

Article

EXPERT: EXchange Rate Prediction Using Encoder Representation from Transformers

Efstratios Bilis ¹, Theophilos Papadimitriou ^{2,*}, Konstantinos Diamantaras ¹ and Konstantinos Goulianas ¹

¹ Department of Information and Electronic Engineering, International Hellenic University, 57400 Thessaloniki, Greece; evstmpil@iee.ihu.gr (E.B.); kdiamant@ihu.gr (K.D.); gouliana@ihu.gr (K.G.)

² Department of Economics, Democritus University of Thrace, 69100 Komotini, Greece

* Correspondence: papadimi@econ.duth.gr

Abstract

This study introduces a Transformer-based forecasting tool termed EXPERT (EXchange rate Prediction using Encoder Representation from Transformers) and applies it to exchange rate forecasting. We developed and trained a Transformer-based forecasting model, then evaluated its performance on nine currency pairs with various characteristics. Finally, we benchmarked its effectiveness against six established forecasting models: Linear Regression, Random Forest, Stochastic Gradient Descent, XGBoost, Bagging Regression, and Long Short-Term Memory. Our dataset covers the period from 1999 to 2022. The models were evaluated for their ability to predict the next day's closing price using three performance metrics. In addition, the EXPERT system was evaluated on its ability to extend forecast horizons and as the core of a trading strategy. The model's robustness was further evaluated using the Multiple Comparisons with the Best (MCB) metric on five dataset samples.

Keywords: exchange rates; time series forecasting; deep learning; Transformers



Academic Editor: Sonia Leva

Received: 6 September 2025

Revised: 23 October 2025

Accepted: 24 October 2025

Published: 29 October 2025

Citation: Bilis, E.; Papadimitriou, T.; Diamantaras, K.; Goulianas, K. EXPERT: EXchange Rate Prediction Using Encoder Representation from Transformers. *Forecasting* **2025**, *7*, 65. <https://doi.org/10.3390/forecast7040065>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The foreign exchange (FOREX) market, a global financial hub, enables continuous 24/5 trade of currencies across participants and time zones, serving as a vital conduit for international trade, investment, and speculation.

The FOREX ecosystem involves a multitude of actors, including multinational corporations engaged in cross-border trade, governments making strategic interventions, central banks shaping monetary policies, financial institutions providing liquidity and market-making services, and individual traders venturing into the online arena. Despite its complexity and volatility, the FOREX market attracts traders and investors through features like short selling and leverage. FOREX allows short selling, which is the act of selling a currency that the agent does not currently own, with the commitment to buy it in the near future.

Exchange rates—reflecting the value of one currency relative to another—are classified by regime (fixed or floating) and by type (nominal or real). Fixed exchange rates, dictated by governments or central banks, remain constant, while floating rates are determined by supply and demand. Nominal exchange rates represent the current market value of a currency pair, while real exchange rates account for inflation. The flexibility of floating rates allows currencies to adapt to changing economic conditions, facilitating trade and investment flows, promoting price stability, and maintaining external balance. The exchange rates of major reserve currencies, such as the US dollar (USD), the euro

(EUR), the Japanese yen (JPY), and the British pound sterling (GBP), hold significant importance in the global economic landscape due to their crucial role in international trade and financial transactions.

Accurate exchange rate forecasts are essential for market participants, namely, traders, investors, businesses, and policymakers. These predictions inform decisions about currency trades, asset allocation, and risk management, ultimately impacting portfolio performance. Businesses and policymakers rely on these models to plan and execute international transactions, manage foreign currency exposure, and mitigate risks associated with currency fluctuations. Additionally, they play a role in formulating effective monetary and fiscal policies aimed at achieving macroeconomic stability, managing inflation, and fostering sustainable economic growth.

Traditionally, exchange rate forecasting has relied on fundamental and technical analyses. Fundamental analysis involves studying economic indicators like interest rates, inflation, GDP growth, trade balances, and even geopolitical developments to understand the underlying factors driving currency movements. Technical analysis, on the other hand, focuses on historical price data, chart patterns, and technical indicators to identify trends and predict future price movements. It is important to note that expert opinions, intuition, and qualitative judgments are sometimes used. However, their success varies considerably.

In recent years, advanced computational techniques such as machine learning (ML) and artificial intelligence (AI) have transformed exchange rate forecasting. These approaches leverage big data analytics, deep learning, and neural networks to process vast amounts of financial data, identify complex patterns, and generate precise predictions. Deep learning models have proven particularly effective, demonstrating superior predictive capabilities compared to traditional methods [1]. They are set apart by their ability to capture nonlinear relationships, temporal dependencies, and high-dimensional features inherent in financial time series data. Classic architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have limitations. For instance, CNN pooling layers disregard crucial part-whole correlations and lose valuable data, while RNNs are prone to gradient vanishing or exploding issues during backpropagation.

Addressing these challenges, Vaswani et al. (2017) [2] introduced the Transformer, a novel deep learning model. This model, originally excelling in natural language processing (NLP) tasks, replaces traditional CNN and RNN frameworks with an attention mechanism. Unlike the sequential structure of RNNs and LSTM, the Transformer's self-attention mechanism can be trained in parallel and requires less complexity to gather global information.

While Transformer architectures have revolutionized NLP tasks, such as machine translation and language modeling, financial institutions are exploring their ability to tackle the complexities of financial time series forecasting. This integration offers significant potential given the challenges inherent in predicting financial market behavior. Additionally, advancements in computing technology and data availability have facilitated the widespread adoption of Transformer-based models by academic researchers seeking an edge in currency trading and investment strategies.

Advancements in technology alone are not a panacea, especially when considering the recent complexities in the FOREX market, such as the fluctuations caused by the COVID-19 pandemic and the Russia–Ukraine conflict. These events posed significant challenges for traders in predicting currency pair movements, as economic factors, sentiment, and geopolitical developments all played a significant role in shaping exchange.

Inspired by the success of Transformers in modeling sequential data in NLP, the same concept is employed in this study to forecast the evolution of exchange rates in the FOREX market. While recent studies have explored Transformer models primarily for trading

strategy back-testing [3–6], our work focuses on applying Transformers directly to next-day closing price prediction in the FOREX market. To the best of our knowledge, this represents the first such application of Transformer architectures for this specific forecasting task in FOREX.

In their analysis, Fisher et al. [5] examine Transformer models with time embeddings for FX-Spot forecasting, comparing results with traditional models like LSTM for major currency pairs (EUR/USD, USD/JPY, GBP/USD) from November 2020 to January 2022. Their method includes both univariate and multivariate models, utilizing historical prices along with technical and fundamental data. Findings reveal that Transformers significantly outperform LSTM. Transformers demonstrated strength in noisy, high-frequency environments, proving effective for complex financial series.

Gradzki & Wojcik [4] focus on high-frequency FOREX trading with Transformers, comparing them to ResNet-LSTM across six currency pairs and five time intervals (60 to 720 min). The study employs a Transformer architecture for forecasting, enhanced by technical analysis for improved accuracy. The findings indicate that Transformers slightly outperform ResNet-LSTM, especially in longer intervals (480, 720 min). However, transaction costs significantly impact performance in shorter intervals (e.g., 60 min), underscoring the necessity for realistic back-testing.

Exploring a Transformer Encoder model for minute-level FOREX trading, ref. [6] specifically focus on EURUSD and GBPUSD. The model integrates Exponential Moving Averages (EMA) with varying smoothing factors to better capture price trends. Trained on data from July 2023, it achieves a cross-entropy loss below 0.2, indicating strong predictive accuracy. However, profitability is limited by high-frequency trading costs, as spreads can negate gains, demonstrating that real-world outcomes are significantly affected by transaction costs.

In a significant contribution, Kantoutsis et al. [3] present the Momentum Transformer, an attention-based deep learning model that outperforms traditional momentum and mean reversion strategies as well as LSTM-based models. By leveraging attention mechanisms, it captures long-term dependencies and adapts to market shifts, such as those seen during the SARS-CoV-2 crisis. Back-testing from 1995 to 2020 reveals superior performance, particularly in recent years and during significant market events. While the hybrid Temporal Fusion Transformer (TFT) performed best overall, pure attention models also demonstrated strong performance. The study suggests an ensemble approach for improved results across asset classes and highlights the model's robustness in commodities trading.

In our approach, we test the forecasting ability of our Transformer-based model, called EXPERT, on nine currency pairs—EUR/USD, AUD/CAD, EUR/AUD, EUR/CAD, GBP/AUD, NZD/USD, USD/JPY, USD/MXN, and BRL/USD—and evaluate it against six widely used forecasting models: the Stochastic Gradient Descent (SGD), the Bagging Regression (BGR), the Extreme Gradient Boosting (XGB), the Random Forest (RF), the Linear Regressor, and the Long Short-Term Memory (LSTM) models.

Each dataset for these nine currency pairs has been individually used in every forecasting model, using the classic training–testing scheme. The training set is utilized to fine-tune the parameters of the model; the performance of the trained models is evaluated on the testing set. All models predict the closing price for the next day. The estimated value is then evaluated with the actual values.

This paper is organized as follows. Section 2 reviews related work, while Section 3 presents the collected dataset. Every aspect of the EXPERT model is analyzed in Section 4. The alternative forecasting models are briefly introduced in Section 5. Section 6 presents the evaluation metrics used. The forecasting performance of the EXPERT model against the competition is presented in Section 7. In the same section, we present the performance

of the EXPERT model for larger forecasting horizons and evaluate its performance using the Diebold–Mariano (DM) test to compare its forecast accuracy with alternative methods, followed by the Multiple Comparisons with the Best method on five samples from our dataset. In Section 8, we evaluate the success of a Transformer-based automatic trading system against other similar systems, and in Section 9, we conclude this paper.

2. Related Work

A systematic review of the existing literature was conducted to provide a comprehensive understanding of machine learning prediction models for exchange rates prediction [7–10].

Islam et al. [10] report improvements from a hybrid GRU–LSTM model that outperforms standalone GRU, LSTM, and simple moving average (SMA) models across multiple metrics, including MSE, RMSE, MAE, and R^2 . Comparisons were made against these benchmarks to demonstrate the efficacy of the proposed model. The models were tested on historical foreign exchange data for four major currency pairs, EUR/USD, GBP/USD, USD/CAD, and USD/CHF, using a dataset that spans 1 January 2017, to 30 June 2020. The hybrid model predicted closing prices for these currency pairs at 10 and 30 min intervals, demonstrating superior predictive capability.

In their study on exchange rate prediction, Panda et al. [9] found that a hybrid GRU–LSTM model effectively predicts future closing prices in the FOREX market. Applied to major currency pairs (EUR/USD, GBP/USD, USD/CAD, USD/CHF), this model outperformed standalone GRU, LSTM, and simple moving average (SMA) models in MSE, RMSE, and MAE for 10 min intervals, and it excelled with GBP/USD and USD/CAD in 30 min intervals. It also achieved a higher R^2 score, indicating a lower prediction risk. Using a dataset of closing prices from 1 January 2017, to 30 June 2020, the model showed strong predictive capabilities, though it struggled during sudden price fluctuations. Future enhancements are planned, including applications to more currency pairs and shorter timeframes.

The findings of [8] emphasize the clear advantages of machine learning algorithms over traditional stochastic models in financial market forecasting. After surveying more than 150 relevant articles, the study demonstrates that machine learning algorithms generally outperform stochastic methods by effectively capturing nonlinear dynamics in financial time series across various asset classes and market geographies. Recurrent neural networks (RNNs) outperform feedforward neural networks and support vector machines, likely due to their ability to capture temporal dependencies.

The paper by Seze et al. [7] reports significant advancements in deep learning (DL) models for financial time series forecasting, showcasing their superiority over traditional machine learning approaches. Long Short-Term Memory (LSTM) networks are favored for their effectiveness in handling time-varying data and capturing temporal dependencies. More than half of the studies surveyed focus on recurrent neural networks (RNNs) for price trend predictions, while deep multilayer perceptrons (DMLPs) are often used for classification tasks. Growing interest in deep reinforcement learning (RL) for algorithmic trading gives us the opportunity to integrate behavioral finance insights.

Fletcher [11] demonstrates that machine learning techniques can be effectively applied to forecast currency movements. Their findings indicate that it is possible to forecast the directional evolution (up, down, or within the bid-ask spread) of the EUR/USD pair between 5 and 200 s into the future, with accuracy rates ranging from 90% to 53%, respectively. Additionally, they have shown that it is feasible to predict price turning points for a basket of currencies in a way that can be profitably exploited.

Goncu [12] applied several machine learning regression methods—Ridge, decision tree, support vector, and Linear Regression—to predict monthly USD/TRY exchange rates.

Key macroeconomic factors, such as domestic money supply, interest rates, and the prior month's exchange rate, are used for prediction. Among the tested models, Ridge regression delivers the most accurate forecasts, with relative errors under 60 basis points. Out-of-sample back-testing over various time periods confirms Ridge's superior performance, suggesting it effectively balances accuracy and overfitting. The model can also support scenario analysis, helping policymakers and investors assess the impact of interest rate changes on exchange rates.

Research by Qi et al. [13] introduces event-driven features to improve FOREX trading predictions by identifying trend changes and retracement points for optimal trade entry. The authors tested deep learning models, including LSTM, BiLSTM, and GRU, against a baseline RNN, with GRU and BiLSTM outperforming the others across various currency pairs. The best model, GRU with 60 time steps for EUR/GBP, achieved an RMSE of 1.50×10^{-3} and a MAPE of 0.12%, surpassing previous studies. These findings show that the proposed models, combined with event-driven features, can provide accurate, low-risk trading strategies.

The development of more advanced models was proposed by Islam & Hosssain [14] when they introduced a network combining a GRU with an LSTM for improved FOREX rate prediction.

Recent studies have increasingly applied advanced deep learning and adaptive learning strategies to improve forecasting accuracy in the FOREX market. Das et al. [15] proposed a deep learning-based framework for trend prediction using multiple LSTM variants, including Vanilla, Stacked, Bidirectional, CNN LSTM, and Conv LSTM, to model short- and long-term price movements of INR-based currency pairs (GBP/INR, AUD/INR, USD/INR). The predicted trends were validated against traditional technical indicators such as ADX, ROC, momentum, CCI, and MACD, demonstrating the reliability of LSTM-based architectures in capturing market dynamics and identifying low-risk trading entry and exit points.

Addressing the limitations of static models, Bousbaa et al. [16] introduced a data stream mining (DSM) approach for financial time series forecasting, integrating online Stochastic Gradient Descent (SGD) with Particle Swarm Optimization (PSO) to adaptively learn from evolving FOREX data. Their model employed sliding windows that adjusted dynamically to changes in data stationarity, enabling it to effectively capture shifting market behaviors. The results showed that this adaptive DSM framework outperformed fixed window methods, achieving higher forecasting accuracy and greater robustness to concept drift.

Extending the application of deep learning models to volatility prediction, Zitis et al. [17] incorporated complexity measures, specifically the Hurst exponent and fuzzy entropy, into RNN, LSTM, and GRU models to enhance FOREX market volatility forecasting. Using intraday data from major currency pairs (EUR/USD, GBP/USD, USD/CAD, and USD/CHF), they found that including these complexity metrics significantly improved predictive accuracy, with LSTM and GRU models outperforming traditional RNNs. The study highlighted the potential of combining complexity-based features with deep architectures to enhance risk assessment and trading decisions.

Zhao et al. [18] evaluated Transformer-based architectures, namely, the original Transformer, Informer, and Temporal Fusion Transformer (TFT), for exchange rate prediction across four NZD currency pairs (NZD/USD, NZD/CNY, NZD/GBP, and NZD/AUD). The TFT achieved the highest predictive accuracy (R^2 up to 0.94) and lowest RMSE and MAE, while the Informer demonstrated faster convergence owing to its sparse attention mechanism. Furthermore, integrating the VIX index into the TFT model further enhanced prediction accuracy. This work underscores the growing effectiveness of Transformer-based models in capturing complex temporal dependencies in exchange rate forecasting.

3. The Dataset

In this study, our objective is twofold: (a) to create a Transformer-based model (EXPERT) for forecasting exchange rates and (b) to test it against a set of well-known forecasting methodologies. To ascertain the overall most accurate forecasting model, we must test them on a rich and diverse dataset that includes exchange rates with different characteristics. Testing our models on multiple pairs helps to mitigate the risk of overfitting to the specific market characteristics present in a single currency pair, increasing the model's reliability and applicability to real-world trading scenarios. The dataset was compiled using the Metatrader application and comprises two major, five minor, and three exotic exchange rates, spanning the period from January 1999 to March 2022 for the weekdays. Each entry in the dataset contains the open, high, low, and close values. In the following, we report the exact dimensions of every exchange rate dataset:

- AUD/CAD: 12 February 2007 to 3 March 2022 (3895 records).
- USD/JPY: 4 January 1999 to 3 March 2022 (5990 records).
- EUR/AUD: 17 June 2004 to 3 March 2022 (4580 records).
- EUR/CAD: 17 June 2004 to 3 March 2022 (4580 records).
- EUR/USD: 4 January 1999 to 3 March 2022 (5990 records).
- GBP/AUD: 21 August 2007 to 3 March 2022 (3760 records).
- NZD/USD: 4 January 1999 to 3 March 2022 (5994 records).
- USD/MXN: 23 July 2007 to 3 March 2022 (3668 records).
- BRL/USD: 17 January 2000 to 21 March 2019 (5000 records).

The dataset was divided into training (68.8%), validation (17.2%), and testing (14%) subsets, with the first 86% of data used for training and validation. The remaining 14%, was used as validation (out-of-sample) data to evaluate the models' ability to generalize to new, unseen data. This data-splitting strategy provided a robust evaluation of the model's performance across both the training and out-of-sample data.

The currency pairs' exchange rates were categorized as major, minor, or exotic based on the European Securities and Markets Authority (ESMA). In this context, the currency pairs in our datasets are classified as follows:

Major currency pairs:

1. EUR/USD (euro/US dollar).
2. USD/JPY (US dollar/Japanese yen).

Minor currency pairs (cross-currency pairs):

3. EUR/AUD (euro/Australian dollar).
4. EUR/CAD (euro/Canadian dollar).
5. AUD/CAD (Australian dollar/Canadian dollar).
6. GBP/AUD (British pound/Australian dollar).
7. NZD/USD (New Zealand dollar/US dollar).

Exotic currency pairs:

8. USD/MXN (US dollar/Mexican peso).
9. BRL/USD (Brazilian real/US dollar).

Major currency pairs are the most widely traded globally. According to ESMA, they include any two of the following: US dollar (USD), euro (EUR), Japanese yen (JPY), British pound (GBP), and Canadian dollar (CAD). All other currencies are considered non-major. Exotic currency pairs involve one major currency and one currency from a smaller or emerging economy; they generally have higher volatility and higher spreads compared to major and minor pairs.

Table 1 provides a summary of the essential descriptive statistics for each currency pair, helping to capture the main characteristics of their price series: minimum and maximum value, mean and standard deviation, first (Q_1) and third (Q_3) quartile, skewness, and kurtosis.

Table 1. Descriptive statistics for the currency pairs.

	Min	Max	Mean	St.D.	Q_1	Q_3	Skewness	Kurtosis
AUD/CAD	0.75	1.07	0.95	0.05	0.92	1.00	−0.46	0.18
EUR/AUD	1.16	2.08	1.54	0.15	1.44	1.64	0.04	0.52
EUR/CAD	1.21	1.72	1.46	0.09	1.40	1.53	−0.17	−0.37
EUR/USD	0.82	1.59	1.19	0.15	1.10	1.31	−0.13	−0.33
GBP/AUD	1.44	2.64	1.81	0.21	1.67	1.91	0.65	0.30
NZD/USD	0.39	0.88	0.66	0.11	0.61	0.74	−0.63	−0.25
USD/JPY	75.81	134.72	107.11	12.51	102.27	116.22	−0.77	0.12
USD/MXN	9.86	25.34	15.93	3.58	12.90	19.13	0.19	−1.29
BRL/USD	0.23	0.65	0.42	0.11	0.33	0.52	0.05	−1.17

For each currency, a dataset was compiled consisting of the open, high, low, and close values (open is the price at the start of the period, close is the price at the end of the period, high is the highest price traded during the period, and low is the lowest price traded during the period). All time series were normalized to the 0–1 range using the classic MinMax normalization. In every case, lagged values of the four time series were employed to forecast the closing price at time instance $t + 1$. The optimal lags for each currency pair were identified through an exhaustive trial-and-error search for lag values up to 20 and can be found in Table 2.

Table 2. Hyperparameter configurations for different currency pairs in the EXPERT model.

Currency Pair	Lag	Optimizer	Batch Size	Head Size	Heads	FF Dim	Blocks	MLP Dropout
AUDCAD	20	ADAM	20	46	60	256	5	0.25
EURUSD	8	ADAMW	18	19	34	512	2	0.48
EURAUD	9	ADAM	20	46	60	256	5	0.25
EURCAD	11	ADAMW	27	105	13	827	6	0.28
GBPAUD	20	ADAM	20	46	60	256	5	0.25
NZDUSD	13	ADAMW	17	121	2	1024	7	0.34
USDJPY	20	ADAM	20	46	60	256	5	0.2
USDMXN	9	ADAM	20	46	60	256	5	0.2
BRLUSD	7	ADAM	20	46	60	256	5	0.2

FF Dim column represents the feedforward dimension.

4. The EXPERT Model

The Transformer model, first introduced by Vaswani et al. [2], revolutionized natural language processing (NLP) by outperforming recurrent neural networks (RNNs) and convolutional neural networks (CNNs) in tasks like machine translation. This architecture's key innovation is the self-attention mechanism, which enables the model to capture long-term dependencies and analyze input sequences more comprehensively.

In time series forecasting, decision-making processes across sectors like finance, retail, and industry often rely on multivariate time series data. Traditionally, statistical models such as Vector Autoregression (VAR) [19] and Autoregressive Moving Average (ARMA) [20] have been used to forecast time series data. Recently, machine learning and particularly deep learning models such as RNNs and CNNs have been explored for this purpose [21,22]. In parallel, the Transformer model was tested as an alternative and promising approach for time series forecasting [23]. Our study lies in the same methodological path.

4.1. Architecture Modifications and Network Structure

The overall architecture for time series forecasting draws inspiration from the Transformer Encoder structure but includes several adjustments, likely based on a combination of good practices in the literature and insights from studies like [23].

Self-attention and layer normalization: Each encoder block starts with layer normalization, a technique that stabilizes the training process by standardizing the input across each layer, ensuring numerical stability and faster convergence. This is followed by multi-head self-attention, a mechanism that enables the model to focus on different parts of the input sequence simultaneously. Unlike traditional models that process sequences step by step, self-attention allows the model to capture diverse temporal patterns by analyzing all positions in the sequence at once. For time series data, the self-attention layers are adapted to capture relationships between time points, focusing on temporal dependencies rather than word-to-word associations, as seen in natural language processing (NLP) models.

Residual connections and feedforward networks: To maintain the flow of important information through the network and address the vanishing gradient problem—where gradients become too small during training, impeding learning—residual connections are employed. These connections add the input of a layer directly to its output, preserving information from earlier layers. After the self-attention step, a feedforward neural network (FFN) processes the output, enabling the model to learn complex, nonlinear relationships within the time series. The FFN often includes convolutional layers that scan over input data and ReLU (Rectified Linear Unit) activations, which introduce nonlinearity and help the model detect intricate temporal patterns. This combination of techniques draws from the original Transformer model by Vaswani et al. [2] and has been adapted for time series analysis [23].

Regularization techniques: To prevent overfitting, where the model performs well on training data but poorly on unseen data, dropout regularization is applied. Dropout randomly “turns off” a fraction of neurons during training, forcing the model to learn more robust features. This regularization is used both after the feedforward layers and within the multilayer perceptron (MLP) used for forecasting. By improving the model’s ability to generalize, these techniques enhance its robustness in handling unseen time series data.

A comprehensive discussion of these terms is provided in Section 4.4.

4.2. Output Processing and Forecasting

Once the input sequence has passed through the encoder blocks, the output is aggregated using global average pooling. This operation condenses the sequence into a fixed-length vector by summarizing information across all positions, making it easier for the model to focus on key features.

The pooled representation is then fed into an MLP for final processing. The MLP consists of a fully connected layer with Exponential Linear Unit (ELU) activation, followed by a linear output layer, which directly regresses the target values, suitable for continuous time series prediction. The linear activation function in the output layer is critical for regression tasks where the goal is to predict real valued outputs.

4.3. EXPERT Unique Features Architecture

The proposed EXPERT model builds upon the core concepts of the Transformer architecture but incorporates several key modifications to address the specific demands of time series forecasting, particularly in financial data. The original Transformer model, designed primarily for natural language processing (NLP) tasks, includes components such as positional encodings, a decoder, and causal masking that are unnecessary for time series forecasting. Below, we outline the unique aspects of our architecture, focusing on how the

EXPERT model diverges from the standard Transformer model and why these changes are necessary for accurate forecasting.

4.3.1. No Positional Encoding

In the original Transformer, positional encoding is applied to account for the lack of inherent order information in input sequences. However, for time series data, where sequential order is inherent, this additional encoding is redundant. Therefore, the EXPERT model omits positional encodings, relying on the inherent structure of the time series to capture temporal relationships. This simplification reduces computational complexity while preserving the time-dependent characteristics of the data.

4.3.2. Encoder-Only Architecture

While the classical Transformer consists of both an encoder and a decoder, the EXPERT model uses an encoder-only structure, as forecasting tasks do not require output sequences to be generated (e.g., in machine translation). The encoder processes the historical time series data, and no decoder is necessary, as the output is a single future prediction rather than a sequence. This architectural choice focuses all learning capacity on extracting meaningful patterns from past data, which is critical for making accurate time series forecasts.

4.3.3. No Masking in Attention Mechanism

In NLP tasks, causal masking is applied in the decoder to ensure the model does not access future tokens when making predictions. However, since time series forecasting only involves predicting future values based on past data, the EXPERT model does not require masking. The attention mechanism is free to focus on any part of the input sequence, optimizing its ability to capture long-range dependencies and interactions within the historical data.

4.3.4. Global Average Pooling for Temporal Feature Aggregation

Our EXPERT model applies global average pooling (GAP) to aggregate the sequence of hidden states generated by the encoder. This aggregation provides a condensed representation of the entire time series, summarizing its overall trend and relevant features. GAP is well suited to time series tasks, as it reduces the sequence into a single feature vector that captures the most salient information for forecasting.

4.3.5. Use of Convolutional Layers in Feedforward Networks

While the original Transformer applies fully connected layers in the feedforward networks, the EXPERT model employs Conv1D layers to capture local temporal dependencies between adjacent time steps. Convolutional layers are more effective in extracting short-term patterns, which are crucial for tasks like financial forecasting where trends and relationships evolve over time. By using Conv1D, the EXPERT model is able to learn finer-grained local structures in the data while still maintaining the benefits of the multihead self-attention mechanism.

4.3.6. Customized Dropout Rates to Mitigate Overfitting

The EXPERT model incorporates custom dropout rates tailored to different layers of the architecture. Specifically, dropout is applied in both the encoder blocks and the multilayer perceptron (MLP) layers. This differentiation helps prevent overfitting, particularly when dealing with highly volatile financial time series data, where overfitting can lead to poor generalization performance. In contrast, the classical Transformer applies uniform dropout across layers, which may not be optimal for time series forecasting.

4.3.7. Data Normalization and Numerical Embedding

In contrast to the word embeddings used in NLP tasks, the EXPERT model applies MinMax scaling to normalize the numerical time series data. This normalization ensures that all input features are on the same scale, which is critical for stabilizing training and improving model performance when forecasting values that vary widely in magnitude. The use of this preprocessing step further highlights the model's adaptation to the specific challenges posed by financial time series data.

These modifications demonstrate that the EXPERT model is uniquely optimized for time series forecasting, particularly for financial applications where patterns, trends, and long-range dependencies must be carefully captured. The encoder-only architecture and adjustments to the attention and feedforward layers enable the model to make accurate predictions of future exchange rates based on historical data.

4.4. The EXPERT Architecture

Initially, the EXPERT model receives input data in the form of historical currency exchange rate values. Each input sequence represents historical exchange rates for a specific currency pair over a period of time. This input data is passed through an embedding layer, converting it into numerical vectors that the model can understand. The embedded input sequences are then passed through multiple Transformer Encoder layers, each consisting of multihead self-attention mechanisms and feedforward neural networks. ("Self-attention" refers to the mechanism that allows the model to weigh the importance of different elements in the input sequence when computing a representation of that sequence. Each element can attend to other elements to capture dependencies and contextual relationships within the sequence. "Multihead" indicates that the self-attention mechanism is executed multiple times (in parallel but independently) with different sets of learned parameters (heads). Each head has access to different representation subspaces, allowing the model to capture diverse patterns or relationships in the data. The results from these multiple heads are then concatenated and combined to form a comprehensive representation of the input sequence.) After processing through these layers, the model generates output sequences.

The model is trained using historical currency financial data, adjusting its parameters to minimize the difference between its predictions and the actual data. Once the model is trained, it attempts to predict the next day's closing price for the exchange rates mentioned in this paper.

The EXPERT architecture consists of the following components (see Figure 1).

4.4.1. Embedding Layer

The embedding layer initiates with data normalization, a technique essential for stabilizing the training process by standardizing the input data. This is achieved using layer normalization, which normalizes the input data as follows:

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sigma + \epsilon}$$

where

$$\mu = \frac{1}{d} \sum_{i=1}^d x_i, \quad \sigma = \sqrt{\frac{1}{d} \sum_{i=1}^d (x_i - \mu)^2}$$

and ϵ is a small constant for numerical stability, and d is the dimension of the input features.

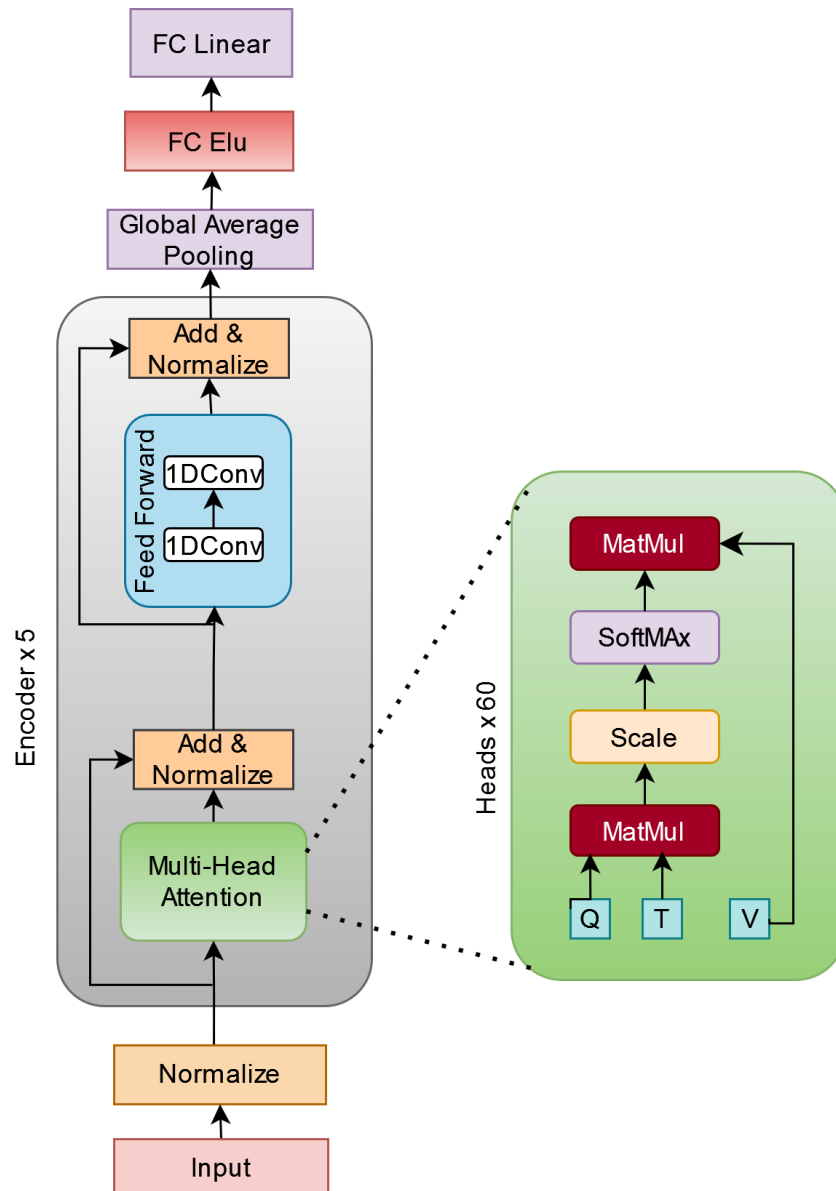


Figure 1. Encoder-based model architecture of EXPERT.

4.4.2. Encoder

The core of the EXPERT model is the encoder, which takes the input time series data and transforms it into a sequence of hidden states. This is performed using a stack of encoder blocks. Each encoder block consists of two sublayers:

- Self-attention layer: This layer allows the model to learn long-range dependencies in the input data. A key part of this is the residual connection, which ensures that the input is passed forward while also allowing the model to learn relationships:

$$\text{Residual}(x) = \text{AttentionOut}(x) + x$$

- Feedforward network: The feedforward network allows the model to learn nonlinear relationships between the input data and the output value. Mathematically, this is computed as

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

where $W_1 \in \mathbb{R}^{d \times f}$, $b_1 \in \mathbb{R}^f$, $W_2 \in \mathbb{R}^{f \times d}$, $b_2 \in \mathbb{R}^d$ and d represents the dimensionality of the input and output of the FFN. It is typically the hidden size of the input sequence after

processing by the encoder. It remains consistent throughout the architecture. f represents the dimensionality of the intermediate layer within the FFN. This is usually larger than d , providing the network with a greater capacity to model complex transformations.

4.4.3. Global Average Pooling

The global average pooling layer takes the sequence of hidden states from the encoder and converts it into a single vector. This vector represents the overall trend of the input time series data. The operation is defined as

$$\text{GlobalAveragePooling1D}(x) = \frac{1}{n} \sum_{i=1}^n x_i$$

where n is the sequence length. The pooling operation compresses the sequence, allowing the model to focus on global trends.

4.4.4. Multilayer Perceptron (MLP)

The MLP takes the output of the global average pooling layer as input and predicts the output value. Each layer of the MLP performs the following transformation:

$$x = \text{ELU}(xW_i + b_i)$$

where $W_i \in \mathbb{R}^{d_i \times m_i}$ and $b_i \in \mathbb{R}^{m_i}$ are the weights and biases, and ELU is the activation function applied at each layer.

4.4.5. Output Layer

The final output layer applies a linear transformation to predict the output value. This is represented by the following equation:

$$\text{output} = xW_{\text{out}} + b_{\text{out}}$$

where $W_{\text{out}} \in \mathbb{R}^{d_{\text{final}} \times 1}$ and $b_{\text{out}} \in \mathbb{R}^1$ are the weights and biases of the output layer.

4.4.6. Encoder-Based Model—Hyperparameter Values

The optimization of the benchmarks in this study was focused on hyperparameter tuning to improve model performance. Key hyperparameters considered for optimization included the number of attention heads, feedforward network dimension, the number of Transformer blocks, learning rate scheduling, and dropout rates. These parameters were chosen due to their significant impact on the performance of Transformer-based models.

The corresponding value ranges for each hyperparameter are delineated as follows:

- Number of attention heads: Tested values ranged from 2 to 60 heads.
- Feedforward network dimension: It varied between 128 and 1024 units.
- Number of Transformer blocks: The number of Transformer layers varied between two and seven blocks.
- Learning rate: A custom learning rate scheduler was implemented to progressively increase the learning rate during the initial warm-up period (30 epochs), followed by gradual decay over 100 epochs. The base learning rate was set at 1×10^{-4} , with a minimum learning rate of 1×10^{-5} .
- Dropout rates: Applied to both the Transformer layers and the fully connected layers, dropout rates ranged between 0.1 and 0.48 to prevent overfitting.

The search for optimal hyperparameters was carried out using a combination of grid search and manual tuning, informed by early experimental results and prior knowledge. Grid search was employed for discrete parameters such as the number of attention heads

and Transformer blocks, while a more manual approach was applied for parameters like learning rate and dropout, as these often required finer control during training iterations. To find the most effective lag length, we gradually increased the number of lags from 0 to 30, noticing that performance improved up to a certain point. Beyond that, adding more lags made the results worse.

The implementation leveraged the TensorFlow and Keras libraries for deep learning, with specific reliance on Keras's Sequential API and the MultiHeadAttention and Conv1D layers for constructing EXPERT model architecture. The MinMaxScaler from scikit-learn was used for feature scaling, and early stopping with learning rate scheduling was implemented using Keras callbacks to prevent overfitting and optimize the learning process.

Table 2 presents the hyperparameter configurations used for different currency pairs in our EXPERT models. While the general structure of the model remains consistent across different setups, key parameters such as the number of attention heads, feedforward dimension, and batch size vary depending on the specific dataset. The proposed architecture consists of multiple stacked Transformer blocks, each incorporating a self-attention mechanism with varying head sizes and feedforward layers. The MLP layer contains 256 units for all currency pairs. To prevent overfitting, dropout is applied to both the MLP layers and the encoder blocks, with an MLP dropout rate varying per experiment, while the encoder dropout remains fixed at 0.1 for all cases. A global average pooling layer aggregates sequence-level representations, which are subsequently passed through fully connected layers to generate the final forecasting output. The model is trained using adaptive optimizers, such as ADAM and ADAMW, with batch sizes optimized for each currency pair to ensure robust performance.

4.5. Rationale for Choosing the EXPERT Model

Our model is designed to forecast currency exchange rates by capturing both broad long-term trends and short-term fluctuations that shape financial data. A key component is the self-attention mechanism, which enables the model to analyze the entire historical data sequence, identifying how past events may continue to influence current market behavior. At the same time, the Conv1D layers focus on local details, such as the daily price movements typical in exchange rate series. Next, the global average pooling layer summarizes the extracted features into a concise representation that reflects the overall data direction. Lastly, custom dropout rates are applied during training to enhance the model's adaptability and robustness when encountering new, unseen data.

5. Alternative Forecasting Models

5.1. Linear Regression

The Linear Regression model is one of the most fundamental and widely utilized statistical tools for predictive analysis. It operates by establishing a linear relationship between input features and the target variable. Essentially, the model fits a straight line to the data points such that the sum of the squared deviations between the observed and the predicted values is minimized. Linear Regression is particularly advantageous for revealing the relationship between two continuous variables and making predictions based on this correlation.

Notwithstanding its simplicity, Linear Regression is frequently the preferred option for regression tasks. The model's analytical form is particularly suited for economic and financial interpretation and for deriving policy implications. Its low computational cost renders it ideal for large datasets or systems with limited computing power, with results being immediate. The principal limitation of Linear Regression is its assumption of a

linear relationship, which may not precisely encapsulate complex real-world phenomena. Moreover, outliers can significantly affect the model's performance.

5.2. Random Forest

Random Forest, introduced by Breiman [24], is a versatile and widely used machine learning framework that combines multiple decision trees to improve accuracy. By training on randomized subsets of data and incorporating stochasticity at each decision node, it overcomes the limitations inherent in solitary decision trees and mitigates overfitting. This adaptable model is effective for both classification and regression tasks, rendering it a favored option across diverse machine learning applications. During the implementation of the Random Forest Regressor model, the emphasis is placed on optimizing two crucial parameters: maximum depth and minimum sample split. In our experiments, we examined maximum depths reaching up to 50 and minimum sample split values of 5 or 10. To ascertain the optimal parameter combination (maximum depth, minimum sample split), grid search and 5-fold cross validation were employed.

The main concept of Random Forest, which involves integrating numerous regressors into a unified system, renders it an ideal and resilient choice capable of effectively managing noise and outliers. Typically, the model averts overfitting and delivers precise predictions. A notable disadvantage of Random Forest is its inability to provide an interpretable model representation. Moreover, as the number of regressors within the forest escalates, the computational cost correspondingly increases at a rapid pace.

5.3. Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) Regressor is a widely used optimization algorithm employed in the training of Linear Regression models [25]. Its operational mechanism involves the iterative update of model parameters in a direction that reduces the error between predicted and actual values, all while employing a subset of the training data in each iteration. This attribute renders it computationally efficient and well suited to large datasets. The algorithm based on gradient descent possesses the ability to traverse local minima, facilitating the attainment of a global minimum.

The SGD Regressor is exceptionally useful for large-scale and sparse datasets, providing adaptability concerning the choice of the loss function and regularization methods, thereby establishing itself as a good instrument for regression tasks in the field of machine learning. Conversely, SGD exhibits suboptimal performance in noisy environments and demands meticulous adjustment of hyperparameters, along with a pronounced sensitivity to feature scaling.

5.4. XGBoost

The XGBoost Regressor, introduced by Chen & Guestrin [26], is a sophisticated enhancement of the gradient boosting algorithm, explicitly engineered for optimal speed and performance. XGBoost, which stands for eXtreme Gradient Boosting, has achieved significant acclaim within machine learning competitions and practical applications, attributed to its exceptional accuracy and efficiency. The model functions through the sequential augmentation of predictors while minimizing errors via gradient descent optimization. The XGBoost Regressor offers numerous benefits, such as its capacity for handling missing data, implementing regularization, and facilitating parallel processing, thereby making it an ideal option for regression tasks where precision and computational speed are paramount. However, it should be noted that the model performs poorly with sparse and unstructured data.

5.5. Bagging Regression

The Bagging Regression model, introduced by Breiman [27], abbreviated from Bootstrap Aggregating, represents an ensemble learning approach designed to enhance the stability and accuracy of regression models. This technique operates by training multiple base regressors on random subsets of the training data, with replacement. The potential base regressors include a variety of regression algorithms, such as decision trees, support vector machines, or Linear Regression. During the prediction phase, the Bagging Regression model synthesizes the predictions of all base regressors, typically by averaging, to arrive at the final output. By mitigating variance and reducing overfitting, the Bagging Regression method significantly enhances overall predictive performance, thus establishing itself as a good method for regression within the field of machine learning. In our study, a specific application of the Bagging Regressor was implemented, employing 100 base regressors as part of the ensemble learning process.

Bagging regression is a low-variance approach that yields robust models. It is capable of managing high-dimensional datasets and mitigating overfitting. The training process can be executed through parallel processing, which reduces computational time; nonetheless, the computational cost can be substantial, as it correlates with the number of constituent models. The effectiveness of the model is enhanced when the base models exhibit diversity; however, this diversity obstructs the possibility of producing models that are easily interpretable.

5.6. Long Short-Term Memory

Long Short-Term Memory (LSTM) constitutes a subset of recurrent neural network (RNN) architectures that are specifically designed to address the vanishing gradient problem associated with traditional RNNs and which are capable of capturing long-term dependencies within sequential data. The model was introduced by Hochreiter & Schmidhuber [28]. In contrast to standard RNNs, LSTM networks exhibit a more elaborate architecture, marked by a persistent cell state that spans the entire sequence, and three gating mechanisms: the input gate, the forget gate, and the output gate. The input gate manages the access of new information into the cell state, the forget gate determines which information should be eliminated from the cell state, and the output gate governs the information to be emitted based on the cell state. This sophisticated gating mechanism endows LSTM networks with the capacity to effectively maintain and utilize long-term dependencies within sequential data, thus making them highly applicable for an array of tasks including time series forecasting, natural language processing, and speech recognition.

The primary benefit of LSTM networks lies in their capacity for long-term dependency, enabling the model to retain information throughout the training phase over extended durations. Conversely, its complexity exceeds that of preceding models, rendering it susceptible to overfitting if not accurately trained and validated.

6. Evaluation Metrics

To assess the performance of the forecasting models created for this paper, we used the Mean Absolute Percentage Error (MAPE), the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) metrics.

MAPE is the average of the absolute percentage differences between the actual and predicted values, and it is calculated as follows:

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

The MAPE is easy to understand because it expresses the accuracy of the model as a simple percentage. In addition, MAPE is scale-invariant, making it optimal for comparing the different exchange rates in our dataset. The main shortcoming of the MAPE metric is that it is sensitive to values close to zero.

MAE is the average of the absolute errors (the differences between the actual and the predicted values), and it is calculated as follows:

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

MAE is easy to implement and understand. However, MAE is not scale-invariant and punishes large errors more than small ones, making it sensitive to outliers.

MSE is the average of the squared differences between the actual and predicted values. It is calculated as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

7. Model's Performance

7.1. Next-Day Forecasting

This section evaluates the forecasting performance of the proposed EXPERT model against six alternatives—Linear Regression, Random Forest, SGD, XGB, Bagging Regression, and LSTM—across nine exchange rates—EUR/USD, AUD/CAD, GBP/AUD, NZD/USD, USD/JPY, EUR/AUD, EUR/CAD, USD/MXN, and BRL/USD—covering the dynamics of various currency pairs on the next-day forecasting.

Table 3 showcases the efficacy of each forecasting model for every currency exchange rate. The bold values show the optimal model, and the underlined values show the second best one.

The random walk model, often referred to as a naïve forecast, assumes that the best prediction for the next time step is simply the current observed value.

Analysis of the results yields several conclusions: (a) all methodologies demonstrate high levels of accuracy in their predictions, with MAPE values consistently below 1%, (b) the MSE values align well with MAPE, enhancing confidence in the conclusions (the BRL/USD is the only exception), and (c) the proposed Transformer-based EXPERT model consistently outperformed all other methodologies (with the exception of the BRL/USD when measured using MAE). The forecasting accuracy of EXPERT as counted by the MAPE ranged from 0.469% in the case of the USD/MXN to 0.275% in the case of USD/JPY.

The simplistic random walk achieved the second best accuracy in most cases, with the exception of USD/MXN and BRL/USD.

Figures 2 and 3 present a series of scatter plots comparing actual versus forecasted prices, offering a comprehensive visual assessment of the model's predictive performance across different scenarios. These plots illustrate the degree of correlation between observed and predicted values, highlighting trends and potential discrepancies in the forecasts.

Table 3. Forecasting performance of each model.

Model	EUR/USD			AUD/CAD			EUR/AUD		
	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE
RW	<u>0.298</u>	<u>0.342</u>	<u>0.002</u>	<u>0.334</u>	<u>0.309</u>	<u>0.002</u>	<u>0.375</u>	<u>0.611</u>	<u>0.007</u>
LIN	0.429	0.513	0.005	0.435	0.415	0.003	0.466	0.726	0.012
SGD	0.686	0.817	0.011	0.798	0.756	0.010	1.064	1.663	0.041
XGB	0.468	0.561	0.006	0.453	0.433	0.003	0.475	0.736	0.010
BGR	0.465	0.555	0.006	0.443	0.422	0.003	0.479	0.745	0.011
RF	0.466	0.555	0.006	0.440	0.420	0.003	0.473	0.735	0.011
LSTM	0.322	0.369	0.003	0.398	0.368	0.003	0.398	0.650	0.008
EXPERT	0.298	0.341	0.002	0.328	0.304	0.002	0.362	0.576	0.005

Model	EUR/CAD			GBP/AUD			NZD/USD		
	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE
RW	<u>0.342</u>	<u>0.513</u>	<u>0.005</u>	<u>0.348</u>	<u>0.643</u>	<u>0.007</u>	<u>0.450</u>	<u>0.301</u>	<u>0.002</u>
LIN	0.433	0.632	0.007	0.522	0.965	0.020	0.591	0.390	0.003
SGD	0.547	0.796	0.011	0.726	1.332	0.039	0.847	0.559	0.006
XGB	0.439	0.640	0.007	0.530	0.984	0.022	0.646	0.424	0.003
BGR	0.438	0.638	0.007	0.528	0.979	0.023	0.621	0.411	0.003
RF	0.435	0.634	0.007	0.529	0.980	0.023	0.617	0.409	0.003
LSTM	0.347	0.520	0.005	0.387	0.717	0.009	0.504	0.336	0.002
EXPERT	0.342	0.512	0.005	0.337	0.616	0.007	0.449	0.301	0.001

Model	USD/JPY			USD/MXN			BRL/USD		
	MAPE	MAE	MSE	MAPE	MAE	MSE	MAPE	MAE	MSE
RW	<u>0.288</u>	<u>31.345</u>	<u>21.401</u>	0.667	14.384	4.157	0.409	0.210	0.084
LIN	0.466	49.710	45.646	0.653	<u>10.594</u>	2.820	<u>0.325</u>	0.131	0.059
SGD	0.530	56.492	59.655	0.912	15.024	5.082	0.924	0.386	0.283
XGB	0.494	52.541	50.992	0.690	11.159	3.066	0.424	0.170	0.073
BGR	0.492	52.588	51.097	0.661	10.680	2.940	0.398	0.161	0.075
RF	0.491	52.446	50.999	0.659	10.675	2.964	0.393	<u>0.158</u>	0.074
LSTM	0.315	34.30	24.16	<u>0.574</u>	11.74	<u>2.450</u>	0.327	0.182	<u>0.056</u>
EXPERT	0.275	29.83	15.80	0.469	9.566	1.552	0.310	0.173	0.051

MAPE values are percentages. Bold denotes the best value. Underline denotes the second best value.

7.2. Diebold–Mariano Statistical Test

To assess whether the predictive accuracy of the proposed machine learning models differs significantly from that of the benchmark (*EXPERT*) model, we employ the Diebold–Mariano (DM) test for equal forecast accuracy. Let $e_{1,t}$ and $e_{2,t}$ denote the forecast errors from two competing models for $t = 1, \dots, n$. The DM test evaluates the null hypothesis $H_0 : E[d_t] = 0$, where

$$d_t = L(e_{1,t}) - L(e_{2,t})$$

is the difference in forecast losses under a chosen loss function $L(\cdot)$. In this study, the loss function is defined as the Mean Squared Error (MSE). This formulation emphasizes the magnitude of large forecast deviations, making the test more sensitive to substantial prediction errors and providing a scale-dependent evaluation of predictive accuracy.

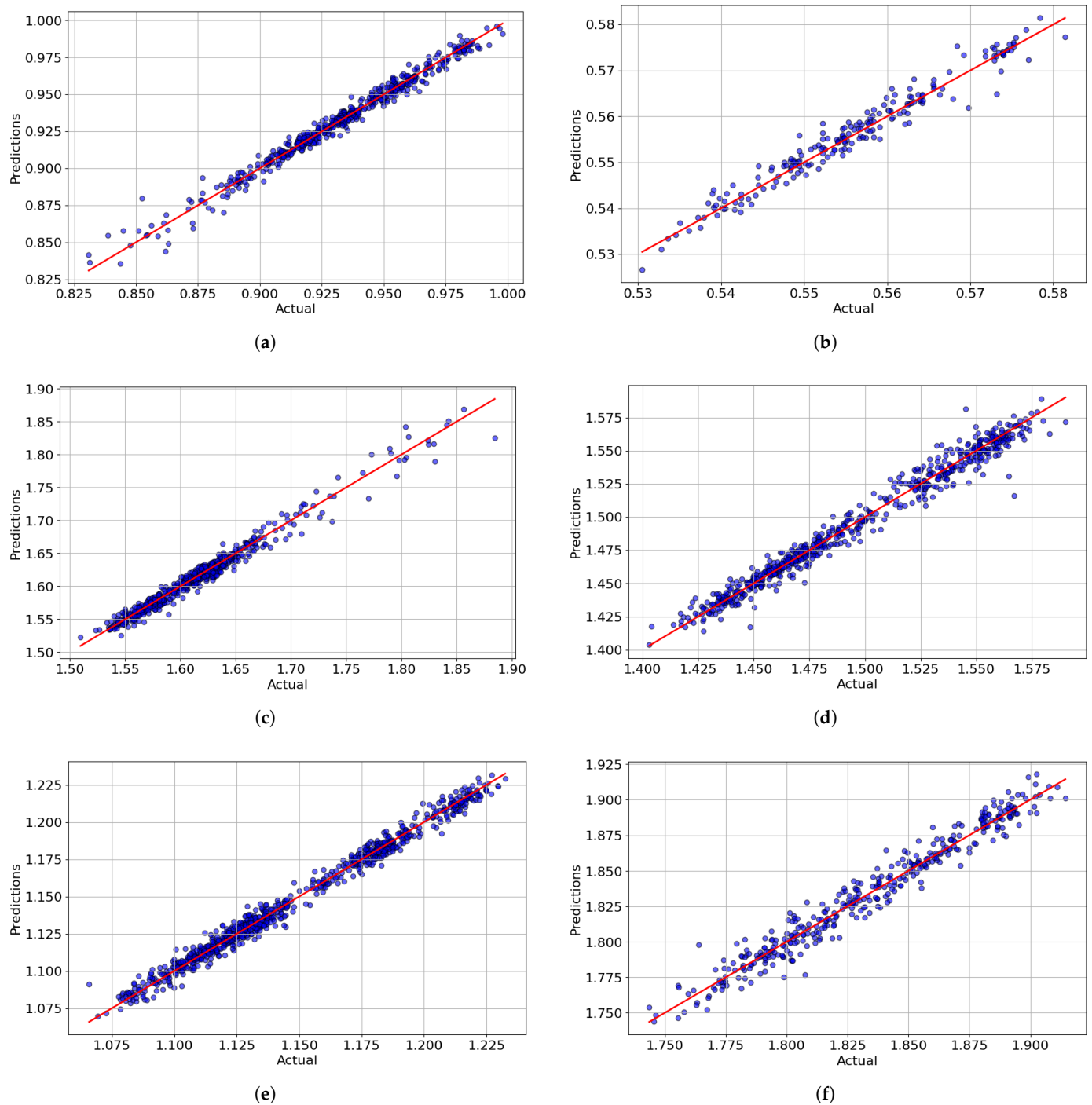


Figure 2. Performance: The blue dots represent individual observations (actual vs. predicted prices), and the red line represents the ideal prediction line ($y = x$), where predicted and actual values are equal. Points lying close to the red line indicate higher prediction accuracy. of the EXPERT model in predicting out-of-sample currency prices (actual vs. predicted) (part A). (a) AUD/CAD scatter plot of EXPERT predictions vs. actual prices; (b) BRL/USD scatter plot of EXPERT predictions vs. actual prices; (c) EUR/AUD scatter plot of EXPERT predictions vs. actual prices; (d) EUR/CAD scatter plot of EXPERT predictions vs. actual prices; (e) EUR/USD scatter plot of EXPERT predictions vs. actual prices; (f) GBP/AUD scatter plot of EXPERT predictions vs. actual prices.

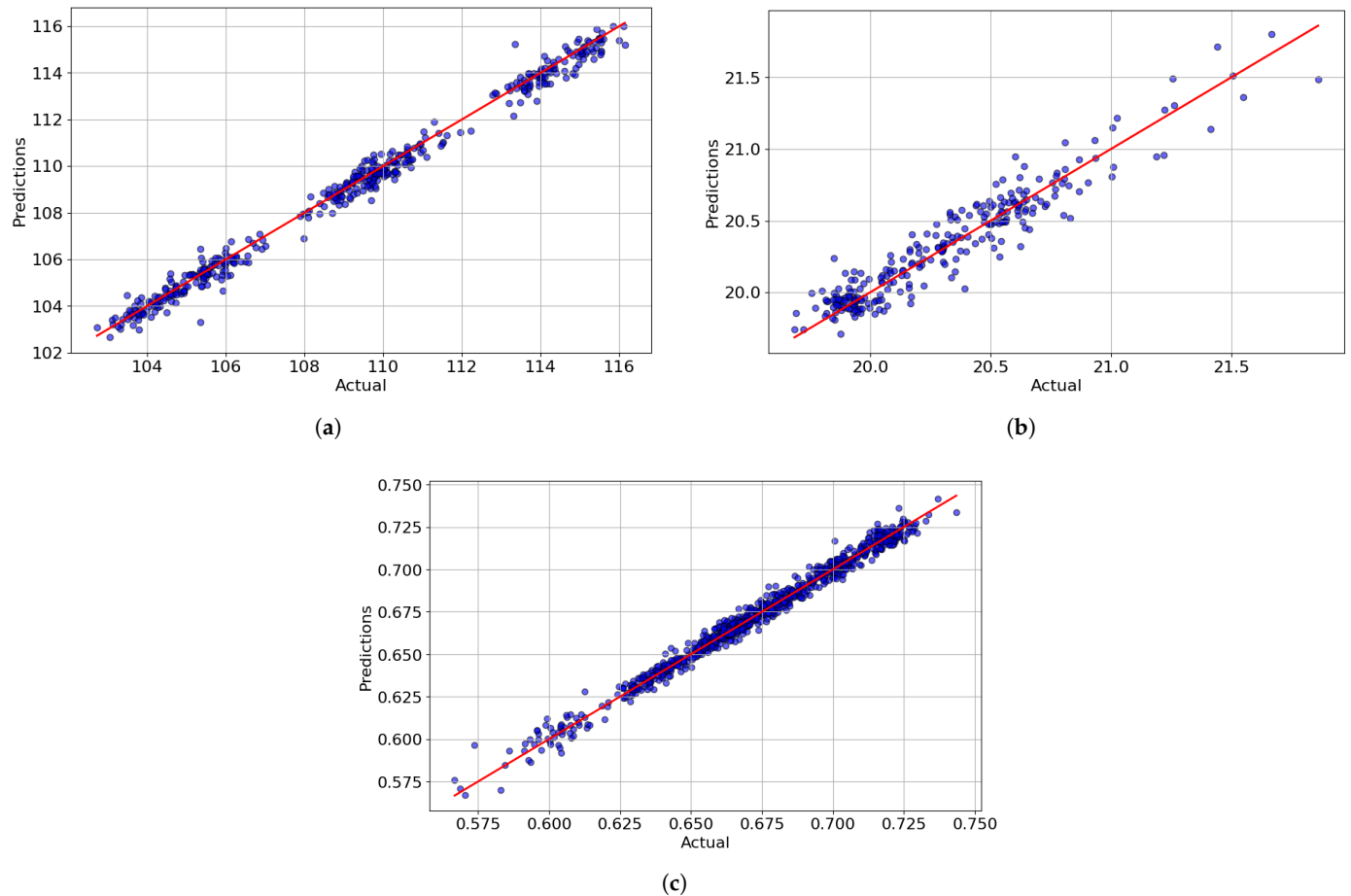


Figure 3. Performance: The blue dots represent individual observations (actual vs. predicted prices), and the red line represents the ideal prediction line ($y = x$), where predicted and actual values are equal. Points lying close to the red line indicate higher prediction accuracy. of the EXPERT model in predicting out-of-sample currency prices (actual vs. predicted) (part B). (a) USD/JPY scatter plot of EXPERT predictions vs. actual prices; (b) USD/MXN scatter plot of EXPERT predictions vs. actual prices; (c) NZD/USD scatter plot of EXPERT predictions vs. actual prices.

The sample mean of the loss differential is given by

$$\bar{d} = \frac{1}{n} \sum_{t=1}^n d_t,$$

which serves as the numerator of the test statistic. The variance of $\bar{d}$ is estimated using the contemporaneous and autocovariance terms of d_t up to lag $h - 1$, where $h = 1$ corresponds to a one-step-ahead forecast horizon. The Diebold–Mariano statistic is computed as

$$DM = \frac{\bar{d}}{\sqrt{\widehat{\text{Var}}(\bar{d})}} = \frac{\bar{d}}{\sqrt{(\gamma_0 + 2 \sum_{k=1}^{h-1} \gamma_k) / n}},$$

where γ_k denotes the sample autocovariance of the loss differential at lag k . Under the null hypothesis of equal predictive accuracy and for large n , the DM statistic follows an asymptotic Student- t distribution with $n - 1$ degrees of freedom. Two-tailed p -values are reported, with statistical significance evaluated at the 5% level. A positive DM statistic indicates that the benchmark EXPERT model achieves lower mean squared forecast errors

and therefore superior predictive accuracy relative to the alternative model, whereas a negative statistic implies the opposite.

Across most of the currency pairs and model comparisons, the Diebold–Mariano statistics were statistically significant at the 5% level (see Table 4), indicating that the EXPERT model consistently achieved superior predictive accuracy relative to all benchmark machine learning models, including Linear, SGD, XGB, BGR, Random Forest, and LSTM. Only in a few isolated cases (e.g., LSTM for EUR/CAD and Linear or LSTM for BRL/USD) was the difference not statistically significant. Overall, the results demonstrate a clear and robust dominance of the EXPERT model in terms of forecast precision across the evaluated exchange rates.

Table 4. Diebold–Mariano test results comparing each model to the EXPERT forecasts (MSE).

Model	EUR/USD			AUD/CAD			EUR/AUD		
	DM Stat.	<i>p</i> -Value	Interpretation	DM Stat.	<i>p</i> -Value	Interpretation	DM Stat.	<i>p</i> -Value	Interpretation
LIN	4.228	$p < 0.05$	Significant	3.534	$p < 0.05$	Significant	4.028	$p < 0.05$	Significant
SGD	4.506	$p < 0.05$	Significant	3.812	$p < 0.05$	Significant	4.472	$p < 0.05$	Significant
XGB	4.331	$p < 0.05$	Significant	3.601	$p < 0.05$	Significant	3.837	$p < 0.05$	Significant
BGR	4.384	$p < 0.05$	Significant	3.723	$p < 0.05$	Significant	3.951	$p < 0.05$	Significant
RF	4.311	$p < 0.05$	Significant	3.578	$p < 0.05$	Significant	3.922	$p < 0.05$	Significant
LSTM	3.427	$p < 0.05$	Significant	3.442	$p < 0.05$	Significant	3.179	$p < 0.05$	Significant
Model	EUR/CAD			GBP/AUD			NZD/USD		
	DM Stat.	<i>p</i> -Value	Interpretation	DM Stat.	<i>p</i> -Value	Interpretation	DM Stat.	<i>p</i> -Value	Interpretation
LIN	3.486	$p < 0.05$	Significant	3.745	$p < 0.05$	Significant	3.841	$p < 0.05$	Significant
SGD	4.001	$p < 0.05$	Significant	4.012	$p < 0.05$	Significant	4.222	$p < 0.05$	Significant
XGB	3.524	$p < 0.05$	Significant	3.609	$p < 0.05$	Significant	3.921	$p < 0.05$	Significant
BGR	3.637	$p < 0.05$	Significant	3.701	$p < 0.05$	Significant	3.982	$p < 0.05$	Significant
RF	3.573	$p < 0.05$	Significant	3.687	$p < 0.05$	Significant	3.945	$p < 0.05$	Significant
LSTM	0.120	0.990	No sig. diff.	2.853	$p < 0.05$	Significant	3.239	$p < 0.05$	Significant
Model	USD/JPY			USD/MXN			BRL/USD		
	DM Stat.	<i>p</i> -Value	Interpretation	DM Stat.	<i>p</i> -Value	Interpretation	DM Stat.	<i>p</i> -Value	Interpretation
LIN	4.823	$p < 0.05$	Significant	3.487	$p < 0.05$	Significant	1.005	0.315	No sig. diff.
SGD	5.017	$p < 0.05$	Significant	4.217	$p < 0.05$	Significant	2.803	$p < 0.05$	Significant
XGB	4.911	$p < 0.05$	Significant	3.653	$p < 0.05$	Significant	2.513	$p < 0.05$	Significant
BGR	4.876	$p < 0.05$	Significant	3.782	$p < 0.05$	Significant	2.593	$p < 0.05$	Significant
RF	4.851	$p < 0.05$	Significant	3.872	$p < 0.05$	Significant	2.507	$p < 0.05$	Significant
LSTM	3.512	$p < 0.05$	Significant	3.214	$p < 0.05$	Significant	1.004	0.315	No sig. diff.

Note: *p*-values < 0.05 indicate EXPERT outperforms the model at the 5% level. “No sig. diff.” indicates differences are not statistically significant.

7.3. Longer Forecasting Horizon

To further assess the proposed EXPERT methodology, we tested its predicting power at multiple forecasting horizons ranging from two days up to fourteen days ahead for the EUR/USD currency pair.

7.3.1. Static Forecasting

Initially, a static forecasting scenario was examined wherein the independent variables were restricted to historical values up to the time instance t , while the target values remained $t + 1, t + 2, t + 3, \dots, t + 14$. For each forecasting horizon, the EXPERT model was retrained following the identical procedure applied to the next-day forecast. For instance, when evaluating a forecasting horizon of +3, data up to time instance t were utilized during model training to predict the value observed at time instance $t + 3$.

7.3.2. Dynamic Forecasting

In the second scenario, we employed the next-day forecasting model in a recursive manner to dynamically extend the forecasting horizon. In this approach, the independent

variables were confined to historical values up to instance t . However, predicted values obtained in $t + 1$ were utilized to make forecasts for $t + 2$, and similarly, predicted values in $t + 1, t + 2$ were used to anticipate $t + 3$, and so forth.

Table 5 shows the performance of the static and the dynamic forecasting scenarios for various horizons.

Table 5. Static and dynamic forecasting performance for various horizons.

Forecasting Horizon	Static Forecasting MAPE	Dynamic Forecasting MAPE
1	0.298	0.298
2	1.674	0.975
3	1.989	1.060
4	2.065	1.255
5	2.153	1.664
6	2.200	1.875
7	2.282	1.996
8	2.406	2.229
9	2.448	2.336
10	2.731	2.395
11	3.008	2.477
12	3.178	2.861
13	3.336	3.220
14	3.547	3.501

MAPE is counted as a percentage.

As expected, accuracy decreases with longer forecasting horizons. However, the error remains acceptable even for the fourteen-day-ahead forecast at just 3.5%. The performance of the dynamic forecast is more accurate than the static forecast for every forecasting horizon, yielding very close to the static forecast after 14 days.

7.4. Multiple Comparisons with the Best Evaluation

Hsu's MCB method [29] is a multiple comparison approach to identify factor levels that are best, insignificantly different, or significantly different from the best (the best is defined as either the highest or lowest mean). When employed with a trained model, it provides precise analysis of level mean differences. It constructs a confidence interval for the difference between each level mean and the best among others.

In this section, we aim to evaluate the performance reliability of the EXPERT model across distinct parts of the EUR/USD dataset. To achieve this, five discrete samples, each comprising 20 data points, were extracted. These samples were chosen to reflect different market conditions, allowing the evaluator to determine how well the EXPERT model performs under varying market scenarios:

- Sample 1: 28 December 2018, to 24 January 2019.
- Sample 2: 2 May 2019, to 29 May 2019.
- Sample 3: 6 February 2020, to 4 March 2020.
- Sample 4: 3 February 2021, to 2 March 2021.
- Sample 5: 5 January 2022, to 1 February 2022.

The evaluation of the EXPERT model's performance was conducted using the MAE between the predicted and the actual values:

$$\text{MAE} = \frac{1}{m} \sum_{i=0}^m |y_{k+i} - \hat{y}_{k+i}| \quad (4)$$

where m is the sample size (in our case $m = 20$) and k is the index of the first observation in the current sample.

To determine the statistical significance of performance differences across samples, Hsu's Multiple Comparisons with the Best (MCB) method was applied with an alpha value of 0.05. In our context, the "best" refers to the sample with the smallest mean, against which other samples are compared.

The differences between the mean of each period and the mean of the "best" period are used to calculate confidence intervals for each period. These intervals determine if the difference between each group and the best group is statistically significant or not according to the following rules shown in Table 6.

Table 6. Evaluation criteria for Hsu's Multiple Comparisons with the Best.

Smallest Is Best	
Lower < 0 < Upper	No difference from best
Lower = 0, Upper > 0	Worse than best
Lower < 0, Upper = 0	Better than other groups

In our case, the smallest mean (the best case) is identified in the second sample; results are shown in Table 7.

Table 7. D-TEST (min) evaluation results.

Group	Mean	Center	Lower	Upper	Evaluation
Sample 1	0.0045	0.0028	−0.0002	0.0058	No difference from best
Sample 2	0.0017	−0.0018	−0.0080	0.0043	No difference from best
Sample 3	0.0029	0.0012	−0.0050	0.0074	No difference from best
Sample 4	0.0034	0.0017	−0.0045	0.0079	No difference from best
Sample 5	0.0040	0.0023	−0.0039	0.0085	No difference from best

The MCB analysis indicates that the proposed model performs comparably to the best sample across all subsets, demonstrating consistent and reliable results. This is a hint of consistency and uniformity in the results, underscoring the reliability of our findings. We can suggest that the EXPERT model will yield accurate, reliable, and equivalent outcomes across diverse scenarios.

8. Trading Scenario

To assess the practical applicability of the proposed model, we used it in the core of a trading strategy, which was subsequently tested on the EUR/USD currency pair from December 2018 to March 2022. The strategy driven by the model operates by a straightforward principle: it initiates a purchase when the forecasted price exceeds the current market price and initiates a sale when the forecasted price is lower than the market price. Such model-driven strategies are prevalent in algorithmic trading and quantitative finance due to their simplicity and efficacy in capturing short-term market trends. Prior research has shown that machine learning-based forecasts can substantially enhance decision making in trading strategies [7,30]. The EXPERT model-based trading strategy was compared against a simplistic buy-and-hold strategy and random strategy. The buy-and-hold approach involves investing the initial capital into EUR/USD at the commencement and maintaining this investment without executing further trades. Its performance reflects the cumulative return that would occur if the investor does not trade. The random strategy opens orders randomly without any criteria.

Step 1: Initial conditions:

- The initial capital allocation (initial_cash) is set at USD 10,000, with no initial position to the EUR/USD currency pair (i.e., the number of contracts held is zero).
- Each iteration (trading day) evaluates whether the model's prediction suggests a buy, sell, or hold action.

Step 2: Trading rules:

- Buy signal: The strategy generates a buy signal when there is available cash in the investing portfolio and the model's predicted price for the currency pair on a given day exceeds the previous day's closing price. In this case, all available cash is allocated to purchase contracts.
- Sell signal: A sell signal occurs when the portfolio contains currency contracts and the predicted price is less than the previous day's closing price. The strategy responds by selling all held contracts, converting these positions back to cash. The return on investment is calculated by determining the relative difference in the exchange rate between the purchase price and the sale price, which generates a profit or loss.
- Hold signal: The strategy implicitly holds its current position in two scenarios:
 - (a) After buying: Once contracts are purchased, the strategy continues to hold them until a sell signal is triggered regardless of further predicted price increases. There is no incremental buying or position scaling.
 - (b) With cash: If no buy signal is triggered (i.e., the prediction indicates a price decrease or no significant change), the strategy holds cash and waits for a favorable buy signal.

This decision-making framework reflects a simple, rule-based “all in, all out” approach, where the entire capital is either fully invested in contracts or held entirely in cash, depending on the model's prediction. Such an approach aligns with similar strategies discussed in quantitative finance and algorithmic trading research [31,32].

Step 3: Portfolio value evaluation:

- The performance of the trading strategy is evaluated based on the profit and loss achieved in each executed trade. The outcome of a trade is determined by comparing the exit price, i.e., the price at which the currency pair is sold, to the entry price, i.e., the price at which the currency pair is bought. A trade is classified as profitable if the exit price is higher than the entry price and as a loss if the exit price is lower than the entry price. The profit and loss for a trade is calculated as follows:

$$\text{Profit/Loss (\%)} = \frac{\text{Exit Price} - \text{Entry Price}}{\text{Entry Price}} \times 100 \quad (5)$$

At each step, the portfolio value is computed based on the state of the portfolio, which can be in one of two mutually exclusive conditions:

- (a) **All cash:** If no contracts are held, the portfolio value equals the cash balance.
- (b) **All contracts:** If a position is held, the portfolio value equals the market value of the position.

Step 4: Comparison strategies:

- **Buy-and-hold strategy:** The buy-and-hold approach invests the initial capital in the currency at the start and holds it without further trades. Its performance reflects the cumulative return if the investor had not traded based on predictive signals.
- **Random strategy:** The random strategy generates buy, sell, or hold signals randomly, without relying on any price data or indicators. The execution follows predefined conditions, buying only when no position is held and selling only when a position is available. Portfolio value, winning and losing trades, and annualized returns are tracked to compare performance against model-driven approaches.

Step 5: Performance metrics:

- **Annualized returns:** To gauge each strategy's effectiveness, we compute annualized returns based on cumulative returns and the number of trading days (assuming 252 trading days per year). The annualized return metric provides insight into the average yearly performance of each approach. The trading scenario was tested for the period from December 2018 to March 2022.

Strategy Performance Analysis

The performance analysis of the tested trading strategies reveals that the EXPERT strategy achieved the highest annualized return, yielding a profit of 2.36%. This result notably surpasses the other strategies, each of which underperformed over the evaluation period. Specifically, the buy-and-hold strategy closely trailed at -0.77% , while the random strategy resulted in an annualized return of -2.34% , demonstrating that purely random trading decisions led to a net loss rather than serving as a neutral benchmark (see Table 8). These findings underscore the superior predictive capability of the EXPERT model in optimizing trading decisions compared to traditional or stochastic approaches.

Table 8. Trading performance of different strategies.

Strategy	Performance
EXPERT	2.36%
Buy and hold	-0.77%
Random	-2.34%

9. Conclusions

In this study, we introduced EXPERT (EXchange rate Prediction using Encoder Representation from Transformers), a Transformer-based framework for forecasting foreign exchange rates. By adapting the encoder-only Transformer architecture to time series data, EXPERT effectively captures both long-term dependencies and short-term dynamics inherent in financial series.

Empirical evaluation across nine major, minor, and exotic currency pairs over more than two decades (1999–2022) demonstrated that EXPERT consistently outperforms traditional econometric, ensemble, and deep learning benchmarks—including random walk, Linear Regression, Random Forest, XGBoost, and LSTM—on various performance metrics (namely, MAPE, MAE, and MSE). Its forecasting advantage is statistically supported by Diebold–Mariano tests, confirming its robustness and reliability across diverse market conditions.

From an academic perspective, this proposal contributes to the literature on attention-based deep learning for time series forecasting. It shows that the encoder-only Transformer

is able to model temporal dependencies through self-attention rather than sequential recurrence and can outperform recurrent architectures such as LSTM in terms of accuracy and efficiency. The proposed design choices pave the way for applying Transformer architectures to other time series domains.

From a practical standpoint, EXPERT offers a scalable and data-efficient forecasting tool suitable for algorithmic trading, portfolio risk management, and policy analysis. Its high accuracy, adaptability to longer horizons, and consistent performance across currency types make it valuable to both financial institutions and decision-makers managing exchange rate exposure.

Future research could extend this work by incorporating macroeconomic indicators, textual or sentiment data, and probabilistic forecasting.

Author Contributions: Conceptualization, E.B., K.D., and K.G.; methodology, E.B., T.P., and K.D.; software, E.B.; validation, E.B.; formal analysis, E.B., K.D., and T.P.; investigation, E.B.; resources, E.B.; data curation, E.B.; writing—original draft preparation, E.B.; writing—review and editing, E.B., K.D., and T.P.; visualization, E.B.; supervision, T.P., K.D., and K.G.; project administration, K.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from the MetaTrader platform (MetaQuotes Ltd.) via the Alpari demo server. The data are available from the authors upon request or by registering for a free demo account with Alpari at <https://alpari.com>.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (OpenAI, GPT-4) to assist with grammar and language editing. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Huang, J.; Chai, J.; Cho, S. Deep learning in finance and banking: A literature review and classification. *Front. Bus. Res. China* **2020**, *14*, 13. [\[CrossRef\]](#)
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. *Attention Is All You Need*; Advances in Neural Information Processing Systems; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
3. Wood, K.; Giegerich, S.; Roberts, S.; Zohren, S. Trading with the Momentum Transformer: An intelligent and interpretable architecture. *arXiv* **2022**, arXiv:2112.08534. [\[CrossRef\]](#)
4. Gradzki, P.; Wójcik, P. Is attention all you need for intraday Forex trading? *Expert Syst.* **2023**, *41*, e13317. [\[CrossRef\]](#)
5. Fischer, T.; Sterling, M.; Lessmann, S. Fx-spot predictions with state-of-the-art transformer and time embeddings. *Expert Syst. Appl.* **2024**, *249*, 123538. [\[CrossRef\]](#)
6. Kantoutsis, K.; Mavrogianni, A.; Theodorakatos, N. Transformers in High-Frequency Trading. *J. Phys. Conf. Ser.* **2024**, *2701*, 012134. [\[CrossRef\]](#)
7. Sezer, O.; Gudelek, M.; Ozbayoglu, A. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [\[CrossRef\]](#)
8. Ryll, L.; Seidens, S. Evaluating the performance of machine learning algorithms in financial market forecasting: A comprehensive survey. *arXiv* **2019**, arXiv:1906.07786. [\[CrossRef\]](#)
9. Panda, M.; Panda, S.; Pattnaik, P. Exchange rate prediction using ANN and deep learning methodologies: A systematic review. In Proceedings of the 2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), Rajpura, India, 7–15 February 2020; pp. 1–6.
10. Islam, M.; Hossain, E.; Rahman, A.; Hossain, M.; Andersson, K. A review on recent advancements in FOREX currency prediction. *Algorithms* **2020**, *13*, 186. [\[CrossRef\]](#)
11. Fletcher, T. Machine Learning for Financial Market Prediction. Doctoral Thesis, University College London, London, UK, 2012.
12. Goncu, A. Prediction of exchange rates with machine learning. In Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, Sanya, China, 19–21 December 2019.

13. Qi, L.; Khushi, M.; Poon, J. Event-driven LSTM for Forex price prediction. In Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 16–18 December 2020; IEEE: Sydney, Australia, 2020.
14. Islam, M.; Hossain, E. Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. *Soft Comput. Lett.* **2021**, *3*, 100009. [[CrossRef](#)]
15. Das, A.K.; Mishra, D.; Das, K.; Mohanty, A.K.; Mohammed, M.A.; Al-Waisy, A.S.; Kadry, S.; Kim, J. A Deep Network-Based Trade and Trend Analysis System to Observe Entry and Exit Points in the Forex Market. *Mathematics* **2022**, *10*, 3632. [[CrossRef](#)]
16. Bousbaa, Z.; Sanchez-Medina, J.; Bencharef, O. Financial Time Series Forecasting: A Data Stream Mining-Based System. *Electronics* **2023**, *12*, 2039. [[CrossRef](#)]
17. Zitis, P.I.; Potirakis, S.M.; Alexandridis, A. Forecasting Forex Market Volatility Using Deep Learning Models and Complexity Measures. *J. Risk Financ. Manag.* **2024**, *17*, 557. [[CrossRef](#)]
18. Zhao, L.; Yan, W.Q. Prediction of Currency Exchange Rate Based on Transformers. *J. Risk Financ. Manag.* **2024**, *17*, 332. [[CrossRef](#)]
19. Freeman, J.; Williams, J.; Lin, T. Vector autoregression and the study of politics. *Am. J. Polit. Sci.* **1989**, *33*, 842–875. [[CrossRef](#)]
20. Benjamin, M.; Rigby, R.; Stasinopoulos, D. Generalized autoregressive moving average models. *J. Am. Stat. Assoc.* **2003**, *98*, 214–223. [[CrossRef](#)]
21. Liu, J.; Liu, X.; Lin, H.; Xu, B.; Ren, Y.; Diao, Y.; Yang, L. Transformer-based capsule network for stock movements prediction. In Proceedings of the First Workshop on Financial Technology and Natural Language Processing, Macao, China, 12 August 2019; pp. 66–73.
22. Vargas, M.R.; Anjos, C.E.M.D.; Bichara, G.L.G.; Evsukoff, A.G. Deep learning for stock market prediction using technical indicators and financial news articles. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; IEEE: Rio de Janeiro, Brazil, 2018. [[CrossRef](#)]
23. Wu, N.; Green, B.; Ben, X.; O'Banion, S. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv* **2020**, arXiv:2001.08317. [[CrossRef](#)]
24. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
25. Chen, X.; Lee, J.D.; Tong, X.T.; Zhang, Y. Statistical inference for model parameters in stochastic gradient descent. *Ann. Stat.* **2020**, *48*, 251–273. [[CrossRef](#)]
26. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
27. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
29. Hsu, J.C. *Multiple Comparisons, Theory and Methods*; Chapman & Hall/CRC: Boca Raton, FL, USA, 1996.
30. Hiransha, M.A.; Al-Khasawneh, M.A.; Khan, S.U.R.; Khan, Z. Stock market trend prediction using deep learning approach. *Comput. Econ.* **2018**, *53*, 123–135.
31. Avellaneda, M.; Stoikov, S. High-frequency trading in a limit order book. *Quant. Financ.* **2008**, *8*, 217–224. [[CrossRef](#)]
32. Krauss, C.; Do, X.A.; Huck, N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *Eur. J. Oper. Res.* **2017**, *259*, 689–702.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.