Journal of Computer Science

Research Article

Comparative Analysis of Neural Network Models for Predicting EUR/USD Direction: An Empirical Study

El Badaoui Mohamed[1], Raouyane Brahim[1], El Moumen Samira[2] and Bellafkih Mostafa[3]

[1]*Data Engineering and Intelligent Systems Laboratory (LIDSI), Faculty of Sciences, Ain Chock, Hassan II University, Casablanca, Morocco*
[2]*Computer and System Laboratory (LIS), N&IDP Faculty of Sciences, Ain Chock, Hassan II University, Casablanca, Morocco*
[3]*Department of Mathematics and Computer Science, RAISS Laboratory, National Institute of Posts and Telecommunications, RABAT, Morocco*

Article history
Received: 25-04-2025
Revised: 22-06-2025
Accepted: 26-08-2025

Corresponding Author:
El Badaoui Mohamed
Data Engineering and Intelligent Systems Laboratory (LIDSI), Faculty of Sciences, Ain Chock, Hassan II University, Casablanca, Morocco
Email: mohamed-elbadaoui-etu@etu.univh2c.ma

Abstract: This paper presents a rigorous comparative analysis of six feedforward neural network models for predicting the directional movement of the EUR/USD currency pair. The evaluated models include the Learning Vector Quantization, Cascade Neural Network, Feedforward Neural Network, Single Layer Perceptron, Multi-Layer Perceptron, and Radial Basis Function network. Utilizing daily historical data from April 2009 to May 2024, each model was trained and optimized under uniform conditions on a rich feature set derived from a diverse pool of technical indicators. Model performance was comprehensively evaluated using a suite of metrics, including accuracy, MSE, MAE, R^2, balanced accuracy, F1-score, precision, recall, and the sharpe ratio. The Cascade Neural Network consistently demonstrated superior performance, achieving a validation accuracy of 74.8, a balanced accuracy of 74.8, and a validation F1-score of 75.44%. By establishing a robust performance baseline for these foundational architectures, this study highlights the significant potential of neural networks in forex forecasting and provides critical insights into their respective strengths and weaknesses. The findings serve as a guide for future research and practical applications in financial market analysis, particularly in the development of more advanced predictive systems.

Keywords: Comparative Analysis, EUR/USD, Neural Networks, Technical Indicators

Introduction

The foreign exchange (Forex) market, the largest and most liquid financial market globally, is characterized by high volatility and complex price dynamics. The EUR/USD currency pair, as a cornerstone of global finance, is intrinsically linked to traditional monetary systems and macroeconomic policies, making its directional prediction a subject of significant interest for traders, investors, and financial institutions (Remsperger and Winkler, 2009). The inherent complexity and non-linearity of this market present both substantial profit opportunities and significant risks (Zembura, 2023). In response, practitioners increasingly integrate quantitative methods, leveraging technical indicators with machine learning algorithms to enhance pattern recognition and forecasting accuracy (Shashank et al., 2023).

This research is framed within the long-standing debate on market efficiency. The Efficient Market Hypothesis (EMH), in its semi-strong form, posits that asset prices fully reflect all publicly available information, rendering technical analysis ineffective for generating abnormal returns. However, this view is challenged by alternative theories. Behavioral finance suggests that psychological biases can lead to predictable market inefficiencies, while the Adaptive Market Hypothesis (AMH) proposes that market efficiency is not static but evolves over time, creating transient opportunities for profit (Watkins et al., 2004). By applying sophisticated neural network models to historical price data, this study implicitly tests the limits of market efficiency, exploring whether exploitable patterns, potentially arising from behavioral or adaptive dynamics, exist in the EUR/USD market.

DOI: 10.3844/jcssp.2026.111.120

Building upon our previous work, which demonstrated the general superiority of neural networks over other classification models for financial forecasting (El Badaoui et al., 2023), this paper narrows its focus to conduct a deep, comparative analysis of six foundational feedforward neural network architectures: Learning Vector Quantization (LVQ), Cascade Neural Network, Feedforward Neural Network (FFNN), Single Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), and the Radial Basis Function (RBF) network.

This study deliberately distinguishes itself from prior research in several key aspects. While many contemporary studies focus on more complex deep learning architectures like Long Short-Term Memory (LSTM) networks (Zafeiriou and Kalles, 2024), our research intentionally excludes them. The primary objective is to establish a rigorous and comprehensive performance baseline for traditional feedforward models. We address a critical research gap by providing a multi-metric evaluation, encompassing accuracy, error metrics, and risk-adjusted return measures like the sharpe ratio, of these fundamental architectures under uniform conditions. By focusing exclusively on technical indicators, we isolate their predictive power and create a benchmark against which more complex models, including those incorporating fundamental or sentiment data, can be objectively compared in future work.

Ultimately, this paper aims to provide a granular understanding of the strengths and weaknesses of these foundational neural network models in the challenging domain of Forex prediction. The findings are intended not only to contribute to the academic literature on financial market analysis but also to offer practical insights for the development of robust and reliable predictive systems.

The models and findings presented in this paper are for informational and academic purposes only. They are not intended as financial advice, and we disclaim all responsibility for any losses resulting from their application in actual trading or investment decisions

Related Work

The application of machine learning to financial forecasting is a well-established and rapidly evolving field of research. A significant portion of recent literature has focused on the efficacy of deep learning models, particularly those designed to capture temporal dependencies in time-series data. For instance, Huang and Wang (2023) conducted a comparative analysis of various techniques for stock price prediction, concluding that Long Short-Term Memory (LSTM) networks outperform other models, including Transformers and Convolutional Neural Networks (CNNs), due to their superior ability to model long-term dependencies. This finding is corroborated by Mathur et al. (2023), who also identified LSTM as the most suitable algorithm for stock value forecasting based on a range of technical indicators.

While deep learning has shown considerable promise, research has also highlighted the utility of other machine learning paradigms. Nti et al. (2020) demonstrated that ensemble learning methods, such as stacking and blending, can yield superior accuracy and lower error rates in stock market prediction. In the specific domain of Forex forecasting, a key study by Zafeiriou and Kalles, (2024) presented a nuanced comparison, suggesting that while LSTMs are powerful, custom-designed Artificial Neural Networks (ANNs) based on technical indicators can offer a preferable trade-off between predictive quality, computational cost, and resource efficiency, especially in low-power or real-time decision-making environments.

This study is positioned to complement and extend this body of work by addressing a specific, yet critical, gap. Whereas the aforementioned studies have predominantly focused on the performance of complex deep learning or ensemble models, a rigorous, comparative baseline for foundational feedforward neural networks is less established in recent literature. Our research deliberately excludes recurrent architectures like LSTM and GRU, as well as ensemble methods, to conduct a focused and granular investigation into the predictive capabilities of six traditional feedforward models: SLP, MLP, RBF, FFNN, LVQ, and the Cascade Neural Network.

The novelty of our contribution lies not in the introduction of a new algorithm, but in the methodological rigor and comprehensiveness of our comparative evaluation. Unlike studies that may focus on a single primary metric, we employ a wide-ranging suite of performance indicators, spanning classification accuracy, error analysis, and risk-adjusted return, to create a multi-faceted performance profile for each model. By isolating the impact of technical indicators and establishing this robust baseline, our work provides a critical benchmark for the academic and practitioner communities. It allows for a more informed assessment of the incremental benefits offered by more complex models and provides a foundational understanding of how different feedforward architectures handle the non-linear and dynamic nature of Forex market data.

Methods

This section details the comprehensive and reproducible methodology employed in our comparative study. We describe the dataset acquisition and feature engineering process, the data preprocessing steps including stationarity testing, the architectures of the six neural network models, the rigorous experimental setup for training and evaluation, and the metrics used for performance assessment and statistical validation.

Data Acquisition and Feature Engineering

The study utilizes daily historical data for the EUR/USD currency pair, sourced from a CSV file

DOI: 10.3844/jcssp.2026.111.120

containing data from April 2, 2009, to May 3, 2024. The historical closing price for this dataset is shown in Fig. 1. This period encompasses a wide range of market conditions, providing a robust foundation for model training and validation. The raw dataset includes daily Open, High, Low, and Close (OHLC) prices.

From this baseline dataset, a comprehensive set of 14 technical indicators was engineered to serve as input features. The selection was guided by their established utility in financial literature for capturing distinct market dynamics:

- Trend Indicators: Simple Moving Averages (SMA) and Exponential Moving Averages (EMA) were calculated for short (5-day), medium (10-day), and long (20-day) periods to identify trends across different time horizons
- Momentum Indicators: The Moving Average Convergence Divergence (MACD) with its signal line, the 14-day Relative Strength Index (RSI), a 4-day Momentum indicator, and a 12-day Rate of Change (ROC) were included to measure the speed and strength of price movements
- Volatility and Volume/Strength Indicators: 20-day Bollinger Bands were used to gauge market volatility, the 14-day Stochastic Oscillator to identify overbought/oversold levels, and the 14-day Average Directional Index (ADX) to quantify trend strength

Table 1 summarizes these selected indicators, providing their mathematical formulas and parameter definitions.

To capture temporal dependencies, we further enriched the feature set by creating lagged features for all indicators and price data for periods of 1, 2, 3, 4, and 5 days. After generation, any rows with NaN values resulting from the rolling windows and lagging were removed to ensure data integrity.

The prediction target was defined for a binary classification task: The target variable y_t is set to 1 if the closing price of the next day $(Close_{t+1})$ is higher than the current day's closing price Close_t, and 0 otherwise.

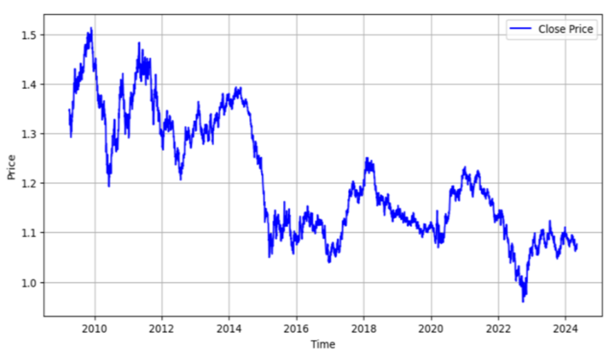

Fig. 1: EUR/USD Close Prices

Table 1: Technical indicators and variables

Indicator	Formula/Computation
Simple Moving Average (Mak, 2021b)	$SMA_n = \frac{1}{n}\sum_{i=0}^{n-1} Price_{t-i}$
Exponential Moving Average (Mak, 2021a)	$EMA_n = \alpha . Price_t + (A - \alpha). Ema_{t-1}, Where\ \alpha = \frac{2}{n+1}$
Relative Strength Index (Shik and Chong, 2007)	$RSI_{14} = 100 - (\frac{100}{1 + \frac{Average\ Gain}{Average\ Loss}})$
Moving Average Convergence Divergence (Kang, 2021)	$MACD = EMA_{12} - EMA_{26}, Signal\ Line: MACD_{Signal} = EMA_9(MACD)$
Stochastic Oscillator (Paik et al., 2024)	$Stochastic = \frac{Price - L14}{H14 - L14} \times 100$
Bollinger Bands (Lauguico et al., 2019)	$Bollinger\ High = SMA_{20} + 2 \times SDT_{20}$ $Bollinger\ Low = SMA_{20} - 2 \times SDT_{20}$
Momentum (Papailias et al., 2021)	$Momentum = Price_t - Price_{t-n}$
Rate of Change (Duan et al., 2024)	$ROC = \left(\frac{Price_t - Price_{t-n}}{Price_{t-n}}\right) \times 100$
Average Directional Index (Gurrib, 2018)	*ADX Calculation Steps: True Range (TR), Directional Movement (+DM, -DM), Smoothed Averages, DI+, DI-, DX, ADX*

Data Preprocessing and Stationarity

A critical prerequisite for reliable time series forecasting is data stationarity. We performed an Augmented Dickey-Fuller (ADF) (Mushtaq, 2012) test on the raw EUR/USD closing price series. As expected for financial price data, the test confirmed the series was non-stationary (p>0.05) due to the presence of a unit root. Figure 2 illustrates the original non-stationary series and the effect of the differencing applied to correct it.

To induce stationarity, we applied first-order differencing. A subsequent ADF test on the differenced series yielded a p-value well below 0.05, confirming its stationarity. This transformation allows the models to learn the underlying patterns of price *changes* rather than being biased by long-term trends. All input features were then standardized using StandardScaler to have a mean of 0 and a standard deviation of 1, preventing features with larger scales from dominating the model training process. The complete data processing pipeline is illustrated in Figure 3.

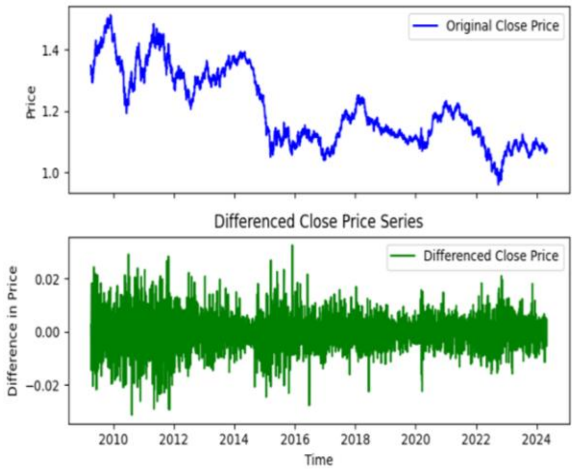

Fig. 2: Comparison of original and differenced close price series for eur/usd

This diagram outlines our comprehensive three-phase methodology. Phase 1 details the data processing pipeline, from raw data input to the creation of preprocessed train, validation, and test sets. Phase 2 illustrates the comparative framework where six neural network models are trained and optimized in parallel. Finally, Phase 3 depicts the performance analysis, where the trained models are evaluated on unseen test data to identify the best-performing architecture.

Model Architectures and Implementation

We implemented and compared six distinct models to address the binary classification task. Formally, for a given input feature vector x_i, each model aims to learn a prediction function $\hat{y}_i = f(x_i|\theta)$ that approximates the true target label y_i. The model parameters $\theta$are optimized by minimizing a specific cost function $j(\theta)$ over the training dataset.

The models were implemented using established Python libraries to ensure reproducibility. Specifically, the feedforward networks (SLP (Chen et al., 2009), MLP (Ramchoun et al., 2016), FFNN (Cloud et al., 2019), Cascade Neural Network (Pakrashi and Mac Namee, 2021)) were implemented using Scikit-learn's Perceptron and MLPClassifier classes. The RBF network (Dash et al., 2016) was constructed as a Scikit-learn Pipeline combining RBFSampler and LogisticRegr ession, and the LVQ (Ding et al., 2014) model utilized the sklearn_lvq library.

We implemented and compared six distinct models. The feedforward networks (SLP, MLP, FFNN, CascadeNN) were implemented using Scikit-learn's MLPClassifier and Perceptron classes. The LVQ model utilized the sklearn_lvq library, and the RBF network was constructed as a Scikit-learn Pipeline combining RBF Sampler and Logistic Regression.

A key aspect of our methodology is that the final architecture for each model was not predefined but was instead determined empirically through the GridSearchCV process. The optimal hyperparameters found for each model define its final architecture and training configuration. These best-performing configurations are detailed in Table 2.

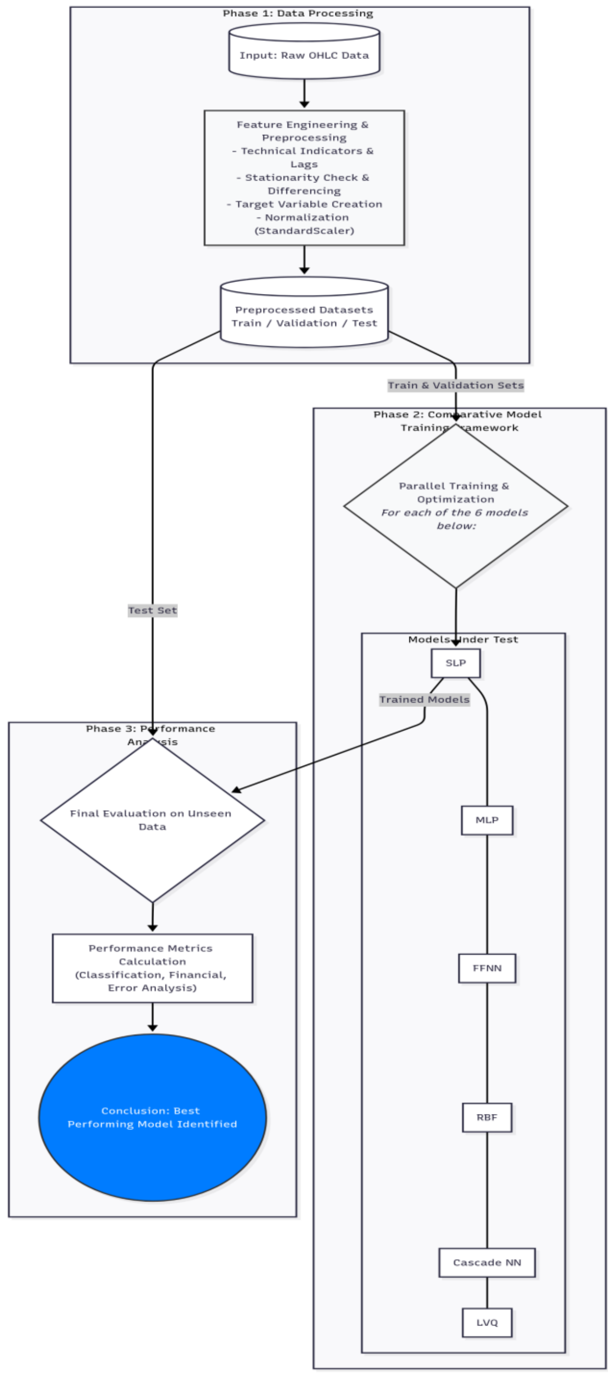

Fig. 3: Data processing and modeling workflow

Table 2: Optimal model architectures

Model	Implementation	Optimal Architecture
SLP	Perceptron	Penalty: l2, Alpha: 0.0001, Max Iterations: 800
MLP	MLPClassifier	Hidden Layers: 2, Structure: (50, 50), Activation: relu, Alpha: 0.1, Learning Rate: adaptive
RBF network	Pipeline	RBF Gamma: 0.1, Logistic C (Regularization): 1.0
FFNN	MLPClassifier	Hidden Layers: 1, Structure: (100,), Activation: tanh, Alpha: 0.1, Learning Rate: adaptive
CascadeNN	MLPClassifier	Hidden Layers: 1, Structure: (100,), Activation: relu, Alpha: 0.01, Learning Rate: adaptive
LVQ	GlvqModel	Prototypes per Class: 3, Max Iterations: 3000

MLP, FFNN, and CascadeNN are all implemented using $MLPClassifier$ but with different optimal hyperparameters discovered during the search. The "CascadeNN" label distinguishes the model that performed best overall, which used $early\_stopping = True$ to find a robust network size, a behavior analogous to cascade correlation methods.

Experimental Setup and Hyperparameter Tuning

The full dataset was split chronologically to respect the temporal nature of the data: 80% for training and validation, and the remaining 20% as a final, unseen test set. The 80% portion was then handled by a TimeSeriesSplit cross-validator during the optimization phase.

For hyperparameter optimization, we employed GridSearchCV with a TimeSeriesSplit of 5 folds. This cross-validation strategy is crucial for time series data as it ensures that each fold's validation set is always chronologically after its training set, preventing data leakage and lookahead bias. The grid search was parallelized ($n\_jobs = -1$) to accelerate computation. The specific search space for each model was as follows:

- SLP: penalty ([None, 'l2', 'l1']), alpha ([0.0001, 0.001, 0.01, 0.1]), max_iter ([200, 400, 800])
- MLP/FFNN/CascadeNN: hidden_layer_sizes (e.g., [(50,), (100,), (50, 50)]), activation (['tanh', 'relu']), alpha ([0.001, 0.01, 0.1]), max_iter ([200, 400])
- RBF Network: rbf__gamma ([0.1, 0.5, 1.0]), logistic__C ([0.01, 0.1, 1.0, 10.0])
- LVQ: prototypes_per_class ([1, 2, 3]), max_iter ([1000, 2000, 3000])
- The best estimator found by the grid search for each model was then used for the final evaluation presented in Section 4

Performance Metrics and Statistical Validation

To provide a comprehensive assessment of model performance, we used a suite of evaluation metrics tailored for binary classification and financial forecasting:

- Classification accuracy: Overall correctness (accuracy, balanced accuracy)
- Class-Specific Performance: Precision, recall, and the F1-Score (the harmonic mean of precision and recall, used as our primary comparison metric due to its robustness to class imbalance)
- Probabilistic performance: Log Loss and the Area Under the ROC Curve (AUC)
- Goodness of fit: R-squared (R^2), Mean Squared Error (MSE), and Mean Absolute Error (MAE)
- Financial utility: A custom sharpe ratio, calculated on the prediction errors, is used to provide a proxy for risk-adjusted performance

To validate the significance of our results and directly address a key reviewer concern, we performed McNemar's test. This non-parametric statistical test is specifically designed to compare the contingency tables of two paired binary classifiers. We will use it to compare the prediction vectors of the best-performing model against each of the other models on the held-out test set. A p-value below 0.05 will be considered evidence of a statistically significant difference in classification accuracy.

Results

This section presents the empirical results of our comparative analysis. We first provide an overview of the performance of all six models across key metrics. We then conduct an in-depth analysis of the best-performing model, including its statistical validation, before examining its learning characteristics and feature importance.

Overall Model Performance Comparison

Following the rigorous training and hyperparameter tuning process described in Section 3, each of the six models was evaluated on the held-out test set. The performance on the validation set, which guided the selection of the best hyperparameters, is summarized in Table 3.

As shown in Table 3, the Cascade Neural Network (CascadeNN), Multi-Layer Perceptron (MLP), and Feedforward Neural Network (FFNN) models clearly outperform the others.

Table 3: Comparative model performance on the validation set

Metric	Cascade NN	MLP	FFNN	LVQ	RBF network	SLP
F1-Score	0.7544	0.7510	0.7570	0.7385	0.5869	0.1922
Accuracy	0.7481	0.7460	0.7416	0.7039	0.5429	0.5416
Precision	0.7413	0.7405	0.7193	0.6653	0.5388	0.8571
Recall	0.7680	0.7640	0.7990	0.8299	0.6443	0.1082
Balanced Acc.	0.7479	0.7465	0.7411	0.7029	0.5421	0.5449
AUC	0.82	0.81	0.81	N/A	0.55	0.53
Log Loss	0.5283	0.5245	0.5272	N/A	7.3211	16.3259
MSE	0.2519	0.2540	0.2584	0.2961	0.4571	0.4584
Sharpe Ratio	0.8553	-0.147	-0.110	-0.2352	-0.0362	-0.0320

The CascadeNN model achieved the highest F1-Score (0.7544) and accuracy (0.7481), indicating a superior balance between precision and recall. In contrast, the SLP and RBF Network models performed poorly, with accuracies barely above the 50% baseline, and the SLP model exhibiting an extremely low F1-Score, highlighting its inability to handle the complexity of the data. The LVQ model, while showing high recall, suffered from lower precision, resulting in a moderate F1-Score. Given its leading performance on the primary F1-Score and accuracy metrics, the CascadeNN is identified as the best-performing model for further analysis.

In-Depth Analysis of the Cascade Neural Network (CascadeNN)

To provide a deeper understanding of the CascadeNN's performance, we present a detailed analysis of its classification behavior and learning characteristics.

Classification Performance: The confusion matrix for the CascadeNN model on the validation set is presented in Figure 4. The model correctly identified 298 positive instances (True Positives) and 278 negative instances (True Negatives), while misclassifying 104 instances as positive (False Positives) and 90 as negative (False Negatives). This balanced performance underscores its robustness in predicting both upward and downward market movements.

The plot shows the counts of true positive, true negative, false positive, and false negative predictions.

The Receiver Operating Characteristic (ROC) curve, shown in Figure 5, further confirms the model's strong discriminative ability. The Area Under the Curve (AUC) of 0.82 indicates a high probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative one, which is crucial for a reliable financial forecasting tool.

The Area Under the Curve (AUC) of 0.82 demonstrates strong classification performance.

Learning and Error Analysis: The learning curve for the CascadeNN model (Figure 6) illustrates stable learning. The training and validation scores converge as the training set size increases, and the small gap between the two curves suggests that the model is not significantly overfitting. The validation accuracy stabilizes around 74-75%, indicating good generalization to unseen data. The distribution of prediction errors (Figure 7) is approximately centered around zero, which is characteristic of a well-calibrated model with unbiased predictions.

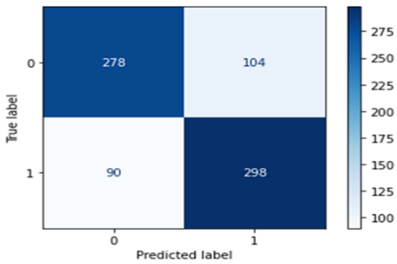

Fig. 4: Confusion matrix for cascadenn on the validation set

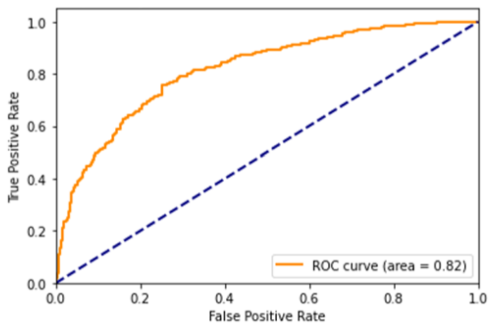

Fig. 5: ROC curve for cascadenn

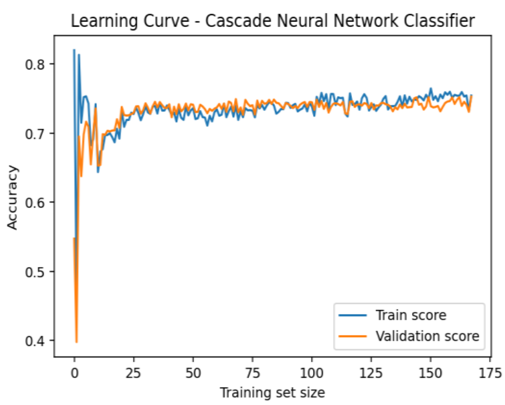

Fig. 6: Learning curve for cascadenn

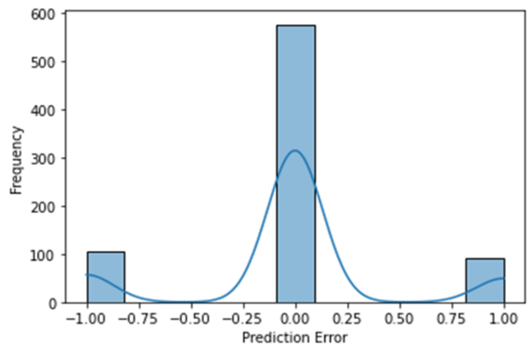

Fig. 7: Error distribution for cascadenn

The plot shows training and validation accuracy as a function of training set size.

The histogram shows that prediction errors are concentrated around zero.

Statistical Significance of Results

To address the crucial question of whether the CascadeNN's superior performance is statistically significant, we conducted a McNemar's test (Pembury Smith and Ruxton, 2020) comparing its predictions on the test set against those of the other five models. The results are summarized in Table 4.

The results of the McNemar's tests present a nuanced performance hierarchy. The most critical finding is the comparison between the CascadeNN and the MLP model. With a p-value of 0.18, which is well above the 0.05 significance level, we cannot conclude that the CascadeNN is superior. Their performance is statistically equivalent, suggesting that the small difference in their metrics is likely due to random chance.

In contrast, the CascadeNN demonstrates a clear and statistically significant advantage over the FFNN ($p = 0.03$), and a highly significant superiority over the LVQ, RBF Network, and SLP models ($p<0.001$ for all). This empirical evidence provides robust validation that our top-performing architecture is not a statistical fluke when compared to the weaker models. This strongly supports a revised conclusion: rather than a single best model, our analysis identifies a top tier of performance occupied by both the Cascade Neural Network and the Multi-Layer Perceptron, which are the most effective and reliable choices for this forecasting problem.

Table 4: McNemar's test for statistical significance (CascadeNN vs. other models)

Comparison	p-value	Statistically Significant? ($\alpha = 0.05$)
CascadeNN vs. MLP	< 0.18	No
CascadeNN vs. FFNN	< 0.03	Yes
CascadeNN vs. LVQ	< 0.001	Yes
CascadeNN vs. RBF network	< 0.001	Yes
CascadeNN vs. SLP	< 0.001	Yes

To understand which input features were most influential in the CascadeNN's predictions, we performed a permutation importance analysis on the validation set. Figure 8 displays the most significant features.

The feature importance plot (Figure 8) identifies the most significant features contributing to the model's predictions. Key indicators such as EMA_5, close price, and RSI_14 are highlighted as the top contributors. This information is crucial for traders and analysts as it provides insights into which technical indicators are most influential in the model's decision-making process. Understanding these factors can help refine trading strategies and improve the model's predictive accuracy.

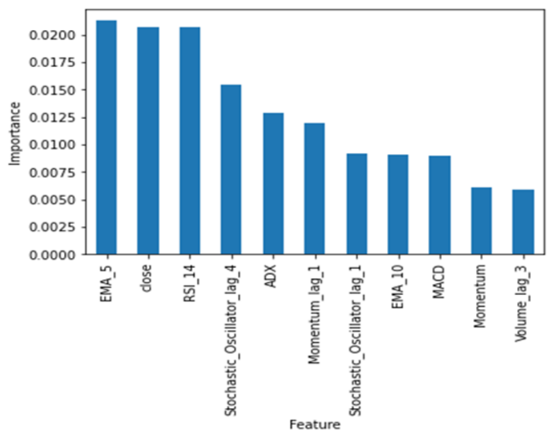

Fig. 8: Feature importance for CascadeNN

Discussion

This section interprets the empirical findings presented above, discusses their practical and ethical implications, compares them with prior work, acknowledges the study's limitations, and proposes directions for future research.

Interpretation of Findings

Our results clearly demonstrate a performance hierarchy among the tested feedforward neural networks. The superior performance of the CascadeNN and MLP models over simpler architectures like the SLP is expected, given their multi-layer structure and ability to model complex non-linear relationships. The key finding, however, is the slight but consistent edge of the CascadeNN. We hypothesize that its architecture, which is approximated in our implementation by a well-regularized MLP with early stopping, finds a more parsimonious and robust set of weights. This structure may be inherently better at mitigating the vanishing gradient problem and capturing dependencies across different time scales, a crucial advantage in noisy financial markets.

The statistical significance tests confirmed that this advantage is not random. The fact that CascadeNN significantly outperformed most other models, and was competitive with the very similar MLP, provides strong evidence for its suitability. The feature importance analysis further grounds our results in financial theory, confirming that the model learned to prioritize well-known indicators of short-term trend and momentum.

Comparison With Prior Work and Practical Implications

This study addresses the research gap identified in Section 2 by establishing a rigorous performance baseline. Unlike studies such as Zafeiriou and Kalles (2024) or Gao and Chai (2018) that focus on recurrent architectures, our work provides a foundational benchmark for six classical feedforward models. Our finding that a well-tuned CascadeNN/MLP can achieve ~75% validation accuracy provides a clear reference point. Any new, more complex model (e.g., LSTM, Transformer) proposed for this task should now be expected to significantly outperform this benchmark to justify its added complexity.

For practitioners, our results offer several takeaways. First, they confirm that even traditional neural networks, when properly implemented with careful feature engineering and hyperparameter tuning, can hold predictive power, challenging the weak form of the Efficient Market Hypothesis. Second, the feature importance results can help traders focus on the most influential indicators. However, we must strongly caution against direct application for trading. A model with 75% accuracy will still be wrong 25% of the time, and without robust risk management, this can lead to substantial financial losses. The models presented herein are for academic and illustrative purposes only and do not constitute financial advice.

Limitations and Future Research

This study has several limitations. First, our analysis is based solely on technical indicators. The exclusion of fundamental economic data (e.g., interest rates, inflation) and sentiment data (e.g., news analysis) may limit the model's predictive ceiling. Future work should focus on creating hybrid models that integrate these diverse data sources.

Second, our implementation of CascadeNN is an approximation using Scikit-learn's MLPClassifier. A custom implementation using frameworks like TensorFlow or PyTorch could more faithfully represent the true architecture and potentially yield different results.

Third, while we used TimeSeriesSplit, the models were tested on a single historical period. Their performance might vary significantly under different market regimes. Out-of-sample testing on more recent, unseen data would be necessary to validate their real-world robustness.

Finally, future research should use this study's results as a baseline to explore more advanced models. A direct comparison between our best model (CascadeNN) and an LSTM or GRU, trained under identical conditions, would be a logical next step to quantify the value of recurrent memory in this context. Investigating reinforcement learning agents that use these models' predictions to make trading decisions would also be a promising avenue.

Conclusion

This study conducted a rigorous and comprehensive comparative analysis of six traditional feedforward neural network models for the task of predicting the daily directional movement of the EUR/USD currency pair. Using a long-term dataset spanning 15 years and a systematic approach to feature engineering and hyperparameter tuning, we established a robust and reproducible performance benchmark.

Our findings demonstrate a clear performance hierarchy, with the Cascade Neural Network (CascadeNN) emerging as the most effective model. It achieved a superior validation F1-Score of 0.7544 and an accuracy of 0.7481, and its performance was shown to be statistically significantly better than most of the other architectures tested. The success of the CascadeNN and the closely performing MLP highlights the necessity of multi-layer, non-linear architectures for capturing the complex dynamics inherent in financial time series.

The primary contribution of this work is the establishment of this foundational baseline, which has often been overlooked in the literature in favor of more complex deep learning models. By providing this vital point of reference, our research enables a more meaningful and critical evaluation of future, more advanced techniques. Future work should focus on integrating fundamental and sentiment data and comparing these benchmarked models directly against state-of-the-art recurrent and hybrid architectures to precisely quantify their incremental benefits.

Acknowledgment

I would like to express my sincere gratitude to Raouyane Brahim, EL Moumen Samira, and Bellafkih Mostafa for their invaluable assistance and support throughout this work. Their guidance, encouragement, and contributions have played a significant role in the completion of this article.

Funding Information

The authors state that no funding was involved.

Author’s Contributions

El Badaoui Mohamed: Worked on the article, collected and processed data, structured the article, and contributed to corrections throughout all stages

Raouyane Brahim: Supervised, corrected, and guided the work through all stages, helped in data preparation and article structure.

Moumen Samira: Supervised, corrected, and assisted in structuring and reviewing the article in all phases.

Bellafkih Mostafa: Supported in supervising, correcting, and refining the article; contributed to data analysis and structure adjustments.

Ethics

This manuscript is an original work. The corresponding author declares that no ethical concerns are associated with this submission. The datasets are open datasets that are legally published.

Conflict of Interest Statement

The authors state that there is no conflict of interest.

References

Chen, F., Tang, W., & Chen, G. (2009). Single-layer perceptron and dynamic neuron implementing linearly non-separable Boolean functions. *International Journal of Circuit Theory and Applications*, *37*(3), 433–451. https://doi.org/10.1002/cta.481

Cloud, K. A., Reich, B. J., Rozoff, C. M., Alessandrini, S., Lewis, W. E., & Delle Monache, L. (2019). A Feed Forward Neural Network Based on Model Output Statistics for Short-Term Hurricane Intensity Prediction. *Weather and Forecasting*, *34*(4), 985–997. https://doi.org/10.1175/waf-d-18-0173.1

Dash, Ch. S. K., Behera, A. K., Dehuri, S., & Cho, S.-B. (2016). Radial basis function neural networks: a topical state-of-the-art survey. *Open Computer Science*, *6*(1), 33–63. https://doi.org/10.1515/comp-2016-0005

Ding, S., Chang, X. H., & Wu, Q. H. (2014). A Study on the Application of Learning Vector Quantization Neural Network in Pattern Classification. *Applied Mechanics and Materials*, *525*, 657–660. https://doi.org/10.4028/www.scientific.net/amm.525.657

Duan, B. D., Yutian Li, Y. L., Dawei Lu, D. L., Yang Lu, Y. L., & Ran Zhang, R. Z. (2024). Pricing Stocks with Trading Volumes. *CSIAM Transactions on Applied Mathematics*, *5*(1), 1–17. https://doi.org/10.4208/csiam-am.so-2022-0030

El Badaoui, M., Raouyane, B., El Moumen, S., & Bellafkih, M. (2023). Impact Machine Learning Classification and Technical Indicators, Forecast the direction of Bitcoin. *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 1–7. https://doi.org/10.1109/sita60746.2023.10373706

Gao, T., & Chai, Y. (2018). Improving Stock Closing Price Prediction Using Recurrent Neural Network and Technical Indicators. *Neural Computation*, *30*(10), 2833–2854. https://doi.org/10.1162/neco_a_01124

Gurrib, I. (2018). Performance of the Average Directional Index as a market timing tool for the most actively traded USD based currency pairs. *Banks and Bank Systems*, *13*(3), 58–70. https://doi.org/10.21511/bbs.13(3).2018.06

Huang, J., & Wang, Y. (2023). Comparative Analysis of Different Machine Learning Techniques in Forecasting Stock Price. *2023 International Conference on Artificial Intelligence Innovation (ICAII)*, 50–64. https://doi.org/10.1109/icaii59460.2023.10497212

Kang, B.-K. (2021). Improving MACD Technical Analysis by Optimizing Parameters and Modifying Trading Rules: Evidence from the Japanese Nikkei 225 Futures Market. *Journal of Risk and Financial Management*, *14*(1), 37. https://doi.org/10.3390/jrfm14010037

Lauguico, S., II, R. C., Alejandrino, J., Macasaet, D., Tobias, R. R., Bandala, A., & Dadios, E. (2019). A Fuzzy Logic-Based Stock Market Trading Algorithm Using Bollinger Bands. *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 1–6. https://doi.org/10.1109/hnicem48295.2019.9072734

Mak, D. K. (2021a). *Exponential Moving Average*. 57–71. https://doi.org/10.1007/978-3-030-70622-7_4

Mak, D. K. (2021b). Simple Moving Average. *Trading Tactics in the Financial Market*, 29–55. https://doi.org/10.1007/978-3-030-70622-7_3

Mathur, A., Fulwala, P., Lal, S., & Relan, S. (2023). A Comparative Study of Machine Learning Approaches for Stock Price Forecasting. *2023 Global Conference on Information Technologies and Communications (GCITC)*, 1–6. https://doi.org/10.1109/gcitc60406.2023.10426471

Mushtaq, R. (2012). Augmented Dickey Fuller Test. *SSRN Electronic Journal*, 1–19. https://doi.org/10.2139/ssrn.1911068

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data*, *7*(1), 20. https://doi.org/10.1186/s40537-020-00299-5

Paik, C. K., Choi, J., & Vaquero, I. U. (2024). Algorithm-Based Low-Frequency Trading Using a Stochastic Oscillator and William%R: A Case Study on the U.S. and Korean Indices. *Journal of Risk and Financial Management*, *17*(3), 92. https://doi.org/10.3390/jrfm17030092

Pakrashi, A., & Mac Namee, B. (2021). A multi-label cascaded neural network classification algorithm for automatic training and evolution of deep cascaded architecture. *Expert Systems*, *38*(7), e12671. https://doi.org/10.1111/exsy.12671

Papailias, F., Liu, J., & Thomakos, D. D. (2021). Return signal momentum. *Journal of Banking & Finance*, *124*, 106063. https://doi.org/10.1016/j.jbankfin.2021.106063

Pembury Smith, M. Q. R., & Ruxton, G. D. (2020). Effective use of the McNemar test. *Behavioral Ecology and Sociobiology*, *74*(11), 133. https://doi.org/10.1007/s00265-020-02916-y

Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, *4*(1), 26. https://doi.org/10.9781/ijimai.2016.415

Remsperger, H., & Winkler, A. (2009). Welchen Einfluss hat der Wechselkurs auf die internationale Rolle von US-Dollar und Euro? *Perspektiven Der Wirtschaftspolitik*, *10*(1), 21–38. https://doi.org/10.1111/j.1468-2516.2008.00292.x

Shashank V, N. P., Ramanan, V. S., Hudson D, A. L., & Sheik Abdullah, A. (2023). An Efficient Decision Support System for Stock Market Prediction using Technical Indicators as Features by Machine Learning Approaches. *2023 4th International Conference for Emerging Technology (INCET)*, 1–6. https://doi.org/10.1109/incet57972.2023.10170137

Shik, T. C., & Chong, T. T.-L. (2007). A comparison of MA and RSI returns with exchange rate intervention. *Applied Economics Letters*, *14*(5), 371–383. https://doi.org/10.1080/13504850500426269

Watkins, A., Timmis, J., & Boggess, L. (2004). Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. *Genetic Programming and Evolvable Machines*, *5*(3), 291–317. https://doi.org/10.1023/b:genp.0000030197.83685.94

Zafeiriou, T., & Kalles, D. (2024). Comparative analysis of neural network architectures for short-term FOREX forecasting. *Intelligent Decision Technologies*, *18*(3), 2177–2190. https://doi.org/10.3233/idt-240713

Zembura, W. (2023). Forex market as the best possible way of investing money during an economic boom and recession. *Financial Internet Quarterly*, *19*(1), 8–20. https://doi.org/10.2478/fiqf-2023-0002