $\tilde{C}_t$, represents the new content to be added:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \quad (3)$$

The cell state, C_t , is then updated as in Equation (4):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t. \quad (4)$$

The output gate, o_t , determines how much of the internal memory is exposed to the next hidden state, h_t :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (5)$$

$$h_t = o_t \odot \tanh(C_t). \quad (6)$$

This gating system enables LSTMs to maintain a dynamic memory of important temporal patterns over long horizons, critical for modeling non-linear dynamics in financial time series.

2.2 Gated Recurrent Unit

The Gated Recurrent Unit (GRU), introduced by Cho et al. [4], is a simplified recurrent neural network architecture designed to tackle challenges such as the problem of vanishing gradients. GRUs use two gates: the update gate, z_t , which balances the retention of past information with the incorporation of new input, and the reset gate, r_t , which controls how much of the previous hidden state is forgotten.

Candidate activation, $\tilde{h}_t$, combines the effect of the reset gate with the current input to generate new information. The final hidden state, h_t , is computed as a weighted sum of the previous hidden state and the candidate activation, modulated by the update gate:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t. \quad (7)$$

GRUs are computationally efficient, requiring fewer parameters than LSTMs, while still effectively capturing dependencies in sequential data. They are widely used in applications like language modeling, time series forecasting, and speech recognition.

While both LSTMs and GRUs present distinct architectural trade-offs, deciding when to deploy each cell structure—or their simplified variants—depends heavily on the underlying statistical behavior of the target time series. Recent taxonomy work by Khaldi et al. [17] systematically evaluated 20 recurrent cell architectures across fundamental time series behaviors (deterministic, random-walk, non-linear, long-memory, and chaotic). Their empirical findings revealed that structural modifications—such as parameter-reduced variants like SLIM—yield statistically significant performance gains tailored to specific series dynamics (e.g., SLIM-LSTM for chaotic dynamics and SLIM-MGU for long-memory processes). This benchmark underscores the necessity of evaluating both standard LSTM and GRU foundations when constructing forecasting frameworks for complex, multi-behavior financial series.

2.3 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are deep learning models that are very effective for tasks involving spatial data, such as images and sequences. The foundational work of LeCun et al. [22] established CNNs as powerful tools for image recognition and classification tasks, demonstrating their ability to achieve high precision in complex data sets.

CNNs are widely recognized for their ability to automatically and adaptively learn spatial hierarchies of features from data through three main types of layers: convolutional layers, pooling layers, and fully connected layers. The convolutional layers are responsible for feature extraction by applying a set of learnable filters (kernels) across the input. This enables the network to detect patterns and features, such as edges, textures, and shapes, which are crucial for tasks such as image recognition [10]. The pooling layers reduce the dimensionality of feature maps, preserving key features while discarding less important information. This not only reduces computational requirements, but also helps to make the network less prone to overfitting [13]. For example, in modern CNNs, max-pooling is widely used to retain the most important aspects of each feature while reducing noise [23]. Finally, the fully connected layer combines the extracted features to produce the final output, typically used for classification or regression tasks. This layer links the learned feature representations to specific output classes, facilitating tasks such as object identification in images. CNNs utilize activation functions, often Rectified Linear Unit (ReLU), to introduce nonlinearities, allowing them to learn complex functions [13].

Beyond image processing, CNNs can be applied effectively to one-dimensional data using 1D convolutional kernels. One-dimensional CNNs (1D CNNs) are particularly well suited for tasks like time series prediction and signal identification, in which the absolute position of features is less critical. For example, combining 1D CNNs with bidirectional LSTM (BiLSTM) networks has been effective in complex prediction tasks that involve volatile data. Lawal et al. [21] developed a hybrid model that combines 1D CNN and BiLSTM to accurately forecast short-term wind speed by extracting spatial and temporal features, demonstrating high performance at various heights above ground level. Similarly, Kim et al. [19] employed a 1D CNN-BiLSTM model for time series predictions of peak electricity demand and marginal price of the system, illustrating that this approach performs better than conventional methods by efficiently capturing sequential dependencies and improving prediction accuracy.

2.4 Ichimoku-Inspired Feature Engineering

The intersections of the Ichimoku indicators Tenkan-sen, Kijun-sen, Senkou Span A, Senkou Span B, and Chikou Span are widely utilized in technical analysis to generate trading signals. Taking inspiration from this approach, we engineered new features based on these intersec-

tions. These features are derived from five key Ichimoku-based trend-identification signals, as described below.

Tenkan-sen and Kijun-sen Cross When the Tenkan-sen (Conversion Line) crosses above the Kijun-sen (Base Line), it signals a potential uptrend. In contrast, a downward cross suggests a possible downward trend (Figure 1).



Figure 1: Tenkan-sen and Kijun-sen Cross

Senkou Span A and Senkou Span B Cross (Cloud Boundary) When Senkou Span A (the faster-moving edge of the cloud) crosses above Senkou Span B, it suggests an uptrend, while a downward cross indicates a downtrend. This intersection defines the “Kumo” (cloud) boundaries and highlights the support and resistance levels (Figure 2).

Tenkan-sen and Kijun-sen Crossing Above or Below the Cloud When both Tenkan-sen and Kijun-sen cross above the cloud (bounded by Senkou Span A and B), it signals a strong uptrend. In contrast, crossing below the cloud indicates a potential downward trend. The cloud serves as a long-term trend indicator, and these crossings provide valuable signals (Figure 3).

Price and Ichimoku Indicators Cross When the price crosses above Tenkan-sen, Kijun-sen, or the cloud, it indicates a potential uptrend and market strength. In contrast, crossing below these indicators signals a potential downward trend, reflecting price movements relative to support and resistance levels (Figure 4).

Chikou Span Cross (Lagging Line) The Chikou Span, a lagging indicator that shifts the closing price backward by 26 periods, provides historical context for trend analysis. When



Figure 2: Senkou Span A and Senkou Span B Cross



Figure 3: Tenkan-sen and Kijun-sen Crossing Above or Below the Cloud



Figure 4: Price and Ichimoku Indicators Cross

the Chikou Span rises above the Tenkan-sen, Kijun-sen, or cloud, it suggests an uptrend. Conversely, when it falls below these indicators, it signals a downtrend, helping identify trend persistence or reversal (Figure 5) [24].



Figure 5: Chikou Span Cross

The mathematical operations used to generate these features include subtraction, maximization, and minimization, which help capture the direction and magnitude of the intersections. Additionally, some features were shifted forward to align with the principles of technical analysis, which rely on analyzing historical data and past price behavior to evaluate current price trends

and forecast future market movements. Table 1 provides a summary of the features derived from these signals.

Table 1: Ichimoku-inspired features.

Feature	Description
T-K	$Tenkan_sen - Kijun_sen$
A-B	$Span\ A - Span\ B$
P-T-K	$Close\ Price - \max(Tenkan_sen, Kijun_sen)$
P-A-B	$Close\ Price - \max(Span\ A, Span\ B)$
T-K-A-B	$\min(Tenkan_sen, Kijun_sen) - \max(Span\ A, Span\ B)$
P-CH	$Close\ Price - Chikou\ Span\ (shifted\ +26)$
CH-T-K	$Chikou\ Span - \max(Tenkan_sen, Kijun_sen)\ (shifted\ +26)$
CH-A-B	$Chikou\ Span - \max(Span\ A, Span\ B)\ (shifted\ +26)$

2.5 Dataset

This study employs a dataset of EUR/USD OHLC prices across three hierarchical timeframes—one-hour (H1), four-hour (H4), and daily (D1)—spanning from September 1, 2019, to September 1, 2024. The data were collected via the MetaTrader 5 platform to ensure high reliability and precision. The proposed multi-timeframe voting framework strategically integrates these complementary resolutions to generate robust, real-world-aligned trading signals. The D1 timeframe serves as the primary decision layer. It effectively filters intraday noise and captures persistent trend structures that align with the medium- to long-term positioning of institutional traders, a concept well-documented in Forex market microstructure research. The H4 timeframe acts as an intermediate confirmatory layer, encapsulating half of each major geographic trading session (Asian, European, and American) within a single candle. This resolution reveals significant intraday momentum shifts and liquidity surges that remain obscured on daily charts yet are excessively noisy on lower timeframes. Finally, the H1 timeframe is incorporated as the highest-frequency component. It enables precise entry and exit timing by detecting short-term support and resistance zones, micro-trends, and order-flow imbalances, while preserving an acceptable signal-to-noise ratio. This hierarchical D1→H4→H1 architecture ensures that the final signal is firmly rooted in dominant market cycles, validated by session-driven dynamics, and optimally timed for execution. Furthermore, the selection of EUR/USD—the world’s most liquid currency pair, exhibiting strong correlations with global risk sentiment—enhances the generalizability and practical relevance of the findings. To improve the predictive accuracy of the models, two distinct datasets are constructed to systematically evaluate the contribution of

Ichimoku-inspired features. The first dataset augments standard OHLC price data with the five canonical components of the Ichimoku Kinko Hyo indicator. The second dataset extends this representation by combining OHLC prices with eight engineered features derived from widely adopted Ichimoku trading strategies. This captures relational and directional market dynamics beyond the original indicator formulation, as detailed in Section 2.4. A sliding window of 50 time steps is applied to both datasets to model recent temporal dependencies while preserving the computational efficiency required for real-time forecasting. This specific window length was selected to cover a complete Ichimoku trend cycle, enabling the model to capture both short-term fluctuations and medium-term directional structures. The data are subsequently partitioned into training and testing subsets using an 80%–20% split, with 20% of the training data allocated as a validation set to monitor performance and ensure proper generalization. Ultimately, the integration of hierarchical timeframes, sophisticated feature engineering, and rigorous data partitioning forms a comprehensive foundation for the robust evaluation of the proposed predictive architectures.

2.6 The Proposed Models

This research contributes to the development of a multivariate time series prediction system that leverages deep learning techniques to forecast Forex price movements. The proposed system consists of three main components trained on the H1, H4, and D1 timeframes. A Convolutional Neural Network (CNN) serves as the base architecture, combined with LSTM and GRU networks for enhanced temporal modeling. The primary focus is on utilizing the Ichimoku technical indicator, which provides high-quality features suitable for these deep learning architectures. To clearly distinguish between different configurations, the following naming conventions are adopted: models using only the original Ichimoku indicators are prefixed with *I* (e.g., IC for Ichimoku-CNN, ICL for Ichimoku-CNN-LSTM, and ICG for Ichimoku-CNN-GRU). Models augmented with the newly engineered Ichimoku-inspired features receive an additional *I* prefix to denote “Improved” status (e.g., IIC, IICL, and IICG).

Furthermore, all configurations undergo hyperparameter optimization via the Starfish Optimization Algorithm (SFOA). To denote this, an *S* is appended to the model name (e.g., ICS, ICLS, and ICGS for standard Ichimoku architectures; IICS, IICLS, and IICGS for the improved versions). The structural frameworks of the baseline IC, ICL, and ICG models are illustrated in Figure 6. Each model employs the Rectified Linear Unit (ReLU) activation function in the hidden layers and a linear activation function in the output layer.

All models were trained for 50 epochs using the Mean Squared Error (MSE) loss function and the Adam optimizer with an initial learning rate of 0.001. Given that the input EUR/USD time-series data were normalized prior to training, this learning rate ensured stable and efficient

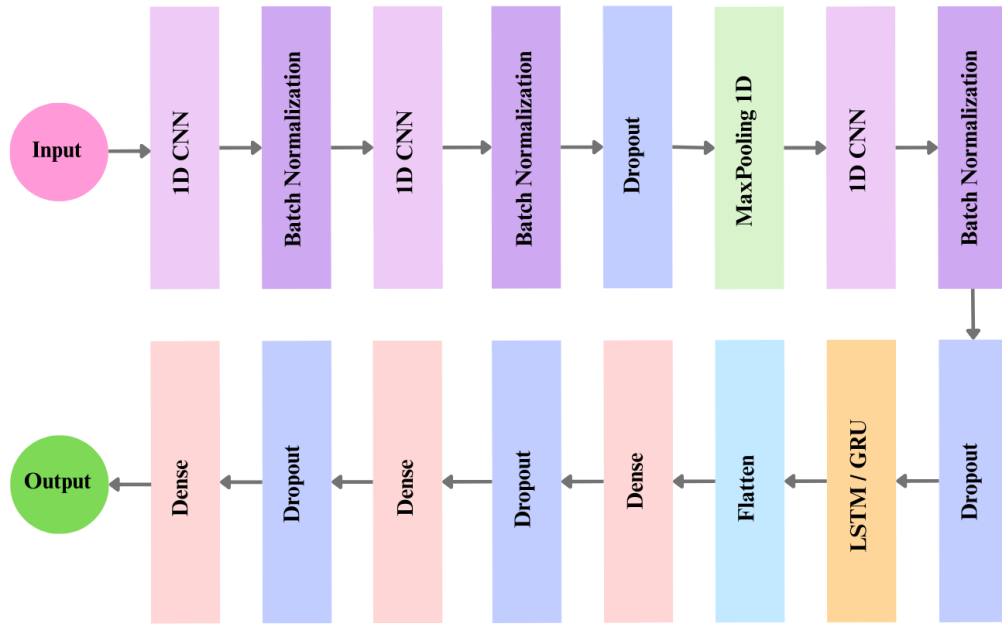


Figure 6: Models' Architecture

convergence. Additionally, the Adam optimizer provides adaptive learning rate adjustments, which stabilizes the training dynamics of deep neural networks. To further enhance convergence and prevent optimization stagnation, a ReduceLROnPlateau learning rate scheduling strategy was applied, dynamically decreasing the learning rate when the validation loss plateaued.

In financial time-series forecasting, overfitting is a common and critical issue, in which a model may achieve strong performance on training data but fail to generalize to unseen samples due to the noisy and non-stationary nature of market data. To mitigate this problem, several regularization techniques were incorporated into the proposed architectures, including Dropout layers, Batch Normalization, and pooling operations. These components collectively improve feature generalization, stabilize training dynamics, and enhance the robustness of the models on out-of-sample data. The system was implemented in Python 3.11.3, utilizing TensorFlow and scikit-learn for deep learning and data preprocessing. All experiments were conducted on a laptop equipped with a 13th Gen Intel® Core™ i7-13620H CPU (2.40 GHz) and 16 GB RAM, running the Windows 11 operating system. To ensure reproducibility and further assess the robustness of the proposed framework, all stochastic components involved in model training and hyperparameter optimization were controlled using fixed random seeds. In particular, each model was trained over 10 independent runs using different fixed random seeds (e.g., 11, 22, 33, 44, 55, 66, 77, 88, 99, and 110) to reduce the influence of random weight initialization and to verify that the reported performance was not associated with a specific initialization.

In addition, the same seed-control strategy was employed in the SFOA-based hyperparameter optimization stage to ensure consistency across the entire experimental procedure.

Table 2 presents the hyperparameter candidate values considered in this study. These parameters determine the model's learning capacity and resistance to overfitting. The SFOA framework systematically optimizes these hyperparameters, ensuring an optimal balance between architectural complexity and generalization performance when training on high-dimensional time series data.

Table 2: Hyperparameter Alternatives

Hyperparameter	Candidate Values
Conv1D Filters	30–150 (step=1)
Conv1D Kernel Size	{3, 5, 7, 9}
Dropout Rate	{0.3, 0.4, 0.5, 0.6}
Neurons (Dense/LSTM/GRU)	30–150 (step=1)

A rule-based aggregation module was developed to harmonize the forecasting outputs obtained from the H1, H4, and D1 timeframes and project them onto a unified daily (24-hour) horizon. Forecasts at different temporal resolutions are treated as complementary trend-following signals that capture short-, medium-, and long-term market dynamics. Within this framework, the H1 and H4 forecasts corresponding to a given 24-hour interval are temporally aligned and consolidated at the daily level together with the direct D1 prediction. Importantly, one-step-ahead forecasts are used exclusively for regression-based performance evaluation and model optimization (MSE, MAE, MAPE, and the SFOA fitness function). For trend extraction, however, a multi-step alignment is applied at H1 and H4 so that the predicted prices corresponding to the daily horizon can be compared, while at D1 the daily trend is obtained by comparing two consecutive one-step-ahead predicted prices. This design reduces trend misclassification: short-horizon predictions may achieve low regression error yet still produce incorrect directional signals. By aligning all timeframes to the daily horizon, the models capture the dominant price movement more reliably. The final daily trend direction is determined by a majority-voting scheme in which the aligned H1, H4, and D1 directional signals participate. This aggregation filters out noisy or conflicting short-term signals and yields a stable estimate of the prevailing daily trend (upward or downward). The overall workflow is summarized in Algorithm 1. Historical price data across the three timeframes are prepared and normalized, hyperparameters are optimized with SFOA, and the models are trained. One-step-ahead predictions are then used to compute regression metrics, while the multi-step-aligned forecasts are converted into directional signals and aggregated by majority voting. The resulting daily trend decisions are evaluated with classification metrics (Accuracy and F1-score).

Algorithm 1 End-to-end multi-timeframe forecasting and decision-making framework using Ichimoku-based features, CNN-based models, and SFOA

- 1: **Input:** Historical EURUSD price data in $\{H1, H4, D1\}$, window length L ,
- 2: CNN-based models ICS, ICLS, ICGS, IICS, IICLS, IICGS,
- 3: optimization algorithm SFOA, number of bootstrap replications B , Random seed set $S = \{s_1, s_2, \dots, s_{10}\}$.
- 4: **Output:** Final classification performance metrics reported as mean $\pm$ SD, and DM test results reported as DM statistics, p -values, and Holm–Bonferroni adjusted p -values.
- 5: **Step 1: Data Preparation**
- 6: **for** each timeframe $tf \in \{H1, H4, D1\}$ **do**
- 7: Load historical price data corresponding to tf .
- 8: Perform Ichimoku-based feature engineering.
- 9: Normalize the extracted features.
- 10: Construct sliding windows of length L .
- 11: Split the dataset into training (80%) and testing (20%) subsets.
- 12: Further split the training subset into training (80%) and validation (20%) subsets.
- 13: **end for**
- 14: **Step 2: Timeframe-specific Hyperparameter Optimization via SFOA**
- 15: **for** each timeframe $tf \in \{H1, H4, D1\}$ **do**
- 16: **for** each model $m \in \{ICS, ICLS, ICGS, IICS, IICLS, IICGS\}$ **do**
- 17: **for** each starfish candidate in SFOA **do**
- 18: Apply candidate hyperparameters to model m for timeframe tf .
- 19: Train model m on the training data of timeframe tf .
- 20: Compute validation MAE.
- 21: Assign the MAE value as the fitness score.
- 22: **end for**
- 23: Select the optimal hyperparameter set for (m, tf) after SFOA convergence.
- 24: **end for**
- 25: **end for**
- 26: **Step 3: Final Training of Optimized Models**
- 27: **for** each timeframe $tf \in \{H1, H4, D1\}$ **do**
- 28: **for** each model m **do**
- 29: Train model m on the entire training set of tf using the optimized hyperparameters.
- 30: **end for**
- 31: **end for**

```

32: Step 4: One-Step-Ahead Price Forecasting and Statistical Regression Evaluation
33: for each timeframe  $tf \in \{H1, H4, D1\}$  do
34:   for each model  $m$  do
35:     Generate one-step-ahead price forecast  $\hat{p}_{t+1}^{tf}$ .
36:     Compute regression metrics: MSE, MAE, and MAPE.
37:   end for
38: end for
39: Step 5: Multi-Step Forecasting for Trend Alignment
40: for each model  $m$  do
41:   Generate aligned multi-step-ahead price forecasts:
42:   H1:  $\hat{p}_{t+24}^{H1}$  (24 steps ahead)
43:   H4:  $\hat{p}_{t+6}^{H4}$  (6 steps ahead)
44:   D1:  $\hat{p}_{t+1}^{D1}$  (1 step ahead)
45: end for
46: Step 6: Trend Extraction
47: for each timeframe  $tf \in \{H1, H4, D1\}$  do
48:   Determine the trend direction by comparing the aligned
49:   forecast with the current actual price.
50: end for
51: Step 7: Daily Trend Decision via Majority Voting
52: Aggregate the aligned trend predictions from  $\{H1, H4, D1\}$ 
53:   using majority voting.
54: Obtain the final daily trend decision.
55: Step 8: Final Classification Evaluation
56: Compare predicted daily trends with ground-truth trends.
57: Compute classification metrics: Accuracy and F1-score.
58: Step 9: Statistical Validation
59: Diebold–Mariano (DM) Test:
60: for each timeframe  $tf \in \{H1, H4, D1\}$  do
61:   for each model pair  $(m_i, m_j)$  do
62:     Compute forecast errors for both models.
63:     Calculate the loss differential:  $d_t = L(e_i) - L(e_j)$ 
64:     Compute the DM statistic and corresponding  $p$ -value.
65:   end for
66: end for

```

- 63: *Holm–Bonferroni Correction:*
- 64: Sort the DM test p -values in ascending order.
- 65: Apply the step-down Holm–Bonferroni procedure to control the family-wise error rate (FWER).
- 66: Compute the Holm-adjusted p -values.
- 67: Determine statistical significance using the adjusted p -values.
- 68: *Moving Block Bootstrap Procedure:*
- 69: **for** $b = 1$ to B **do**
- 70: Generate a bootstrap sample using moving blocks.
- 71: Compute evaluation metrics on the bootstrap sample.
- 72: **end for**
- 73: Calculate the mean and standard deviation of Accuracy and F1-score over bootstrap replications.
- 74: Report the final performance metrics as $\text{mean} \pm \text{SD}$ to assess model robustness.
- 75: **Output:**
- 76: Report Mean $\pm$ SD results, DM test outcomes, and Holm–Bonferroni adjusted p -values.

Statistical reliability is assessed by the Diebold–Mariano (DM) test with Holm–Bonferroni correction for multiple comparisons, and the stability of classification performance is examined via moving-block bootstrap analysis. Together, these steps provide an end-to-end evaluation pipeline that clearly separates the regression training objective from the final directional decision-making objective. To evaluate the cross-market generalization capability of the proposed framework, the models trained exclusively on the EUR/USD currency pair were applied to eight additional currency pairs without any retraining or fine-tuning. For each unseen currency pair, the same Ichimoku-based feature engineering pipeline, data normalization procedure, and multi-timeframe forecasting and voting mechanism were employed. This protocol enables assessment of the framework’s zero-shot transfer performance across different Forex markets.

Remark 1. The stochastic initialization and search phases of Algorithm 1 are controlled via fixed random seeds, with results reported as the mean and standard deviation over 10 independent runs.

Remark 2. The DM test was conducted at a significance level of $\alpha = 0.05$. A p -value lower than 0.05 indicates a statistically significant difference between the forecasting errors of the compared models.

3 Experimental Results

3.1 Evaluation Criteria

The proposed forecasting framework is evaluated using a comprehensive set of regression, classification, and statistical validation metrics. Regression performance is assessed using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), while trend prediction performance is evaluated using Accuracy and F1-score. To further investigate the statistical significance and robustness of the obtained results, the Diebold–Mariano (DM) test and Bootstrap resampling analysis are additionally employed. The DM test determines whether the forecasting accuracy of two competing models differs significantly, whereas Bootstrap analysis evaluates the stability and reliability of the reported performance estimates across repeated samples. To account for the increased probability of false-positive findings resulting from multiple DM comparisons, the Holm–Bonferroni procedure is applied to adjust the resulting p -values while controlling the family-wise error rate (FWER). Together, these evaluation criteria provide a comprehensive assessment of predictive accuracy, classification effectiveness, statistical significance, multiple-comparison control, and model robustness.

MSE measures the average squared difference between predicted ($\hat{y}_i$) and actual values (y_i) as shown in Equation (8):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2. \quad (8)$$

By penalizing larger errors more heavily, MSE is sensitive to outliers and is widely used for regression tasks. It traces its origins to Gauss and Legendre’s foundational work on least squares regression [9].

MAE represents the average magnitude of errors, treating all deviations equally as shown in Equation (9):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|. \quad (9)$$

MAE is more robust to outliers than MSE and provides interpretable results in the same units as the data. It has its roots in robust regression and descriptive statistics [7].

MAPE expresses errors as a percentage of actual values, making it scale-independent as shown in Equation (10):

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100. \quad (10)$$

MAPE is useful for comparing performance across datasets but can produce inflated errors when actual values are near zero. It gained prominence in time-series forecasting and business modeling [26].

Accuracy is a straightforward metric that measures the proportion of correct predictions including true positives and true negatives, relative to the total predictions made as shown in Equation (11).

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \quad (11)$$

While accuracy indicates overall performance, it can be misleading under class imbalance with a dominant class [30].

F1-score is a performance metric used in classification tasks to evaluate the balance between precision and recall. It is defined as the harmonic mean of precision (P) and recall (R), formulated as Equation (12):

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}. \quad (12)$$

Precision (P) is defined as Equation (13):

$$P = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (13)$$

Recall (R) is defined as Equation (14):

$$R = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}. \quad (14)$$

The F1-score provides a single metric that balances the trade-off between precision and recall, particularly useful in situations with imbalanced datasets [30]. This metric has since been adapted widely in machine learning and classification tasks for evaluating model performance. Together, these metrics provide a well-rounded evaluation of the model's effectiveness in predicting financial time series movements.

Diebold-Mariano (DM) Test is a statistical method used to compare the accuracy of two competing forecast models. Developed by Diebold and Mariano [6], this test assesses whether the difference in forecast performance between two models is statistically significant. It is particularly useful in econometrics and time series analysis when evaluating the performance of different forecasting techniques. Let Y_t be the actual value at time t , and let $\hat{Y}_{1,t}$ and $\hat{Y}_{2,t}$ be the forecasts generated by model 1 and model 2, respectively. The forecast errors for each model are defined as Equations (15) and (16):

$$e_{1,t} = Y_t - \hat{Y}_{1,t}, \quad (15)$$

$$e_{2,t} = Y_t - \hat{Y}_{2,t}. \quad (16)$$

The Diebold–Mariano test focuses on the loss differential series, defined as $d_t = L(e_{1,t}) - L(e_{2,t})$, here $L(\cdot)$ denotes a specific loss function, such as squared error or absolute error. The null hypothesis (H_0) states that both competing models possess equal forecast accuracy (i.e., the expected value of the loss differential is zero), while the alternative hypothesis (H_1) implies that one model statistically outperforms the other. The test statistic is based on the sample mean of the loss differentials, $\bar{d}$, and its variance. A modified t-statistic is typically utilized to account for potential autocorrelation in the forecast errors, especially in finite samples. The DM statistic is calculated as Equation (17):

$$DM = \frac{\bar{d}}{\sqrt{\hat{V}(\bar{d})}}, \quad (17)$$

in which $\bar{d} = \frac{1}{T} \sum_{t=1}^T d_t$ and $\hat{V}(\bar{d})$ is a consistent estimate of the asymptotic variance of $\bar{d}$. Under the null hypothesis, this statistic asymptotically follows a standard normal distribution [6].

Moving Block Bootstrap (MBB) is a resampling technique designed for dependent data, particularly time-series observations, where the assumption of independent and identically distributed samples is not appropriate [20]. Unlike the conventional bootstrap, which independently resamples individual observations with replacement, MBB resamples consecutive blocks of observations to preserve the temporal dependence structure of the original series.

Let $X = x_1, x_2, \dots, x_n$ denote an ordered time series and let L represent the block length. A collection of overlapping blocks is first constructed from consecutive observations as

$$B_j = \{x_j, x_{j+1}, \dots, x_{j+L-1}\}, \quad j = 1, 2, \dots, n - L + 1.$$

Each block therefore contains L consecutive observations, and adjacent blocks overlap. During each bootstrap replication, blocks are randomly selected with replacement from the collection of available overlapping blocks and concatenated sequentially to construct a bootstrap sample of the desired length. The statistic of interest is then calculated from the resulting bootstrap sample.

Repeating this procedure generates an empirical bootstrap distribution of the statistic, which can be used to assess its sampling variability and statistical uncertainty. By resampling contiguous blocks rather than individual observations, MBB retains the local dependence structure of the original time series and provides a suitable nonparametric resampling framework for dependent observations [20].

Holm–Bonferroni Correction When multiple statistical hypotheses are tested simultaneously, the probability of obtaining at least one false-positive result increases with the number

of comparisons. To control the family-wise error rate (FWER), the Holm–Bonferroni procedure provides a sequential (step-down) multiple-comparison correction that is less conservative than the classical Bonferroni method while still guaranteeing strong FWER control [14].

Let m denote the total number of statistical tests and let $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(m)}$ be the ordered raw p -values. The Holm procedure sequentially compares each ordered p -value with its adjusted significance threshold:

$$p_{(i)} \leq \frac{\alpha}{m - i + 1}, \quad i = 1, \dots, m, \quad (18)$$

where α is the chosen significance level. Equivalently, the adjusted p -values are computed as

$$p_{(i)}^{\text{Holm}} = \min \left(1, \max_{j \leq i} \{ (m - j + 1) p_{(j)} \} \right). \quad (19)$$

A hypothesis is declared statistically significant if its corresponding adjusted p -value falls below the pre-specified significance level α .

In the present study, this procedure is applied to the complete set of Diebold–Mariano test p -values obtained across all currency pairs, loss functions (MSE, MAE, MAPE), and forecasting horizons (H1, H4, D1), thereby ensuring rigorous control of the multiplicity problem.

3.2 Analysis of Model Performance Based on Metric Values

In this subsection, we evaluate the performance of all previously introduced models across three distinct timeframes (H1, H4, and D1), with the primary objective of predicting trends in the D1 timeframe. This evaluation demonstrates the effectiveness of the proposed framework for Forex trend forecasting while also assessing its robustness under varying market conditions. All experiments were repeated independently 10 times, and all reported results are presented as the mean $\pm$ standard deviation (Mean $\pm$ SD) over these runs.

The predictions generated by these models are subsequently processed by a decision-making module that aggregates the multi-timeframe outputs using a voting-based strategy to produce the final trading signal. The models are trained using historical EUR/USD data covering the period from 2019 to 2024. Before extending the evaluation to additional currency pairs, we first analyze the performance of the six proposed models on the EUR/USD dataset used for training and initial testing. This analysis establishes the benchmark for subsequent experiments. The evaluated models (ICS, ICLS, ICGS, IICS, IICLS, and IICGS) represent different combinations of engineered Ichimoku-based features and optimization strategies. As summarized in

Table 3, the proposed IICGS model consistently achieves the best overall performance across all evaluated timeframes. At the H1 timeframe, it attains the lowest regression errors, with an MSE of $(1.5 \pm 0.1) \times 10^{-5}$, an MAE of 0.0035 ± 0.0001 , and a MAPE of 0.0041 ± 0.0002 , averaged over 10 independent runs. Similar behavior is observed at the H4 timeframe, with IICGS again reporting the lowest errors, achieving an MSE of $(1.5 \pm 0.1) \times 10^{-5}$, an MAE of 0.0029 ± 0.0001 , and a MAPE of 0.0040 ± 0.0002 . At the D1 timeframe, the proposed model maintains its superiority, yielding an MSE of $(1.7 \pm 0.1) \times 10^{-5}$, an MAE of 0.0028 ± 0.0001 , and a MAPE of 0.0033 ± 0.0002 . At the framework level, IICGS also achieves the highest classification performance, reaching an Accuracy of $78.97 \pm 0.50\%$ and an F1-score of 0.66 ± 0.01 . Although other models, particularly IICLS, achieve competitive regression performance in some individual timeframes, their overall framework-level Accuracy and F1-score remain consistently lower than those of IICGS. This observation indicates that superior regression performance at a single temporal resolution does not necessarily translate into better final trading decisions after multi-timeframe voting. Overall, these findings demonstrate that the proposed IICGS architecture provides an effective balance between regression accuracy, cross-timeframe consistency, and classification robustness, making it particularly well suited for multi-timeframe EUR/USD trend prediction. To ensure transparency and reproducibility, representative case studies have been made publicly available.

To rigorously examine the robustness of the proposed approach and assess its applicability across diverse market conditions, the experimental evaluation is extended to eight additional currency pairs: AUD/USD, EUR/SGD, GBP/NZD, NZD/USD, USD/CHF, USD/CNH, USD/SEK, and USD/SGD. These pairs were selected to construct a heterogeneous zero-shot validation set spanning different regional markets, currency compositions, and levels of international FX activity, thereby enabling a more comprehensive assessment of model stability and generalization under varying market structures and data distributions. The set includes highly traded pairs such as AUD/USD and USD/CHF, as well as regionally important markets represented by USD/CNH and USD/SGD. Consistent with the BIS Triennial Central Bank Surveys, AUD/USD, USD/CHF, USD/CNY, USD/SGD, and USD/NZD accounted for substantial shares of global FX turnover in 2019 and 2022 (Bank for International Settlements [2, 3]). The inclusion of EUR/SGD and GBP/NZD further introduces non-USD cross-currency relationships, while USD/SEK extends the evaluation to the Scandinavian market. Collectively, these currency pairs provide geographic and market-structure diversity beyond EUR/USD, while the availability of sufficiently long and consistent historical data over 2019–2024 ensures a reliable out-of-sample evaluation. For all eight additional currency pairs, the testing period is consistently defined from September 1, 2019, to September 1, 2024. Accordingly, all subsequent references to the testing or evaluation of these currency pairs throughout the manuscript refer to this same period.

The performance of the models is evaluated using regression metrics, including MSE, MAE, and MAPE, to quantify the accuracy of forecasting future closing prices. Given that the test data are scaled, all reported results reflect this normalization. These metrics were analyzed across multiple currency pairs for H1, H4, and D1 timeframes, with detailed results summarized in Tables 4, 5, and 6. Visual representations of the comparative results are provided in Figures 7 through 12 in the discussion section.

The comparative evaluation of the proposed models, as shown in Table 4, considers the H1, H4, and D1 timeframes and uses the MSE criterion for evaluation. From the empirical data, a clear performance hierarchy can be established among the models, following the order of increasing error: IICGS < IICLS < IICS < ICGS < ICLS < ICS. At the H1 timeframe, the IICGS model consistently achieves the lowest MSE values across most currency pairs, such as USD/SGD $((0.4 \pm 0.0) \times 10^{-5})$, GBP/NZD $((0.8 \pm 0.0) \times 10^{-5})$, and $((0.9 \pm 0.0) \times 10^{-5})$, highlighting its strong capability in modeling short-term price fluctuations. The IICLS and IICS models also demonstrate competitive performance but remain inferior to IICGS, while the baseline models (ICS and ICLS) exhibit noticeably higher error levels. A similar trend is evident in the H4 timeframe, in which IICGS again outperforms all competing approaches for the majority of currency pairs, including USD/SGD $((1.5 \pm 0.1) \times 10^{-5})$, GBP/NZD $((2.8 \pm 0.1) \times 10^{-5})$, and USD/CHF $((2.9 \pm 0.1) \times 10^{-5})$. The gradual performance improvement from ICS toward IICGS indicates the positive impact of progressively enriching the feature space and model architecture. In the D1 timeframe, IICGS maintains its dominance by achieving the lowest MSE for most currency pairs, such as USD/SGD $((2.0 \pm 0.1) \times 10^{-5})$, EUR/SGD $((4.0 \pm 0.2) \times 10^{-5})$, and GBP/NZD $((4.8 \pm 0.2) \times 10^{-5})$, reflecting strong generalization capability under longer-term market dynamics. Overall, the results confirm that integrating engineered Ichimoku-inspired features within the forecasting framework substantially enhances regression accuracy across all time horizons.

Similarly, Table 5 evaluates the forecasting performance using the Mean Absolute Error (MAE) metric. A highly consistent performance hierarchy is observed across all assets, following the error minimizing pattern IICGS < IICLS < IICS < ICGS < ICLS < ICS. In the H1 timeframe, the IICGS model systematically achieves the lowest MAE values across all currency pairs, such as USD/SGD (0.0014 ± 0.0001) , GBP/NZD (0.0023 ± 0.0001) , and EUR/SGD (0.0023 ± 0.0001) , highlighting its superior ability to capture short-term price movements. This performance ordering is preserved in the H4 timeframe, in which IICGS again dominates for all assets, including USD/SGD (0.0028 ± 0.0001) , EUR/SGD (0.0032 ± 0.0002) , and GBP/NZD (0.0042 ± 0.0002) . The gradual decrease in MAE from ICS toward IICGS reflects the cumulative impact of both SFOA-based optimization and engineered Ichimoku-inspired features. In the D1 timeframe, IICGS continues to deliver the most accurate forecasts, achieving the lowest MAE values for pairs such as USD/SGD (0.0040 ± 0.0002) , EUR/SGD (0.0041 ± 0.0002) , and GBP/NZD (0.0049 ± 0.0002) . This confirms the robustness of the proposed architecture un-

der longer-term forecasting horizons, in which error propagation and regime changes typically challenge model stability.

Table 6 reports the Mean Absolute Percentage Error (MAPE) of the proposed models. The same error minimizing hierarchy ($IICGS < IICLS < IICS < ICGS < ICLS < ICS$) is observed, confirming the robustness of the proposed modeling strategy. In the H1 timeframe, the IICGS model systematically achieves the lowest MAPE values for all currency pairs, including USD/SGD (0.0015 ± 0.0001), GBP/NZD (0.0025 ± 0.0001), and EUR/SGD (0.0025 ± 0.0001). A similar trend is preserved in the H4 timeframe, in which IICGS consistently outperforms competing models across all assets, noting particularly low MAPE values for USD/SGD (0.0030 ± 0.0001), EUR/SGD (0.0042 ± 0.0002), and GBP/NZD (0.0046 ± 0.0002). At the D1 timeframe, IICGS continues to demonstrate superior relative accuracy, achieving the lowest MAPE values for USD/SGD (0.0044 ± 0.0002), EUR/SGD (0.0048 ± 0.0002), and USD/CHF (0.0059 ± 0.0003). While optimization alone contributes to moderate improvements, the integration of engineered Ichimoku-inspired features plays the dominant role in reducing relative forecasting errors.

Table 7 presents the trend prediction performance of the evaluated models, measured using Accuracy and F1-score metrics derived based on the decision-making voting mechanism. Overall, the results indicate that models incorporating the improved Ichimoku-based feature construction generally outperform their original counterparts. In terms of Accuracy, the IICGS model consistently achieves the highest values for most currency pairs, including USD/SGD ($82.7 \pm 0.5\%$), USD/CHF ($79.3 \pm 0.8\%$), and AUD/USD ($78.7 \pm 0.7\%$). From the F1-score perspective, which balances precision and recall, IICGS again demonstrates competitive performance, attaining the highest scores for AUD/USD (0.63 ± 0.02) and GBP/NZD (0.65 ± 0.01). For USD/CNH, IICS attains an F1-score of 0.66 ± 0.01 and IICGS attains 0.65 ± 0.02 , indicating a highly comparable and competitive level of classification balance among the improved models.

As evidenced by the results in Table 8, the statistical robustness of the proposed IICGS framework is validated through rigorous comparative testing across diverse market conditions. The Diebold–Mariano (DM) test was evaluated at significance levels $\alpha = 0.05$ and $\alpha = 0.01$. Accordingly, DM statistics exceeding the critical asymptotic normal thresholds ($|DM| > 1.96$ for $\alpha = 0.05$ and $|DM| > 2.58$ for $\alpha = 0.01$) indicate statistically significant differences in predictive performance. To rigorously control the family-wise error rate arising from multiple pairwise comparisons across currency pairs, metrics, and forecasting horizons, the sequential Holm–Bonferroni step-down adjustment procedure was applied to all computed p -values. As detailed in Table 8, even after controlling for multiple testing, all adjusted p -values satisfy the strict criteria ($p < 0.05$ and $p < 0.01$), confirming that the reductions in MSE, MAE, and MAPE achieved by the proposed IICGS framework over the baseline ICS model are statistically significant rather than the result of random variation. These statistical tests were performed on

the best-performing model obtained from ten independent training runs (each run consisting of full end-to-end training and evaluation under identical hyperparameter settings). This predictive superiority is further supported by the Bootstrap analysis, where the stable mean accuracy values and relatively small standard deviations demonstrate the framework's robustness, consistent generalization capability, and resistance to overfitting across different historical market conditions. These findings collectively confirm that the proposed multi-timeframe voting framework captures meaningful market dynamics beyond short-term stochastic fluctuations.

3.3 Analysis of Model Performance Based on Prediction Charts

A general analysis of the comparative performance across multiple models, as illustrated in Figures 7–12 reveals nuanced insights into predictive accuracy and error dynamics in financial forecasting for various currency pairs. An examination of the comparative charts between actual and predicted prices in ICS models (Figure 7) indicates that, across all three timeframes (H1, H4, and D1), the overall price direction is generally well captured by the proposed model. In the one-hour timeframe, the discrepancy between the predicted values and the actual prices is relatively small, reflecting the model's stronger ability to track short-term price movements. As the timeframe increases to four hours, this gap becomes more pronounced, and a temporal lag in the model's response to price changes can be observed in certain intervals. In the daily timeframe, the spread of the predicted price range further increases; although the general trend direction is largely preserved, occasional inversions between predicted and actual trends are evident. In addition, for the USD/SEK pair, there exists a specific period during which the model is unable to adequately reflect the prevailing market behavior across all three timeframes. During this interval, the predicted prices remain largely range-bound, while the actual market exhibits a clear directional trend. This behavior suggests a reduced sensitivity of the model to gradual trend development under low-volatility conditions rather than an inability to capture abrupt price movements.

Overall, these observations highlight that shorter timeframes yield closer alignment between predicted and actual prices, whereas higher timeframes tend to emphasize broader trend characteristics at the expense of local accuracy. For the ICLS model (Figure 8), the gap between the predicted prices and the actual market values is generally reduced compared to the previous model, and the forecasts exhibit a higher degree of directional consistency with the true price movement. This indicates an improved capability of the model to follow the overall market trend. However, despite this enhanced alignment, the model demonstrates a limited ability to accurately represent periods of heightened price variability, often smoothing out pronounced fluctuations observed in the actual price series. Across different temporal resolutions, a systematic bias pattern can be observed. In the one-hour timeframe, the predicted prices are

Table 3: Performance comparison of hybrid models on the benchmark EUR/USD pair (Mean $\pm$ SD over 10 runs).

Model	Framework									
	H1		H4		D1		Acc. (%)			
	MSE ($\times 10^{-5}$)	MAE	MAPE	MSE ($\times 10^{-5}$)	MAE	MAPE	MSE ($\times 10^{-5}$)	MAE	MAPE	F1
ICS	7.7 $\pm$ 0.3	.0084 $\pm$.0004	.0096 $\pm$.0005	13.4 $\pm$ 0.6	.0103 $\pm$.0005	.0119 $\pm$.0007	9.9 $\pm$ 0.4	.0086 $\pm$.0004	.0098 $\pm$.0005	54.73 $\pm$ 1.2
ICLS	2.7 $\pm$ 0.1	.0048 $\pm$.0002	.0050 $\pm$.0002	10.8 $\pm$ 0.5	.0087 $\pm$.0004	.0100 $\pm$.0005	7.4 $\pm$ 0.3	.0072 $\pm$.0003	.0083 $\pm$.0004	59.79 $\pm$ 1.1
ICGS	2.5 $\pm$ 0.1	.0043 $\pm$.0002	.0045 $\pm$.0002	4.3 $\pm$ 0.2	.0060 $\pm$.0003	.0076 $\pm$.0004	5.7 $\pm$ 0.2	.0063 $\pm$.0003	.0071 $\pm$.0003	61.70 $\pm$ 1.0
ICCS	2.0 $\pm$ 0.1	.0043 $\pm$.0002	.0049 $\pm$.0003	4.0 $\pm$ 0.2	.0055 $\pm$.0002	.0069 $\pm$.0004	2.2 $\pm$ 0.1	.0038 $\pm$.0002	.0043 $\pm$.0002	75.40 $\pm$ 0.8
ICLS	1.9 $\pm$ 0.1	.0042 $\pm$.0002	.0049 $\pm$.0003	3.6 $\pm$ 0.1	.0052 $\pm$.0002	.0061 $\pm$.0003	1.9 $\pm$ 0.1	.0034 $\pm$.0001	.0040 $\pm$.0002	77.16 $\pm$ 0.7
ICGS	1.5 $\pm$ 0.1	.0035 $\pm$.0001	.0041 $\pm$.0002	1.5 $\pm$ 0.1	.0029 $\pm$.0001	.0040 $\pm$.0002	1.7 $\pm$ 0.1	.0028 $\pm$.0001	.0033 $\pm$.0002	78.97 $\pm$ 0.5
ICCS										0.66 $\pm$.01

Table 4: Evaluation of models using MSE ($\times 10^{-5}$), (Mean $\pm$ SD over 10 independent runs).

Ticker	H1									
	ICS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS
AUD/USD	18.9 $\pm$.9	12.6 $\pm$.6	7.4 $\pm$.4	5.5 $\pm$.3	3.6 $\pm$.2	2.2 $\pm$.1	33.6 $\pm$ 1.6	27.3 $\pm$ 1.3	14.7 $\pm$.7	9.7 $\pm$.5
EUR/SGD	6.5 $\pm$.3	5.7 $\pm$.3	3.4 $\pm$.2	2.3 $\pm$.1	1.5 $\pm$.1	0.9 $\pm$.0	8.2 $\pm$.4	7.2 $\pm$.3	5.5 $\pm$.3	4.3 $\pm$.2
GBP/NZD	6.2 $\pm$.3	5.0 $\pm$.2	3.3 $\pm$.2	2.3 $\pm$.1	1.5 $\pm$.1	0.8 $\pm$.0	21.0 $\pm$ 1.0	11.6 $\pm$.6	8.7 $\pm$.4	7.0 $\pm$.3
NZD/USD	20.0 $\pm$ 1.0	13.7 $\pm$.7	7.6 $\pm$.4	5.1 $\pm$.2	3.3 $\pm$.2	2.2 $\pm$.1	36.8 $\pm$ 1.8	30.5 $\pm$ 1.5	16.8 $\pm$.8	11.6 $\pm$.6
USD/CHF	6.3 $\pm$.3	5.1 $\pm$.2	3.9 $\pm$.2	2.4 $\pm$.1	1.8 $\pm$.1	1.4 $\pm$.1	23.1 $\pm$ 1.1	14.7 $\pm$.7	10.1 $\pm$.5	7.1 $\pm$.3
USD/SEK	75.6 $\pm$ 3.6	60.9 $\pm$ 2.9	32.6 $\pm$ 1.6	18.9 $\pm$.9	10.1 $\pm$.5	3.6 $\pm$.2	126.0 $\pm$ 6.0	102.9 $\pm$ 4.9	44.1 $\pm$ 2.1	26.3 $\pm$ 1.3
USD/SGD	6.1 $\pm$.3	4.5 $\pm$.2	2.7 $\pm$.1	1.7 $\pm$.1	0.9 $\pm$.0	0.4 $\pm$.0	22.1 $\pm$ 1.1	12.6 $\pm$.6	9.3 $\pm$.4	6.5 $\pm$.3

Table 5: Evaluation of models using MAE (Mean $\pm$ SD over 10 independent runs).

Ticker	H1									
	ICS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS
AUD/USD	01.49 $\pm$ 0.007	01.22 $\pm$.0006	.0078 $\pm$.0004	.0064 $\pm$.0003	.0051 $\pm$.0002	.0038 $\pm$.0002	02.05 $\pm$.0010	01.66 $\pm$.0008	.0111 $\pm$.0005	.0088 $\pm$.0004
EUR/SGD	.0071 $\pm$.0003	.0039 $\pm$.0003	.0014 $\pm$.0002	.0036 $\pm$.0002	.0029 $\pm$.0001	.0023 $\pm$.0001	.0086 $\pm$.0004	.0070 $\pm$.0003	.0034 $\pm$.0003	.0042 $\pm$.0002
GBP/NZD	.0078 $\pm$.0004	.0066 $\pm$.0003	.0050 $\pm$.0002	.0041 $\pm$.0002	.0033 $\pm$.0002	.0023 $\pm$.0001	.0122 $\pm$.0006	.0099 $\pm$.0005	.0068 $\pm$.0003	.0056 $\pm$.0003
NZD/USD	.0085 $\pm$.0004	.0072 $\pm$.0003	.0060 $\pm$.0003	.0048 $\pm$.0002	.0041 $\pm$.0002	.0035 $\pm$.0002	.0176 $\pm$.0008	.0146 $\pm$.0007	.0101 $\pm$.0005	.0082 $\pm$.0004
USD/CHF	.0076 $\pm$.0004	.0064 $\pm$.0003	.0051 $\pm$.0002	.0040 $\pm$.0002	.0033 $\pm$.0002	.0027 $\pm$.0001	.0130 $\pm$.0006	.0101 $\pm$.0005	.0071 $\pm$.0003	.0059 $\pm$.0003
USD/SEK	.0218 $\pm$.0010	.0181 $\pm$.0009	.0132 $\pm$.0006	.0108 $\pm$.0005	.0085 $\pm$.0004	.0067 $\pm$.0003	.0205 $\pm$.0014	.0258 $\pm$.0012	.0177 $\pm$.0008	.0141 $\pm$.0007
USD/SGD	.0077 $\pm$.0004	.0065 $\pm$.0003	.0045 $\pm$.0002	.0035 $\pm$.0002	.0025 $\pm$.0001	.0014 $\pm$.0001	.0142 $\pm$.0007	.0106 $\pm$.0005	.0073 $\pm$.0004	.0061 $\pm$.0003

Ticker	H4									
	ICS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS
AUD/USD	01.49 $\pm$ 0.007	01.22 $\pm$.0006	.0078 $\pm$.0004	.0064 $\pm$.0003	.0051 $\pm$.0002	.0038 $\pm$.0002	02.05 $\pm$.0010	01.66 $\pm$.0008	.0111 $\pm$.0005	.0088 $\pm$.0004
EUR/SGD	.0071 $\pm$.0003	.0039 $\pm$.0003	.0014 $\pm$.0002	.0036 $\pm$.0002	.0029 $\pm$.0001	.0023 $\pm$.0001	.0086 $\pm$.0004	.0070 $\pm$.0003	.0034 $\pm$.0003	.0042 $\pm$.0002
GBP/NZD	.0078 $\pm$.0004	.0066 $\pm$.0003	.0050 $\pm$.0002	.0041 $\pm$.0002	.0033 $\pm$.0002	.0023 $\pm$.0001	.0122 $\pm$.0006	.0099 $\pm$.0005	.0068 $\pm$.0003	.0056 $\pm$.0003
NZD/USD	.0085 $\pm$.0004	.0072 $\pm$.0003	.0060 $\pm$.0003	.0048 $\pm$.0002	.0041 $\pm$.0002	.0035 $\pm$.0002	.0176 $\pm$.0008	.0146 $\pm$.0007	.0101 $\pm$.0005	.0082 $\pm$.0004
USD/CHF	.0076 $\pm$.0004	.0064 $\pm$.0003	.0051 $\pm$.0002	.0040 $\pm$.0002	.0033 $\pm$.0002	.0027 $\pm$.0001	.0130 $\pm$.0006	.0101 $\pm$.0005	.0071 $\pm$.0003	.0059 $\pm$.0003
USD/SEK	.0218 $\pm$.0010	.0181 $\pm$.0009	.0132 $\pm$.0006	.0108 $\pm$.0005	.0085 $\pm$.0004	.0067 $\pm$.0003	.0205 $\pm$.0014	.0258 $\pm$.0012	.0177 $\pm$.0008	.0141 $\pm$.0007
USD/SGD	.0077 $\pm$.0004	.0065 $\pm$.0003	.0045 $\pm$.0002	.0035 $\pm$.0002	.0025 $\pm$.0001	.0014 $\pm$.0001	.0142 $\pm$.0007	.0106 $\pm$.0005	.0073 $\pm$.0004	.0061 $\pm$.0003

Ticker	D1									
	ICS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS
AUD/USD	01.49 $\pm$ 0.007	01.22 $\pm$.0006	.0078 $\pm$.0004	.0064 $\pm$.0003	.0051 $\pm$.0002	.0038 $\pm$.0002	02.05 $\pm$.0010	01.66 $\pm$.0008	.0111 $\pm$.0005	.0088 $\pm$.0004
EUR/SGD	.0071 $\pm$.0003	.0039 $\pm$.0003	.0014 $\pm$.0002	.0036 $\pm$.0002	.0029 $\pm$.0001	.0023 $\pm$.0001	.0086 $\pm$.0004	.0070 $\pm$.0003	.0034 $\pm$.0003	.0042 $\pm$.0002
GBP/NZD	.0078 $\pm$.0004	.0066 $\pm$.0003	.0050 $\pm$.0002	.0041 $\pm$.0002	.0033 $\pm$.0002	.0023 $\pm$.0001	.0122 $\pm$.0006	.0099 $\pm$.0005	.0068 $\pm$.0003	.0056 $\pm$.0003
NZD/USD	.0085 $\pm$.0004	.0072 $\pm$.0003	.0060 $\pm$.0003	.0048 $\pm$.0002	.0041 $\pm$.0002	.0035 $\pm$.0002	.0176 $\pm$.0008	.0146 $\pm$.0007	.0101 $\pm$.0005	.0082 $\pm$.0004
USD/CHF	.0076 $\pm$.0004	.0064 $\pm$.0003	.0051 $\pm$.0002	.0040 $\pm$.0002	.0033 $\pm$.0002	.0027 $\pm$.0001	.0130 $\pm$.0006	.0101 $\pm$.0005	.0071 $\pm$.0003	.0059 $\pm$.0003
USD/SEK	.0218 $\pm$.0010	.0181 $\pm$.0009	.0132 $\pm$.0006	.0108 $\pm$.0005	.0085 $\pm$.0004	.0067 $\pm$.0003	.0205 $\pm$.0014	.0258 $\pm$.0012	.0177 $\pm$.0008	.0141 $\pm$.0007
USD/SGD	.0077 $\pm$.0004	.0065 $\pm$.0003	.0045 $\pm$.0002	.0035 $\pm$.0002	.0025 $\pm$.0001	.0014 $\pm$.0001	.0142 $\pm$.0007	.0106 $\pm$.0005	.0073 $\pm$.0004	.0061 $\pm$.0003

Table 6: Evaluation of models using MAPE (Mean $\pm$ SD over 10 independent runs).

Ticker	H1									
	ICS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS
AUD/USD	01.88 $\pm$.0009	01.49 $\pm$.0007	.0096 $\pm$.0005	.0080 $\pm$.0004	.0066 $\pm$.0003	.0043 $\pm$.0002	02.29 $\pm$.0011	01.95 $\pm$.0009	.0139 $\pm$.0006	.0106 $\pm$.0005
EUR/SGD	.0086 $\pm$.0004	.0072 $\pm$.0003	.0054 $\pm$.0003	.0042 $\pm$.0002	.0033 $\pm$.0002	.0025 $\pm$.0001	.0101 $\pm$.0005	.0085 $\pm$.0004	.0067 $\pm$.0003	.0056 $\pm$.0003
GBP/NZD	.0091 $\pm$.0004	.0075 $\pm$.0004	.0059 $\pm$.0003	.0048 $\pm$.0002	.0037 $\pm$.0002	.0025 $\pm$.0001	.0146 $\pm$.0007	.0117 $\pm$.0006	.0088 $\pm$.0004	.0070 $\pm$.0003
NZD/USD	.0156 $\pm$.0007	.0129 $\pm$.0006	.0097 $\pm$.0005	.0080 $\pm$.0004	.0062 $\pm$.0003	.0043 $\pm$.0002	.0217 $\pm$.0010	.0185 $\pm$.0007	.0132 $\pm$.0006	.0106 $\pm$.0005
USD/CHF	.0096 $\pm$.0005	.0082 $\pm$.0004	.0064 $\pm$.0003	.0052 $\pm$.0002	.0040 $\pm$.0002	.0029 $\pm$.0001	.0143 $\pm$.0007	.0117 $\pm$.0006	.0088 $\pm$.0004	.0072 $\pm$.0003
USD/CNH	.0109 $\pm$.0005	.0091 $\pm$.0004	.0075 $\pm$.0004	.0059 $\pm$.0003	.0043 $\pm$.0002	.0026 $\pm$.0001	.0176 $\pm$.0008	.0139 $\pm$.0007	.0109 $\pm$.0006	.0090 $\pm$.0004
USD/SEK	.0233 $\pm$.0014	.0258 $\pm$.0012	.0182 $\pm$.0009	.0143 $\pm$.0007	.0106 $\pm$.0005	.0066 $\pm$.0003	.0353 $\pm$.0017	.0316 $\pm$.0015	.0222 $\pm$.0011	.0183 $\pm$.0009
USD/SGD	.0093 $\pm$.0004	.0078 $\pm$.0004	.0054 $\pm$.0003	.0041 $\pm$.0002	.0027 $\pm$.0001	.0015 $\pm$.0001	.0153 $\pm$.0007	.0119 $\pm$.0006	.0080 $\pm$.0004	.0067 $\pm$.0003

Ticker	H4									
	ICS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS	ICLS	ICGS	ICCS
AUD/USD	01.88 $\pm$.0009	01.49 $\pm$.0007	.0096 $\pm$.0005	.0080 $\pm$.0004	.0066 $\pm$.0003	.0043 $\pm$.0002	02.29 $\pm$.0011	01.95 $\pm$.0009	.0139 $\pm$.0006	.0106 $\pm$.0005
EUR/SGD	.0086 $\pm$.0004	.0072 $\pm$.0003	.0054 $\pm$.0003	.0042 $\pm$.0002	.0033 $\pm$.0002	.0025 $\pm$.0001	.0101 $\pm$.0005	.0085 $\pm$.0004	.0067 $\pm$.0003	.0056 $\pm$.0003
GBP/NZD	.0091 $\pm$.0004	.0075 $\pm$.0004	.0059 $\pm$.0003	.0048 $\pm$.0002	.0037 $\pm$.0002	.0025 $\pm$.0001	.0146 $\pm$.0007	.0117 $\pm$.0006	.0088 $\pm$.0004	.0070 $\pm$.0003
NZD/USD	.0156 $\pm$.0007	.0129 $\pm$.0006	.0097 $\pm$.0005	.0080 $\pm$.0004	.0062 $\pm$.0003	.0043 $\pm$.0002	.0217 $\pm$.0010	.0185 $\pm$.0007	.0132 $\pm$.0006	.0106 $\pm$.0005
USD/CHF	.0096 $\pm$.0005	.0082 $\pm$.0004	.0064 $\pm$.0003	.0052 $\pm$.0002	.0040 $\pm$.0002	.0029 $\pm$.0001	.0143 $\pm$.0007	.0117 $\pm$.0006	.0088 $\pm$.0004	.0072 $\pm$.0003
USD/CNH	.0109 $\pm$.0005	.0091 $\pm$.0004	.0075 $\pm$.0004	.0059 $\pm$.0003	.0043 $\pm$.0002	.0026 $\pm$.0001	.0176 $\pm$.0008	.0139 $\pm$.0007	.0109 $\pm$.0006	.0090 $\pm$.

Table 7: Evaluation results with Accuracy (%) and F1-score (Mean $\pm$ SD over 10 independent runs).

Ticker	Accuracy (%)						F1-score					
	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS
AUD/USD	74.4 $\pm$ 1.1	74.1 $\pm$ 0.9	78.6 $\pm$ 0.8	76.0 $\pm$ 1.2	74.1 $\pm$ 1.0	78.7 $\pm$ 0.7	.59 $\pm$.02	.58 $\pm$.02	.60 $\pm$.01	.58 $\pm$.02	.55 $\pm$.03	.63 $\pm$.02
EUR/SGD	76.1 $\pm$ 0.8	74.1 $\pm$ 1.2	75.3 $\pm$ 0.9	76.1 $\pm$ 1.0	77.8 $\pm$ 0.7	77.9 $\pm$ 0.8	.61 $\pm$.02	.49 $\pm$.04	.59 $\pm$.02	.53 $\pm$.03	.57 $\pm$.02	.59 $\pm$.02
GBP/NZD	69.9 $\pm$ 1.3	70.3 $\pm$ 1.0	72.2 $\pm$ 0.9	71.8 $\pm$ 1.1	72.2 $\pm$ 0.8	74.3 $\pm$ 0.8	.59 $\pm$.02	.56 $\pm$.03	.56 $\pm$.02	.60 $\pm$.02	.60 $\pm$.02	.65 $\pm$.01
NZD/USD	74.7 $\pm$ 0.9	73.6 $\pm$ 1.1	75.8 $\pm$ 1.0	75.6 $\pm$ 0.8	74.1 $\pm$ 1.2	77.5 $\pm$ 0.6	.57 $\pm$.02	.58 $\pm$.02	.60 $\pm$.01	.58 $\pm$.02	.54 $\pm$.03	.61 $\pm$.02
USD/CHF	75.6 $\pm$ 1.0	76.5 $\pm$ 0.8	76.8 $\pm$ 0.7	77.4 $\pm$ 0.9	75.8 $\pm$ 1.1	79.3 $\pm$ 0.8	.53 $\pm$.03	.57 $\pm$.02	.57 $\pm$.02	.58 $\pm$.02	.54 $\pm$.03	.61 $\pm$.02
USD/CNH	67.4 $\pm$ 1.4	67.5 $\pm$ 1.2	66.6 $\pm$ 1.5	69.4 $\pm$ 1.0	67.5 $\pm$ 1.3	69.6 $\pm$ 0.9	.62 $\pm$.02	.63 $\pm$.02	.62 $\pm$.02	.66 $\pm$.01	.61 $\pm$.03	.65 $\pm$.02
USD/SEK	62.1 $\pm$ 1.5	61.5 $\pm$ 1.6	64.9 $\pm$ 1.2	63.8 $\pm$ 1.4	61.8 $\pm$ 1.5	65.7 $\pm$ 1.1	.58 $\pm$.03	.55 $\pm$.03	.62 $\pm$.02	.61 $\pm$.02	.57 $\pm$.03	.62 $\pm$.02
USD/SGD	80.5 $\pm$ 0.7	80.0 $\pm$ 0.9	80.8 $\pm$ 0.8	81.6 $\pm$ 0.6	80.2 $\pm$ 0.8	82.7 $\pm$ 0.5	.49 $\pm$.03	.54 $\pm$.02	.55 $\pm$.02	.55 $\pm$.02	.51 $\pm$.03	.58 $\pm$.01

Table 8: Statistical significance of the ICGS framework across H1, H4, and D1 horizons via DM statistics, two-sided p -values, Holm-Bonferroni adjusted p -values, and bootstrap analysis of overall majority-voted daily trend accuracy.

Ticker	MSE – DM Statistic			MAE – DM Statistic			MAPE – DM Statistic			Bootstrap Accuracy (%)	
	H1	H4	D1	H1	H4	D1	H1	H4	D1	Mean	SD
AUD/USD	3.98	4.82	3.12	3.65	4.41	2.98	3.55	4.30	2.88	79.32 $\pm$ 0.47	
	$p = 6.892 \times 10^{-5}$ $p_H = 3.230 \times 10^{-3}$	$p = 1.436 \times 10^{-6}$ $p_H = 9.475 \times 10^{-5}$	$p = 1.809 \times 10^{-3}$ $p_H = 2.532 \times 10^{-2}$	$p = 2.622 \times 10^{-4}$ $p_H = 8.654 \times 10^{-3}$	$p = 1.034 \times 10^{-5}$ $p_H = 5.995 \times 10^{-4}$	$p = 2.882 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$	$p = 3.852 \times 10^{-4}$ $p_H = 1.117 \times 10^{-2}$	$p = 1.708 \times 10^{-5}$ $p_H = 9.565 \times 10^{-4}$	$p = 3.977 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$		
EUR/SGD	3.88	4.25	3.22	3.52	3.96	2.95	3.41	3.82	2.84	78.67 $\pm$ 0.21	
	$p = 1.045 \times 10^{-4}$ $p_H = 4.492 \times 10^{-3}$	$p = 2.138 \times 10^{-5}$ $p_H = 1.154 \times 10^{-3}$	$p = 1.282 \times 10^{-3}$ $p_H = 2.307 \times 10^{-2}$	$p = 4.315 \times 10^{-4}$ $p_H = 3.373 \times 10^{-3}$	$p = 7.495 \times 10^{-5}$ $p_H = 2.790 \times 10^{-2}$	$p = 3.178 \times 10^{-3}$ $p_H = 1.566 \times 10^{-2}$	$p = 6.496 \times 10^{-4}$ $p_H = 1.566 \times 10^{-2}$	$p = 1.335 \times 10^{-4}$ $p_H = 6.605 \times 10^{-3}$	$p = 4.511 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$		
GBP/NZD	3.76	4.54	3.05	3.48	4.18	2.82	3.35	4.02	2.71	74.97 $\pm$ 0.35	
	$p = 1.609 \times 10^{-4}$ $p_H = 2.87 \times 10^{-3}$	$p = 5.625 \times 10^{-6}$ $p_H = 3.488 \times 10^{-4}$	$p = 2.288 \times 10^{-3}$ $p_H = 2.746 \times 10^{-2}$	$p = 5.014 \times 10^{-4}$ $p_H = 1.54 \times 10^{-2}$	$p = 2.915 \times 10^{-5}$ $p_H = 1.545 \times 10^{-3}$	$p = 4.802 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$	$p = 8.081 \times 10^{-4}$ $p_H = 1.793 \times 10^{-2}$	$p = 5.820 \times 10^{-5}$ $p_H = 2.852 \times 10^{-3}$	$p = 6.728 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$		
NZD/USD	4.15	5.22	3.36	3.82	4.75	3.10	3.70	4.61	2.99	78.07 $\pm$ 0.31	
	$p = 3.325 \times 10^{-5}$ $p_H = 1.729 \times 10^{-3}$	$p = 1.789 \times 10^{-7}$ $p_H = 1.252 \times 10^{-5}$	$p = 7.794 \times 10^{-4}$ $p_H = 1.793 \times 10^{-2}$	$p = 1.335 \times 10^{-4}$ $p_H = 5.605 \times 10^{-3}$	$p = 2.034 \times 10^{-6}$ $p_H = 1.322 \times 10^{-4}$	$p = 1.935 \times 10^{-3}$ $p_H = 2.532 \times 10^{-2}$	$p = 2.156 \times 10^{-4}$ $p_H = 7.546 \times 10^{-3}$	$p = 4.027 \times 10^{-6}$ $p_H = 2.537 \times 10^{-4}$	$p = 2.790 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$		
USD/CHF	4.05	5.10	3.58	3.78	4.65	3.32	3.66	4.52	3.20	80.01 $\pm$ 0.27	
	$p = 5.122 \times 10^{-5}$ $p_H = 2.561 \times 10^{-3}$	$p = 3.397 \times 10^{-7}$ $p_H = 2.310 \times 10^{-5}$	$p = 3.436 \times 10^{-4}$ $p_H = 1.065 \times 10^{-2}$	$p = 1.568 \times 10^{-4}$ $p_H = 6.116 \times 10^{-3}$	$p = 3.319 \times 10^{-6}$ $p_H = 2.124 \times 10^{-4}$	$p = 9.002 \times 10^{-4}$ $p_H = 1.800 \times 10^{-2}$	$p = 2.522 \times 10^{-4}$ $p_H = 8.575 \times 10^{-3}$	$p = 6.184 \times 10^{-6}$ $p_H = 3.772 \times 10^{-4}$	$p = 1.374 \times 10^{-3}$ $p_H = 2.307 \times 10^{-2}$		
USD/CNH	3.62	4.02	2.88	3.35	3.71	2.65	3.22	3.58	2.54	70.31 $\pm$ 0.48	
	$p = 2.946 \times 10^{-4}$ $p_H = 9.427 \times 10^{-3}$	$p = 5.820 \times 10^{-5}$ $p_H = 2.852 \times 10^{-3}$	$p = 3.977 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$	$p = 8.081 \times 10^{-4}$ $p_H = 1.793 \times 10^{-2}$	$p = 2.073 \times 10^{-4}$ $p_H = 7.461 \times 10^{-3}$	$p = 8.049 \times 10^{-3}$ $p_H = 2.790 \times 10^{-2}$	$p = 1.292 \times 10^{-3}$ $p_H = 2.307 \times 10^{-2}$	$p = 3.436 \times 10^{-4}$ $p_H = 1.065 \times 10^{-2}$	$p = 1.109 \times 10^{-2}$ $p_H = 2.790 \times 10^{-2}$		
USD/SEK	4.28	4.95	3.42	3.95	4.52	3.18	3.82	4.38	3.05	67.13 $\pm$ 0.42	
	$p = 1.869 \times 10^{-5}$ $p_H = 1.028 \times 10^{-3}$	$p = 7.421 \times 10^{-7}$ $p_H = 4.972 \times 10^{-5}$	$p = 2.602 \times 10^{-4}$ $p_H = 1.566 \times 10^{-2}$	$p = 7.815 \times 10^{-5}$ $p_H = 4.393 \times 10^{-3}$	$p = 6.184 \times 10^{-6}$ $p_H = 3.772 \times 10^{-4}$	$p = 1.473 \times 10^{-3}$ $p_H = 2.307 \times 10^{-2}$	$p = 1.335 \times 10^{-4}$ $p_H = 5.605 \times 10^{-3}$	$p = 1.187 \times 10^{-5}$ $p_H = 6.765 \times 10^{-4}$	$p = 2.288 \times 10^{-3}$ $p_H = 2.746 \times 10^{-2}$		
USD/SGD	4.45	5.88	3.78	4.12	5.35	3.45	3.98	5.18	3.32	83.27 $\pm$ 0.39	
	$p = 8.587 \times 10^{-6}$ $p_H = 5.066 \times 10^{-4}$	$p = 4.103 \times 10^{-9}$ $p_H = 2.954 \times 10^{-7}$	$p = 1.568 \times 10^{-4}$ $p_H = 6.116 \times 10^{-3}$	$p = 3.789 \times 10^{-5}$ $p_H = 1.932 \times 10^{-3}$	$p = 8.795 \times 10^{-8}$ $p_H = 6.245 \times 10^{-6}$	$p = 5.606 \times 10^{-4}$ $p_H = 1.458 \times 10^{-2}$	$p = 6.892 \times 10^{-5}$ $p_H = 3.239 \times 10^{-3}$	$p = 2.219 \times 10^{-7}$ $p_H = 1.531 \times 10^{-5}$	$p = 9.002 \times 10^{-4}$ $p_H = 1.800 \times 10^{-2}$		

Note that p denotes the two-sided asymptotic p -value obtained from the DM statistic under the standard normal approximation, and p_H denotes the Holm-Bonferroni adjusted p -value based on all 72 DM tests (8 currency pairs $\times$ 3 error metrics $\times$ 3 forecasting horizons). Statistical significance was assessed at $\alpha = 0.05$. The bootstrap accuracy is reported as mean $\pm$ standard deviation over the bootstrap samples.

predominantly lower than the corresponding actual values, whereas in the four-hour and daily timeframes, the forecasts tend to remain above the real market prices for most periods. This behavior suggests a timeframe-dependent bias in the model's estimations. Similar to previous observations, a specific interval is identified for the USD/SEK currency pair across all three timeframes, during which the ICLS predictions remain largely range-bound. In this interval, the model fails to capture both the prevailing trend and the associated price variations, effectively overlooking the directional movement present in the actual market data. Overall, these findings indicate that while the ICLS model improves directional alignment, it remains less responsive to trend development when such movements are accompanied by limited or structurally constrained volatility.

The ICGS model (Figure 9) is able to effectively follow the overall price movement across all three timeframes, demonstrating consistent directional alignment with the actual market behavior. This indicates a strong capability in capturing the general trend dynamics of the price series. However, in certain intervals, the model produces forecasts with excessively high variability, resulting in amplified oscillations that exceed those observed in the real price data. Conversely, there are also periods in which the model fails to adequately capture sudden price changes, smoothing out abrupt movements present in the actual series. Similar to the previous models, the previously identified issue for the USD/SEK currency pair remains observable across all three timeframes. Nevertheless, in the case of the ICGS model, this behavior appears to be partially mitigated, as the range-bound predictions exhibit improved responsiveness to the underlying price trend compared to the other models. Overall, these results suggest that while the ICGS model achieves better trend-following performance, it still faces challenges in balancing sensitivity to abrupt price movements and stability in low-volatility or structurally constrained market conditions.

For the IICS (Figure 10) model, a noticeable improvement is observed compared to the preceding models across all three timeframes. The forecasts exhibit stronger directional consistency with the actual price movements, and the prediction errors gradually reveal more structured and recognizable patterns rather than purely random deviations. This behavior suggests enhanced stability and learning capacity in the model's predictive process. Despite these improvements, certain limitations remain evident. Specifically, the model continues to struggle with accurately capturing price fluctuations characterized by large amplitudes, and in some intervals, it generates predictions with excessively high variability. These behaviors indicate that, while the IICS model achieves superior overall performance, challenges persist in balancing responsiveness to high-magnitude price movements and the suppression of spurious oscillations in the predicted series.

For the IICLS model (Figure 11), a substantial improvement in predictive accuracy is observed across all three timeframes. In most time intervals, the forecasts exhibit bias-dominated errors, indicating systematic deviations rather than random fluctuations. Despite this bias, the

model demonstrates a strong capability in capturing price variability, successfully reflecting market oscillations in the majority of cases. Except for a limited number of specific intervals, in which minor discrepancies between the predicted and actual prices are observed, the distance between the forecasts and the true price values remains relatively small. In the remaining periods, the predicted and actual price trajectories are nearly overlapping, highlighting the model's enhanced ability to closely track market dynamics. Overall, these results indicate that the IICLS model achieves a high level of alignment between predicted and actual prices, with residual errors being predominantly structured and limited in magnitude.

Figure 12 shows the strongest overall performance among the six evaluated models. Across all three timeframes, the predictions exhibit a high degree of alignment with the actual price movements, with the one-hour timeframe showing slightly superior performance compared to the four-hour and daily resolutions. The prediction errors are predominantly bias-driven and follow consistent, structured patterns rather than appearing random. Especially, the performance for the USD/SEK currency pair shows a substantial improvement in this model. Unlike earlier models, the IICGS framework effectively captures both the underlying trend and price fluctuations for this pair across all timeframes. Overall, no significant temporal lag is observed between the predicted and actual series, and the model demonstrates a robust ability to track price dynamics, successfully capturing market oscillations. These results indicate a consistent and meaningful improvement in predictive performance across all timeframes and all evaluated currency pairs.

Overall, the evaluated models can be categorized into two distinct groups based on the type of Ichimoku-related features incorporated into the learning framework. The first group of models utilizes the five canonical Ichimoku indicators directly alongside OHLC price data. While this configuration enhances the models' ability to capture general trend-following behavior, their performance remains limited in terms of accurately representing complex price dynamics, particularly under varying volatility regimes and across higher timeframes. In contrast, the second group of models leverages engineered features derived from the Ichimoku indicator, combined with OHLC prices. These engineered representations provide a more informative and structured description of market dynamics, enabling the models to achieve stronger directional alignment, improved stability, and reduced temporal lag across all timeframes. The empirical results consistently demonstrate that feature engineering plays a critical role in enhancing both price-tracking capability and trend prediction robustness. Consequently, models employing engineered Ichimoku features exhibit superior overall performance compared to those relying solely on the raw indicator components.

Figure 7: Ichimoku-CNN-SFOA results

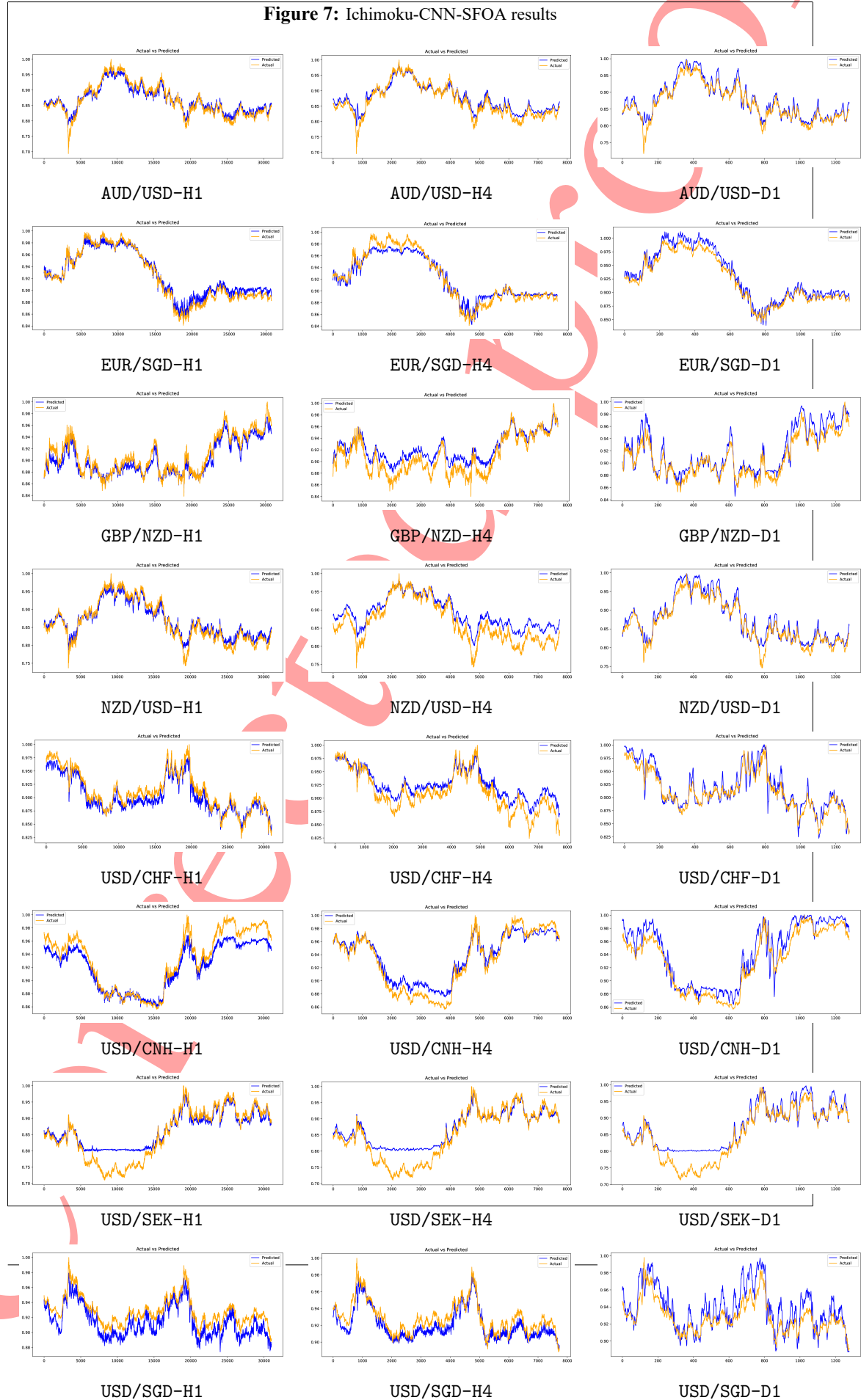


Figure 8: Ichimoku-CNNLSTM-SFOA results

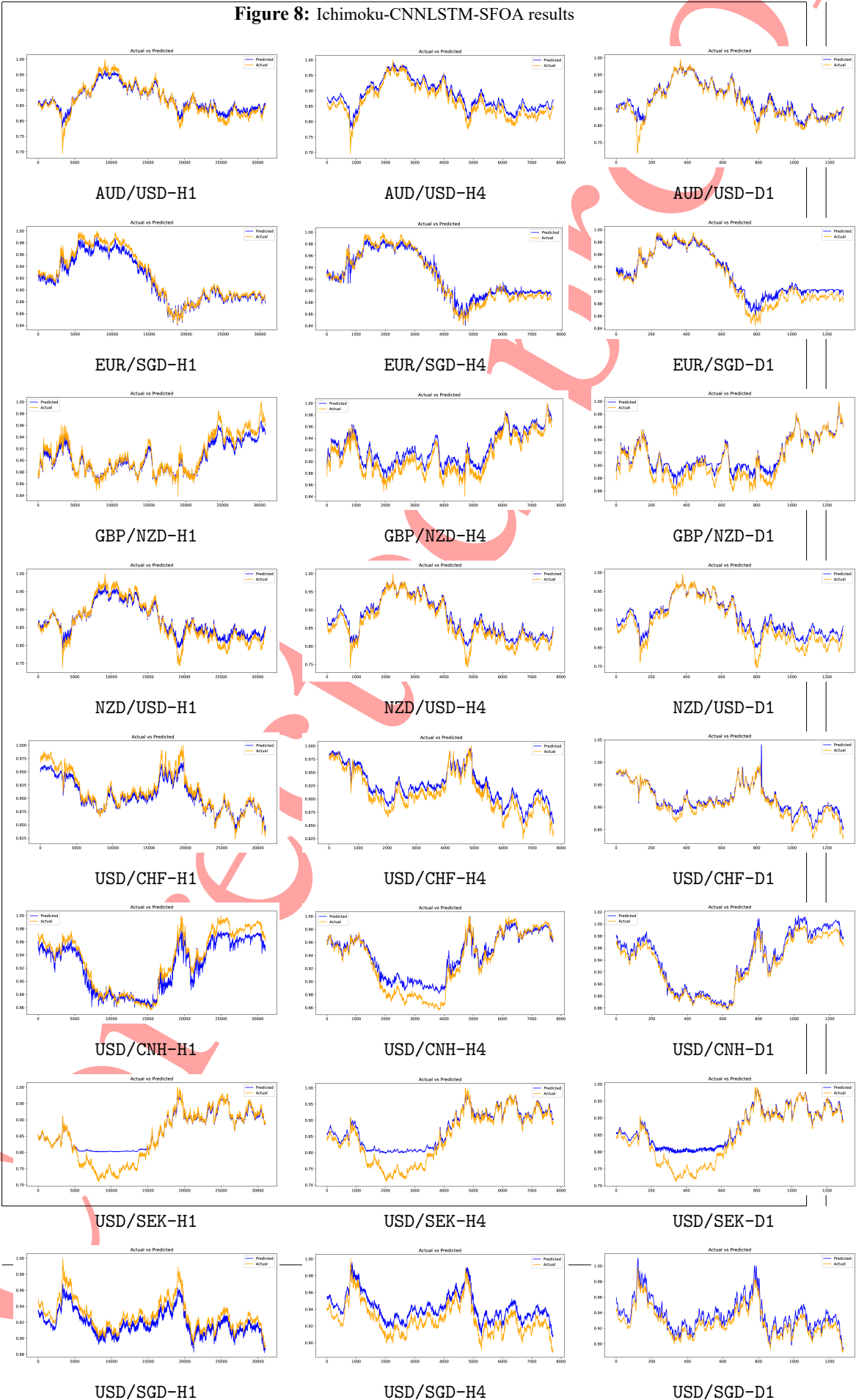


Figure 9: Ichimoku-CNNGRU-SFOA results

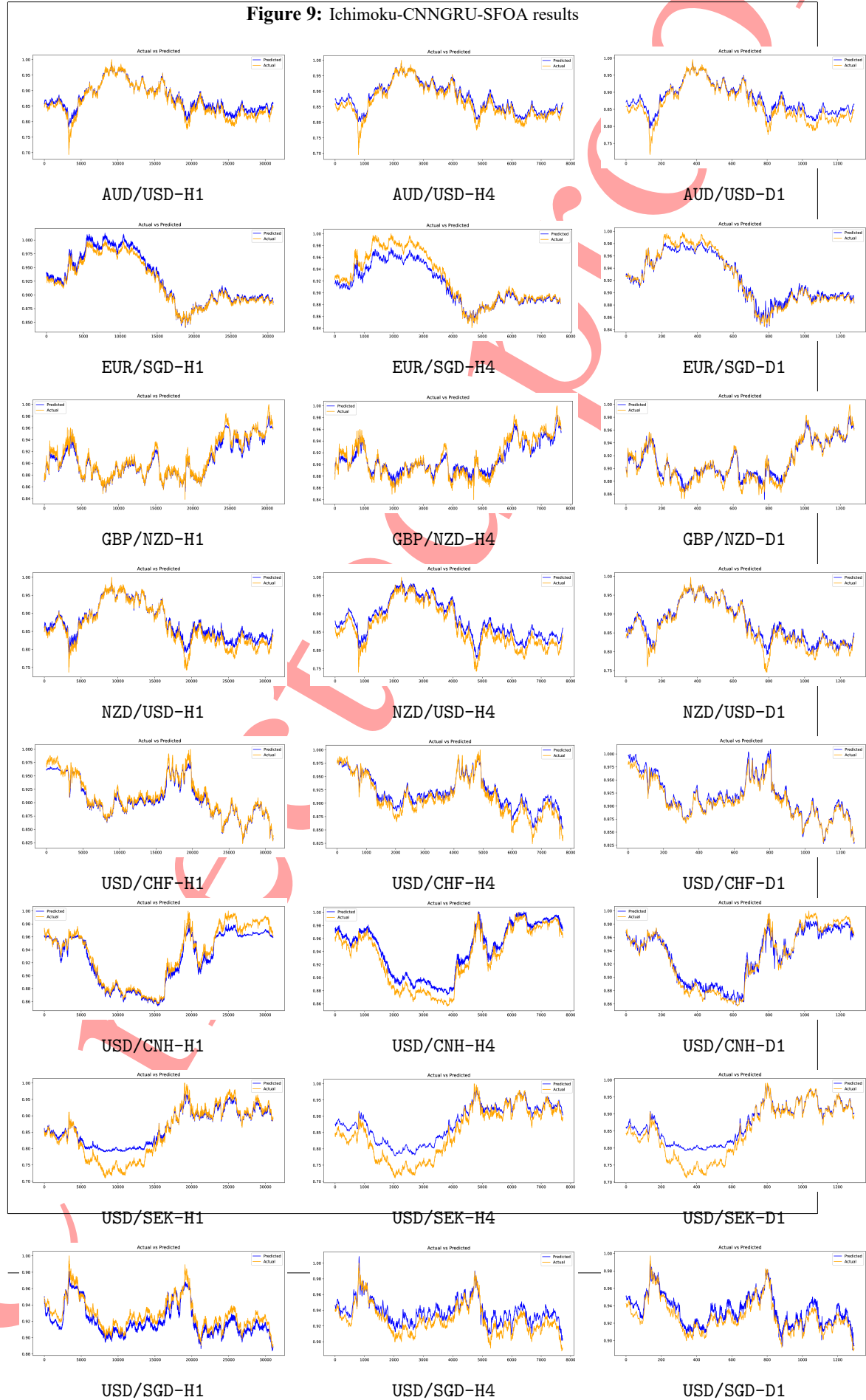


Figure 10: Improved-Ichimoku-CNN-SFOA results

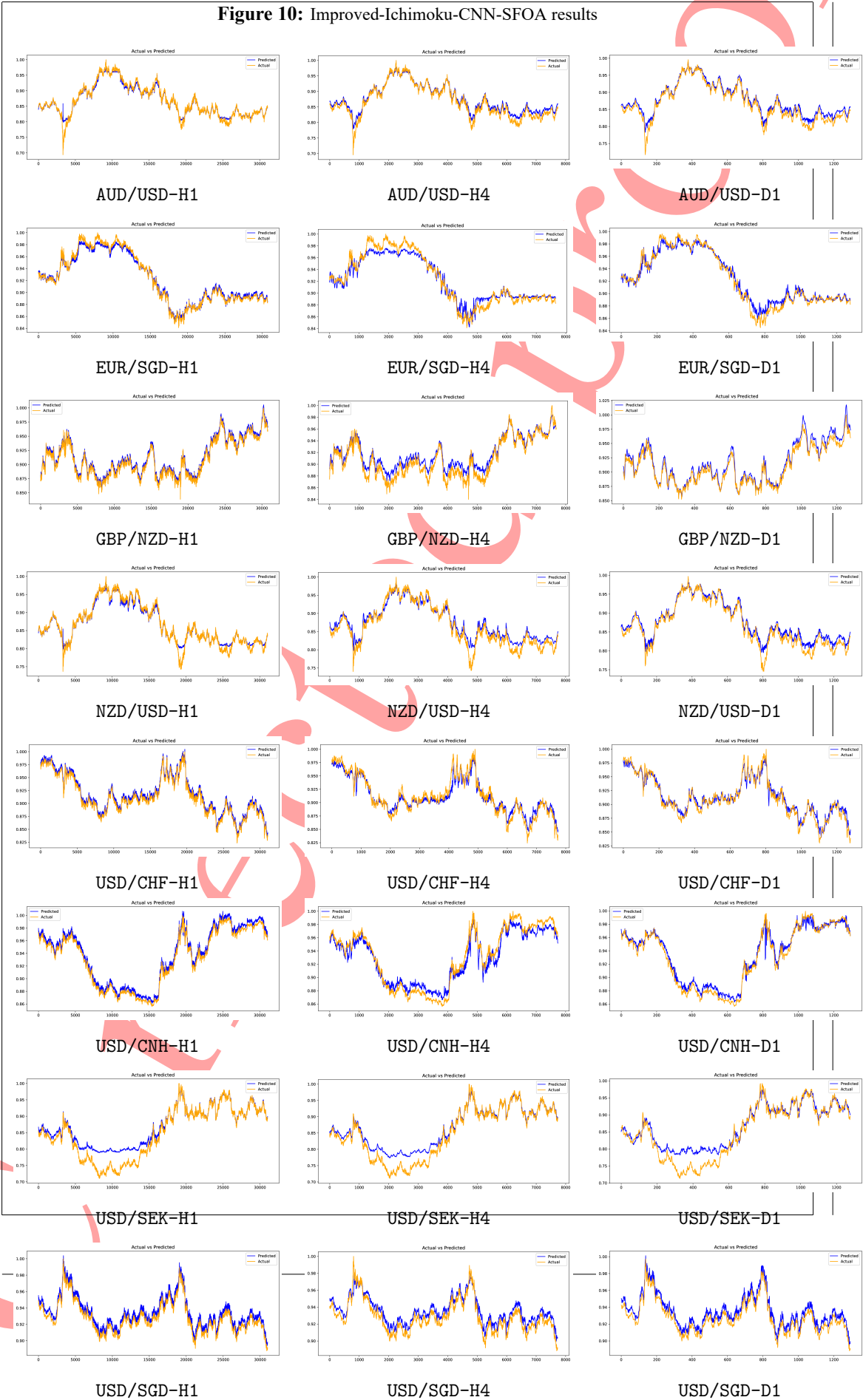
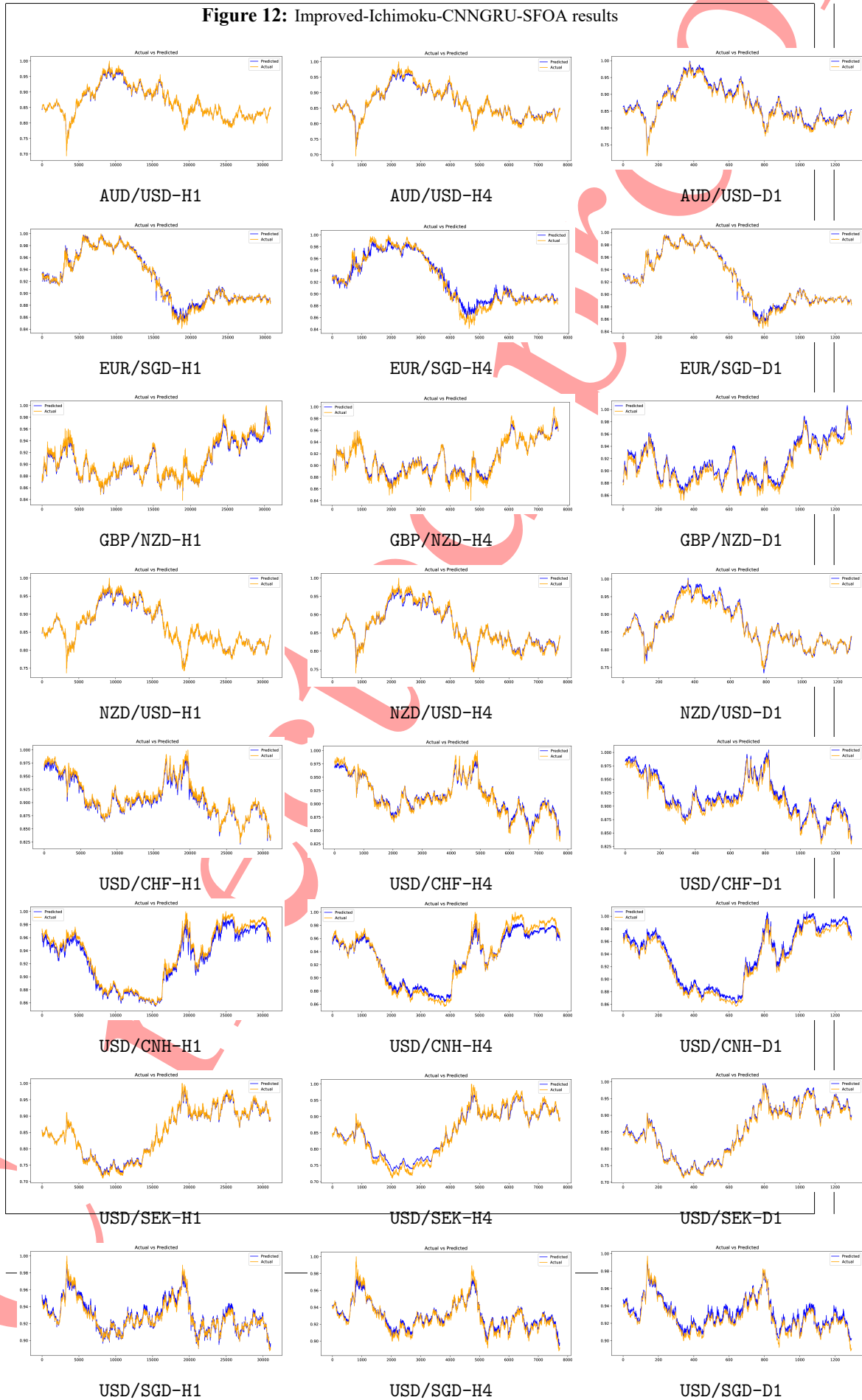


Figure 11: Improved-Ichimoku-CNNLSTM-SFOA results

Figure 12: Improved-Ichimoku-CNNGRU-SFOA results

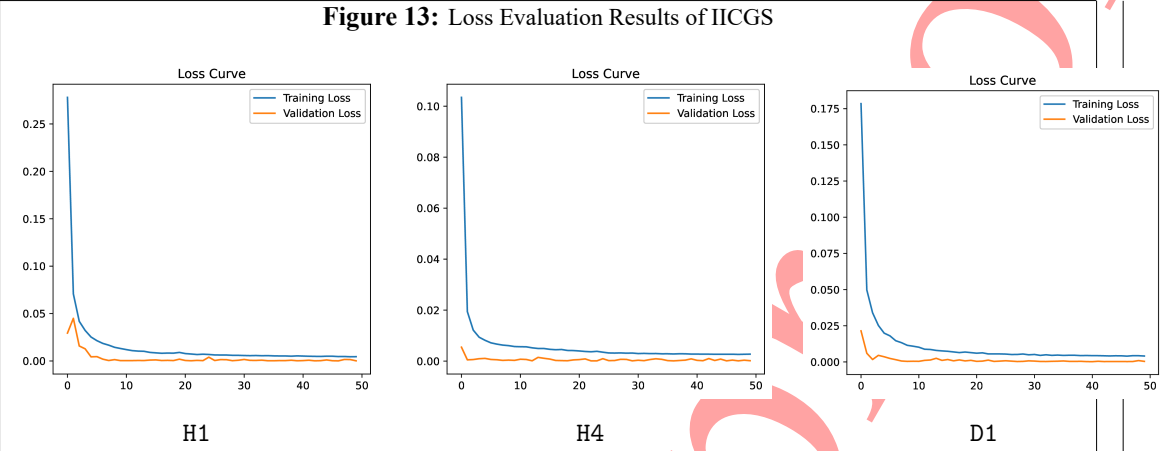
4 Discussion

4.1 Motivation for using Ichimoku Kinko Hyo

The selection of Ichimoku Kinko Hyo over conventional indicators such as RSI, MACD, and Bollinger Bands provides a distinct advantage due to its forward-looking characteristics. Traditional indicators primarily extract insights from historical data, generating features that represent previous market conditions. Although these methods are effective to a degree, they are predominantly lagging in nature and provide more limited information about future market movements. In contrast, the Ichimoku indicator offers unique feature-engineering capabilities through its components, particularly the Kumo (cloud), which by default is displaced 26 periods forward. In classical technical analysis, this forward displacement allows traders to examine the position of price relative to the cloud that appears ahead of the current price and to interpret potential interactions with it as anticipatory signals about future trend direction. It is important to clarify, however, that the 26-period forward shift is solely a plotting convention used for visual alignment. In the feature-engineering process of this study, all Ichimoku components at time t are computed exclusively from price information available up to time t (or $t - 1$); no actual future observations are used. Consequently, while the resulting features preserve the rich forward-oriented information valued by classical traders—capturing shifts in momentum, equilibrium, and support/resistance levels more effectively than traditional lagging indicators—they remain completely free of data leakage. Integrating these carefully constructed features enables the predictive model to overcome the limitations of conventional indicators and deliver robust predictions suited to the complexities of volatile market environments.

4.2 Training Stability and Loss Behavior Analysis

Figure 13 illustrates the successful convergence behavior of the proposed model across the H1, H4, and D1 timeframes. As observed, the training and validation loss consistently decrease over epochs, indicating stable learning dynamics without significant oscillations. The relatively small gap between training and validation loss suggests that the model maintains strong generalization capability and does not suffer from severe overfitting. This stability is particularly important in financial time-series forecasting, in which data is highly noisy and non-stationary. Furthermore, the smooth convergence pattern reflects the effectiveness of the proposed architecture along with the applied regularization techniques, such as Dropout and Batch Normalization, in controlling model complexity. Overall, the results demonstrate that the training process is successful and that the proposed framework is robust and reliable across different forecasting horizons.

Figure 13: Loss Evaluation Results of IICGS

4.3 Comparison with Baseline Models

In this subsection, the proposed IICGS framework is empirically evaluated against several baseline models, including statistical methods (ARIMA and GARCH), traditional machine learning classifiers (SVM, Random Forest, and XGBoost), and deep learning architectures (LSTM, GRU, CNN, and their hybrid variants). All baseline models are evaluated under the same experimental environment and data partitions to ensure a consistent and reproducible comparison. The evaluation considers two complementary aspects of Forex forecasting: (i) directional trend prediction, assessed using classification accuracy, F1-score, and bootstrap mean $\pm$ SD, and (ii) continuous price-level prediction across H1, H4, and D1 horizons, assessed using MAE and the Diebold–Mariano (DM) test. It should be noted that the results reported in this subsection are obtained by evaluating the single best-performing model selected from the ten independent runs. To provide broader context against existing literature, we also consider relevant benchmarking frameworks such as the study by Jarusek et al. [16]. They proposed a short-term Forex forecasting methodology integrating Elliott Wave patterns, Fast Fourier Transform (FFT), and neural networks on EUR/USD. While their evaluation reported an aggregated performance score of approximately 77% based on averaged percentage-converted error metrics (MSE, RMSE, MAE, and NCC), their output represents a composite regression-based index rather than standard directional classification accuracy. Consequently, the findings of Jarusek et al. [16] serve as background literature and a contextual reference for continuous price prediction rather than a direct benchmark. This distinction is necessary due to differences in datasets, sampling frequencies, time horizons, and evaluation protocols. Primary emphasis is therefore placed on controlled comparisons with baseline models evaluated under identical experimental conditions.

4.3.1 Directional Trend Forecasting Evaluation

The directional prediction capability is evaluated through both the Full-Dataset Classification Accuracy (%) and the Bootstrap simulation metrics reported as mean $\pm$ SD, as summarized in Table 9. According to the discussion of negative overfitting in the established literature [29], exceptionally high performance during the optimization phase does not necessarily translate into comparable performance during subsequent testing periods and may indicate overfitting. Therefore, evaluating model performance beyond the optimization period is essential for assessing its generalization capability. This methodological consideration is consistent with our empirical findings. Across 10 independent evaluation runs on the dedicated testing data (Table 3), the proposed IICGS framework achieves a stable mean accuracy of $78.97 \pm 0.5\%$. When evaluated on the entire dataset, including both training and testing subsets (Table 9), the accuracy is 78.47%. The small difference between these results suggests that the proposed framework does not exhibit substantial performance degradation when evaluated beyond the dedicated testing subset, supporting its ability to generalize effectively across different market conditions.

Crucially, the IICGS framework achieves the highest classification accuracy among all compared methods (78.47%), substantially outperforming traditional machine learning baselines such as XGBoost (53.61%), Random Forest (51.26%), and SVM (44.10%). Furthermore, empirical simulation demonstrates that our framework exhibits the lowest bootstrap standard deviation ($78.31 \pm 0.35\%$). This low SD indicates a highly concentrated distribution, reflecting superior prediction consistency and robustness under resampled market environments. This statistical stability proves that integrating Ichimoku-based features with an optimized CNN-GRU architecture effectively mitigates the inherent noise and high volatility of Forex time series. In addition to directional classification, continuous price-level tracking across multiple timeframes (H1, H4, and D1) demonstrates the capability of the regression architecture to capture price dynamics. Together with the 78.47% classification accuracy, these results indicate strong performance in both price tracking and directional classification, suggesting that the IICGS framework can effectively capture continuous price movements while providing reliable directional trend predictions under volatile market conditions.

Table 9: Comparative Analysis of Model Performances Based on Full-Dataset Classification Accuracy vs. Bootstrap mean $\pm$ SD (%) for EUR/USD.

Reference	Methodology	Accuracy (%)	Bootstrap [mean $\pm$ SD (%)]
Our Framework	IICGS	78.47	78.31 $\pm$ 0.35
xgboost Library	XGBoost Classifier	53.61	53.39 $\pm$ 0.58
sklearn Library	Random Forest Classifier	51.26	51.38 $\pm$ 0.61
sklearn Library	Support Vector Machine (SVM)	44.10	43.92 $\pm$ 0.65

4.3.2 Multi-Horizon Price Forecasting & Statistical Significance

To evaluate price-level prediction and verify whether the proposed framework's superiority is statistically significant rather than a result of random chance, we compute the Mean Absolute Error (MAE) and conduct the Diebold–Mariano (DM) test across multiple trading horizons (H1, H4, and D1). The proposed IICGS framework serves as the target model against which all baseline methods are individually compared. The empirical results are detailed in Table 10, which shows that the proposed IICGS framework consistently achieves the lowest MAE values among all compared methods, recording 0.0034, 0.0028, and 0.0027 for the H1, H4, and D1 horizons, respectively. Crucially, all computed DM statistics for the baseline models relative to IICGS are positive and substantially exceed the critical threshold of 1.96. The corresponding raw p -values and Holm–Bonferroni adjusted p -values ($m = 24$) remain far below $\alpha = 0.05$, formally rejecting the null hypothesis of equal predictive accuracy. This statistically confirms that the predictive gains of the proposed framework are highly significant and not attributable to sampling variation.

Table 10: Performance comparison, DM statistics, raw p -values and Holm–Bonferroni adjusted p -values across multiple forecasting horizons on EUR/USD dataset.

Model	MAE (H1)	MAE (H4)	MAE (D1)	DM (H1)	p (H1)	Holm (H1)	DM (H4)	p (H4)	Holm (H4)	DM (D1)	p (D1)	Holm (D1)
ARIMA	0.0191	0.0183	0.0195	14.32	1.64×10^{-46}	3.61×10^{-45}	15.11	1.39×10^{-51}	3.20×10^{-50}	16.04	6.72×10^{-58}	1.61×10^{-56}
GARCH	0.0168	0.0179	0.0182	12.05	1.94×10^{-33}	3.69×10^{-32}	12.89	5.12×10^{-38}	1.03×10^{-36}	13.91	5.51×10^{-44}	1.16×10^{-42}
Plain LSTM	0.0161	0.0175	0.0169	9.84	7.57×10^{-23}	1.14×10^{-21}	10.12	4.50×10^{-24}	7.21×10^{-23}	11.05	2.19×10^{-28}	3.95×10^{-27}
Plain GRU	0.0153	0.0165	0.0160	8.91	5.10×10^{-19}	6.12×10^{-18}	9.45	3.39×10^{-21}	4.74×10^{-20}	10.18	2.44×10^{-24}	4.14×10^{-23}
Plain CNN	0.0149	0.0162	0.0158	8.12	4.66×10^{-16}	4.20×10^{-15}	8.76	1.95×10^{-18}	2.15×10^{-17}	9.34	9.63×10^{-21}	1.25×10^{-19}
Plain CNN-LSTM	0.0137	0.0157	0.0151	7.23	4.83×10^{-13}	3.38×10^{-12}	7.89	3.02×10^{-15}	2.42×10^{-14}	8.42	3.77×10^{-17}	3.77×10^{-16}
Plain LSTM-GRU	0.0124	0.0128	0.0148	6.04	1.54×10^{-9}	6.17×10^{-9}	6.32	2.62×10^{-10}	1.31×10^{-9}	7.11	1.16×10^{-12}	6.96×10^{-12}
Plain CNN-GRU	0.0102	0.0110	0.0146	4.87	1.12×10^{-6}	1.12×10^{-6}	5.12	3.06×10^{-7}	6.11×10^{-7}	5.95	2.68×10^{-9}	8.04×10^{-9}
IICGS (Proposed)	0.0034	0.0028	0.0027	—	—	—	—	—	—	—	—	—

Note: All raw and Holm–Bonferroni adjusted p -values ($m = 24$) remain far below 0.05, confirming statistical significance at the 5% level.

Moreover, an analysis of the error trends across horizons reveals a distinctive property of IICGS: while baseline models (e.g., ARIMA, GARCH, and Plain CNN-GRU) exhibit performance degradation or increased error volatility when moving from short-term (H1) to longer-term (D1) horizons, the proposed framework shows a progressive reduction in MAE. This behavior indicates that the Ichimoku-inspired multi-scale feature extraction effectively isolates macro trend structures from intraday noise, enabling the optimized CNN-GRU layers to maintain exceptional prediction stability and error containment over broader temporal horizons.

4.3.3 Effectiveness Analysis of Model Components

To isolate and investigate the individual empirical contribution of each core component within our pipeline, an ablation study was conducted. Four distinct configurations were evaluated: (A) the baseline CNN-GRU network without enhancements, (B) the CNN-GRU network equipped with Ichimoku-based engineered features, (C) the CNN-GRU network optimized via the Starfish Optimization Algorithm (SFOA), and (D) the complete integrated IICGS framework.

The relative improvements reported in Table 11 quantify the percentage reduction in MAE with respect to the baseline CNN-GRU model, calculated as:

$$\text{Improvement (\%)} = \frac{MAE_{\text{Base}} - MAE_{\text{Model}}}{MAE_{\text{Base}}} \times 100. \quad (20)$$

The empirical findings from the ablation study indicate that integrating Ichimoku-based feature engineering independently drops the MAE from 0.0045 to 0.0040 (an 11.11% improvement), validating that multi-scale technical features enrich the input space beyond raw OHLC data. On the other hand, applying SFOA-based hyperparameter optimization independently reduces the MAE to 0.0038 (a 15.56% improvement), proving the optimization algorithm's strength in discovering superior architectural parameters by balancing exploitation and exploration. The best-so-far solution is obtained when both components are deployed simultaneously in Configuration D (IICGS), yielding an MAE of 0.0034. This corresponds to a total error reduction of 24.44% relative to the base network, indicating that the combination of structured financial indicators and metaheuristic optimization can contribute to improved forecasting performance.

Table 11: Ablation study of the proposed framework to evaluate the contribution of Ichimoku-based feature engineering and SFOA-based hyperparameter optimization.

Model	Configuration	MAE	Improvement (%)
Model A	CNN-GRU (Base model)	0.0045	—
Model B	CNN-GRU + Ichimoku	0.0040	11.11%
Model C	CNN-GRU + SFOA	0.0038	15.56%
Model D	CNN-GRU + Ichimoku + SFOA (Proposed)	0.0034	24.44%

4.4 Comparative Analysis of Metaheuristic Optimizers

In this study, a comparative evaluation of metaheuristic algorithms for hyperparameter tuning in deep learning models is conducted for Forex trend forecasting. The evaluated algorithms include the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer

(GWO), and the proposed Ichimoku-CNN-GRU-SFOA (IICGS) framework, which incorporates the Starfish Optimization Algorithm (SFOA). Performance is assessed using Accuracy (%), F1-score, and bootstrap statistics reported as mean $\pm$ SD across eight currency pairs. For each method, the results reported in Table 12 correspond to the best-performing model obtained from ten independent runs and subsequently evaluated on the complete dataset. Therefore, the IICGS results in this table may differ from those reported in Table 7, where the results are presented as the mean $\pm$ standard deviation across the ten independent runs. This difference reflects the distinct reporting procedures used in the two tables rather than an inconsistency in the experimental results. Overall, the proposed IICGS framework achieves higher predictive performance than GA, PSO, and GWO across all evaluated currency pairs. For example, for AUD/USD, IICGS achieves an Accuracy of 79.41%, with a bootstrap estimate of $79.28 \pm 0.38\%$ and an F1-score of 0.65. In comparison, GWO achieves 68.23% Accuracy ($68.11 \pm 0.59\%$), PSO achieves 68.56% ($68.41 \pm 0.58\%$), and GA achieves 59.12% ($58.94 \pm 0.65\%$). Similarly, for EUR/SGD, IICGS achieves 78.77% Accuracy and an F1-score of 0.61, compared with 70.12% and 0.57 for GWO, 69.88% and 0.55 for PSO, and 64.43% and 0.51 for GA. Across the evaluated currency pairs, IICGS consistently outperforms the three baseline optimization algorithms in both Accuracy and F1-score. The bootstrap results also show relatively small standard deviations for IICGS, indicating stable performance under resampling. Overall, these findings demonstrate the competitive performance and robustness of the proposed SFOA-based hyperparameter optimization framework.

Table 12: Comparison of hyperparameter optimization algorithms based on Accuracy (%) with Bootstrap mean $\pm$ SD (%) and F1-score.

Ticker	Accuracy (%) [Bootstrap mean $\pm$ SD]				F1-score			
	GA	PSO	GWO	IICGS (Ours)	GA	PSO	GWO	IICGS (Ours)
AUD/USD	59.12 (58.94 ± 0.65)	68.56 (68.41 ± 0.58)	68.23 (68.11 ± 0.59)	79.41 (79.28 ± 0.38)	0.50	0.55	0.55	0.65
EUR/SGD	64.43 (64.25 ± 0.64)	69.88 (69.73 ± 0.60)	70.12 (69.95 ± 0.58)	78.77 (78.61 ± 0.39)	0.51	0.55	0.57	0.61
GBP/NZD	57.77 (57.59 ± 0.66)	67.89 (67.74 ± 0.59)	67.46 (67.31 ± 0.61)	75.19 (75.05 ± 0.41)	0.52	0.57	0.58	0.66
NZD/USD	61.35 (61.18 ± 0.63)	70.12 (69.97 ± 0.57)	70.57 (70.42 ± 0.59)	78.17 (78.04 ± 0.37)	0.53	0.59	0.60	0.63
USD/CHF	63.57 (63.38 ± 0.65)	72.35 (72.19 ± 0.58)	70.89 (70.73 ± 0.60)	80.11 (79.96 ± 0.39)	0.50	0.54	0.54	0.63
USD/CNH	58.89 (58.71 ± 0.64)	65.43 (65.28 ± 0.61)	67.12 (66.97 ± 0.58)	70.57 (70.43 ± 0.40)	0.55	0.62	0.61	0.67
USD/SEK	54.23 (54.06 ± 0.62)	60.57 (60.42 ± 0.59)	62.35 (62.19 ± 0.57)	66.88 (66.74 ± 0.38)	0.54	0.60	0.58	0.64
USD/SGD	69.13 (68.95 ± 0.65)	76.47 (76.31 ± 0.58)	76.89 (76.74 ± 0.60)	83.23 (83.09 ± 0.36)	0.50	0.55	0.55	0.59

4.5 Limitations and Metrics in Evaluating Trading Models

In financial forecasting, traditional evaluation metrics such as MAE, MSE, and MAPE are commonly used to assess the performance of regression models. However, in algorithmic trading contexts, these metrics may be insufficient. A model with low average errors can still generate small fluctuations that inadvertently trigger stop-loss mechanisms, leading to premature trade

exits and undermining the overall strategy. This problem primarily stems from relying on one-step-ahead predictions, in which the trend direction is determined by merely comparing two consecutive steps (the immediate next prediction versus the current step). In lower timeframes, such a consecutive comparison is highly susceptible to localized anomalies; minor, sudden price fluctuations or market noise can easily flip the predicted direction, leading to severe trend misclassifications. As illustrated in Figure 14, while the predicted price closely tracks the actual price according to regression metrics, these micro-fluctuations frequently cross stop-loss thresholds, resulting in unnecessary losses. To mitigate this, we adopt a classification-based

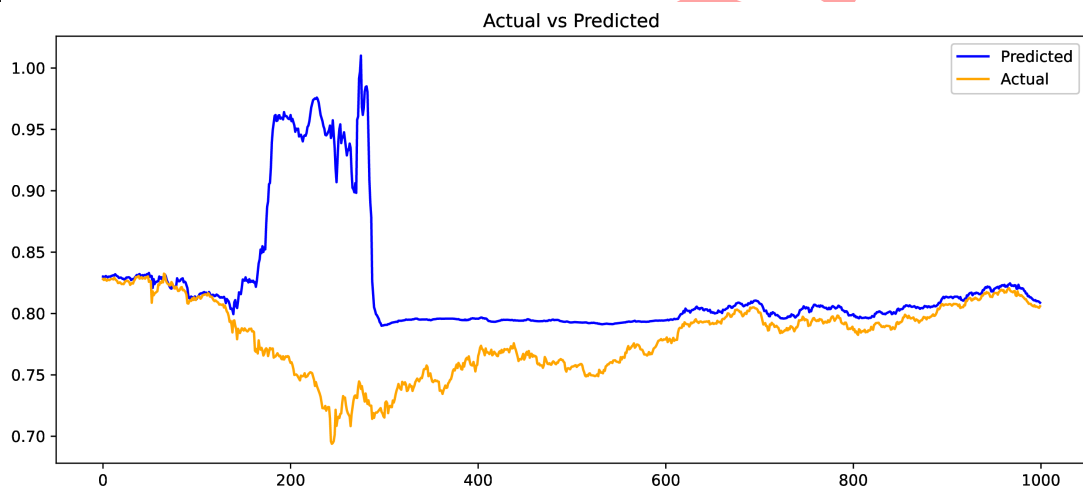


Figure 14: Predicted vs. actual prices with stop-loss crossings.

evaluation approach powered by a multi-step forecasting formulation. Instead of focusing on the magnitude of numerical errors between consecutive steps, we prioritize the accurate identification of the broader trend direction by evaluating multiple future steps against the current horizon. This multi-step approach inherently filters out sudden intraday noises and prevents premature stop-loss triggers, aligning the model more effectively with real-world trading needs in which strategic success depends on directional accuracy. Consequently, we utilize classification metrics, such as Accuracy and the F1-score, as key indicators of model performance across short-term (H1), medium-term (H4), and daily (D1) timeframes. To support this robust evaluation, the trend direction in each dataset is derived from historical price movements, resulting in a balanced distribution in which approximately 50% of the instances represent an upward trend and 50% a downward trend. This balanced class distribution, maintained across all three timeframes, reduces the effect of class imbalance and the dominance of any single class, thereby making Accuracy a more straightforward and reliable overall performance measure. The F1-score is additionally reported to provide a complementary assessment of the balance between precision and recall.

5 Conclusions

This study presented a robust multi-timeframe and multi-step framework for Forex trend prediction by integrating CNN-based deep learning architectures with Ichimoku-driven feature engineering and the Starfish Optimization Algorithm (SFOA). In this framework, the outputs of the regression models are treated as dynamic, price-following indicators. While higher regression accuracy pulls these indicators closer to the actual price and better reflects market tendencies, experimental observations revealed that relying solely on a single daily-scale regression model can create a false sense of reliability, as low regression errors may still result in substantial trend misclassifications. To address this limitation, a multi-timeframe and multi-step forecasting strategy was implemented using the EUR/USD currency pair as the primary training baseline. By extending the prediction horizon into multiple future steps for lower timeframes, such as evaluating the next several steps against the current one in the H1 dimension, the influence of minor, localized prediction errors can be reduced. The same process was applied to the H4 medium-term timeframe to further validate the directional trend before aggregating the signals into a final daily market decision through a voting mechanism. Among the tested configurations, the IICGS model achieved the most reliable performance in capturing complex sequential dependencies. Importantly, the proposed framework demonstrated strong out-of-sample generalization across various currency pairs. Without retraining, the framework maintained stable trend classification performance across these currency pairs, indicating zero-shot-like learning capabilities across the broader Forex market. Statistical validation using the Diebold–Mariano and Bootstrap tests further supported the significance and robustness of the observed performance improvements.

Limitations and Future Work. Although the proposed IICGS framework demonstrates consistent improvements in both regression and classification performance across multiple currency pairs and timeframes, several limitations should be acknowledged. First, the experimental evaluation is primarily conducted on the Forex market. Although the selected currency pairs represent diverse market characteristics, the generalizability of the proposed framework to other financial assets, such as equities, commodities, cryptocurrencies, or fixed-income markets, has not yet been systematically investigated. Future studies should therefore evaluate the framework across a broader range of asset classes to further assess its robustness and transferability. Second, although the Starfish Optimization Algorithm (SFOA) provides effective exploration and exploitation capabilities, its stochastic nature, like that of other evolutionary optimization algorithms, introduces an inherent degree of uncertainty into the optimization process. Therefore, future studies could investigate hybrid adaptive optimization strategies that combine evolutionary algorithms such as SFOA with classical deterministic optimization methods, potentially achieving a better balance between global exploration and local refinement while reducing dependence on stochastic search. Since the optimization process is performed only during model

training and does not affect the inference stage, such strategies could also contribute to improving the computational efficiency and stability of the optimization process. Additionally, future work could incorporate new technical features, such as the VWAP indicator, or indices derived from sentiment and fundamental analysis to further enrich the feature space and potentially improve predictive performance. Future research will also investigate transformer-based architectures, adaptive trading systems, and the application of the proposed framework to higher-volatility assets such as XAU/USD and cryptocurrencies. Addressing these limitations may further improve the scalability, transferability, and practical applicability of the proposed framework in real-world financial forecasting.

Appendix. Starfish Optimization Algorithm (SFOA)

The Starfish Optimization Algorithm (SFOA) is a bio-inspired metaheuristic designed to solve global optimization problems. By mimicking the behaviors of starfish, SFOA introduces a hybrid search strategy that effectively balances exploration and exploitation. The algorithm comprises five primary phases, each of which is mathematically defined as follows:

Initialization Phase

In the initialization phase, the positions of the starfish are randomly generated within the boundaries of the design variables. The positions are represented as Equation (21):

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1D} \\ X_{21} & X_{22} & \dots & X_{2D} \\ \vdots & \vdots & \dots & \vdots \\ X_{N1} & X_{N2} & \dots & X_{ND} \end{bmatrix}. \quad (21)$$

where $\mathbf{X}$ is a matrix of size $N \times D$, with N representing the population size and D the number of dimensions of the design variables. The position of each starfish is calculated using Equation (22):

$$X_{ij} = l_j + r \cdot (u_j - l_j), \quad i = 1, 2, \dots, N, j = 1, 2, \dots, D. \quad (22)$$

where X_{ij} denotes the position of the i -th starfish in the j -th dimension, r is a random number between 0 and 1, and l_j and u_j represent the lower and upper bounds of the j -th dimension, respectively. After initializing the positions, the fitness values for all starfish are computed using the objective function. These fitness values are stored in a vector as Equation (23):

$$\mathbf{F} = \begin{bmatrix} F(X_1) \\ F(X_2) \\ \vdots \\ F(X_N) \end{bmatrix}, \quad (23)$$

where $\mathbf{F}$ is a vector of size $N \times 1$, containing the fitness values of the starfish. Once the initialization is complete, the algorithm proceeds to the exploration and exploitation phases.

Exploration Phase

The exploration phase simulates the searching capabilities of the starfish, modeled as a five-armed structure with eyes at the tips of each arm. The exploration behavior is governed by the following principles:

- For optimization problems with dimensions greater than 5 ($D > 5$), the algorithm uses a five-dimensional search pattern. Each arm moves in one of five randomly selected dimensions to explore the search space more efficiently. This exploration is guided by the best position among all starfish, and the update for the position is given by Equation (24):

$$Y_{i,p}^T = \begin{cases} X_{i,p}^T + a_1(X_{best,p}^T - X_{i,p}^T) \cos \theta, & \text{for } r \leq 0.5, \\ X_{i,p}^T - a_1(X_{best,p}^T - X_{i,p}^T) \sin \theta, & \text{for } r > 0.5, \end{cases} \quad (24)$$

where $Y_{i,p}^T$ and $X_{i,p}^T$ represent the obtained and current positions, respectively, $X_{best,p}^T$ is the best known position in the p -th dimension, r is a random number, and a_1 and θ are defined as Equation (25) and Equation (26):

$$a_1 = (2r - 1)\pi, \quad (25)$$

$$\theta = \frac{\pi}{2} \cdot \frac{T}{T_{max}}, \quad (26)$$

where T is the current iteration and T_{max} is the maximum number of iterations.

- For $D > 5$, the five-dimensional search pattern is used to update only five dimensions at a time. If the updated position exceeds the design variable boundaries, the starfish remains in the previous position as Equation (27):

$$X_{i,p}^{T+1} = \begin{cases} Y_{i,p}^T, & \text{if } l_{b,p} \geq Y_{i,p}^T \leq u_{b,p}l_{b,p}, \\ X_{i,p}^T, & \text{otherwise.} \end{cases} \quad (27)$$

- For optimization problems with $D \leq 5$, a unidimensional search pattern is applied, with only one arm of the starfish moving to explore the environment. The updated position is given by Equation (28):

$$Y_{i,q}^T = E_t X_{i,p}^T + A_1(X_{k_1,p}^T - X_{i,p}^T) + A_2(X_{k_2,p}^T - X_{i,p}^T). \quad (28)$$

where $X_{k_1,p}^T$ and $X_{k_2,p}^T$ are the positions of two randomly selected starfish, and A_1 and A_2 are random numbers between $(-1, 1)$. The energy E_t is computed as Equation (29):

$$E_t = \frac{T_{max} - T}{T_{max}} \cos \theta. \quad (29)$$

If the obtained position is outside the design variable boundaries, the starfish stays in its previous position.

Exploitation Phase

The exploitation phase models the preying and regeneration behaviors of the starfish to refine the search for global solutions. This phase involves two strategies: preying and regeneration.

Preying Phase The distances between the global best position and the positions of other starfish are computed. Two distances are selected randomly to update each starfish's position Equation (30):

$$d_m = (X_{best}^T - X_m^T), \quad m = 1, \dots, 5. \quad (30)$$

The update for each starfish during the preying phase is given by Equation (31):

$$Y_i^T = X_i^T + r_1 d_{m_1} + r_2 d_{m_2}, \quad (31)$$

where r_1 and r_2 are random numbers between $(0,1)$, and d_{m_1} and d_{m_2} are the randomly selected distances.

Regeneration Phase For the last starfish in the population, the position is updated using a slow movement model Equation (32):

$$Y_i^T = \exp\left(-T \times \frac{N}{T_{max}}\right) X_i^T. \quad (32)$$

If the position obtained from either the preying or regeneration phases is outside the design variable boundaries, it is adjusted as Equation (33):

$$X_i^{T+1} = \begin{cases} Y_i^T, & l_b \leq Y_i^T \leq u_b, \\ l_b, & Y_i^T < l_b, \\ u_b, & u_b < Y_i^T. \end{cases} \quad (33)$$

Algorithm 2 Pseudo-code of SFOA algorithm**Input:** Algorithmic parameters (N , $T_{\max}$, and GP)**Output:** The global solution

```

1: Initialize input parameters, problem information, boundary, etc.
2: Initialize positions by Equation (21) and evaluate the fitness values by Equation (23)
3: while  $T < T_{\max}$  do
4:   if  $\text{rand} > GP$  then
5:     Calculate  $\theta$  by Equation (26) and energy  $E_t$  by Equation (29)
6:     for each starfish do
7:       if  $D > 5$  then
8:         Generate five randomly selected  $p$ -dimensions among  $D$ 
9:         for  $j = 1 : 5$  do
10:          Calculate  $a_1$  by Equation (25)
11:          Update the dimension of position in starfish using Equation 24
12:        end for
13:        Check the boundary of position by Equation (27)
14:      else
15:        Generate two random numbers  $A_1$  and  $A_2$ 
16:        Select a random dimension  $p$ 
17:        Update the  $p$ -dimension of position using Equation (28)
18:        Check the boundary of starfish by Equation (33)
19:      end if
20:    end for
21:  else
22:    Generate distances between the best position and starfish
23:    for each starfish do
24:      Generate random number  $r_k$ , and update the position using Equation (31)
25:      if  $i = N$  then
26:        Update the position using Equation (32)
27:      end if
28:      Check the boundary of starfish by Equation (33)
29:    end for
30:  end if
31:  Calculate the fitness values for all starfish
32:  Obtain the global best solution in iteration  $T$  step
33:   $T = T + 1$ 
34: end while
35: Obtain global solution

```

The SFOA continues iterating through these phases, progressively improving the starfish positions and updating the best-so-far solution throughout the optimization process [31]. In the SFOA, GP denotes the exploration control parameter, which regulates the selection between the exploration and exploitation phases. At each iteration, a random number is generated and compared with GP . When $\text{rand} > GP$, the exploration phase is performed; otherwise, the exploitation phase is executed. Thus, GP controls the balance between exploration of the search space and exploitation of promising solutions. The pseudocode for the SFOA is presented in Algorithm 2.

Declarations

Availability of Supporting Data

The complete implementations of two models, along with their corresponding testing procedures, are released via the following GitHub repository: <https://github.com/YasHmi06/Price-Prediction.git>.

Funding

The authors received no specific funding for this research.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Authors' Contributions

Seyedeh Yasamin Hashemi: Conceptualization; Methodology; Data curation; Formal analysis; Model implementation; Software; Writing – original draft; Visualization. **Omid Solaymani Fard:** Conceptualization; Methodology; Validation; Writing – review & editing; Supervision; Theoretical development; Corresponding author.

Artificial Intelligence Statement

Artificial intelligence (AI) tools were used solely for language editing and manuscript refinement, including grammatical correction, sentence restructuring for clarity, and improvement of phrasing and readability. The AI tools were not used to generate scientific ideas, design experiments, perform analyses, interpret results, or develop scientific conclusions. All scientific content, methodological decisions, analyses, interpretations, and conclusions were independently developed and verified by the authors, who retain full responsibility for the final manuscript.

Publisher's Note

The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

References

- [1] Ayitey Junior, M., Appiahene, P., Appiah, O., Bombie, C. N. (2023). "Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis". *Journal of Big Data*, 10(1), 9. <https://doi.org/10.1186/s40537-022-00676-2>
- [2] Bank for International Settlements. (2019). "Triennial Central Bank Survey of foreign exchange and OTC derivatives markets in 2019: Foreign exchange turnover in April 2019". *Bank for International Settlements*. https://www.bis.org/statistics/rpfx19_fx.pdf
- [3] Bank for International Settlements. (2022). "Triennial Central Bank Survey of foreign exchange and over-the-counter (OTC) derivatives markets in 2022". *Bank for International Settlements*. https://www.bis.org/statistics/rpfx22_fx.pdf.
- [4] Cho, K., Van Merriënboer, B., Gulçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). "Learning phrase representations using RNN encoder–decoder for statistical machine translation". *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
- [5] Deng, S., Xiao, C., Zhu, Y., Peng, J., Li, J., Liu, Z. (2023). "High-frequency direction forecasting and simulation trading of the crude oil futures using Ichimoku KinkoHyo and Fuzzy Rough Set". *Expert Systems with Applications*, 215, 119326. <https://doi.org/10.1016/j.eswa.2022.119326>
- [6] Diebold, F. X., Mariano, R. S. (2002). "Comparing predictive accuracy". *Journal of Business & Economic Statistics*, 20(1), 134–144. <https://doi.org/10.1198/073500102753410444>
- [7] Dodge, Y. (Ed.). (2003). "The Oxford dictionary of statistical terms". *Oxford University Press*.
- [8] Elliott, N. (2007). "Ichimoku charts: An introduction to Ichimoku Kinko clouds". *Harri-man House Limited*.

- [9] Gauss, C. F. (1877). "Theoria motus corporum coelestium in sectionibus conicis solem ambientium (Vol. 7)". *Cambridge: Cambridge University Press*. <https://doi.org/10.1017/CB09780511841705>
- [10] Goodfellow, I., Bengio, Y., Courville, A. (2016). "Deep learning". *MIT Press*.
- [11] Gülmez, B. (2023). "Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm". *Expert Systems with Applications*, 227, 120346. <https://doi.org/10.1016/j.eswa.2023.120346>
- [12] Hochreiter, S., Schmidhuber, J. (1997). "Long short-term memory". *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [13] Hoffmann, J., Navarro, O., Kastner, F., Janßen, B., Hubner, M. (2017). "A survey on CNN and RNN implementations". In *PESARO 2017: The Seventh International Conference on Performance, Safety and Robustness in Complex Systems and Applications*.
- [14] Holm, S. (1979). "A simple sequentially rejective multiple test procedure". *Scandinavian journal of statistics*, 65-70.
- [15] Islam, M. S., Hossain, E. (2021). "Foreign exchange currency rate prediction using a GRU-LSTM hybrid network". *Soft Computing Letters*, 3, 100009. <https://doi.org/10.1016/j.socl.2020.100009>
- [16] Jarusek, R., Volna, E., Kotyrba, M. (2022). "FOREX rate prediction improved by Elliott waves patterns based on neural networks". *Neural Networks*, 145, 342-355. <https://doi.org/10.1016/j.neunet.2021.10.024>
- [17] Khaldi, R., El Afia, A., Chiheb, R., Tabik, S. (2023). "What is the best RNN-cell structure to forecast each time series behavior?". *Expert Systems with Applications*, 215, 119140. <https://doi.org/10.1016/j.eswa.2022.119140>
- [18] Khattak, B. H. A., Shafi, I., Rashid, C. H., Safran, M., Alfarhood, S., Ashraf, I. (2024). "Profitability trend prediction in crypto financial markets using Fibonacci technical indicator and hybrid CNN model". *Journal of Big Data*, 11(1), 58. <https://doi.org/10.1186/s40537-024-00908-7>
- [19] Kim, J., Oh, S., Kim, H., Choi, W. (2023). "Tutorial on time series prediction using 1D-CNN and BiLSTM: A case example of peak electricity demand and system marginal price prediction". *Engineering Applications of Artificial Intelligence*, 126, 106817. <https://doi.org/10.1016/j.engappai.2023.106817>

- [20] Künsch, H. R. (1989). "The jackknife and the bootstrap for general stationary observations". In *The Annals of Statistics*, 1217-1241. <https://doi.org/10.1214/aos/1176347265>
- [21] Lawal, A., Rehman, S., Alhems, L. M., Alam, M. M. (2021). "Wind speed prediction using hybrid 1D CNN and BLSTM network". *IEEE Access*, 9, 156672–156679. <https://doi.org/10.1109/ACCESS.2021.3129883>
- [22] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (2002). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- [23] Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2021). "A survey of convolutional neural networks: Analysis, applications, and prospects". *IEEE transactions on neural networks and learning systems*, 33(12), 6999-7019. <https://doi.org/10.1109/TNNLS.2021.3084827>
- [24] Lim, M. A. (2015). "The handbook of technical analysis+ test bank: The practitioner's comprehensive guide to technical analysis". *John Wiley & Sons*.
- [25] Mabrouk, N., Chihab, M., Hachkar, Z., Chihab, Y. (2022). "Intraday trading strategy based on gated recurrent unit and convolutional neural network: Forecasting daily price direction". *International Journal of Advanced Computer Science and Applications*, 13(3). <https://doi.org/10.14569/IJACSA.2022.0130369>
- [26] Makridakis, S., Wheelwright, S. C., Hyndman, R. J. (2008). "Forecasting methods and applications". *John Wiley & Sons*.
- [27] Mirzajani, S., Roohi, M. (2026). "Feedback long short-term memory: A long short-term memory-based framework for multivariate time series prediction in chaotic systems". *Control and Optimization in Applied Mathematics*, 11(1), 199–218. <https://doi.org/10.30473/coam.2025.72730.1312>
- [28] Özorhan, M. O., Toroslu, İ. H., Şehitoğlu, O. T. (2019). "Short-term trend prediction in financial time series data". *Knowledge and Information Systems*, 61(1), 397-429. <https://doi.org/10.1007/s10115-018-1303-x>
- [29] Sevastjanov, P., Kaczmarek, K., Rutkowski, L. (2024). "A multi-model approach to the development of algorithmic trading systems for the Forex market". *Expert Systems with Applications*, 236, 121310. <https://doi.org/10.1016/j.eswa.2023.121310>

- [30] Sokolova, M., Lapalme, G. (2009). "A systematic analysis of performance measures for classification tasks". *Information Processing & Management*, 45(4), 427-437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [31] Zhong, C., Li, G., Meng, Z., Li, H., Yildiz, A. R., Mirjalili, S. (2025). "Starfish optimization algorithm (SFOA): A bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers". *Neural Computing and Applications*, 37(5), 3641-3683. <https://doi.org/10.1007/s00521-024-10694-1>

Authors' Bio-sketches

Seyedeh Yasamin Hashemi is a Ph.D. candidate in Applied Mathematics, specializing in Operations Research, at the Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran, where she has been pursuing her doctoral studies since September 2020. Her primary research interests include optimization and deep learning.

Omid Solaymani Fard received his Ph.D. degree in Applied Mathematics from Ferdowsi University of Mashhad, Iran, in 2004. He is currently an Associate Professor of Optimization at the same institution. His primary research interests include optimization, fuzzy systems, and machine learning. Corresponding author. Email: soleimani@ferdowsi.um.ac.ir

Received: xxx Accepted: xxx Published: xxx.

DOI: [10.30473/coam.2026.77493.1402](https://doi.org/10.30473/coam.2026.77493.1402)

Volume xxx, Issue xxx, p.p. 53-55: xxx

Research Article



Open Access

Control and Optimization in Applied Mathematics - COAM

Multi-Timeframe Forex Trend Prediction via Ichimoku-Based Feature Engineering and Starfish-Optimized CNN-LSTM/CNN-GRU Networks

Seyedeh Yasamin Hashemi[✉], Omid Solaymani Fard[✉]

Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

✉ Correspondence:

Omid Solaymani Fard

E-mail:

soleimani@ferdowsi.um.ac.ir

How to Cite

Hashemi, S. Y., Solaymani Fard, O. (2027). "Multi-timeframe Forex trend prediction via Ichimoku-based feature engineering and starfish-optimized CNN-LSTM/CNN-GRU networks". *Control and Optimization in Applied Mathematics*, X(x), 53–55. <https://doi.org/10.30473/coam.2026.77493.1402>

Abstract. As the Forex market becomes increasingly complex, accurate directional forecasting is essential for effective trading decisions. This study proposes a multi-timeframe framework to evaluate directional trend prediction for the EUR/USD exchange rate using Ichimoku Cloud-inspired feature engineering and CNN-based hybrid architectures, including CNN, CNN-LSTM, and CNN-GRU. The study optimizes model hyperparameters using the Starfish Optimization Algorithm (SFOA). Specifically, the study investigates multi-step trend prediction for the H1 and H4 timeframes, and one-step-ahead prediction for the daily (D1) timeframe. Historical EUR/USD data are used to train continuous forecasting models, whose outputs are transformed into discrete trend classification labels. The framework ultimately employs a voting mechanism to determine the final trend decision at the daily (D1) timeframe. To assess the model's zero-shot classification capability and generalization, eight additional currency pairs are evaluated using the same evaluation criteria at the relevant stages to assess both individual model performance and overall framework effectiveness. Regression performance is assessed using MSE, MAE, and MAPE, while classification performance is evaluated using Accuracy and F1-score. Statistical robustness is further examined using the Diebold–Mariano test and bootstrap analysis. Experimental results demonstrate that incorporating GRU layers into CNN architectures consistently improves forecasting performance across multiple timeframes and currency pairs, with the proposed IICGS architecture achieving the most competitive overall performance. Selected results and implementation code are publicly available on GitHub.

Keywords. Forex trend prediction, Deep learning (CNN, LSTM, GRU), Ichimoku-based feature engineering, Starfish Optimization Algorithm, Diebold–Mariano test.

MSC. 68T07; 91G15; 90C59.

<https://mathco.journals.pnu.ac.ir>

©2026 by the authors. Licensee PNU, Tehran, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY4.0) (<http://creativecommons.org/licenses/by/4.0>)

A Introduction

The foreign exchange (Forex) market is widely recognized for its high liquidity, substantial volatility, decentralization, and rapid transaction execution. These characteristics have attracted considerable attention from both researchers and market practitioners, particularly in algorithmic trading, wherein automated systems use historical market data and computational models to optimize trading strategies. In this context, technical analysis serves as a foundational methodology, utilizing historical price data—such as open, high, low, and close prices—along with mathematical indicators to identify market trends, price behavior patterns, and potential reversals. Among these indicators, the Ichimoku Kinko Hyo (Ichimoku Cloud) stands out for its comprehensive nature, offering insights into support and resistance levels, trend direction, momentum, and trading signals. Developed in pre-World War II Japan by journalist Goichi Hosoda under the pseudonym “Ichimoku Sanjin” (meaning “at a glance”), this tool was designed to provide a holistic “mountain view” of market dynamics by integrating multiple moving averages and forward-looking components. Despite its historical origins and initially complex manual calculations, the Ichimoku Cloud is still widely used in modern trading applications for both short-term and long-term market analysis [8].

Recent advances in deep learning have substantially increased interest in financial forecasting and trend prediction, particularly in the foreign exchange (Forex) market. Models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Networks (CNNs) have demonstrated strong capabilities in capturing nonlinear and complex temporal patterns. Their effectiveness in modeling challenging dynamical systems has also been demonstrated beyond financial applications. For instance, Mirzajani et al. [27] proposed feedback-based LSTM architectures for multivariate prediction of the chaotic Lorenz system, achieving higher predictive accuracy than conventional RNN and LSTM models. These findings highlight the potential of recurrent architectures to capture complex temporal dependencies in nonlinear and chaotic systems, which are also characteristic of financial markets. In the Forex domain, a systematic literature review by Ayitey Junior et al. [1] identified a growing shift toward hybrid machine learning and deep learning approaches, emphasizing their potential to improve forecasting accuracy in noisy financial time series. Consistent with this trend, Islam et al. [15] proposed a hybrid GRU-LSTM architecture that outperformed standalone GRU and LSTM models in forecasting several major currency pairs. Similarly, Mabrouk et al. [25] developed a CNN-GRU-based trading framework that integrated technical indicators with deep learning for daily trading-signal prediction, achieving superior forecasting accuracy and trading performance compared with conventional machine learning methods. Collectively, these studies demonstrate the growing potential of deep and hybrid learning architectures for capturing the nonlinear and dynamic behavior of financial markets. However, the predictive performance of these models depends heavily on appropriate hyperparameter selection. Consequently, meta-

heuristic optimization algorithms have become increasingly popular for automatically tuning hyperparameters and improving forecasting performance. For example, Gülmez [11] proposed a metaheuristic-based LSTM model for stock price prediction and reported significant improvements in MAE, MSE, and MAPE through optimized hyperparameter selection. Owing to their ability to efficiently explore large and complex search spaces, metaheuristic algorithms are particularly well suited for financial prediction problems. Motivated by these advantages, this study employs the Starfish Optimization Algorithm (SFOA) [31] to optimize the hyperparameters of the proposed deep learning models. Inspired by the adaptive and regenerative behavior of starfish, SFOA performs population-based exploration to identify suitable hyperparameter configurations, including network architecture, thereby improving convergence stability and reducing the risk of premature convergence to local optima.

Beyond model architecture and hyperparameter optimization, recent studies have also emphasized the importance of feature engineering in enhancing forecasting performance. Deng et al. [5], for instance, proposed a hybrid trading system that combined Ichimoku-based feature extraction with fuzzy rough sets and NSGA-II optimization for crude oil futures forecasting, demonstrating improvements in both profitability and risk management. Likewise, Sevastjanov et al. [29] showed that integrating multiple forecasting models with novel technical indicators can reduce overfitting and improve adaptability under changing market conditions. Özorhan et al. [28] improved EUR/USD trend prediction using ZigZag-based motif discovery and SVM classification, while Khattak et al. [18] integrated Fibonacci indicators with a hybrid CNN-LSTM architecture for cryptocurrency trend prediction, highlighting the importance of profitability-oriented evaluation in practical trading environments. Collectively, these studies indicate that combining advanced deep learning architectures with effective optimization techniques and carefully designed technical indicators provides a promising direction for developing more accurate and robust financial forecasting systems.

Although these studies have reported encouraging results, several limitations remain. Many existing approaches focus mainly on price forecasting accuracy while paying less attention to directional trend prediction, which is often more relevant in trading applications. In addition, most studies rely heavily on regression metrics such as MAE, MSE, and MAPE. While these measures evaluate numerical forecasting error, they do not necessarily reflect whether the predicted market direction is correct. In practical trading, even a small prediction error may still lead to an incorrect trading signal and financial loss. Another limitation is the lack of rigorous statistical validation in many forecasting frameworks. Comparisons between models are often reported without statistical tests capable of determining whether performance differences are truly significant.

To address these limitations, this study proposes a hybrid forecasting framework that combines Ichimoku Cloud feature engineering with CNN-based deep learning architectures for multi-timeframe trend prediction. The proposed framework considers H1, H4, and D1 time-

frames to capture short-, medium-, and long-term market behavior. Regression models are trained to perform one-step-ahead price forecasting at each timeframe. At H1 and H4, successive one-step-ahead forecasts are generated, and the predicted prices corresponding to the daily forecasting horizon are compared to determine the associated daily directional trend. At D1, the daily trend is obtained by directly comparing two consecutive one-step-ahead predicted prices. The directional signals from H1, H4, and D1 are then integrated through a voting mechanism to produce a single final daily trend decision. Thus, regression constitutes the training and optimization objective, whereas directional classification represents the final decision-making objective of the framework. Three deep learning architectures—CNN, CNN-LSTM, and CNN-GRU—are investigated. The Starfish Optimization Algorithm (SFOA) is employed for hyperparameter optimization and is compared with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO). Regression performance is evaluated using MSE, MAE, and MAPE, with validation MAE serving as the fitness function for SFOA. At the final decision stage, the directional outputs are assessed using Accuracy and F1-score. The statistical reliability of the regression results is examined using the Diebold–Mariano (DM) test with Holm–Bonferroni correction, together with a moving block bootstrap procedure to assess the stability of the final classification performance. In addition to EUR/USD, eight previously unseen currency pairs are considered to evaluate the zero-shot classification capability of the trained framework. The main contributions of this study are as follows:

- A multi-timeframe hybrid framework is developed that explicitly separates the regression training objective from the final directional decision-making objective. One-step-ahead price forecasts at H1, H4, and D1 are converted into directional signals (by comparing prices aligned with the daily horizon at H1/H4 and consecutive predictions at D1) and aggregated via a voting mechanism to obtain the final daily trend.
- A unified evaluation strategy is established in which MSE, MAE, and MAPE guide model training and optimization, while Accuracy and F1-score evaluate the ultimate trend-prediction performance of the overall framework.
- The modeling and optimization strategy is systematically compared across CNN, CNN-LSTM, and CNN-GRU architectures and against established metaheuristic methods (GA, PSO, GWO), with statistical significance assessed by the DM test (Holm–Bonferroni adjusted) and performance stability examined via moving block bootstrap.
- The zero-shot cross-market robustness of the trained framework is evaluated on eight currency pairs that were not used during training, assessing both regression forecasting quality and the final daily trend-classification performance obtained through the voting mechanism.

The remainder of this paper is organized as follows. Section 2 describes the proposed methodology, data collection process, and model development procedure. Section 3 presents the experimental results and statistical analyses. Section 4 discusses the findings and practical implications of the study, and Section 5 concludes the paper and outlines directions for future research.

B Material and methods

The way in which the creator of Ichimoku views and analyzes the market forms the basis, motivation, and goal of designing our proposed neural networks. Before exploring the main topic, a brief overview of the concepts behind LSTM, GRU, and CNN is essential. In addition, a detailed explanation of the SFOA metaheuristic algorithm is provided in Appendix.

B.1 Long Short-Term Memory

Hochreiter et al. [12] introduced long short-term memory (LSTM) networks, a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data such as time series, natural language, and audio. Unlike traditional RNNs, which suffer from vanishing or exploding gradients, LSTMs overcome this limitation through a gating mechanism consisting of a forget gate, input gate, and output gate. These gates enable the model to selectively retain, update, or discard information, making LSTMs particularly suitable for financial forecasting tasks such as stock prediction, Forex trend analysis, and cryptocurrency modeling.

Let x_t represent the input vector at time step t and h_{t-1} the hidden state of the previous step. The forget gate, f_t , controls the proportion of the previous cell state, C_{t-1} , to retain:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (1)$$

The input gate, i_t , determines the extent to which new information is stored in the cell state. The candidate cell state, $\tilde{C}_t$, represents the new content to be added:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \quad (3)$$

The cell state, C_t , is then updated as in Equation (4):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t. \quad (4)$$

The output gate, o_t , determines how much of the internal memory is exposed to the next hidden state, h_t :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (5)$$

$$h_t = o_t \odot \tanh(C_t). \quad (6)$$

This gating system enables LSTMs to maintain a dynamic memory of important temporal patterns over long horizons, critical for modeling non-linear dynamics in financial time series.

B.2 Gated Recurrent Unit

The Gated Recurrent Unit (GRU), introduced by Cho et al. [4], is a simplified recurrent neural network architecture designed to tackle challenges such as the problem of vanishing gradients. GRUs use two gates: the update gate, z_t , which balances the retention of past information with the incorporation of new input, and the reset gate, r_t , which controls how much of the previous hidden state is forgotten.

Candidate activation, $\tilde{h}_t$, combines the effect of the reset gate with the current input to generate new information. The final hidden state, h_t , is computed as a weighted sum of the previous hidden state and the candidate activation, modulated by the update gate:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t. \quad (7)$$

GRUs are computationally efficient, requiring fewer parameters than LSTMs, while still effectively capturing dependencies in sequential data. They are widely used in applications like language modeling, time series forecasting, and speech recognition.

While both LSTMs and GRUs present distinct architectural trade-offs, deciding when to deploy each cell structure—or their simplified variants—depends heavily on the underlying statistical behavior of the target time series. Recent taxonomy work by Khaldi et al. [17] systematically evaluated 20 recurrent cell architectures across fundamental time series behaviors (deterministic, random-walk, non-linear, long-memory, and chaotic). Their empirical findings revealed that structural modifications—such as parameter-reduced variants like SLIM—yield statistically significant performance gains tailored to specific series dynamics (e.g., SLIM-LSTM for chaotic dynamics and SLIM-MGU for long-memory processes). This benchmark underscores the necessity of evaluating both standard LSTM and GRU foundations when constructing forecasting frameworks for complex, multi-behavior financial series.

B.3 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are deep learning models that are very effective for tasks involving spatial data, such as images and sequences. The foundational work of LeCun et al. [22] established CNNs as powerful tools for image recognition and classification tasks, demonstrating their ability to achieve high precision in complex data sets.

CNNs are widely recognized for their ability to automatically and adaptively learn spatial hierarchies of features from data through three main types of layers: convolutional layers, pooling layers, and fully connected layers. The convolutional layers are responsible for feature extraction by applying a set of learnable filters (kernels) across the input. This enables the network to detect patterns and features, such as edges, textures, and shapes, which are crucial for tasks such as image recognition [10]. The pooling layers reduce the dimensionality of feature maps, preserving key features while discarding less important information. This not only reduces computational requirements, but also helps to make the network less prone to overfitting [13]. For example, in modern CNNs, max-pooling is widely used to retain the most important aspects of each feature while reducing noise [23]. Finally, the fully connected layer combines the extracted features to produce the final output, typically used for classification or regression tasks. This layer links the learned feature representations to specific output classes, facilitating tasks such as object identification in images. CNNs utilize activation functions, often Rectified Linear Unit (ReLU), to introduce nonlinearities, allowing them to learn complex functions [13].

Beyond image processing, CNNs can be applied effectively to one-dimensional data using 1D convolutional kernels. One-dimensional CNNs (1D CNNs) are particularly well suited for tasks like time series prediction and signal identification, in which the absolute position of features is less critical. For example, combining 1D CNNs with bidirectional LSTM (BiLSTM) networks has been effective in complex prediction tasks that involve volatile data. Lawal et al. [21] developed a hybrid model that combines 1D CNN and BiLSTM to accurately forecast short-term wind speed by extracting spatial and temporal features, demonstrating high performance at various heights above ground level. Similarly, Kim et al. [19] employed a 1D CNN-BiLSTM model for time series predictions of peak electricity demand and marginal price of the system, illustrating that this approach performs better than conventional methods by efficiently capturing sequential dependencies and improving prediction accuracy.

B.4 Ichimoku-Inspired Feature Engineering

The intersections of the Ichimoku indicators Tenkan-sen, Kijun-sen, Senkou Span A, Senkou Span B, and Chikou Span are widely utilized in technical analysis to generate trading signals. Taking inspiration from this approach, we engineered new features based on these intersec-

tions. These features are derived from five key Ichimoku-based trend-identification signals, as described below.

Tenkan-sen and Kijun-sen Cross When the Tenkan-sen (Conversion Line) crosses above the Kijun-sen (Base Line), it signals a potential uptrend. In contrast, a downward cross suggests a possible downward trend (Figure 1).



Figure 1: Tenkan-sen and Kijun-sen Cross

Senkou Span A and Senkou Span B Cross (Cloud Boundary) When Senkou Span A (the faster-moving edge of the cloud) crosses above Senkou Span B, it suggests an uptrend, while a downward cross indicates a downtrend. This intersection defines the “Kumo” (cloud) boundaries and highlights the support and resistance levels (Figure 2).

Tenkan-sen and Kijun-sen Crossing Above or Below the Cloud When both Tenkan-sen and Kijun-sen cross above the cloud (bounded by Senkou Span A and B), it signals a strong uptrend. In contrast, crossing below the cloud indicates a potential downward trend. The cloud serves as a long-term trend indicator, and these crossings provide valuable signals (Figure 3).

Price and Ichimoku Indicators Cross When the price crosses above Tenkan-sen, Kijun-sen, or the cloud, it indicates a potential uptrend and market strength. In contrast, crossing below these indicators signals a potential downward trend, reflecting price movements relative to support and resistance levels (Figure 4).

Chikou Span Cross (Lagging Line) The Chikou Span, a lagging indicator that shifts the closing price backward by 26 periods, provides historical context for trend analysis. When



Figure 2: Senkou Span A and Senkou Span B Cross



Figure 3: Tenkan-sen and Kijun-sen Crossing Above or Below the Cloud



Figure 4: Price and Ichimoku Indicators Cross

the Chikou Span rises above the Tenkan-sen, Kijun-sen, or cloud, it suggests an uptrend. Conversely, when it falls below these indicators, it signals a downtrend, helping identify trend persistence or reversal (Figure 5) [24].



Figure 5: Chikou Span Cross

The mathematical operations used to generate these features include subtraction, maximization, and minimization, which help capture the direction and magnitude of the intersections. Additionally, some features were shifted forward to align with the principles of technical analysis, which rely on analyzing historical data and past price behavior to evaluate current price trends

and forecast future market movements. Table 1 provides a summary of the features derived from these signals.

Table 1: Ichimoku-inspired features.

Feature	Description
T-K	$Tenkan_sen - Kijun_sen$
A-B	$Span\ A - Span\ B$
P-T-K	$Close\ Price - \max(Tenkan_sen, Kijun_sen)$
P-A-B	$Close\ Price - \max(Span\ A, Span\ B)$
T-K-A-B	$\min(Tenkan_sen, Kijun_sen) - \max(Span\ A, Span\ B)$
P-CH	$Close\ Price - Chikou\ Span\ (shifted\ +\ 26)$
CH-T-K	$Chikou\ Span - \max(Tenkan_sen, Kijun_sen)\ (shifted\ +\ 26)$
CH-A-B	$Chikou\ Span - \max(Span\ A, Span\ B)\ (shifted\ +\ 26)$

B.5 Dataset

This study employs a dataset of EUR/USD OHLC prices across three hierarchical timeframes—one-hour (H1), four-hour (H4), and daily (D1)—spanning from September 1, 2019, to September 1, 2024. The data were collected via the MetaTrader 5 platform to ensure high reliability and precision. The proposed multi-timeframe voting framework strategically integrates these complementary resolutions to generate robust, real-world-aligned trading signals. The D1 timeframe serves as the primary decision layer. It effectively filters intraday noise and captures persistent trend structures that align with the medium- to long-term positioning of institutional traders, a concept well-documented in Forex market microstructure research. The H4 timeframe acts as an intermediate confirmatory layer, encapsulating half of each major geographic trading session (Asian, European, and American) within a single candle. This resolution reveals significant intraday momentum shifts and liquidity surges that remain obscured on daily charts yet are excessively noisy on lower timeframes. Finally, the H1 timeframe is incorporated as the highest-frequency component. It enables precise entry and exit timing by detecting short-term support and resistance zones, micro-trends, and order-flow imbalances, while preserving an acceptable signal-to-noise ratio. This hierarchical D1→H4→H1 architecture ensures that the final signal is firmly rooted in dominant market cycles, validated by session-driven dynamics, and optimally timed for execution. Furthermore, the selection of EUR/USD—the world’s most liquid currency pair, exhibiting strong correlations with global risk sentiment—enhances the generalizability and practical relevance of the findings. To improve the predictive accuracy of the models, two distinct datasets are constructed to systematically evaluate the contribution of

Ichimoku-inspired features. The first dataset augments standard OHLC price data with the five canonical components of the Ichimoku Kinko Hyo indicator. The second dataset extends this representation by combining OHLC prices with eight engineered features derived from widely adopted Ichimoku trading strategies. This captures relational and directional market dynamics beyond the original indicator formulation, as detailed in Section 2.4. A sliding window of 50 time steps is applied to both datasets to model recent temporal dependencies while preserving the computational efficiency required for real-time forecasting. This specific window length was selected to cover a complete Ichimoku trend cycle, enabling the model to capture both short-term fluctuations and medium-term directional structures. The data are subsequently partitioned into training and testing subsets using an 80%–20% split, with 20% of the training data allocated as a validation set to monitor performance and ensure proper generalization. Ultimately, the integration of hierarchical timeframes, sophisticated feature engineering, and rigorous data partitioning forms a comprehensive foundation for the robust evaluation of the proposed predictive architectures.

B.6 The Proposed Models

This research contributes to the development of a multivariate time series prediction system that leverages deep learning techniques to forecast Forex price movements. The proposed system consists of three main components trained on the H1, H4, and D1 timeframes. A Convolutional Neural Network (CNN) serves as the base architecture, combined with LSTM and GRU networks for enhanced temporal modeling. The primary focus is on utilizing the Ichimoku technical indicator, which provides high-quality features suitable for these deep learning architectures. To clearly distinguish between different configurations, the following naming conventions are adopted: models using only the original Ichimoku indicators are prefixed with *I* (e.g., IC for Ichimoku-CNN, ICL for Ichimoku-CNN-LSTM, and ICG for Ichimoku-CNN-GRU). Models augmented with the newly engineered Ichimoku-inspired features receive an additional *I* prefix to denote “Improved” status (e.g., IIC, IICL, and IICG).

Furthermore, all configurations undergo hyperparameter optimization via the Starfish Optimization Algorithm (SFOA). To denote this, an *S* is appended to the model name (e.g., ICS, ICLS, and ICGS for standard Ichimoku architectures; IICS, IICLS, and IICGS for the improved versions). The structural frameworks of the baseline IC, ICL, and ICG models are illustrated in Figure 6. Each model employs the Rectified Linear Unit (ReLU) activation function in the hidden layers and a linear activation function in the output layer.

All models were trained for 50 epochs using the Mean Squared Error (MSE) loss function and the Adam optimizer with an initial learning rate of 0.001. Given that the input EUR/USD time-series data were normalized prior to training, this learning rate ensured stable and efficient

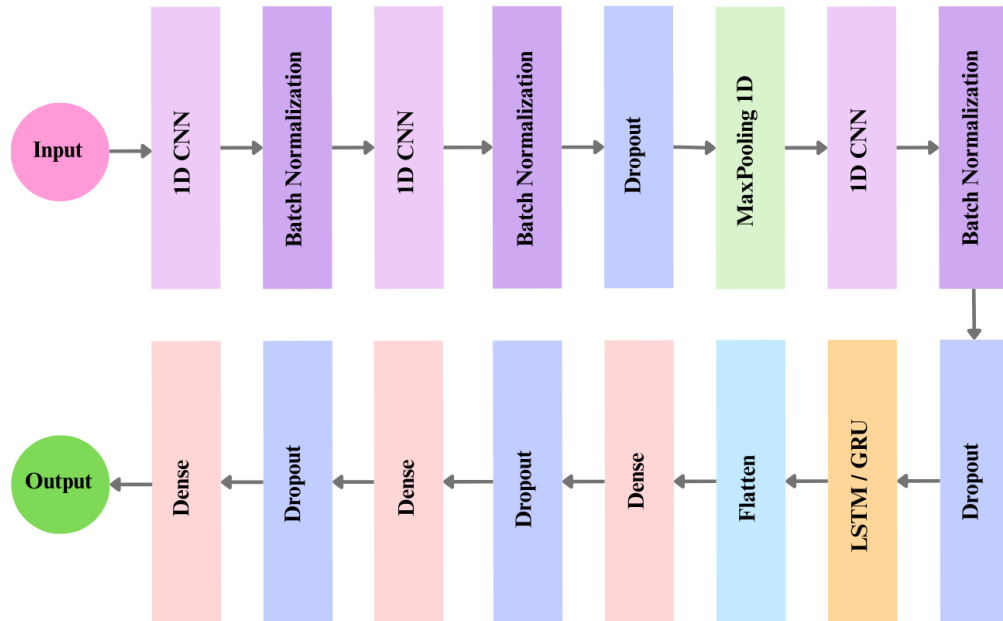


Figure 6: Models' Architecture

convergence. Additionally, the Adam optimizer provides adaptive learning rate adjustments, which stabilizes the training dynamics of deep neural networks. To further enhance convergence and prevent optimization stagnation, a ReduceLROnPlateau learning rate scheduling strategy was applied, dynamically decreasing the learning rate when the validation loss plateaued.

In financial time-series forecasting, overfitting is a common and critical issue, in which a model may achieve strong performance on training data but fail to generalize to unseen samples due to the noisy and non-stationary nature of market data. To mitigate this problem, several regularization techniques were incorporated into the proposed architectures, including Dropout layers, Batch Normalization, and pooling operations. These components collectively improve feature generalization, stabilize training dynamics, and enhance the robustness of the models on out-of-sample data. The system was implemented in Python 3.11.3, utilizing TensorFlow and scikit-learn for deep learning and data preprocessing. All experiments were conducted on a laptop equipped with a 13th Gen Intel® Core™ i7-13620H CPU (2.40 GHz) and 16 GB RAM, running the Windows 11 operating system. To ensure reproducibility and further assess the robustness of the proposed framework, all stochastic components involved in model training and hyperparameter optimization were controlled using fixed random seeds. In particular, each model was trained over 10 independent runs using different fixed random seeds (e.g., 11, 22, 33, 44, 55, 66, 77, 88, 99, and 110) to reduce the influence of random weight initialization and to verify that the reported performance was not associated with a specific initialization.

In addition, the same seed-control strategy was employed in the SFOA-based hyperparameter optimization stage to ensure consistency across the entire experimental procedure.

Table 2 presents the hyperparameter candidate values considered in this study. These parameters determine the model's learning capacity and resistance to overfitting. The SFOA framework systematically optimizes these hyperparameters, ensuring an optimal balance between architectural complexity and generalization performance when training on high-dimensional time series data.

Table 2: Hyperparameter Alternatives

Hyperparameter	Candidate Values
Conv1D Filters	30–150 (step=1)
Conv1D Kernel Size	{3, 5, 7, 9}
Dropout Rate	{0.3, 0.4, 0.5, 0.6}
Neurons (Dense/LSTM/GRU)	30–150 (step=1)

A rule-based aggregation module was developed to harmonize the forecasting outputs obtained from the H1, H4, and D1 timeframes and project them onto a unified daily (24-hour) horizon. Forecasts at different temporal resolutions are treated as complementary trend-following signals that capture short-, medium-, and long-term market dynamics. Within this framework, the H1 and H4 forecasts corresponding to a given 24-hour interval are temporally aligned and consolidated at the daily level together with the direct D1 prediction. Importantly, one-step-ahead forecasts are used exclusively for regression-based performance evaluation and model optimization (MSE, MAE, MAPE, and the SFOA fitness function). For trend extraction, however, a multi-step alignment is applied at H1 and H4 so that the predicted prices corresponding to the daily horizon can be compared, while at D1 the daily trend is obtained by comparing two consecutive one-step-ahead predicted prices. This design reduces trend misclassification: short-horizon predictions may achieve low regression error yet still produce incorrect directional signals. By aligning all timeframes to the daily horizon, the models capture the dominant price movement more reliably. The final daily trend direction is determined by a majority-voting scheme in which the aligned H1, H4, and D1 directional signals participate. This aggregation filters out noisy or conflicting short-term signals and yields a stable estimate of the prevailing daily trend (upward or downward). The overall workflow is summarized in Algorithm 1. Historical price data across the three timeframes are prepared and normalized, hyperparameters are optimized with SFOA, and the models are trained. One-step-ahead predictions are then used to compute regression metrics, while the multi-step-aligned forecasts are converted into directional signals and aggregated by majority voting. The resulting daily trend decisions are evaluated with classification metrics (Accuracy and F1-score).

Algorithm 3 End-to-end multi-timeframe forecasting and decision-making framework using Ichimoku-based features, CNN-based models, and SFOA

```

1: Input: Historical EURUSD price data in {H1, H4, D1}, window length  $L$ ,
2: CNN-based models ICS, ICLS, ICGS, IICS, IICLS, IICGS,
3: optimization algorithm SFOA, number of bootstrap replications  $B$ , Random seed set  $\mathcal{S} = \{s_1, s_2, \dots, s_{10}\}$ .
4: Output: Final classification performance metrics reported as mean  $\pm$  SD, and DM test results reported as DM statistics,  $p$ -values, and Holm–Bonferroni adjusted  $p$ -values.
5: Step 1: Data Preparation
6: for each timeframe  $tf \in \{H1, H4, D1\}$  do
7:   Load historical price data corresponding to  $tf$ .
8:   Perform Ichimoku-based feature engineering.
9:   Normalize the extracted features.
10:  Construct sliding windows of length  $L$ .
11:  Split the dataset into training (80%) and testing (20%) subsets.
12:  Further split the training subset into training (80%) and validation (20%) subsets.
13: end for
14: Step 2: Timeframe-specific Hyperparameter Optimization via SFOA
15: for each timeframe  $tf \in \{H1, H4, D1\}$  do
16:   for each model  $m \in \{ICS, ICLS, ICGS, IICS, IICLS, IICGS\}$  do
17:     for each starfish candidate in SFOA do
18:       Apply candidate hyperparameters to model  $m$  for timeframe  $tf$ .
19:       Train model  $m$  on the training data of timeframe  $tf$ .
20:       Compute validation MAE.
21:       Assign the MAE value as the fitness score.
22:     end for
23:     Select the optimal hyperparameter set for  $(m, tf)$  after SFOA convergence.
24:   end for
25: end for
26: Step 3: Final Training of Optimized Models
27: for each timeframe  $tf \in \{H1, H4, D1\}$  do
28:   for each model  $m$  do
29:     Train model  $m$  on the entire training set of  $tf$  using the optimized hyperparameters.
30:   end for
31: end for

```

```

32: Step 4: One-Step-Ahead Price Forecasting and Statistical Regression Evaluation
33: for each timeframe  $tf \in \{H1, H4, D1\}$  do
34:     for each model  $m$  do
35:         Generate one-step-ahead price forecast  $\hat{p}_{t+1}^{tf}$ .
36:         Compute regression metrics: MSE, MAE, and MAPE.
37:     end for
38: end for
39: Step 5: Multi-Step Forecasting for Trend Alignment
40: for each model  $m$  do
41:     Generate aligned multi-step-ahead price forecasts:
42:     H1:  $\hat{p}_{t+24}^{H1}$  (24 steps ahead)
43:     H4:  $\hat{p}_{t+6}^{H4}$  (6 steps ahead)
44:     D1:  $\hat{p}_{t+1}^{D1}$  (1 step ahead)
45: end for
46: Step 6: Trend Extraction
47: for each timeframe  $tf \in \{H1, H4, D1\}$  do
48:     Determine the trend direction by comparing the aligned
49:     forecast with the current actual price.
50: end for
51: Step 7: Daily Trend Decision via Majority Voting
52: Aggregate the aligned trend predictions from  $\{H1, H4, D1\}$ 
53: using majority voting.
54: Obtain the final daily trend decision.
55: Step 8: Final Classification Evaluation
56: Compare predicted daily trends with ground-truth trends.
57: Compute classification metrics: Accuracy and F1-score.
58: Step 9: Statistical Validation
59: Diebold–Mariano (DM) Test:
60: for each timeframe  $tf \in \{H1, H4, D1\}$  do
61:     for each model pair  $(m_i, m_j)$  do
62:         Compute forecast errors for both models.
63:         Calculate the loss differential:  $d_t = L(e_i) - L(e_j)$ 
64:         Compute the DM statistic and corresponding  $p$ -value.
65:     end for
66: end for

```


- 63: *Holm–Bonferroni Correction:*
- 64: Sort the DM test p -values in ascending order.
- 65: Apply the step-down Holm–Bonferroni procedure to control the family-wise error rate (FWER).
- 66: Compute the Holm-adjusted p -values.
- 67: Determine statistical significance using the adjusted p -values.
- 68: *Moving Block Bootstrap Procedure:*
- 69: **for** $b = 1$ to B **do**
- 70: Generate a bootstrap sample using moving blocks.
- 71: Compute evaluation metrics on the bootstrap sample.
- 72: **end for**
- 73: Calculate the mean and standard deviation of Accuracy and F1-score over bootstrap replications.
- 74: Report the final performance metrics as mean $\pm$ SD to assess model robustness.
- 75: **Output:**
- 76: Report Mean $\pm$ SD results, DM test outcomes, and Holm–Bonferroni adjusted p -values.

Statistical reliability is assessed by the Diebold–Mariano (DM) test with Holm–Bonferroni correction for multiple comparisons, and the stability of classification performance is examined via moving-block bootstrap analysis. Together, these steps provide an end-to-end evaluation pipeline that clearly separates the regression training objective from the final directional decision-making objective. To evaluate the cross-market generalization capability of the proposed framework, the models trained exclusively on the EUR/USD currency pair were applied to eight additional currency pairs without any retraining or fine-tuning. For each unseen currency pair, the same Ichimoku-based feature engineering pipeline, data normalization procedure, and multi-timeframe forecasting and voting mechanism were employed. This protocol enables assessment of the framework’s zero-shot transfer performance across different Forex markets.

Remark 1. The stochastic initialization and search phases of Algorithm 1 are controlled via fixed random seeds, with results reported as the mean and standard deviation over 10 independent runs.

Remark 2. The DM test was conducted at a significance level of $\alpha = 0.05$. A p -value lower than 0.05 indicates a statistically significant difference between the forecasting errors of the compared models.

C Experimental Results

C.1 Evaluation Criteria

The proposed forecasting framework is evaluated using a comprehensive set of regression, classification, and statistical validation metrics. Regression performance is assessed using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), while trend prediction performance is evaluated using Accuracy and F1-score. To further investigate the statistical significance and robustness of the obtained results, the Diebold–Mariano (DM) test and Bootstrap resampling analysis are additionally employed. The DM test determines whether the forecasting accuracy of two competing models differs significantly, whereas Bootstrap analysis evaluates the stability and reliability of the reported performance estimates across repeated samples. To account for the increased probability of false-positive findings resulting from multiple DM comparisons, the Holm–Bonferroni procedure is applied to adjust the resulting p -values while controlling the family-wise error rate (FWER). Together, these evaluation criteria provide a comprehensive assessment of predictive accuracy, classification effectiveness, statistical significance, multiple-comparison control, and model robustness.

MSE measures the average squared difference between predicted ($\hat{y}_i$) and actual values (y_i) as shown in Equation (8):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2. \quad (8)$$

By penalizing larger errors more heavily, MSE is sensitive to outliers and is widely used for regression tasks. It traces its origins to Gauss and Legendre’s foundational work on least squares regression [9].

MAE represents the average magnitude of errors, treating all deviations equally as shown in Equation (9):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|. \quad (9)$$

MAE is more robust to outliers than MSE and provides interpretable results in the same units as the data. It has its roots in robust regression and descriptive statistics [7].

MAPE expresses errors as a percentage of actual values, making it scale-independent as shown in Equation (10) :

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100. \quad (10)$$

MAPE is useful for comparing performance across datasets but can produce inflated errors when actual values are near zero. It gained prominence in time-series forecasting and business modeling [26].

Accuracy is a straightforward metric that measures the proportion of correct predictions including true positives and true negatives, relative to the total predictions made as shown in Equation (11).

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \quad (11)$$

While accuracy indicates overall performance, it can be misleading under class imbalance with a dominant class [30].

F1-score is a performance metric used in classification tasks to evaluate the balance between precision and recall. It is defined as the harmonic mean of precision (P) and recall (R), formulated as Equation (12):

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}. \quad (12)$$

Precision (P) is defined as Equation (13):

$$P = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (13)$$

Recall (R) is defined as Equation (14):

$$R = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}. \quad (14)$$

The F1-score provides a single metric that balances the trade-off between precision and recall, particularly useful in situations with imbalanced datasets [30]. This metric has since been adapted widely in machine learning and classification tasks for evaluating model performance. Together, these metrics provide a well-rounded evaluation of the model's effectiveness in predicting financial time series movements.

Diebold-Mariano (DM) Test is a statistical method used to compare the accuracy of two competing forecast models. Developed by Diebold and Mariano [6], this test assesses whether the difference in forecast performance between two models is statistically significant. It is particularly useful in econometrics and time series analysis when evaluating the performance of different forecasting techniques. Let Y_t be the actual value at time t , and let $\hat{Y}_{1,t}$ and $\hat{Y}_{2,t}$ be the forecasts generated by model 1 and model 2, respectively. The forecast errors for each model are defined as Equations (15) and (16):

$$e_{1,t} = Y_t - \hat{Y}_{1,t}, \quad (15)$$

$$e_{2,t} = Y_t - \hat{Y}_{2,t}. \quad (16)$$

The Diebold–Mariano test focuses on the loss differential series, defined as $d_t = L(e_{1,t}) - L(e_{2,t})$, here $L(\cdot)$ denotes a specific loss function, such as squared error or absolute error. The null hypothesis (H_0) states that both competing models possess equal forecast accuracy (i.e., the expected value of the loss differential is zero), while the alternative hypothesis (H_1) implies that one model statistically outperforms the other. The test statistic is based on the sample mean of the loss differentials, $\bar{d}$, and its variance. A modified t-statistic is typically utilized to account for potential autocorrelation in the forecast errors, especially in finite samples. The DM statistic is calculated as Equation (17):

$$DM = \frac{\bar{d}}{\sqrt{\hat{V}(\bar{d})}}, \quad (17)$$

in which $\bar{d} = \frac{1}{T} \sum_{t=1}^T d_t$ and $\hat{V}(\bar{d})$ is a consistent estimate of the asymptotic variance of $\bar{d}$. Under the null hypothesis, this statistic asymptotically follows a standard normal distribution [6].

Moving Block Bootstrap (MBB) is a resampling technique designed for dependent data, particularly time-series observations, where the assumption of independent and identically distributed samples is not appropriate [20]. Unlike the conventional bootstrap, which independently resamples individual observations with replacement, MBB resamples consecutive blocks of observations to preserve the temporal dependence structure of the original series.

Let $X = x_1, x_2, \dots, x_n$ denote an ordered time series and let L represent the block length. A collection of overlapping blocks is first constructed from consecutive observations as

$$B_j = \{x_j, x_{j+1}, \dots, x_{j+L-1}\}, \quad j = 1, 2, \dots, n - L + 1.$$

Each block therefore contains L consecutive observations, and adjacent blocks overlap. During each bootstrap replication, blocks are randomly selected with replacement from the collection of available overlapping blocks and concatenated sequentially to construct a bootstrap sample of the desired length. The statistic of interest is then calculated from the resulting bootstrap sample.

Repeating this procedure generates an empirical bootstrap distribution of the statistic, which can be used to assess its sampling variability and statistical uncertainty. By resampling contiguous blocks rather than individual observations, MBB retains the local dependence structure of the original time series and provides a suitable nonparametric resampling framework for dependent observations [20].

Holm–Bonferroni Correction When multiple statistical hypotheses are tested simultaneously, the probability of obtaining at least one false-positive result increases with the number

of comparisons. To control the family-wise error rate (FWER), the Holm–Bonferroni procedure provides a sequential (step-down) multiple-comparison correction that is less conservative than the classical Bonferroni method while still guaranteeing strong FWER control [14].

Let m denote the total number of statistical tests and let $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(m)}$ be the ordered raw p -values. The Holm procedure sequentially compares each ordered p -value with its adjusted significance threshold:

$$p_{(i)} \leq \frac{\alpha}{m - i + 1}, \quad i = 1, \dots, m, \quad (18)$$

where α is the chosen significance level. Equivalently, the adjusted p -values are computed as

$$p_{(i)}^{\text{Holm}} = \min \left(1, \max_{j \leq i} \{ (m - j + 1) p_{(j)} \} \right). \quad (19)$$

A hypothesis is declared statistically significant if its corresponding adjusted p -value falls below the pre-specified significance level α .

In the present study, this procedure is applied to the complete set of Diebold–Mariano test p -values obtained across all currency pairs, loss functions (MSE, MAE, MAPE), and forecasting horizons (H1, H4, D1), thereby ensuring rigorous control of the multiplicity problem.

C.2 Analysis of Model Performance Based on Metric Values

In this subsection, we evaluate the performance of all previously introduced models across three distinct timeframes (H1, H4, and D1), with the primary objective of predicting trends in the D1 timeframe. This evaluation demonstrates the effectiveness of the proposed framework for Forex trend forecasting while also assessing its robustness under varying market conditions. All experiments were repeated independently 10 times, and all reported results are presented as the mean $\pm$ standard deviation (Mean $\pm$ SD) over these runs.

The predictions generated by these models are subsequently processed by a decision-making module that aggregates the multi-timeframe outputs using a voting-based strategy to produce the final trading signal. The models are trained using historical EUR/USD data covering the period from 2019 to 2024. Before extending the evaluation to additional currency pairs, we first analyze the performance of the six proposed models on the EUR/USD dataset used for training and initial testing. This analysis establishes the benchmark for subsequent experiments. The evaluated models (ICS, ICLS, ICGS, IICS, IICLS, and IICGS) represent different combinations of engineered Ichimoku-based features and optimization strategies. As summarized in

Table 3, the proposed IICGS model consistently achieves the best overall performance across all evaluated timeframes. At the H1 timeframe, it attains the lowest regression errors, with an MSE of $(1.5 \pm 0.1) \times 10^{-5}$, an MAE of 0.0035 ± 0.0001 , and a MAPE of 0.0041 ± 0.0002 , averaged over 10 independent runs. Similar behavior is observed at the H4 timeframe, with IICGS again reporting the lowest errors, achieving an MSE of $(1.5 \pm 0.1) \times 10^{-5}$, an MAE of 0.0029 ± 0.0001 , and a MAPE of 0.0040 ± 0.0002 . At the D1 timeframe, the proposed model maintains its superiority, yielding an MSE of $(1.7 \pm 0.1) \times 10^{-5}$, an MAE of 0.0028 ± 0.0001 , and a MAPE of 0.0033 ± 0.0002 . At the framework level, IICGS also achieves the highest classification performance, reaching an Accuracy of $78.97 \pm 0.50\%$ and an F1-score of 0.66 ± 0.01 . Although other models, particularly IICLS, achieve competitive regression performance in some individual timeframes, their overall framework-level Accuracy and F1-score remain consistently lower than those of IICGS. This observation indicates that superior regression performance at a single temporal resolution does not necessarily translate into better final trading decisions after multi-timeframe voting. Overall, these findings demonstrate that the proposed IICGS architecture provides an effective balance between regression accuracy, cross-timeframe consistency, and classification robustness, making it particularly well suited for multi-timeframe EUR/USD trend prediction. To ensure transparency and reproducibility, representative case studies have been made publicly available.

To rigorously examine the robustness of the proposed approach and assess its applicability across diverse market conditions, the experimental evaluation is extended to eight additional currency pairs: AUD/USD, EUR/SGD, GBP/NZD, NZD/USD, USD/CHF, USD/CNH, USD/SEK, and USD/SGD. These pairs were selected to construct a heterogeneous zero-shot validation set spanning different regional markets, currency compositions, and levels of international FX activity, thereby enabling a more comprehensive assessment of model stability and generalization under varying market structures and data distributions. The set includes highly traded pairs such as AUD/USD and USD/CHF, as well as regionally important markets represented by USD/CNH and USD/SGD. Consistent with the BIS Triennial Central Bank Surveys, AUD/USD, USD/CHF, USD/CNY, USD/SGD, and USD/NZD accounted for substantial shares of global FX turnover in 2019 and 2022 (Bank for International Settlements [2, 3]). The inclusion of EUR/SGD and GBP/NZD further introduces non-USD cross-currency relationships, while USD/SEK extends the evaluation to the Scandinavian market. Collectively, these currency pairs provide geographic and market-structure diversity beyond EUR/USD, while the availability of sufficiently long and consistent historical data over 2019–2024 ensures a reliable out-of-sample evaluation. For all eight additional currency pairs, the testing period is consistently defined from September 1, 2019, to September 1, 2024. Accordingly, all subsequent references to the testing or evaluation of these currency pairs throughout the manuscript refer to this same period.

The performance of the models is evaluated using regression metrics, including MSE, MAE, and MAPE, to quantify the accuracy of forecasting future closing prices. Given that the test data are scaled, all reported results reflect this normalization. These metrics were analyzed across multiple currency pairs for H1, H4, and D1 timeframes, with detailed results summarized in Tables 4, 5, and 6. Visual representations of the comparative results are provided in Figures 7 through 12 in the discussion section.

The comparative evaluation of the proposed models, as shown in Table 4, considers the H1, H4, and D1 timeframes and uses the MSE criterion for evaluation. From the empirical data, a clear performance hierarchy can be established among the models, following the order of increasing error: IICGS < IICLS < IICS < ICGS < ICLS < ICS. At the H1 timeframe, the IICGS model consistently achieves the lowest MSE values across most currency pairs, such as USD/SGD $((0.4 \pm 0.0) \times 10^{-5})$, GBP/NZD $((0.8 \pm 0.0) \times 10^{-5})$, and $((0.9 \pm 0.0) \times 10^{-5})$, highlighting its strong capability in modeling short-term price fluctuations. The IICLS and IICS models also demonstrate competitive performance but remain inferior to IICGS, while the baseline models (ICS and ICLS) exhibit noticeably higher error levels. A similar trend is evident in the H4 timeframe, in which IICGS again outperforms all competing approaches for the majority of currency pairs, including USD/SGD $((1.5 \pm 0.1) \times 10^{-5})$, GBP/NZD $((2.8 \pm 0.1) \times 10^{-5})$, and USD/CHF $((2.9 \pm 0.1) \times 10^{-5})$. The gradual performance improvement from ICS toward IICGS indicates the positive impact of progressively enriching the feature space and model architecture. In the D1 timeframe, IICGS maintains its dominance by achieving the lowest MSE for most currency pairs, such as USD/SGD $((2.0 \pm 0.1) \times 10^{-5})$, EUR/SGD $((4.0 \pm 0.2) \times 10^{-5})$, and GBP/NZD $((4.8 \pm 0.2) \times 10^{-5})$, reflecting strong generalization capability under longer-term market dynamics. Overall, the results confirm that integrating engineered Ichimoku-inspired features within the forecasting framework substantially enhances regression accuracy across all time horizons.

Similarly, Table 5 evaluates the forecasting performance using the Mean Absolute Error (MAE) metric. A highly consistent performance hierarchy is observed across all assets, following the error minimizing pattern IICGS < IICLS < IICS < ICGS < ICLS < ICS. In the H1 timeframe, the IICGS model systematically achieves the lowest MAE values across all currency pairs, such as USD/SGD (0.0014 ± 0.0001) , GBP/NZD (0.0023 ± 0.0001) , and EUR/SGD (0.0023 ± 0.0001) , highlighting its superior ability to capture short-term price movements. This performance ordering is preserved in the H4 timeframe, in which IICGS again dominates for all assets, including USD/SGD (0.0028 ± 0.0001) , EUR/SGD (0.0032 ± 0.0002) , and GBP/NZD (0.0042 ± 0.0002) . The gradual decrease in MAE from ICS toward IICGS reflects the cumulative impact of both SFOA-based optimization and engineered Ichimoku-inspired features. In the D1 timeframe, IICGS continues to deliver the most accurate forecasts, achieving the lowest MAE values for pairs such as USD/SGD (0.0040 ± 0.0002) , EUR/SGD (0.0041 ± 0.0002) , and GBP/NZD (0.0049 ± 0.0002) . This confirms the robustness of the proposed architecture un-

der longer-term forecasting horizons, in which error propagation and regime changes typically challenge model stability.

Table 6 reports the Mean Absolute Percentage Error (MAPE) of the proposed models. The same error minimizing hierarchy (IICGS < IICLS < IICS < ICGS < ICLS < ICS) is observed, confirming the robustness of the proposed modeling strategy. In the H1 timeframe, the IICGS model systematically achieves the lowest MAPE values for all currency pairs, including USD/SGD (0.0015 ± 0.0001), GBP/NZD (0.0025 ± 0.0001), and EUR/SGD (0.0025 ± 0.0001). A similar trend is preserved in the H4 timeframe, in which IICGS consistently outperforms competing models across all assets, noting particularly low MAPE values for USD/SGD (0.0030 ± 0.0001), EUR/SGD (0.0042 ± 0.0002), and GBP/NZD (0.0046 ± 0.0002). At the D1 timeframe, IICGS continues to demonstrate superior relative accuracy, achieving the lowest MAPE values for USD/SGD (0.0044 ± 0.0002), EUR/SGD (0.0048 ± 0.0002), and USD/CHF (0.0059 ± 0.0003). While optimization alone contributes to moderate improvements, the integration of engineered Ichimoku-inspired features plays the dominant role in reducing relative forecasting errors.

Table 7 presents the trend prediction performance of the evaluated models, measured using Accuracy and F1-score metrics derived based on the decision-making voting mechanism. Overall, the results indicate that models incorporating the improved Ichimoku-based feature construction generally outperform their original counterparts. In terms of Accuracy, the IICGS model consistently achieves the highest values for most currency pairs, including USD/SGD ($82.7 \pm 0.5\%$), USD/CHF ($79.3 \pm 0.8\%$), and AUD/USD ($78.7 \pm 0.7\%$). From the F1-score perspective, which balances precision and recall, IICGS again demonstrates competitive performance, attaining the highest scores for AUD/USD (0.63 ± 0.02) and GBP/NZD (0.65 ± 0.01). For USD/CNH, IICS attains an F1-score of 0.66 ± 0.01 and IICGS attains 0.65 ± 0.02 , indicating a highly comparable and competitive level of classification balance among the improved models.

As evidenced by the results in Table 8, the statistical robustness of the proposed IICGS framework is validated through rigorous comparative testing across diverse market conditions. The Diebold–Mariano (DM) test was evaluated at significance levels $\alpha = 0.05$ and $\alpha = 0.01$. Accordingly, DM statistics exceeding the critical asymptotic normal thresholds ($|DM| > 1.96$ for $\alpha = 0.05$ and $|DM| > 2.58$ for $\alpha = 0.01$) indicate statistically significant differences in predictive performance. To rigorously control the family-wise error rate arising from multiple pairwise comparisons across currency pairs, metrics, and forecasting horizons, the sequential Holm–Bonferroni step-down adjustment procedure was applied to all computed p -values. As detailed in Table 8, even after controlling for multiple testing, all adjusted p -values satisfy the strict criteria ($p < 0.05$ and $p < 0.01$), confirming that the reductions in MSE, MAE, and MAPE achieved by the proposed IICGS framework over the baseline ICS model are statistically significant rather than the result of random variation. These statistical tests were performed on

the best-performing model obtained from ten independent training runs (each run consisting of full end-to-end training and evaluation under identical hyperparameter settings). This predictive superiority is further supported by the Bootstrap analysis, where the stable mean accuracy values and relatively small standard deviations demonstrate the framework's robustness, consistent generalization capability, and resistance to overfitting across different historical market conditions. These findings collectively confirm that the proposed multi-timeframe voting framework captures meaningful market dynamics beyond short-term stochastic fluctuations.

C.3 Analysis of Model Performance Based on Prediction Charts

A general analysis of the comparative performance across multiple models, as illustrated in Figures 7–12 reveals nuanced insights into predictive accuracy and error dynamics in financial forecasting for various currency pairs. An examination of the comparative charts between actual and predicted prices in ICS models (Figure 7) indicates that, across all three timeframes (H1, H4, and D1), the overall price direction is generally well captured by the proposed model. In the one-hour timeframe, the discrepancy between the predicted values and the actual prices is relatively small, reflecting the model's stronger ability to track short-term price movements. As the timeframe increases to four hours, this gap becomes more pronounced, and a temporal lag in the model's response to price changes can be observed in certain intervals. In the daily timeframe, the spread of the predicted price range further increases; although the general trend direction is largely preserved, occasional inversions between predicted and actual trends are evident. In addition, for the USD/SEK pair, there exists a specific period during which the model is unable to adequately reflect the prevailing market behavior across all three timeframes. During this interval, the predicted prices remain largely range-bound, while the actual market exhibits a clear directional trend. This behavior suggests a reduced sensitivity of the model to gradual trend development under low-volatility conditions rather than an inability to capture abrupt price movements.

Overall, these observations highlight that shorter timeframes yield closer alignment between predicted and actual prices, whereas higher timeframes tend to emphasize broader trend characteristics at the expense of local accuracy. For the ICLS model (Figure 8), the gap between the predicted prices and the actual market values is generally reduced compared to the previous model, and the forecasts exhibit a higher degree of directional consistency with the true price movement. This indicates an improved capability of the model to follow the overall market trend. However, despite this enhanced alignment, the model demonstrates a limited ability to accurately represent periods of heightened price variability, often smoothing out pronounced fluctuations observed in the actual price series. Across different temporal resolutions, a systematic bias pattern can be observed. In the one-hour timeframe, the predicted prices are

Table 3: Performance comparison of hybrid models on the benchmark EUR/USD pair (Mean $\pm$ SD over 10 runs).

Model	Framework									
	H1			H4			D1			
	MSE ($\times 10^{-5}$)	MAE	MAPE	MSE ($\times 10^{-5}$)	MAE	MAPE	MSE ($\times 10^{-5}$)	MAE	MAPE	Acc. (%)
ICS	7.7 ± 0.3	.0084 $\pm$.0004	.0096 $\pm$.0005	13.4 ± 0.6	.0103 $\pm$.0005	.0119 $\pm$.0007	9.9 ± 0.4	.0086 $\pm$.0004	.0098 $\pm$.0005	54.73 ± 1.2
ICLS	2.7 ± 0.1	.0048 $\pm$.0002	.0050 $\pm$.0002	10.8 ± 0.5	.0087 $\pm$.0004	.0100 $\pm$.0005	7.4 ± 0.3	.0072 $\pm$.0003	.0083 $\pm$.0004	59.79 ± 1.1
ICGS	2.5 ± 0.1	.0043 $\pm$.0002	.0045 $\pm$.0002	4.3 ± 0.2	.0060 $\pm$.0003	.0076 $\pm$.0004	5.7 ± 0.2	.0063 $\pm$.0003	.0071 $\pm$.0003	61.70 ± 1.0
IICS	2.0 ± 0.1	.0043 $\pm$.0002	.0049 $\pm$.0003	4.0 ± 0.2	.0055 $\pm$.0002	.0069 $\pm$.0004	2.2 ± 0.1	.0038 $\pm$.0002	.0043 $\pm$.0002	75.40 ± 0.8
IICLS	1.9 ± 0.1	.0042 $\pm$.0002	.0049 $\pm$.0003	3.6 ± 0.1	.0052 $\pm$.0002	.0061 $\pm$.0003	1.9 ± 0.1	.0034 $\pm$.0001	.0040 $\pm$.0002	77.16 ± 0.7
IICGS	1.5 ± 0.1	.0035 $\pm$.0001	.0041 $\pm$.0002	1.5 ± 0.1	.0029 $\pm$.0001	.0040 $\pm$.0002	1.7 ± 0.1	.0028 $\pm$.0001	.0033 $\pm$.0002	78.97 ± 0.5

Table 4: Evaluation of models using MSE ($\times 10^{-5}$), (Mean $\pm$ SD over 10 independent runs).

Ticker	H1						H4						D1					
	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS
AUD/USD	$18.9 \pm .9$	$12.6 \pm .6$	$7.4 \pm .4$	$5.5 \pm .3$	$3.6 \pm .2$	$2.2 \pm .1$	33.6 ± 1.6	27.3 ± 1.3	$14.7 \pm .7$	$9.7 \pm .5$	$7.5 \pm .4$	$6.1 \pm .3$	37.8 ± 1.8	29.4 ± 1.4	$15.8 \pm .8$	$10.3 \pm .5$	$7.2 \pm .3$	$5.0 \pm .2$
EUR/SGD	$6.5 \pm .3$	$5.7 \pm .3$	$3.4 \pm .2$	$2.3 \pm .1$	$1.5 \pm .1$	0.9 ± 0	$8.2 \pm .4$	$7.2 \pm .3$	$5.5 \pm .3$	$4.3 \pm .2$	$3.8 \pm .2$	$3.3 \pm .2$	$13.7 \pm .7$	$9.7 \pm .5$	$7.5 \pm .4$	$6.3 \pm .3$	$5.1 \pm .2$	$4.0 \pm .2$
GBP/NZD	$6.2 \pm .3$	$5.0 \pm .2$	$3.3 \pm .2$	$2.3 \pm .1$	$1.5 \pm .1$	0.8 ± 0	21.0 ± 1.0	$11.6 \pm .6$	$8.7 \pm .4$	$7.0 \pm .3$	$4.3 \pm .2$	$2.8 \pm .1$	$18.9 \pm .9$	$12.6 \pm .6$	$8.8 \pm .4$	$7.2 \pm .3$	$5.7 \pm .3$	$4.8 \pm .2$
NZD/USD	20.0 ± 1.0	$13.7 \pm .7$	$7.6 \pm .4$	$5.1 \pm .2$	$3.3 \pm .2$	$2.2 \pm .1$	36.8 ± 1.8	30.5 ± 1.5	$16.8 \pm .8$	$11.6 \pm .6$	$8.6 \pm .4$	$6.9 \pm .3$	31.5 ± 1.5	25.2 ± 1.2	$14.7 \pm .7$	$10.2 \pm .5$	$6.8 \pm .3$	$4.8 \pm .2$
USD/CHF	$6.3 \pm .3$	$5.1 \pm .2$	$3.9 \pm .2$	$2.4 \pm .1$	$1.8 \pm .1$	$1.4 \pm .1$	23.1 ± 1.1	$14.7 \pm .7$	$10.1 \pm .5$	$7.1 \pm .3$	$5.1 \pm .2$	$2.9 \pm .1$	$19.9 \pm .9$	$12.6 \pm .6$	$9.0 \pm .4$	$6.4 \pm .3$	$6.1 \pm .3$	$5.1 \pm .2$
USD/SEK	75.6 ± 3.6	60.9 ± 2.9	32.6 ± 1.6	$18.9 \pm .9$	$10.1 \pm .5$	$3.6 \pm .2$	126.0 ± 6.0	102.9 ± 4.9	44.1 ± 2.1	26.3 ± 1.3	$14.7 \pm .7$	$9.6 \pm .5$	100.8 ± 4.8	77.7 ± 3.7	41.0 ± 2.0	22.1 ± 1.1	$11.6 \pm .6$	$5.1 \pm .2$
USD/SGD	$6.1 \pm .3$	$4.5 \pm .2$	$2.7 \pm .1$	$1.7 \pm .1$	$0.9 \pm .0$	0.4 ± 0	22.1 ± 1.1	$12.6 \pm .6$	$9.3 \pm .4$	$6.5 \pm .3$	$4.0 \pm .2$	$1.5 \pm .1$	$12.6 \pm .6$	$8.5 \pm .4$	$6.5 \pm .3$	$4.9 \pm .2$	$3.3 \pm .2$	$2.0 \pm .1$

Table 5: Evaluation of models using MAE (Mean $\pm$ SD over 10 independent runs).

Ticker	H1						H4						D1					
	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS
AUD/USD	0.149 ± 0.007	$0.122 \pm .006$	$0.078 \pm .004$	$0.064 \pm .003$	$0.051 \pm .002$	$0.038 \pm .002$	0.0205 ± 0.010	$0.166 \pm .008$	$0.111 \pm .005$	$0.088 \pm .004$	$0.072 \pm .003$	$0.059 \pm .003$	$0.190 \pm .009$	$0.119 \pm .007$	$0.113 \pm .005$	$0.090 \pm .004$	$0.077 \pm .004$	$0.063 \pm .003$
EUR/SGD	$0.071 \pm .0003$	$0.039 \pm .0003$	$0.014 \pm .0002$	$0.036 \pm .0002$	$0.029 \pm .0001$	$0.023 \pm .0001$	$0.086 \pm .0004$	$0.070 \pm .003$	$0.054 \pm .003$	$0.042 \pm .002$	$0.030 \pm .002$	$0.032 \pm .002$	$0.096 \pm .005$	$0.076 \pm .004$	$0.064 \pm .003$	$0.053 \pm .003$	$0.045 \pm .002$	$0.041 \pm .002$
GBP/NZD	$0.078 \pm .0004$	$0.066 \pm .0003$	$0.050 \pm .0002$	$0.041 \pm .0002$	$0.033 \pm .0002$	$0.023 \pm .0001$	$0.122 \pm .0006$	$0.099 \pm .005$	$0.068 \pm .003$	$0.056 \pm .003$	$0.048 \pm .002$	$0.042 \pm .002$	$0.113 \pm .005$	$0.063 \pm .004$	$0.078 \pm .004$	$0.065 \pm .003$	$0.057 \pm .003$	$0.049 \pm .002$
NZD/USD	$0.085 \pm .0004$	$0.072 \pm .0003$	$0.060 \pm .0003$	$0.048 \pm .0002$	$0.041 \pm .0002$	$0.035 \pm .0002$	$0.176 \pm .0008$	$0.146 \pm .007$	$0.100 \pm .005$	$0.082 \pm .004$	$0.072 \pm .003$	$0.064 \pm .003$	$0.156 \pm .007$	$0.128 \pm .006$	$0.096 \pm .005$	$0.086 \pm .004$	$0.070 \pm .003$	$0.063 \pm .003$
USD/CHF	$0.076 \pm .0004$	$0.064 \pm .0003$	$0.051 \pm .0002$	$0.040 \pm .0002$	$0.033 \pm .0002$	$0.027 \pm .0001$	$0.130 \pm .0006$	$0.101 \pm .005$	$0.071 \pm .003$	$0.059 \pm .003$	$0.051 \pm .002$	$0.045 \pm .002$	$0.121 \pm .0006$	$0.097 \pm .005$	$0.078 \pm .004$	$0.065 \pm .003$	$0.058 \pm .003$	$0.054 \pm .002$
USD/SEK	$0.218 \pm .0010$	$0.181 \pm .0009$	$0.132 \pm .0006$	$0.08 \pm .0005$	$0.035 \pm .0004$	$0.067 \pm .0003$	$0.205 \pm .0014$	$0.258 \pm .0012$	$0.177 \pm .008$	$0.141 \pm .007$	$0.113 \pm .005$	$0.093 \pm .004$	$0.267 \pm .0013$	$0.222 \pm .0011$	$0.184 \pm .009$	$0.146 \pm .007$	$0.111 \pm .005$	$0.063 \pm .003$
USD/SGD	$0.077 \pm .0004$	$0.065 \pm .0003$	$0.045 \pm .0002$	$0.035 \pm .0002$	$0.025 \pm .0001$	$0.014 \pm .0001$	$0.112 \pm .0007$	$0.106 \pm .005$	$0.078 \pm .004$	$0.061 \pm .003$	$0.043 \pm .002$	$0.028 \pm .0001$	$0.127 \pm .0006$	$0.103 \pm .005$	$0.083 \pm .004$	$0.064 \pm .003$	$0.050 \pm .002$	$0.040 \pm .002$

Table 6: Evaluation of models using MAPE (Mean $\pm$ SD over 10 independent runs).

Ticker	H1						H4						D1					
	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS
AUD/USD	$0.188 \pm .0009$	$0.149 \pm .0007$	$0.096 \pm .0005$	$0.080 \pm .0004$	$0.051 \pm .0003$	$0.043 \pm .0002$	$0.0229 \pm .0011$	$0.195 \pm .0009$	$0.139 \pm .0006$	$0.106 \pm .0005$	$0.090 \pm .0004$	$0.075 \pm .0004$	$0.214 \pm .0010$	$0.177 \pm .0008$	$0.128 \pm .0006$	$0.106 \pm .0005$	$0.090 \pm .0004$	$0.078 \pm .0003$
EUR/SGD	$0.086 \pm .0004$	$0.072 \pm .0003$	$0.054 \pm .0003$	$0.042 \pm .0002$	$0.033 \pm .0002$	$0.025 \pm .0001$	$0.101 \pm .0005$	$0.085 \pm .0004$	$0.067 \pm .0003$	$0.056 \pm .0003$	$0.048 \pm .0002$	$0.042 \pm .0002$	$0.113 \pm .0005$	$0.095 \pm .0004$	$0.080 \pm .0004$	$0.067 \pm .0003$	$0.056 \pm .0003$	$0.048 \pm .0002$
GBP/NZD	$0.091 \pm .0004$	$0.075 \pm .0004$	$0.059 \pm .0003$	$0.048 \pm .0002$	$0.037 \pm .0002$	$0.025 \pm .0001$	$0.146 \pm .0007$	$0.117 \pm .0006$	$0.088 \pm .0005$	$0.070 \pm .0004$	$0.063 \pm .0003$	$0.046 \pm .0002$	$0.138 \pm .0007$	$0.111 \pm .0005$	$0.092 \pm .0004$	$0.079 \pm .0004$	$0.071 \pm .0003$	$0.073 \pm .0003$
NZD/USD	$0.156 \pm .0007$	$0.129 \pm .0006$	$0.087 \pm .0005$	$0.080 \pm .0004$	$0.062 \pm .0003$	$0.043 \pm .0002$	$0.217 \pm .0010$	$0.188 \pm .0007$	$0.132 \pm .0006$	$0.106 \pm .0005$	$0.080 \pm .0004$	$0.081 \pm .0004$	$0.195 \pm .0009$	$0.160 \pm .0008$	$0.123 \pm .0006$	$0.100 \pm .0005$	$0.087 \pm .0004$	$0.073 \pm .0003$
USD/CHF	$0.096 \pm .0005$	$0.082 \pm .0004$	$0.064 \pm .0003$	$0.052 \pm .0002$	$0.040 \pm .0002$	$0.029 \pm .0001$	$0.143 \pm .0007$	$0.117 \pm .0006$	$0.088 \pm .0004$	$0.072 \pm .0003$	$0.065 \pm .0003$	$0.046 \pm .0002$	$0.134 \pm .0006$	$0.111 \pm .0005$	$0.083 \pm .0004$	$0.079 \pm .0004$	$0.068 \pm .0003$	$0.053 \pm .0003$
USD/CNH	$0.109 \pm .0005$	$0.091 \pm .0004$	$0.075 \pm .0004$	$0.059 \pm .0003$	$0.043 \pm .0002$	$0.026 \pm .0001$	$0.176 \pm .0008$	$0.139 \pm .0007$	$0.100 \pm .0006$	$0.090 \pm .0004$	$0.070 \pm .0003$	$0.048 \pm .0002$	$0.156 \pm .0007$	$0.127 \pm .0006$	$0.101 \pm .0005$	$0.085 \pm .0004$	$0.075 \pm .0004$	$0.065 \pm .0003$
USD/SEK	$0.233 \pm .0014$	$0.258 \pm .0012$	$0.182 \pm .0009$	$0.143 \pm .0007$	$0.106 \pm .0005$	$0.066 \pm .0003$	$0.353 \pm .0017$	$0.316 \pm .0015$	$0.222 \pm .0011$	$0.183 \pm .0009$	$0.143 \pm .0007$	$0.090 \pm .0004$	$0.313 \pm .0015$	$0.274 \pm .0013$	$0.198 \pm .0009$	$0.159 \pm .0008$	$0.112 \pm .0005$	$0.073 \pm .0004$
USD/SGD	$0.093 \pm .0004$	$0.078 \pm .0004$	$0.054 \pm .0003$	$0.041 \pm .0002$	$0.027 \pm .0001$	$0.015 \pm .0001$	$0.153 \pm .0007$	$0.119 \pm .0006$	$0.080 \pm .0004$	$0.067 \pm .0003$	$0.051 \pm .0002$	$0.030 \pm .0001$	$0.159 \pm .0008$	$0.130 \pm .0006$	$0.096 \pm .0005$	$0.070 \pm .0004$	$0.057 \pm .0003$	$0.044 \pm .0002$

Table 7: Evaluation results with Accuracy (%) and F1-score (Mean $\pm$ SD over 10 independent runs).

Ticker	Accuracy (%)						F1-score					
	ICS	ICLS	ICGS	IICS	IICLS	IICGS	ICS	ICLS	ICGS	IICS	IICLS	IICGS
AUD/USD	74.4 $\pm$ 1.1	74.1 $\pm$ 0.9	78.6 $\pm$ 0.8	76.0 $\pm$ 1.2	74.1 $\pm$ 1.0	78.7 $\pm$ 0.7	.59 $\pm$.02	.58 $\pm$.02	.60 $\pm$.01	.58 $\pm$.02	.55 $\pm$.03	.63 $\pm$.02
EUR/SGD	76.1 $\pm$ 0.8	74.1 $\pm$ 1.2	75.3 $\pm$ 0.9	76.1 $\pm$ 1.0	77.8 $\pm$ 0.7	77.9 $\pm$ 0.8	.61 $\pm$.02	.49 $\pm$.04	.59 $\pm$.02	.53 $\pm$.03	.57 $\pm$.02	.59 $\pm$.02
GBP/NZD	69.9 $\pm$ 1.3	70.3 $\pm$ 1.0	72.2 $\pm$ 0.9	71.8 $\pm$ 1.1	72.2 $\pm$ 0.8	74.3 $\pm$ 0.8	.59 $\pm$.02	.56 $\pm$.03	.56 $\pm$.02	.60 $\pm$.02	.60 $\pm$.02	.65 $\pm$.01
NZD/USD	74.7 $\pm$ 0.9	73.6 $\pm$ 1.1	75.8 $\pm$ 1.0	75.6 $\pm$ 0.8	74.1 $\pm$ 1.2	77.5 $\pm$ 0.6	.57 $\pm$.02	.58 $\pm$.02	.60 $\pm$.01	.58 $\pm$.02	.54 $\pm$.03	.61 $\pm$.02
USD/CHF	75.6 $\pm$ 1.0	76.5 $\pm$ 0.8	76.8 $\pm$ 0.7	77.4 $\pm$ 0.9	75.8 $\pm$ 1.1	79.3 $\pm$ 0.8	.53 $\pm$.03	.57 $\pm$.02	.57 $\pm$.02	.58 $\pm$.02	.54 $\pm$.03	.61 $\pm$.02
USD/CNH	67.4 $\pm$ 1.4	67.5 $\pm$ 1.2	66.6 $\pm$ 1.5	69.4 $\pm$ 1.0	67.5 $\pm$ 1.3	69.6 $\pm$ 0.9	.62 $\pm$.02	.63 $\pm$.02	.62 $\pm$.02	.66 $\pm$.01	.61 $\pm$.03	.65 $\pm$.02
USD/SEK	62.1 $\pm$ 1.5	61.5 $\pm$ 1.6	64.9 $\pm$ 1.2	63.8 $\pm$ 1.4	61.8 $\pm$ 1.5	65.7 $\pm$ 1.1	.58 $\pm$.03	.55 $\pm$.03	.62 $\pm$.02	.61 $\pm$.02	.57 $\pm$.03	.62 $\pm$.02
USD/SGD	80.5 $\pm$ 0.7	80.0 $\pm$ 0.9	80.8 $\pm$ 0.8	81.6 $\pm$ 0.6	80.2 $\pm$ 0.8	82.7 $\pm$ 0.5	.49 $\pm$.03	.54 $\pm$.02	.55 $\pm$.02	.55 $\pm$.02	.51 $\pm$.03	.58 $\pm$.01

Table 8: Statistical significance of the IICGS framework across H1, H4, and D1 horizons via DM statistics, two-sided p -values, Holm–Bonferroni adjusted p -values, and bootstrap analysis of overall majority-voted daily trend accuracy.

Ticker	MSE – DM Statistic			MAE – DM Statistic			MAPE – DM Statistic			Bootstrap Accuracy (%)
	H1	H4	D1	H1	H4	D1	H1	H4	D1	
AUD/USD	3.98 $p = 6.892 \times 10^{-5}$	4.82 $p = 1.436 \times 10^{-6}$	3.12 $p = 1.809 \times 10^{-3}$	3.65 $p = 2.622 \times 10^{-4}$	4.41 $p = 1.034 \times 10^{-5}$	2.98 $p = 2.882 \times 10^{-3}$	3.55 $p = 3.652 \times 10^{-4}$	4.30 $p = 1.708 \times 10^{-5}$	2.88 $p = 3.977 \times 10^{-3}$	79.32 $\pm$ 0.47
	3.88 $p = 3.239 \times 10^{-3}$	4.25 $p = 9.475 \times 10^{-5}$	3.22 $p = 2.532 \times 10^{-2}$	3.52 $p = 8.654 \times 10^{-3}$	3.96 $p = 5.995 \times 10^{-4}$	2.95 $p = 2.790 \times 10^{-2}$	3.41 $p = 1.117 \times 10^{-2}$	3.82 $p = 9.565 \times 10^{-4}$	2.84 $p = 2.790 \times 10^{-2}$	
EUR/SGD	3.76 $p = 1.045 \times 10^{-4}$	4.54 $p = 2.138 \times 10^{-5}$	3.05 $p = 1.282 \times 10^{-3}$	3.48 $p = 4.315 \times 10^{-4}$	4.18 $p = 7.495 \times 10^{-5}$	2.82 $p = 3.178 \times 10^{-3}$	3.35 $p = 6.496 \times 10^{-4}$	4.02 $p = 1.335 \times 10^{-4}$	2.71 $p = 4.511 \times 10^{-3}$	78.67 $\pm$ 0.21
	3.76 $p = 4.492 \times 10^{-3}$	4.54 $p = 1.154 \times 10^{-3}$	3.05 $p = 2.307 \times 10^{-2}$	3.48 $p = 1.208 \times 10^{-2}$	4.18 $p = 3.373 \times 10^{-3}$	2.82 $p = 2.790 \times 10^{-2}$	3.35 $p = 1.566 \times 10^{-2}$	4.02 $p = 5.605 \times 10^{-3}$	2.71 $p = 2.790 \times 10^{-2}$	
GBP/NZD	4.15 $p = 1.699 \times 10^{-4}$	5.22 $p = 5.625 \times 10^{-6}$	3.36 $p = 2.288 \times 10^{-3}$	3.82 $p = 5.014 \times 10^{-4}$	4.75 $p = 2.915 \times 10^{-5}$	3.10 $p = 4.802 \times 10^{-3}$	3.70 $p = 8.081 \times 10^{-4}$	4.61 $p = 5.820 \times 10^{-5}$	2.99 $p = 6.728 \times 10^{-3}$	74.97 $\pm$ 0.35
	4.15 $p = 6.287 \times 10^{-3}$	5.22 $p = 3.488 \times 10^{-4}$	3.36 $p = 2.766 \times 10^{-2}$	3.82 $p = 1.354 \times 10^{-2}$	4.75 $p = 1.545 \times 10^{-3}$	3.10 $p = 2.790 \times 10^{-2}$	3.70 $p = 1.793 \times 10^{-2}$	4.61 $p = 2.852 \times 10^{-3}$	2.99 $p = 2.790 \times 10^{-2}$	
NZD/USD	4.05 $p = 3.325 \times 10^{-5}$	5.10 $p = 1.789 \times 10^{-7}$	3.58 $p = 7.794 \times 10^{-4}$	3.78 $p = 1.358 \times 10^{-4}$	4.65 $p = 2.034 \times 10^{-6}$	3.32 $p = 1.335 \times 10^{-3}$	3.66 $p = 2.160 \times 10^{-4}$	4.52 $p = 4.027 \times 10^{-6}$	3.20 $p = 2.790 \times 10^{-3}$	78.07 $\pm$ 0.31
	4.05 $p = 1.729 \times 10^{-3}$	5.10 $p = 1.252 \times 10^{-5}$	3.58 $p = 1.793 \times 10^{-2}$	3.78 $p = 5.605 \times 10^{-3}$	4.65 $p = 1.322 \times 10^{-4}$	3.32 $p = 2.532 \times 10^{-2}$	3.66 $p = 7.546 \times 10^{-3}$	4.52 $p = 2.537 \times 10^{-4}$	3.20 $p = 2.790 \times 10^{-2}$	
USD/CHF	3.62 $p = 5.122 \times 10^{-5}$	4.02 $p = 3.397 \times 10^{-7}$	2.88 $p = 3.436 \times 10^{-4}$	3.35 $p = 1.668 \times 10^{-3}$	3.71 $p = 3.319 \times 10^{-6}$	2.65 $p = 9.002 \times 10^{-4}$	3.22 $p = 2.522 \times 10^{-4}$	3.58 $p = 6.184 \times 10^{-6}$	2.54 $p = 1.374 \times 10^{-3}$	80.01 $\pm$ 0.27
	3.62 $p = 2.561 \times 10^{-3}$	4.02 $p = 2.310 \times 10^{-5}$	2.88 $p = 1.065 \times 10^{-2}$	3.35 $p = 6.116 \times 10^{-3}$	3.71 $p = 2.124 \times 10^{-4}$	2.65 $p = 1.800 \times 10^{-2}$	3.22 $p = 8.575 \times 10^{-3}$	3.58 $p = 3.772 \times 10^{-4}$	2.54 $p = 2.307 \times 10^{-2}$	
USD/CNH	4.28 $p = 2.946 \times 10^{-4}$	4.95 $p = 5.820 \times 10^{-5}$	3.42 $p = 3.977 \times 10^{-3}$	3.95 $p = 8.081 \times 10^{-4}$	4.52 $p = 2.073 \times 10^{-4}$	3.18 $p = 8.049 \times 10^{-3}$	3.82 $p = 1.282 \times 10^{-3}$	4.38 $p = 3.436 \times 10^{-4}$	3.05 $p = 1.109 \times 10^{-2}$	70.31 $\pm$ 0.48
	4.28 $p = 9.427 \times 10^{-3}$	4.95 $p = 2.852 \times 10^{-3}$	3.42 $p = 2.790 \times 10^{-2}$	3.95 $p = 1.793 \times 10^{-2}$	4.52 $p = 7.461 \times 10^{-3}$	3.18 $p = 2.790 \times 10^{-2}$	3.82 $p = 2.307 \times 10^{-2}$	4.38 $p = 1.065 \times 10^{-2}$	3.05 $p = 2.790 \times 10^{-2}$	
USD/SEK	4.45 $p = 1.869 \times 10^{-5}$	5.88 $p = 7.421 \times 10^{-7}$	3.78 $p = 6.292 \times 10^{-4}$	4.12 $p = 7.815 \times 10^{-5}$	5.35 $p = 1.826 \times 10^{-6}$	3.45 $p = 1.473 \times 10^{-3}$	3.98 $p = 1.335 \times 10^{-4}$	5.18 $p = 1.187 \times 10^{-5}$	3.32 $p = 2.288 \times 10^{-3}$	67.13 $\pm$ 0.42
	4.45 $p = 1.028 \times 10^{-3}$	5.88 $p = 4.972 \times 10^{-5}$	3.78 $p = 1.566 \times 10^{-2}$	4.12 $p = 3.439 \times 10^{-3}$	5.35 $p = 3.772 \times 10^{-4}$	3.45 $p = 2.307 \times 10^{-2}$	3.98 $p = 5.605 \times 10^{-3}$	5.18 $p = 6.765 \times 10^{-4}$	3.32 $p = 2.746 \times 10^{-2}$	
USD/SGD	4.45 $p = 8.587 \times 10^{-6}$	5.88 $p = 4.103 \times 10^{-9}$	3.78 $p = 1.568 \times 10^{-4}$	4.12 $p = 3.789 \times 10^{-5}$	5.35 $p = 8.795 \times 10^{-8}$	3.45 $p = 5.606 \times 10^{-4}$	3.98 $p = 6.802 \times 10^{-5}$	5.18 $p = 2.219 \times 10^{-7}$	3.32 $p = 9.002 \times 10^{-4}$	83.27 $\pm$ 0.39
	4.45 $p = 5.066 \times 10^{-4}$	5.88 $p = 2.934 \times 10^{-7}$	3.78 $p = 6.116 \times 10^{-3}$	4.12 $p = 1.932 \times 10^{-3}$	5.35 $p = 6.245 \times 10^{-6}$	3.45 $p = 1.458 \times 10^{-2}$	3.98 $p = 3.239 \times 10^{-3}$	5.18 $p = 1.531 \times 10^{-5}$	3.32 $p = 1.800 \times 10^{-2}$	

Note that p denotes the two-sided asymptotic p -value obtained from the DM statistic under the standard normal approximation, and p_H denotes the Holm–Bonferroni adjusted p -value based on all 72 DM tests (8 currency pairs $\times$ 3 error metrics $\times$ 3 forecasting horizons). Statistical significance was assessed at $\alpha = 0.05$. The bootstrap accuracy is reported as mean $\pm$ standard deviation over the bootstrap samples.

predominantly lower than the corresponding actual values, whereas in the four-hour and daily timeframes, the forecasts tend to remain above the real market prices for most periods. This behavior suggests a timeframe-dependent bias in the model's estimations. Similar to previous observations, a specific interval is identified for the USD/SEK currency pair across all three timeframes, during which the ICLS predictions remain largely range-bound. In this interval, the model fails to capture both the prevailing trend and the associated price variations, effectively overlooking the directional movement present in the actual market data. Overall, these findings indicate that while the ICLS model improves directional alignment, it remains less responsive to trend development when such movements are accompanied by limited or structurally constrained volatility.

The ICGS model (Figure 9) is able to effectively follow the overall price movement across all three timeframes, demonstrating consistent directional alignment with the actual market behavior. This indicates a strong capability in capturing the general trend dynamics of the price series. However, in certain intervals, the model produces forecasts with excessively high variability, resulting in amplified oscillations that exceed those observed in the real price data. Conversely, there are also periods in which the model fails to adequately capture sudden price changes, smoothing out abrupt movements present in the actual series. Similar to the previous models, the previously identified issue for the USD/SEK currency pair remains observable across all three timeframes. Nevertheless, in the case of the ICGS model, this behavior appears to be partially mitigated, as the range-bound predictions exhibit improved responsiveness to the underlying price trend compared to the other models. Overall, these results suggest that while the ICGS model achieves better trend-following performance, it still faces challenges in balancing sensitivity to abrupt price movements and stability in low-volatility or structurally constrained market conditions.

For the IICS (Figure 10) model, a noticeable improvement is observed compared to the preceding models across all three timeframes. The forecasts exhibit stronger directional consistency with the actual price movements, and the prediction errors gradually reveal more structured and recognizable patterns rather than purely random deviations. This behavior suggests enhanced stability and learning capacity in the model's predictive process. Despite these improvements, certain limitations remain evident. Specifically, the model continues to struggle with accurately capturing price fluctuations characterized by large amplitudes, and in some intervals, it generates predictions with excessively high variability. These behaviors indicate that, while the IICS model achieves superior overall performance, challenges persist in balancing responsiveness to high-magnitude price movements and the suppression of spurious oscillations in the predicted series.

For the IICLS model (Figure 11), a substantial improvement in predictive accuracy is observed across all three timeframes. In most time intervals, the forecasts exhibit bias-dominated errors, indicating systematic deviations rather than random fluctuations. Despite this bias, the

model demonstrates a strong capability in capturing price variability, successfully reflecting market oscillations in the majority of cases. Except for a limited number of specific intervals, in which minor discrepancies between the predicted and actual prices are observed, the distance between the forecasts and the true price values remains relatively small. In the remaining periods, the predicted and actual price trajectories are nearly overlapping, highlighting the model's enhanced ability to closely track market dynamics. Overall, these results indicate that the IICLS model achieves a high level of alignment between predicted and actual prices, with residual errors being predominantly structured and limited in magnitude.

Figure 12 shows the strongest overall performance among the six evaluated models. Across all three timeframes, the predictions exhibit a high degree of alignment with the actual price movements, with the one-hour timeframe showing slightly superior performance compared to the four-hour and daily resolutions. The prediction errors are predominantly bias-driven and follow consistent, structured patterns rather than appearing random. Especially, the performance for the USD/SEK currency pair shows a substantial improvement in this model. Unlike earlier models, the IICGS framework effectively captures both the underlying trend and price fluctuations for this pair across all timeframes. Overall, no significant temporal lag is observed between the predicted and actual series, and the model demonstrates a robust ability to track price dynamics, successfully capturing market oscillations. These results indicate a consistent and meaningful improvement in predictive performance across all timeframes and all evaluated currency pairs.

Overall, the evaluated models can be categorized into two distinct groups based on the type of Ichimoku-related features incorporated into the learning framework. The first group of models utilizes the five canonical Ichimoku indicators directly alongside OHLC price data. While this configuration enhances the models' ability to capture general trend-following behavior, their performance remains limited in terms of accurately representing complex price dynamics, particularly under varying volatility regimes and across higher timeframes. In contrast, the second group of models leverages engineered features derived from the Ichimoku indicator, combined with OHLC prices. These engineered representations provide a more informative and structured description of market dynamics, enabling the models to achieve stronger directional alignment, improved stability, and reduced temporal lag across all timeframes. The empirical results consistently demonstrate that feature engineering plays a critical role in enhancing both price-tracking capability and trend prediction robustness. Consequently, models employing engineered Ichimoku features exhibit superior overall performance compared to those relying solely on the raw indicator components.

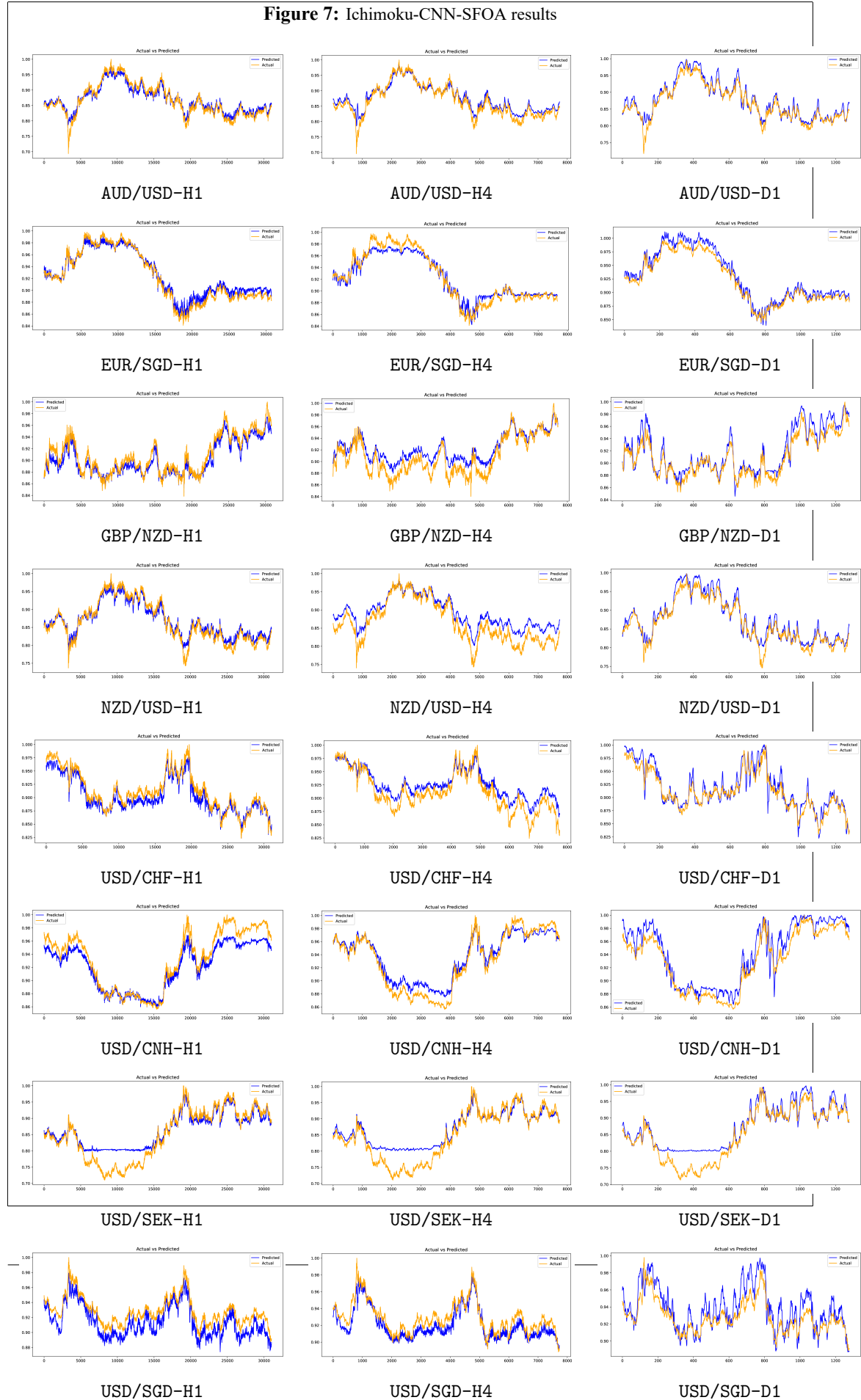
Figure 7: Ichimoku-CNN-SFOA results

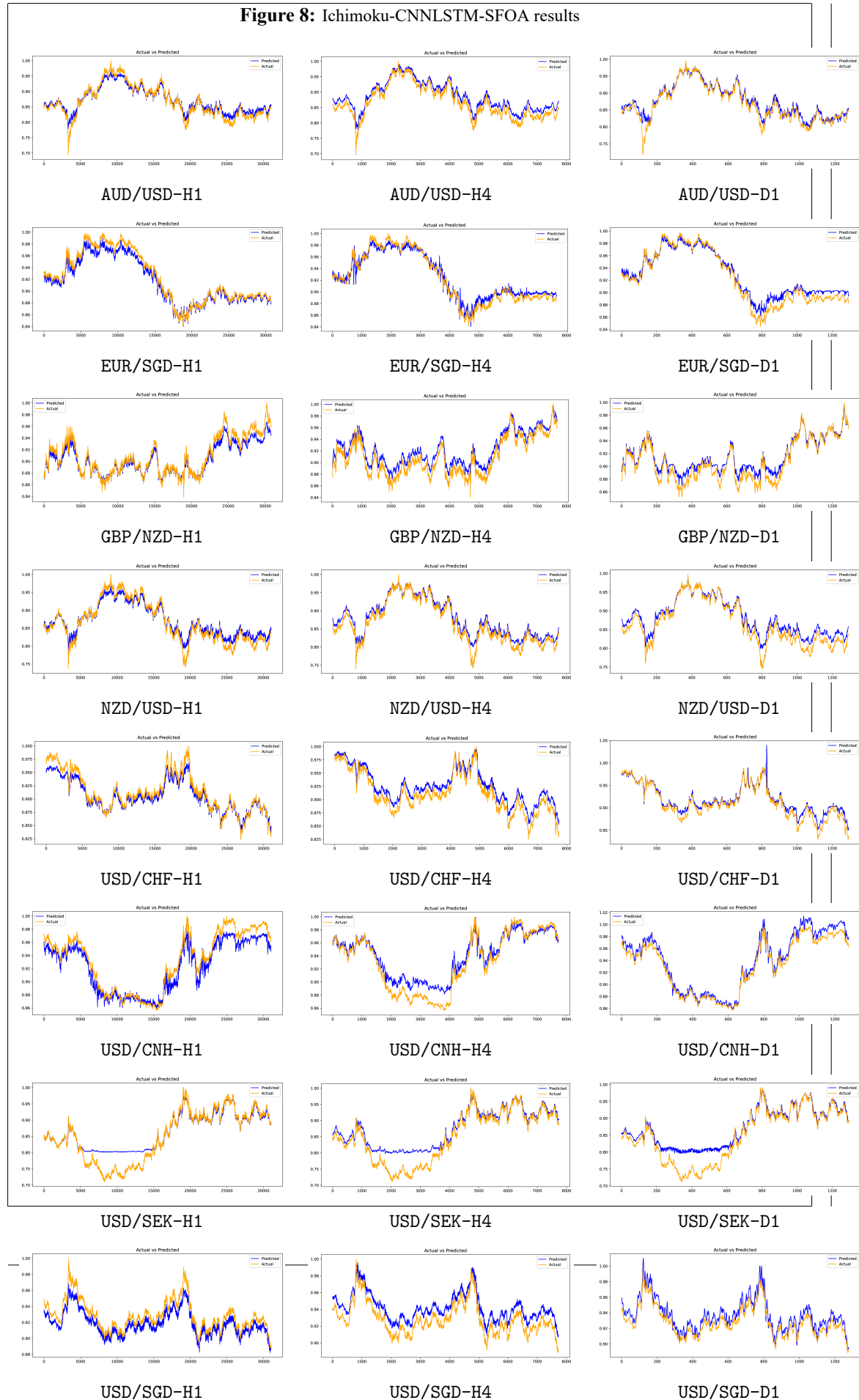
Figure 8: Ichimoku-CNNLSTM-SFOA results

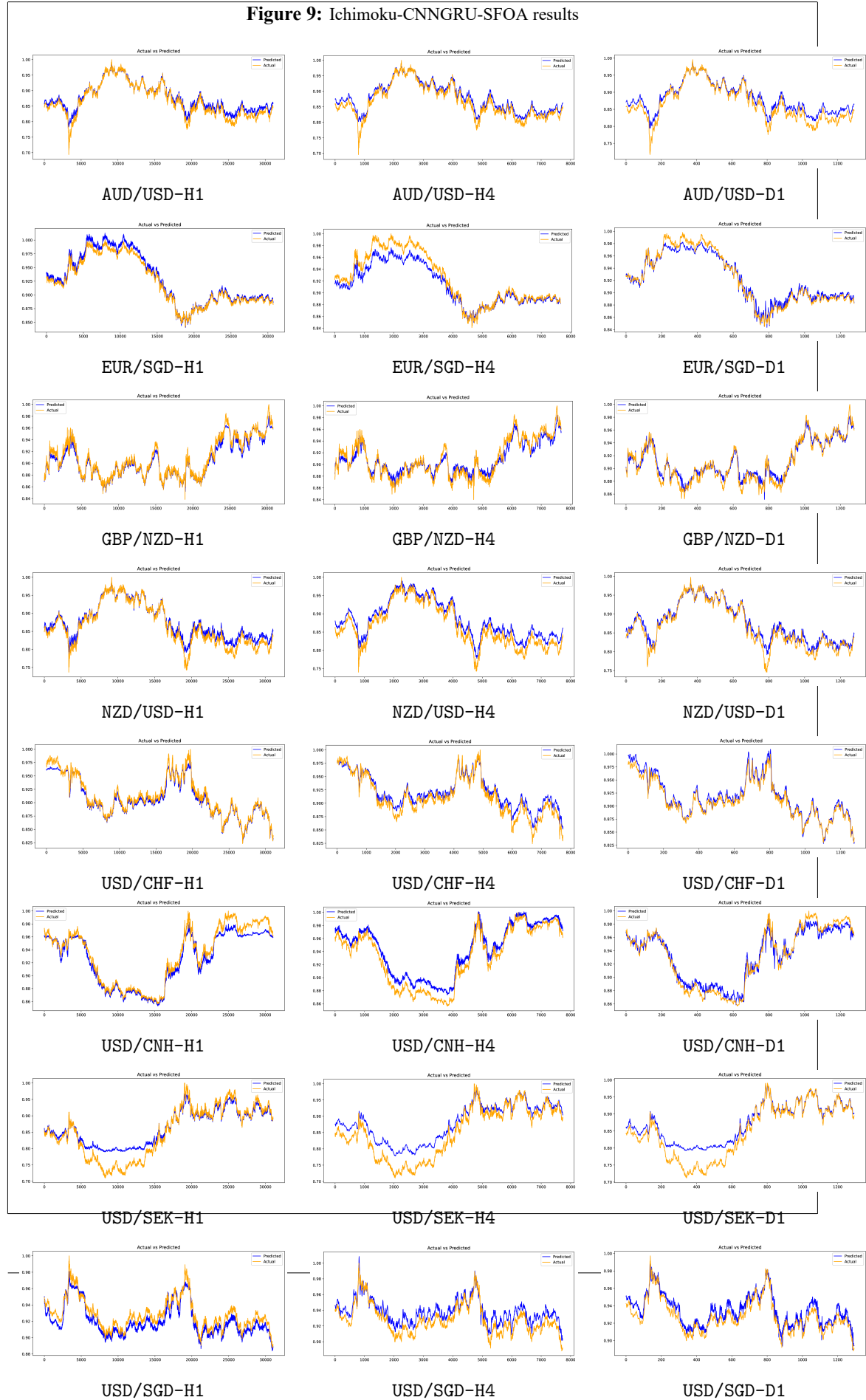
Figure 9: Ichimoku-CNNGRU-SFOA results

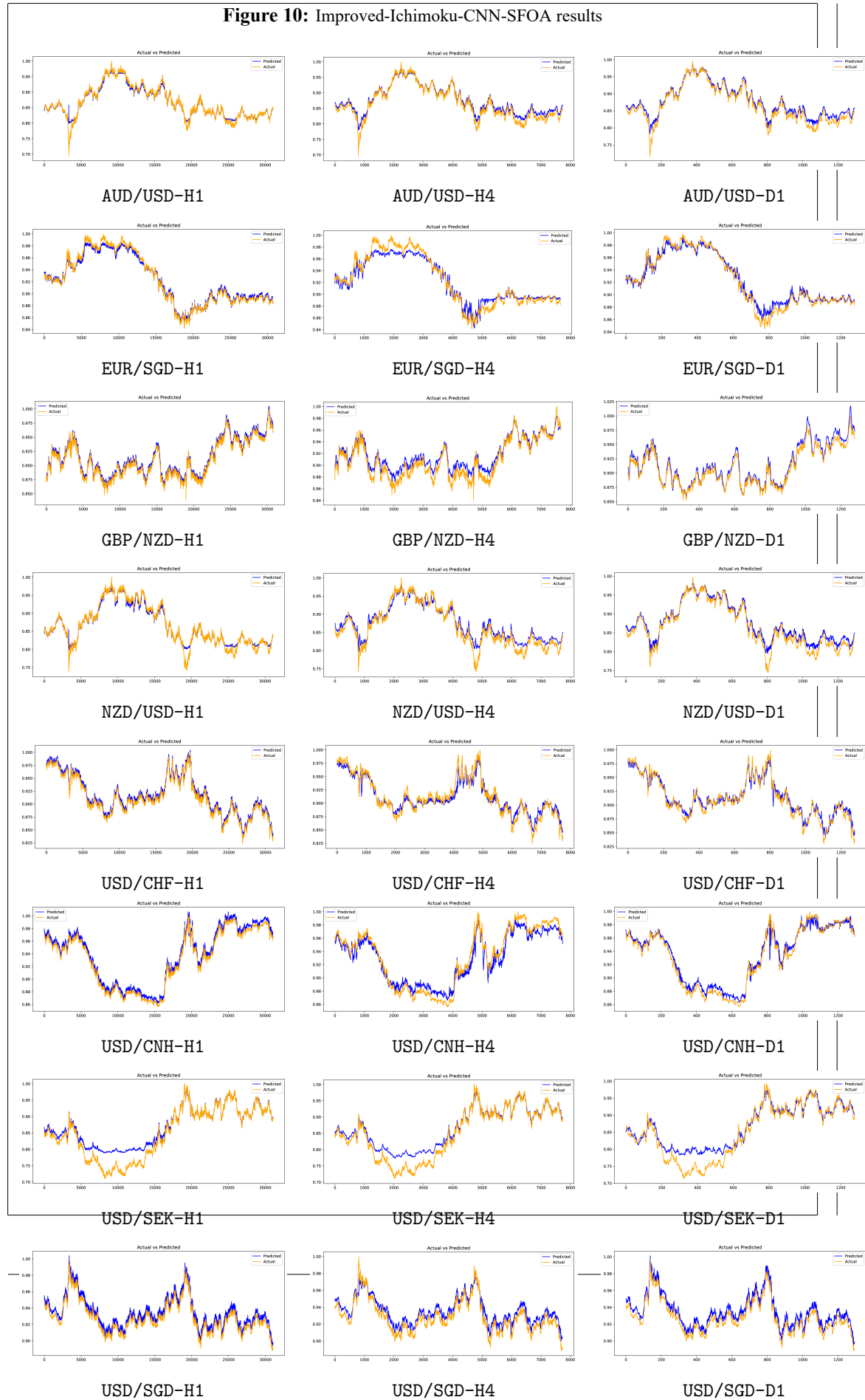
Figure 10: Improved-Ichimoku-CNN-SFOA results

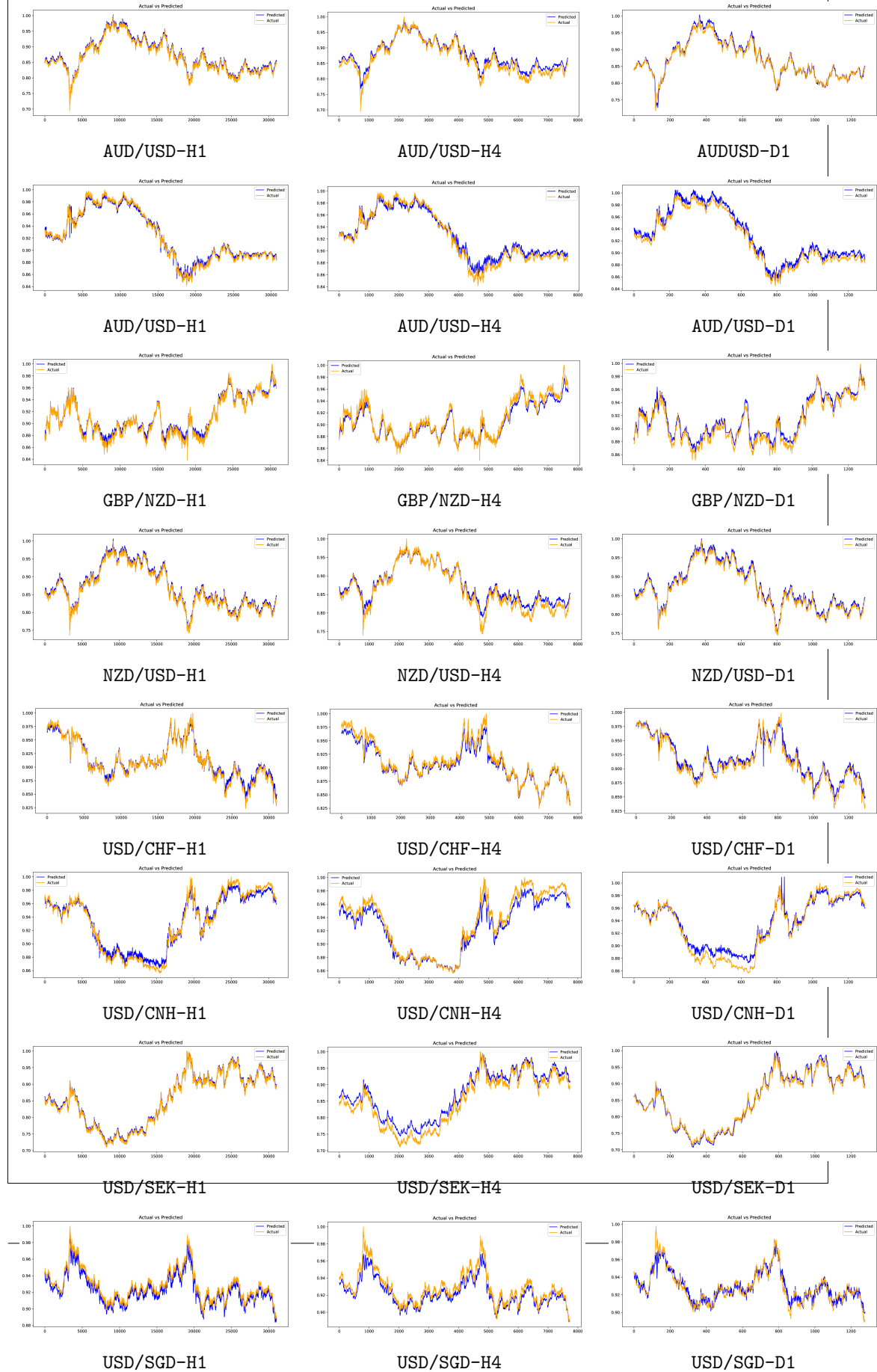
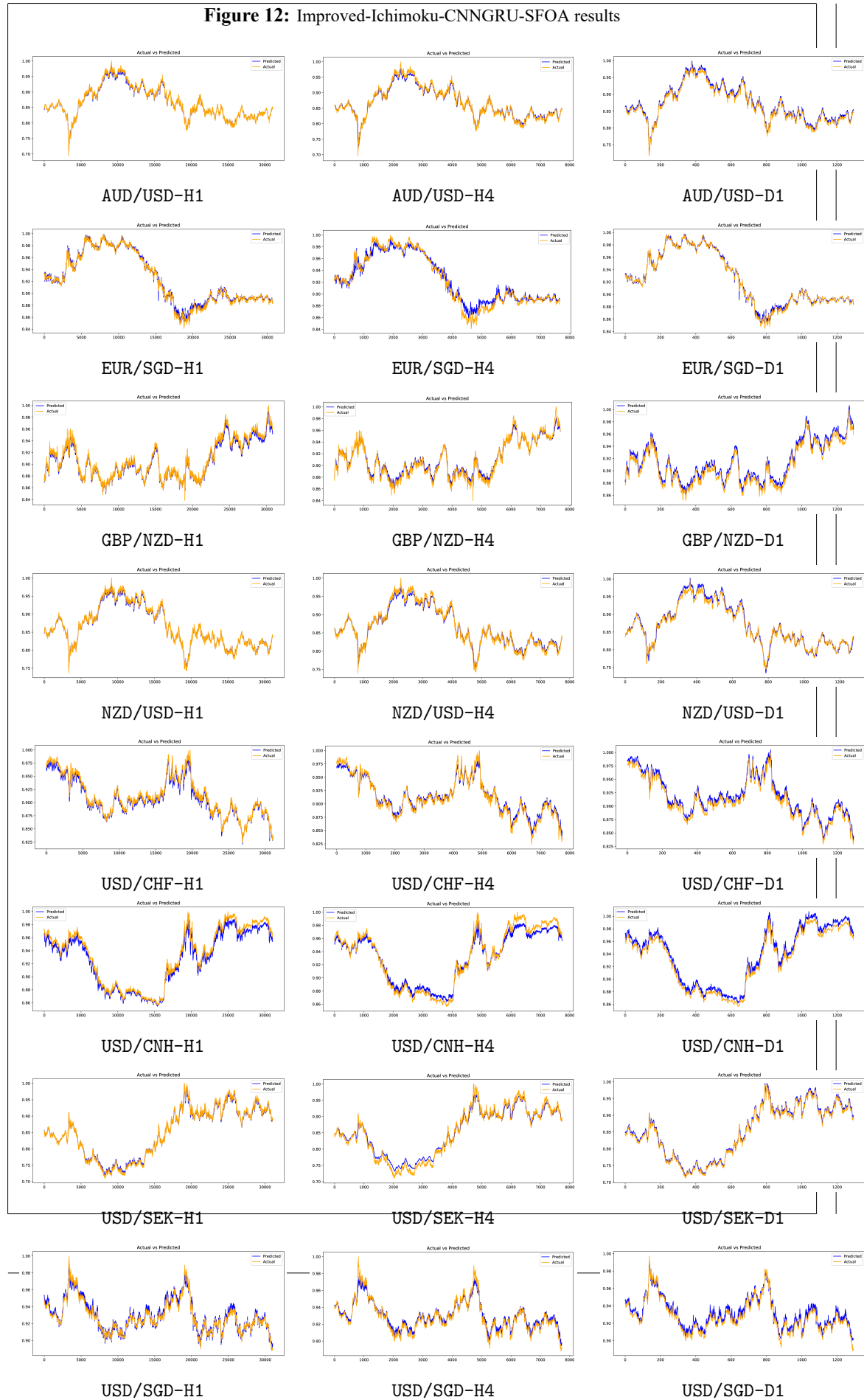
Figure 11: Improved-Ichimoku-CNNLSTM-SFOA results

Figure 12: Improved-Ichimoku-CNNGRU-SFOA results

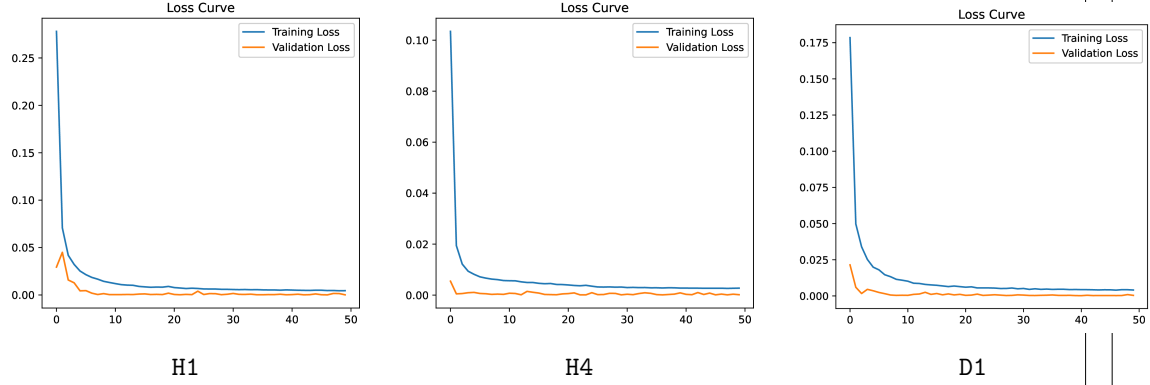
D Discussion

D.1 Motivation for using Ichimoku Kinko Hyo

The selection of Ichimoku Kinko Hyo over conventional indicators such as RSI, MACD, and Bollinger Bands provides a distinct advantage due to its forward-looking characteristics. Traditional indicators primarily extract insights from historical data, generating features that represent previous market conditions. Although these methods are effective to a degree, they are predominantly lagging in nature and provide more limited information about future market movements. In contrast, the Ichimoku indicator offers unique feature-engineering capabilities through its components, particularly the Kumo (cloud), which by default is displaced 26 periods forward. In classical technical analysis, this forward displacement allows traders to examine the position of price relative to the cloud that appears ahead of the current price and to interpret potential interactions with it as anticipatory signals about future trend direction. It is important to clarify, however, that the 26-period forward shift is solely a plotting convention used for visual alignment. In the feature-engineering process of this study, all Ichimoku components at time t are computed exclusively from price information available up to time t (or $t - 1$); no actual future observations are used. Consequently, while the resulting features preserve the rich forward-oriented information valued by classical traders—capturing shifts in momentum, equilibrium, and support/resistance levels more effectively than traditional lagging indicators—they remain completely free of data leakage. Integrating these carefully constructed features enables the predictive model to overcome the limitations of conventional indicators and deliver robust predictions suited to the complexities of volatile market environments.

D.2 Training Stability and Loss Behavior Analysis

Figure 13 illustrates the successful convergence behavior of the proposed model across the H1, H4, and D1 timeframes. As observed, the training and validation loss consistently decrease over epochs, indicating stable learning dynamics without significant oscillations. The relatively small gap between training and validation loss suggests that the model maintains strong generalization capability and does not suffer from severe overfitting. This stability is particularly important in financial time-series forecasting, in which data is highly noisy and non-stationary. Furthermore, the smooth convergence pattern reflects the effectiveness of the proposed architecture along with the applied regularization techniques, such as Dropout and Batch Normalization, in controlling model complexity. Overall, the results demonstrate that the training process is successful and that the proposed framework is robust and reliable across different forecasting horizons.

Figure 13: Loss Evaluation Results of IICGS

D.3 Comparison with Baseline Models

In this subsection, the proposed IICGS framework is empirically evaluated against several baseline models, including statistical methods (ARIMA and GARCH), traditional machine learning classifiers (SVM, Random Forest, and XGBoost), and deep learning architectures (LSTM, GRU, CNN, and their hybrid variants). All baseline models are evaluated under the same experimental environment and data partitions to ensure a consistent and reproducible comparison. The evaluation considers two complementary aspects of Forex forecasting: (i) directional trend prediction, assessed using classification accuracy, F1-score, and bootstrap mean $\pm$ SD, and (ii) continuous price-level prediction across H1, H4, and D1 horizons, assessed using MAE and the Diebold–Mariano (DM) test. It should be noted that the results reported in this subsection are obtained by evaluating the single best-performing model selected from the ten independent runs. To provide broader context against existing literature, we also consider relevant benchmarking frameworks such as the study by Jarusek et al. [16]. They proposed a short-term Forex forecasting methodology integrating Elliott Wave patterns, Fast Fourier Transform (FFT), and neural networks on EUR/USD. While their evaluation reported an aggregated performance score of approximately 77% based on averaged percentage-converted error metrics (MSE, RMSE, MAE, and NCC), their output represents a composite regression-based index rather than standard directional classification accuracy. Consequently, the findings of Jarusek et al. [16] serve as background literature and a contextual reference for continuous price prediction rather than a direct benchmark. This distinction is necessary due to differences in datasets, sampling frequencies, time horizons, and evaluation protocols. Primary emphasis is therefore placed on controlled comparisons with baseline models evaluated under identical experimental conditions.

D.3.1 Directional Trend Forecasting Evaluation

The directional prediction capability is evaluated through both the Full-Dataset Classification Accuracy (%) and the Bootstrap simulation metrics reported as mean $\pm$ SD, as summarized in Table 9. According to the discussion of negative overfitting in the established literature [29], exceptionally high performance during the optimization phase does not necessarily translate into comparable performance during subsequent testing periods and may indicate overfitting. Therefore, evaluating model performance beyond the optimization period is essential for assessing its generalization capability. This methodological consideration is consistent with our empirical findings. Across 10 independent evaluation runs on the dedicated testing data (Table 3), the proposed IICGS framework achieves a stable mean accuracy of $78.97 \pm 0.5\%$. When evaluated on the entire dataset, including both training and testing subsets (Table 9), the accuracy is 78.47%. The small difference between these results suggests that the proposed framework does not exhibit substantial performance degradation when evaluated beyond the dedicated testing subset, supporting its ability to generalize effectively across different market conditions.

Crucially, the IICGS framework achieves the highest classification accuracy among all compared methods (78.47%), substantially outperforming traditional machine learning baselines such as XGBoost (53.61%), Random Forest (51.26%), and SVM (44.10%). Furthermore, empirical simulation demonstrates that our framework exhibits the lowest bootstrap standard deviation ($78.31 \pm 0.35\%$). This low SD indicates a highly concentrated distribution, reflecting superior prediction consistency and robustness under resampled market environments. This statistical stability proves that integrating Ichimoku-based features with an optimized CNN-GRU architecture effectively mitigates the inherent noise and high volatility of Forex time series. In addition to directional classification, continuous price-level tracking across multiple timeframes (H1, H4, and D1) demonstrates the capability of the regression architecture to capture price dynamics. Together with the 78.47% classification accuracy, these results indicate strong performance in both price tracking and directional classification, suggesting that the IICGS framework can effectively capture continuous price movements while providing reliable directional trend predictions under volatile market conditions.

Table 9: Comparative Analysis of Model Performances Based on Full-Dataset Classification Accuracy vs. Bootstrap mean $\pm$ SD (%) for EUR/USD.

Reference	Methodology	Accuracy (%)	Bootstrap [mean $\pm$ SD (%)]
Our Framework	IICGS	78.47	78.31 $\pm$ 0.35
xgboost Library	XGBoost Classifier	53.61	53.39 $\pm$ 0.58
sklearn Library	Random Forest Classifier	51.26	51.38 $\pm$ 0.61
sklearn Library	Support Vector Machine (SVM)	44.10	43.92 $\pm$ 0.65

D.3.2 Multi-Horizon Price Forecasting & Statistical Significance

To evaluate price-level prediction and verify whether the proposed framework's superiority is statistically significant rather than a result of random chance, we compute the Mean Absolute Error (MAE) and conduct the Diebold–Mariano (DM) test across multiple trading horizons (H1, H4, and D1). The proposed IICGS framework serves as the target model against which all baseline methods are individually compared. The empirical results are detailed in Table 10, which shows that the proposed IICGS framework consistently achieves the lowest MAE values among all compared methods, recording 0.0034, 0.0028, and 0.0027 for the H1, H4, and D1 horizons, respectively. Crucially, all computed DM statistics for the baseline models relative to IICGS are positive and substantially exceed the critical threshold of 1.96. The corresponding raw p -values and Holm–Bonferroni adjusted p -values ($m = 24$) remain far below $\alpha = 0.05$, formally rejecting the null hypothesis of equal predictive accuracy. This statistically confirms that the predictive gains of the proposed framework are highly significant and not attributable to sampling variation.

Table 10: Performance comparison, DM statistics, raw p -values and Holm–Bonferroni adjusted p -values across multiple forecasting horizons on EUR/USD dataset.

Model	MAE (H1)	MAE (H4)	MAE (D1)	DM (H1)	p (H1)	Holm (H1)	DM (H4)	p (H4)	Holm (H4)	DM (D1)	p (D1)	Holm (D1)
ARIMA	0.0191	0.0183	0.0195	14.32	1.64×10^{-46}	3.61×10^{-45}	15.11	1.39×10^{-51}	3.20×10^{-50}	16.04	6.72×10^{-58}	1.61×10^{-56}
GARCH	0.0168	0.0179	0.0182	12.05	1.94×10^{-33}	3.69×10^{-32}	12.89	5.12×10^{-38}	1.03×10^{-36}	13.91	5.51×10^{-44}	1.16×10^{-42}
Plain LSTM	0.0161	0.0175	0.0169	9.84	7.57×10^{-23}	1.14×10^{-21}	10.12	4.50×10^{-24}	7.21×10^{-23}	11.05	2.19×10^{-28}	3.95×10^{-27}
Plain GRU	0.0153	0.0165	0.0160	8.91	5.10×10^{-19}	6.12×10^{-18}	9.45	3.39×10^{-21}	4.74×10^{-20}	10.18	2.44×10^{-24}	4.14×10^{-23}
Plain CNN	0.0149	0.0162	0.0158	8.12	4.66×10^{-16}	4.20×10^{-15}	8.76	1.95×10^{-18}	2.15×10^{-17}	9.34	9.63×10^{-21}	1.25×10^{-19}
Plain CNN-LSTM	0.0137	0.0157	0.0151	7.23	4.83×10^{-13}	3.38×10^{-12}	7.89	3.02×10^{-15}	2.42×10^{-14}	8.42	3.77×10^{-17}	3.77×10^{-16}
Plain LSTM-GRU	0.0124	0.0128	0.0148	6.04	1.54×10^{-9}	6.17×10^{-9}	6.32	2.62×10^{-10}	1.31×10^{-9}	7.11	1.16×10^{-12}	6.96×10^{-12}
Plain CNN-GRU	0.0102	0.0110	0.0146	4.87	1.12×10^{-6}	1.12×10^{-6}	5.12	3.06×10^{-7}	6.11×10^{-7}	5.95	2.68×10^{-9}	8.04×10^{-9}
IICGS (Proposed)	0.0034	0.0028	0.0027	—	—	—	—	—	—	—	—	—

Note: All raw and Holm–Bonferroni adjusted p -values ($m = 24$) remain far below 0.05, confirming statistical significance at the 5% level.

Moreover, an analysis of the error trends across horizons reveals a distinctive property of IICGS: while baseline models (e.g., ARIMA, GARCH, and Plain CNN-GRU) exhibit performance degradation or increased error volatility when moving from short-term (H1) to longer-term (D1) horizons, the proposed framework shows a progressive reduction in MAE. This behavior indicates that the Ichimoku-inspired multi-scale feature extraction effectively isolates macro trend structures from intraday noise, enabling the optimized CNN-GRU layers to maintain exceptional prediction stability and error containment over broader temporal horizons.

D.3.3 Effectiveness Analysis of Model Components

To isolate and investigate the individual empirical contribution of each core component within our pipeline, an ablation study was conducted. Four distinct configurations were evaluated: (A) the baseline CNN-GRU network without enhancements, (B) the CNN-GRU network equipped with Ichimoku-based engineered features, (C) the CNN-GRU network optimized via the Starfish Optimization Algorithm (SFOA), and (D) the complete integrated IICGS framework.

The relative improvements reported in Table 11 quantify the percentage reduction in MAE with respect to the baseline CNN-GRU model, calculated as:

$$\text{Improvement (\%)} = \frac{MAE_{\text{Base}} - MAE_{\text{Model}}}{MAE_{\text{Base}}} \times 100. \quad (20)$$

The empirical findings from the ablation study indicate that integrating Ichimoku-based feature engineering independently drops the MAE from 0.0045 to 0.0040 (an 11.11% improvement), validating that multi-scale technical features enrich the input space beyond raw OHLC data. On the other hand, applying SFOA-based hyperparameter optimization independently reduces the MAE to 0.0038 (a 15.56% improvement), proving the optimization algorithm's strength in discovering superior architectural parameters by balancing exploitation and exploration. The best-so-far solution is obtained when both components are deployed simultaneously in Configuration D (IICGS), yielding an MAE of 0.0034. This corresponds to a total error reduction of 24.44% relative to the base network, indicating that the combination of structured financial indicators and metaheuristic optimization can contribute to improved forecasting performance.

Table 11: Ablation study of the proposed framework to evaluate the contribution of Ichimoku-based feature engineering and SFOA-based hyperparameter optimization.

Model	Configuration	MAE	Improvement (%)
Model A	CNN-GRU (Base model)	0.0045	—
Model B	CNN-GRU + Ichimoku	0.0040	11.11%
Model C	CNN-GRU + SFOA	0.0038	15.56%
Model D	CNN-GRU + Ichimoku + SFOA (Proposed)	0.0034	24.44%

D.4 Comparative Analysis of Metaheuristic Optimizers

In this study, a comparative evaluation of metaheuristic algorithms for hyperparameter tuning in deep learning models is conducted for Forex trend forecasting. The evaluated algorithms include the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer

(GWO), and the proposed Ichimoku-CNN-GRU-SFOA (IICGS) framework, which incorporates the Starfish Optimization Algorithm (SFOA). Performance is assessed using Accuracy (%), F1-score, and bootstrap statistics reported as mean $\pm$ SD across eight currency pairs. For each method, the results reported in Table 12 correspond to the best-performing model obtained from ten independent runs and subsequently evaluated on the complete dataset. Therefore, the IICGS results in this table may differ from those reported in Table 7, where the results are presented as the mean $\pm$ standard deviation across the ten independent runs. This difference reflects the distinct reporting procedures used in the two tables rather than an inconsistency in the experimental results. Overall, the proposed IICGS framework achieves higher predictive performance than GA, PSO, and GWO across all evaluated currency pairs. For example, for AUD/USD, IICGS achieves an Accuracy of 79.41%, with a bootstrap estimate of $79.28 \pm 0.38\%$ and an F1-score of 0.65. In comparison, GWO achieves 68.23% Accuracy ($68.11 \pm 0.59\%$), PSO achieves 68.56% ($68.41 \pm 0.58\%$), and GA achieves 59.12% ($58.94 \pm 0.65\%$). Similarly, for EUR/SGD, IICGS achieves 78.77% Accuracy and an F1-score of 0.61, compared with 70.12% and 0.57 for GWO, 69.88% and 0.55 for PSO, and 64.43% and 0.51 for GA. Across the evaluated currency pairs, IICGS consistently outperforms the three baseline optimization algorithms in both Accuracy and F1-score. The bootstrap results also show relatively small standard deviations for IICGS, indicating stable performance under resampling. Overall, these findings demonstrate the competitive performance and robustness of the proposed SFOA-based hyperparameter optimization framework.

Table 12: Comparison of hyperparameter optimization algorithms based on Accuracy (%) with Bootstrap mean $\pm$ SD (%) and F1-score.

Ticker	Accuracy (%) [Bootstrap mean $\pm$ SD]				F1-score			
	GA	PSO	GWO	IICGS (Ours)	GA	PSO	GWO	IICGS (Ours)
AUD/USD	59.12 (58.94 ± 0.65)	68.56 (68.41 ± 0.58)	68.23 (68.11 ± 0.59)	79.41 (79.28 ± 0.38)	0.50	0.55	0.55	0.65
EUR/SGD	64.43 (64.25 ± 0.64)	69.88 (69.73 ± 0.60)	70.12 (69.95 ± 0.58)	78.77 (78.61 ± 0.39)	0.51	0.55	0.57	0.61
GBP/NZD	57.77 (57.59 ± 0.66)	67.89 (67.74 ± 0.59)	67.46 (67.31 ± 0.61)	75.19 (75.05 ± 0.41)	0.52	0.57	0.58	0.66
NZD/USD	61.35 (61.18 ± 0.63)	70.12 (69.97 ± 0.57)	70.57 (70.42 ± 0.59)	78.17 (78.04 ± 0.37)	0.53	0.59	0.60	0.63
USD/CHF	63.57 (63.38 ± 0.65)	72.35 (72.19 ± 0.58)	70.89 (70.73 ± 0.60)	80.11 (79.96 ± 0.39)	0.50	0.54	0.54	0.63
USD/CNH	58.89 (58.71 ± 0.64)	65.43 (65.28 ± 0.61)	67.12 (66.97 ± 0.58)	70.57 (70.43 ± 0.40)	0.55	0.62	0.61	0.67
USD/SEK	54.23 (54.06 ± 0.62)	60.57 (60.42 ± 0.59)	62.35 (62.19 ± 0.57)	66.88 (66.74 ± 0.38)	0.54	0.60	0.58	0.64
USD/SGD	69.13 (68.95 ± 0.65)	76.47 (76.31 ± 0.58)	76.89 (76.74 ± 0.60)	83.23 (83.09 ± 0.36)	0.50	0.55	0.55	0.59

D.5 Limitations and Metrics in Evaluating Trading Models

In financial forecasting, traditional evaluation metrics such as MAE, MSE, and MAPE are commonly used to assess the performance of regression models. However, in algorithmic trading contexts, these metrics may be insufficient. A model with low average errors can still generate small fluctuations that inadvertently trigger stop-loss mechanisms, leading to premature trade

exits and undermining the overall strategy. This problem primarily stems from relying on one-step-ahead predictions, in which the trend direction is determined by merely comparing two consecutive steps (the immediate next prediction versus the current step). In lower timeframes, such a consecutive comparison is highly susceptible to localized anomalies; minor, sudden price fluctuations or market noise can easily flip the predicted direction, leading to severe trend misclassifications. As illustrated in Figure 14, while the predicted price closely tracks the actual price according to regression metrics, these micro-fluctuations frequently cross stop-loss thresholds, resulting in unnecessary losses. To mitigate this, we adopt a classification-based

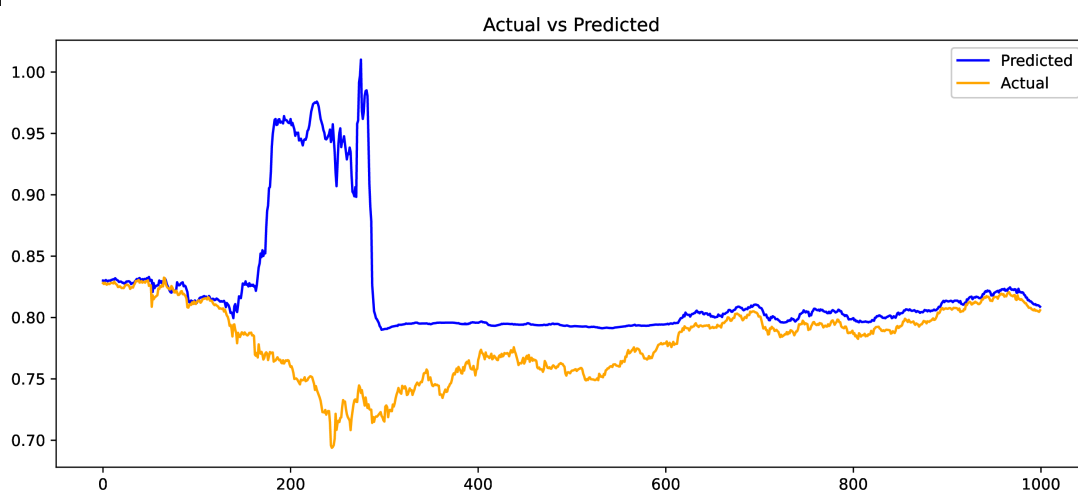


Figure 14: Predicted vs. actual prices with stop-loss crossings.

evaluation approach powered by a multi-step forecasting formulation. Instead of focusing on the magnitude of numerical errors between consecutive steps, we prioritize the accurate identification of the broader trend direction by evaluating multiple future steps against the current horizon. This multi-step approach inherently filters out sudden intraday noises and prevents premature stop-loss triggers, aligning the model more effectively with real-world trading needs in which strategic success depends on directional accuracy. Consequently, we utilize classification metrics, such as Accuracy and the F1-score, as key indicators of model performance across short-term (H1), medium-term (H4), and daily (D1) timeframes. To support this robust evaluation, the trend direction in each dataset is derived from historical price movements, resulting in a balanced distribution in which approximately 50% of the instances represent an upward trend and 50% a downward trend. This balanced class distribution, maintained across all three timeframes, reduces the effect of class imbalance and the dominance of any single class, thereby making Accuracy a more straightforward and reliable overall performance measure. The F1-score is additionally reported to provide a complementary assessment of the balance between precision and recall.

E Conclusions

This study presented a robust multi-timeframe and multi-step framework for Forex trend prediction by integrating CNN-based deep learning architectures with Ichimoku-driven feature engineering and the Starfish Optimization Algorithm (SFOA). In this framework, the outputs of the regression models are treated as dynamic, price-following indicators. While higher regression accuracy pulls these indicators closer to the actual price and better reflects market tendencies, experimental observations revealed that relying solely on a single daily-scale regression model can create a false sense of reliability, as low regression errors may still result in substantial trend misclassifications. To address this limitation, a multi-timeframe and multi-step forecasting strategy was implemented using the EUR/USD currency pair as the primary training baseline. By extending the prediction horizon into multiple future steps for lower timeframes, such as evaluating the next several steps against the current one in the H1 dimension, the influence of minor, localized prediction errors can be reduced. The same process was applied to the H4 medium-term timeframe to further validate the directional trend before aggregating the signals into a final daily market decision through a voting mechanism. Among the tested configurations, the IICGS model achieved the most reliable performance in capturing complex sequential dependencies. Importantly, the proposed framework demonstrated strong out-of-sample generalization across various currency pairs. Without retraining, the framework maintained stable trend classification performance across these currency pairs, indicating zero-shot-like learning capabilities across the broader Forex market. Statistical validation using the Diebold–Mariano and Bootstrap tests further supported the significance and robustness of the observed performance improvements.

Limitations and Future Work. Although the proposed IICGS framework demonstrates consistent improvements in both regression and classification performance across multiple currency pairs and timeframes, several limitations should be acknowledged. First, the experimental evaluation is primarily conducted on the Forex market. Although the selected currency pairs represent diverse market characteristics, the generalizability of the proposed framework to other financial assets, such as equities, commodities, cryptocurrencies, or fixed-income markets, has not yet been systematically investigated. Future studies should therefore evaluate the framework across a broader range of asset classes to further assess its robustness and transferability. Second, although the Starfish Optimization Algorithm (SFOA) provides effective exploration and exploitation capabilities, its stochastic nature, like that of other evolutionary optimization algorithms, introduces an inherent degree of uncertainty into the optimization process. Therefore, future studies could investigate hybrid adaptive optimization strategies that combine evolutionary algorithms such as SFOA with classical deterministic optimization methods, potentially achieving a better balance between global exploration and local refinement while reducing dependence on stochastic search. Since the optimization process is performed only during model

training and does not affect the inference stage, such strategies could also contribute to improving the computational efficiency and stability of the optimization process. Additionally, future work could incorporate new technical features, such as the VWAP indicator, or indices derived from sentiment and fundamental analysis to further enrich the feature space and potentially improve predictive performance. Future research will also investigate transformer-based architectures, adaptive trading systems, and the application of the proposed framework to higher-volatility assets such as XAU/USD and cryptocurrencies. Addressing these limitations may further improve the scalability, transferability, and practical applicability of the proposed framework in real-world financial forecasting.

Appendix. Starfish Optimization Algorithm (SFOA)

The Starfish Optimization Algorithm (SFOA) is a bio-inspired metaheuristic designed to solve global optimization problems. By mimicking the behaviors of starfish, SFOA introduces a hybrid search strategy that effectively balances exploration and exploitation. The algorithm comprises five primary phases, each of which is mathematically defined as follows:

Initialization Phase

In the initialization phase, the positions of the starfish are randomly generated within the boundaries of the design variables. The positions are represented as Equation (21):

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1D} \\ X_{21} & X_{22} & \dots & X_{2D} \\ \vdots & \vdots & \dots & \vdots \\ X_{N1} & X_{N2} & \dots & X_{ND} \end{bmatrix}. \quad (21)$$

where $\mathbf{X}$ is a matrix of size $N \times D$, with N representing the population size and D the number of dimensions of the design variables. The position of each starfish is calculated using Equation (22):

$$X_{ij} = l_j + r \cdot (u_j - l_j), \quad i = 1, 2, \dots, N, j = 1, 2, \dots, D. \quad (22)$$

where X_{ij} denotes the position of the i -th starfish in the j -th dimension, r is a random number between 0 and 1, and l_j and u_j represent the lower and upper bounds of the j -th dimension, respectively. After initializing the positions, the fitness values for all starfish are computed using the objective function. These fitness values are stored in a vector as Equation (23):

$$\mathbf{F} = \begin{bmatrix} F(X_1) \\ F(X_2) \\ \vdots \\ F(X_N) \end{bmatrix}, \quad (23)$$

where $\mathbf{F}$ is a vector of size $N \times 1$, containing the fitness values of the starfish. Once the initialization is complete, the algorithm proceeds to the exploration and exploitation phases.

Exploration Phase

The exploration phase simulates the searching capabilities of the starfish, modeled as a five-armed structure with eyes at the tips of each arm. The exploration behavior is governed by the following principles:

- For optimization problems with dimensions greater than 5 ($D > 5$), the algorithm uses a five-dimensional search pattern. Each arm moves in one of five randomly selected dimensions to explore the search space more efficiently. This exploration is guided by the best position among all starfish, and the update for the position is given by Equation (24):

$$Y_{i,p}^T = \begin{cases} X_{i,p}^T + a_1(X_{best,p}^T - X_{i,p}^T) \cos \theta, & \text{for } r \leq 0.5, \\ X_{i,p}^T - a_1(X_{best,p}^T - X_{i,p}^T) \sin \theta, & \text{for } r > 0.5, \end{cases} \quad (24)$$

where $Y_{i,p}^T$ and $X_{i,p}^T$ represent the obtained and current positions, respectively, $X_{best,p}^T$ is the best known position in the p -th dimension, r is a random number, and a_1 and θ are defined as Equation (25) and Equation (26):

$$a_1 = (2r - 1)\pi, \quad (25)$$

$$\theta = \frac{\pi}{2} \cdot \frac{T}{T_{max}}, \quad (26)$$

where T is the current iteration and T_{max} is the maximum number of iterations.

- For $D > 5$, the five-dimensional search pattern is used to update only five dimensions at a time. If the updated position exceeds the design variable boundaries, the starfish remains in the previous position as Equation (27):

$$X_{i,p}^{T+1} = \begin{cases} Y_{i,p}^T, & \text{if } l_{b,p} \geq Y_{i,p}^T \leq u_{b,p} l_{b,p}, \\ X_{i,p}^T, & \text{otherwise.} \end{cases} \quad (27)$$

- For optimization problems with $D \leq 5$, a unidimensional search pattern is applied, with only one arm of the starfish moving to explore the environment. The updated position is given by Equation (28):

$$Y_{i,q}^T = E_t X_{i,p}^T + A_1(X_{k_1,p}^T - X_{i,p}^T) + A_2(X_{k_2,p}^T - X_{i,p}^T). \quad (28)$$

where $X_{k_1,p}^T$ and $X_{k_2,p}^T$ are the positions of two randomly selected starfish, and A_1 and A_2 are random numbers between $(-1, 1)$. The energy E_t is computed as Equation (29):

$$E_t = \frac{T_{max} - T}{T_{max}} \cos \theta. \quad (29)$$

If the obtained position is outside the design variable boundaries, the starfish stays in its previous position.

Exploitation Phase

The exploitation phase models the preying and regeneration behaviors of the starfish to refine the search for global solutions. This phase involves two strategies: preying and regeneration.

Preying Phase The distances between the global best position and the positions of other starfish are computed. Two distances are selected randomly to update each starfish's position Equation (30):

$$d_m = (X_{best}^T - X_m^T), \quad m = 1, \dots, 5. \quad (30)$$

The update for each starfish during the preying phase is given by Equation (31):

$$Y_i^T = X_i^T + r_1 d_{m_1} + r_2 d_{m_2}, \quad (31)$$

where r_1 and r_2 are random numbers between $(0,1)$, and d_{m_1} and d_{m_2} are the randomly selected distances.

Regeneration Phase For the last starfish in the population, the position is updated using a slow movement model Equation (32):

$$Y_i^T = \exp\left(-T \times \frac{N}{T_{max}}\right) X_i^T. \quad (32)$$

If the position obtained from either the preying or regeneration phases is outside the design variable boundaries, it is adjusted as Equation (33):

$$X_i^{T+1} = \begin{cases} Y_i^T, & l_b \leq Y_i^T \leq u_b, \\ l_b, & Y_i^T < l_b, \\ u_b, & u_b < Y_i^T. \end{cases} \quad (33)$$

Algorithm 4 Pseudo-code of SFOA algorithm**Input:** Algorithmic parameters (N , $T_{\max}$, and GP)**Output:** The global solution

```

1: Initialize input parameters, problem information, boundary, etc.
2: Initialize positions by Equation (21) and evaluate the fitness values by Equation (23)
3: while  $T < T_{\max}$  do
4:   if  $\text{rand} > GP$  then
5:     Calculate  $\theta$  by Equation (26) and energy  $E_t$  by Equation (29)
6:     for each starfish do
7:       if  $D > 5$  then
8:         Generate five randomly selected  $p$ -dimensions among  $D$ 
9:         for  $j = 1 : 5$  do
10:          Calculate  $a_1$  by Equation (25)
11:          Update the dimension of position in starfish using Equation 24
12:        end for
13:        Check the boundary of position by Equation (27)
14:      else
15:        Generate two random numbers  $A_1$  and  $A_2$ 
16:        Select a random dimension  $p$ 
17:        Update the  $p$ -dimension of position using Equation (28)
18:        Check the boundary of starfish by Equation (33)
19:      end if
20:    end for
21:  else
22:    Generate distances between the best position and starfish
23:    for each starfish do
24:      Generate random number  $r_k$ , and update the position using Equation (31)
25:      if  $i = N$  then
26:        Update the position using Equation (32)
27:      end if
28:      Check the boundary of starfish by Equation (33)
29:    end for
30:  end if
31:  Calculate the fitness values for all starfish
32:  Obtain the global best solution in iteration  $T$  step
33:   $T = T + 1$ 
34: end while
35: Obtain global solution

```

The SFOA continues iterating through these phases, progressively improving the starfish positions and updating the best-so-far solution throughout the optimization process [31]. In the SFOA, GP denotes the exploration control parameter, which regulates the selection between the exploration and exploitation phases. At each iteration, a random number is generated and compared with GP . When $\text{rand} > GP$, the exploration phase is performed; otherwise, the exploitation phase is executed. Thus, GP controls the balance between exploration of the search space and exploitation of promising solutions. The pseudocode for the SFOA is presented in Algorithm 2.

Declarations

Availability of Supporting Data

The complete implementations of two models, along with their corresponding testing procedures, are released via the following GitHub repository: <https://github.com/YasHmi06/Price-Prediction.git>.

Funding

The authors received no specific funding for this research.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Authors' Contributions

Seyedeh Yasamin Hashemi: Conceptualization; Methodology; Data curation; Formal analysis; Model implementation; Software; Writing – original draft; Visualization. **Omid Solaymani Fard:** Conceptualization; Methodology; Validation; Writing – review & editing; Supervision; Theoretical development; Corresponding author.

Artificial Intelligence Statement

Artificial intelligence (AI) tools were used solely for language editing and manuscript refinement, including grammatical correction, sentence restructuring for clarity, and improvement of phrasing and readability. The AI tools were not used to generate scientific ideas, design experiments, perform analyses, interpret results, or develop scientific conclusions. All scientific content, methodological decisions, analyses, interpretations, and conclusions were independently developed and verified by the authors, who retain full responsibility for the final manuscript.

Publisher's Note

The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

References

- [1] Ayitey Junior, M., Appiahene, P., Appiah, O., Bombie, C. N. (2023). "Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis". *Journal of Big Data*, 10(1), 9. <https://doi.org/10.1186/s40537-022-00676-2>
- [2] Bank for International Settlements. (2019). "Triennial Central Bank Survey of foreign exchange and OTC derivatives markets in 2019: Foreign exchange turnover in April 2019". *Bank for International Settlements*. https://www.bis.org/statistics/rpfx19_fx.pdf
- [3] Bank for International Settlements. (2022). "Triennial Central Bank Survey of foreign exchange and over-the-counter (OTC) derivatives markets in 2022". *Bank for International Settlements*. https://www.bis.org/statistics/rpfx22_fx.pdf.
- [4] Cho, K., Van Merriënboer, B., Gulçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). "Learning phrase representations using RNN encoder–decoder for statistical machine translation". *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
- [5] Deng, S., Xiao, C., Zhu, Y., Peng, J., Li, J., Liu, Z. (2023). "High-frequency direction forecasting and simulation trading of the crude oil futures using Ichimoku KinkoHyo and Fuzzy Rough Set". *Expert Systems with Applications*, 215, 119326. <https://doi.org/10.1016/j.eswa.2022.119326>
- [6] Diebold, F. X., Mariano, R. S. (2002). "Comparing predictive accuracy". *Journal of Business & Economic Statistics*, 20(1), 134–144. <https://doi.org/10.1198/073500102753410444>
- [7] Dodge, Y. (Ed.). (2003). "The Oxford dictionary of statistical terms". *Oxford University Press*.
- [8] Elliott, N. (2007). "Ichimoku charts: An introduction to Ichimoku Kinko clouds". *Harri-man House Limited*.

- [9] Gauss, C. F. (1877). "Theoria motus corporum coelestium in sectionibus conicis solem ambientium (Vol. 7)". *Cambridge: Cambridge University Press*. <https://doi.org/10.1017/CB09780511841705>
- [10] Goodfellow, I., Bengio, Y., Courville, A. (2016). "Deep learning". *MIT Press*.
- [11] Gülmez, B. (2023). "Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm". *Expert Systems with Applications*, 227, 120346. <https://doi.org/10.1016/j.eswa.2023.120346>
- [12] Hochreiter, S., Schmidhuber, J. (1997). "Long short-term memory". *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [13] Hoffmann, J., Navarro, O., Kastner, F., Janßen, B., Hubner, M. (2017). "A survey on CNN and RNN implementations". In *PESARO 2017: The Seventh International Conference on Performance, Safety and Robustness in Complex Systems and Applications*.
- [14] Holm, S. (1979). "A simple sequentially rejective multiple test procedure". *Scandinavian journal of statistics*, 65-70.
- [15] Islam, M. S., Hossain, E. (2021). "Foreign exchange currency rate prediction using a GRU-LSTM hybrid network". *Soft Computing Letters*, 3, 100009. <https://doi.org/10.1016/j.socl.2020.100009>
- [16] Jarusek, R., Volna, E., Kotyrba, M. (2022). "FOREX rate prediction improved by Elliott waves patterns based on neural networks". *Neural Networks*, 145, 342-355. <https://doi.org/10.1016/j.neunet.2021.10.024>
- [17] Khaldi, R., El Afia, A., Chiheb, R., Tabik, S. (2023). "What is the best RNN-cell structure to forecast each time series behavior?". *Expert Systems with Applications*, 215, 119140. <https://doi.org/10.1016/j.eswa.2022.119140>
- [18] Khattak, B. H. A., Shafi, I., Rashid, C. H., Safran, M., Alfarhood, S., Ashraf, I. (2024). "Profitability trend prediction in crypto financial markets using Fibonacci technical indicator and hybrid CNN model". *Journal of Big Data*, 11(1), 58. <https://doi.org/10.1186/s40537-024-00908-7>
- [19] Kim, J., Oh, S., Kim, H., Choi, W. (2023). "Tutorial on time series prediction using 1D-CNN and BiLSTM: A case example of peak electricity demand and system marginal price prediction". *Engineering Applications of Artificial Intelligence*, 126, 106817. <https://doi.org/10.1016/j.engappai.2023.106817>

- [20] Künsch, H. R. (1989). "The jackknife and the bootstrap for general stationary observations". In *The Annals of Statistics*, 1217-1241. <https://doi.org/10.1214/aos/1176347265>
- [21] Lawal, A., Rehman, S., Alhems, L. M., Alam, M. M. (2021). "Wind speed prediction using hybrid 1D CNN and BLSTM network". *IEEE Access*, 9, 156672–156679. <https://doi.org/10.1109/ACCESS.2021.3129883>
- [22] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (2002). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- [23] Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2021). "A survey of convolutional neural networks: Analysis, applications, and prospects". *IEEE transactions on neural networks and learning systems*, 33(12), 6999-7019. <https://doi.org/10.1109/TNNLS.2021.3084827>
- [24] Lim, M. A. (2015). "The handbook of technical analysis+ test bank: The practitioner's comprehensive guide to technical analysis". *John Wiley & Sons*.
- [25] Mabrouk, N., Chihab, M., Hachkar, Z., Chihab, Y. (2022). "Intraday trading strategy based on gated recurrent unit and convolutional neural network: Forecasting daily price direction". *International Journal of Advanced Computer Science and Applications*, 13(3). <https://doi.org/10.14569/IJACSA.2022.0130369>
- [26] Makridakis, S., Wheelwright, S. C., Hyndman, R. J. (2008). "Forecasting methods and applications". *John Wiley & Sons*.
- [27] Mirzajani, S., Roohi, M. (2026). "Feedback long short-term memory: A long short-term memory-based framework for multivariate time series prediction in chaotic systems". *Control and Optimization in Applied Mathematics*, 11(1), 199–218. <https://doi.org/10.30473/coam.2025.72730.1312>
- [28] Özorhan, M. O., Toroslu, İ. H., Şehitoğlu, O. T. (2019). "Short-term trend prediction in financial time series data". *Knowledge and Information Systems*, 61(1), 397-429. <https://doi.org/10.1007/s10115-018-1303-x>
- [29] Sevastjanov, P., Kaczmarek, K., Rutkowski, L. (2024). "A multi-model approach to the development of algorithmic trading systems for the Forex market". *Expert Systems with Applications*, 236, 121310. <https://doi.org/10.1016/j.eswa.2023.121310>

- [30] Sokolova, M., Lapalme, G. (2009). “A systematic analysis of performance measures for classification tasks”. *Information Processing & Management*, 45(4), 427-437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [31] Zhong, C., Li, G., Meng, Z., Li, H., Yildiz, A. R., Mirjalili, S. (2025). “Starfish optimization algorithm (SFOA): A bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers”. *Neural Computing and Applications*, 37(5), 3641-3683. <https://doi.org/10.1007/s00521-024-10694-1>

Authors' Bio-sketches

Seyedeh Yasamin Hashemi is a Ph.D. candidate in Applied Mathematics, specializing in Operations Research, at the Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran, where she has been pursuing her doctoral studies since September 2020. Her primary research interests include optimization and deep learning.

Omid Solaymani Fard received his Ph.D. degree in Applied Mathematics from Ferdowsi University of Mashhad, Iran, in 2004. He is currently an Associate Professor of Optimization at the same institution. His primary research interests include optimization, fuzzy systems, and machine learning. Corresponding author. Email: soleimani@ferdowsi.um.ac.ir