Exchange Rates Forecasting and Trend Analysis from Machine Learning

Haiwen Huang[1], Kexin Cui[2]

[1] *Business school, The University of Sydney, Australia*

[2] *College of Business & Public Management, Wenzhou-Kean University, China,*

Abstract:Exchange rates play a pivotal role in global economic and financial activities, influencing macroeconomic adjustments through nominal and real rates. Accurate exchange rate forecasting has become increasingly vital for investors, policymakers, and multinational enterprises, enabling effective trading strategies and proactive currency risk management. Despite the theoretical insights offered by fundamental models, their practical application in short-term forecasts remains limited. Statistical models like GARCH, ARIMA, ECM, and VAR have been widely utilized but struggle to capture the nonlinear dynamics and complex relationships in exchange rates, especially over extended forecasting horizons. Artificial intelligence (AI) models have demonstrated significant promise, although challenges like parameter optimization and overfitting persist. Recent empirical studies highlight the superior robustness of hybrid models over single-model approaches. Furthermore, volatility forecasting has gained importance for risk management, investment analysis, and policymaking. This study leverages high-frequency EUR/USD exchange rate data to evaluate minute-based volatility and assess the performance of various forecasting models, contributing to the advancement of predictive methodologies in currency markets.

Keywords: Exchange Rates; Machine Learning; LSTM; CEEMDAN

1. Introduction

Exchange rates are crucial for economic activities and financial markets, impacting macroeconomic adjustments through both the nominal and real exchange rates (Demir & Razmi, 2022). Recently, the importance of hedging against exchange rate risk has been increasingly recognized. Accurate exchange rate forecasts can significantly affect international transactions and the global currency market (Lal & Lim, 2023). For investors, policymakers, and entrepreneurs in multinational companies, precise exchange rate predictions are crucial. They allow for the developing of effective trading strategies and proactive currency risk management (Wang, 2023). Therefore, accurately forecasting exchange rates is critically important, and there is a growing consensus on the need to apply methods for predicting exchange rates.

Focusing on model types, numerous studies aim to demonstrate the viability of using various models to forecast exchange rates. According to Lubecke (1998), Exchange rate forecasting models are generally divided into four primary categories: fundamental theories, statistical models, artificial intelligence (AI), and hybrid forecasting approaches. As Meese and Rogoff (1983) stated, fundamental models cannot fit better than random walks and the outcome is still accepted until now. Overall, while fundamental models offer theoretical insights into exchange rate mechanisms, they are not effective for predicting specific numerical values, particularly in short-term forecasts, which remain a significant challenge (Wang, 2023). According to West (1995) and Wei (2019), statistical models such as generalized autoregressive conditional heteroscedasticity (GARCH) (Chortareas et al., 2011), autoregressive integrated moving average (ARIMA) models (Chortareas et al., 2011), error correction models (ECM) (Moosa & Vaz, 2016) , vector auto-regression (VAR) (Carriero, Kapetanios, & Marcellino, 2009)and Bayesian theory. However, these models face challenges in detecting the nonlinear patterns and complex relationships between exchange rates and other economic factors that lead to poor performance over longer forecasting horizons in conventional tests of forecast efficiency. AI approaches have demonstrated superiority over traditional models in empirical investigations, but they face challenges in settings with large time series fluctuations, including issues like optimal parameter setting, misspecification, underfitting, and overfitting (Wang, 2023). Also, the empirical studies show that hybrid models are more effective and robust than single models (Wei, 2019).

According to Research (Poon & Granger, 2003) forecasting volatility in asset prices, especially exchange rates, is critical for investment analysis, derivative pricing, and risk management. Furthermore, as market volatility directly affects policymaking, volatility forecasts serve as indicators of financial market and economic vulnerability. Evaluate minute based volatility will be assessed in this paper to evaluate its performances of high frequency data. High-frequency data, rich in daily transaction details, are instrumental not only for measuring volatility but also for directly estimating and evaluating models. Recent methodological improvements have centered on utilizing high-frequency data, with numerous studies leveraging this advancement to assess various volatility forecasting models (Chortareas et al., 2011, Carriero, Kapetanios, & Marcellino, 2009). Current research primarily concentrates on general exchange rate trends without specific currency analysis; therefore, this study specifically targets the EUR/USD exchange rate.

The remainder of the paper is arranged as follows. Section 2 reviews some of the main findings and current arguments in the volatility forecasting literature. Section 3 focuses on the data and methodology used in this paper. Section 4 discusses forecast evaluation methods. Section 5 evaluates the estimation results and compares the out-of-sample forecast performances of the models. Finally, Section 6 concludes.

2. Literature Review

2.1 LSTM & CEEMDAN and Their Combination

As one of the most significant methods in time interval, Long Short-Term memory (LSTM), a special kind of recurrent neural network (RNN), was first introduced by Hochreiter and Schmidhuber (1997) to solve the problem of gradient vanishing. Being compared with other methods, such as BPTT and RTRL, LSTM is able to bridge time lags over 1000 steps. It can rapidly acquire the ability to differentiate between two or more widely spaced instances of a certain element in a given sequence, without relying on suitable training examples with short time intervals. Besides, when dealing with long time lag problems, LSTM has the capability to dispose of continuous value, distributed representations and noise. It can even deal with infinite state numbers in principle. According to Gonzalez and Yu (2018), LSTM has a better performance on modeling of time series than other feedforward and recurrent models.

As mentioned before, LSTM is able to deal with problems brought by RNN, such as long-term dependencies. It is a better choice for researchers to apply to predict the trades of currency exchange rate. WigesingheI (2020) chose LSTM to forecast exchange rate because of its ability of defining complicated non-linear interactions between factors and outcomes. According to WigesingheI (2020)'s research, with a 95% confidence level, LSTM model had better performance on predicting five days exchange rates between GBR and USD, USD and CAD, as well as AUS and USD than ARIMA and the Exponential Moving Average.

First introduced by Torres, Colominas, Schlotthauer and Flandrin (2011), Complete EEMD with Adaptive Noise (CEEMDAN) is an algorithm developed by ensemble empirical mode decomposition (EEMD). Based on EEMD, CEEMDAN makes up for EEMD's shortage of fully data-driven number of models and completeness. Besides, the number of sifting iterations it needs is less than EEMD while reconstructing original signal through summing models.

Guan (2022) used CEEMDAN-LSTM model to forecast the stock price of Ping An Bank. In Guan (2022)'s research, CCEEMDAN-LSTM is proved that it can keep practical information when vanishing noise in high frequency in order to gain denoised time series. These characteries enable CEEMDAN-LSTM performs better than simplex CEEMDAN method and LSTM method.

2.2 Exchange Rate

In the field of forecasting currency exchange rate, numerous methods are introduced and created. For example, Li et al. (2019) tried to combine deep Convolutional Neural Network with deep-Recurrent Neural Network to make advantages of their merit. The method they proposed is named C-RNN，which is used to predict the time series data of nine currencies' exchange rates. It was verified that compared to LSTM and CNN, the outcome came from C-RNN is more precise as well as closer to reality. Additionally, several researchers paid attention to improve the LSTM model, which is familiar to this research, to make the prediction of currency exchange rates to be more precise than before. Cao, Zhu, Wang, Demazeau and Zhang (2020) develops a new deep coupled LSTM approach, namely DC-LSTM, to capture the complex couplings for USD/CNY exchange rate forecasting. Similarly, Jung and Choi (2018) introduced autoencoder models and LSTM in their research. Auto encoder models and LSTM were combined to be a hybrid method, called autoencoder-LSTM model to predict FX volatility more accurately than the original LSTM model.

This paper is committed to use the CEEMDAN-LSTM method, which combines the advantages of LSTM with CEEMDAN, to predict as well as analyse the trend of the exchange rate of euro.

3. Data and Methodology

3.1 Data, Properties and the Stylized Facts

The original data sets we use are 1 minute interval spot foreign exchange rates of the euro against the US dollar, provided by Oriental Fortune Terminal.

The EUR/USD currency pair is one of the most traded pairs in the forex market. It is characterized by high liquidity, moderate volatility, and low trading costs, which is due to its narrow spreads. This currency pair is highly sensitive to economic data as well as policy announcements, especially from the European Central Bank (ECB) and the Federal Reserve. Apergis et al. (2012) indicate that the European Central Bank and the American Federal Reserve are able to adjust their monetary policies and changing interest rates in order to influence the EUR/USD exchange rate. However, Apergis et al. (2012) also point out that comparing to the US Federal Reserve, ECB has a disadvantage because of Euro Zone countries do not have the same financial policy. In addition, key economic indicators from the Eurozone and the US, such as GDP and inflation figures also have huge influence on it. For example, Clostermann and Schnatz (2000), in their study, indicate that the change of oil price will lead to the variation of exchange rate because of the dependence on oil imports. Nearly all euro area countries get oil depend on imports, thus the increasing of oil price will cause the euro falls. The EUR/USD pair experiences its highest trading volumes and volatility during the European and North American sessions, making it becomes a favoured choice for traders seeking to capitalize on these active periods. Often viewed as a safe-haven pair during market uncertainty, EUR/USD benefits from being composed of two major reserve currencies. Compared with other currencies, the exchange rate of EUR/USD has the highest liquidity (Mancini et al.,2009). Its price movements are generally well-suited to technical analysis due to its high liquidity and broad market participation. Political events and policy changes in the Eurozone and the US also significantly impact the EUR/USD exchange rate. This makes it an attractive and dynamic pair for various trading strategies, from short-term day trading to long-term positions.

The use of high-frequency data in this study allows for capturing the intraday dynamics and subtle market movements of the EUR/USD pair. High-frequency data, characterized by a very fine granularity, provides detailed information about market behaviour over short time intervals, such as one-minute ticks used in this analysis. This level of detail is particularly useful for understanding how quickly markets react to new information, for example, economic indicator and policy changes. High-frequency data can highlight patterns and market anomalies that may not visible in lower-frequency data, making it invaluable for trading strategy development and risk management. Additionally, the richness of high-frequency data enables more sophisticated modelling techniques. For example, those used in machine learning and high-frequency trading algorithms, to predict price movements and assess market volatility with greater accuracy.

Empirical Mode Decomposition (EMD) is a versatile and adaptive signal-processing technique that has proven effective in handling non-stationary signals. By decomposing complex signals into simpler components known as Intrinsic Mode Functions (IMFs), EMD facilitates a detailed analysis of various signal characteristics (Jiusheng et al., 2006). This research aims to utilize IMFs to examine both simulated and real-world signals, providing insights into their underlying patterns and behaviours (Figure 1). Through this approach, the study seeks to enhance our understanding of the dynamic features in non-stationary data, offering potential applications across various fields in forex market signal analysis.

By using EMD method, each signal can be decomposed as follows:

$$x(t) = \sum_{j=1}^{n} C_j + r_n \# \qquad (1)$$

In this context, C_j represents an Intrinsic Mode Function (IMF). Therefore, the signal can be decomposed into n empirical modes and a residual r_n, which represents the mean trend of $x(t)$. The IMFs $c_1, c_2, \ldots, c_n$ span different frequency bands, from high to low frequencies. Each frequency band contains distinct frequency components that vary with changes in the signal.

Consider a non-stationary signal $x(t)$ at consists of a finite number of mutually independent components $xi(t)$, each having an average value of zero.

$$x(t) = x_1(t) + x_2(t) + \ldots + x_n(t) = \sum_{i=1}^{n} x_i(t) \#(2)$$

The total energy of signal $x(t)$ is computed by integrating the square of the signal over its full-time duration.

$$E_x = \int_{-\infty}^{\infty} x(t)^2 \, dt = \int_{-\infty}^{\infty} \left(\sum_{i=1}^{n} x_i(t)^2 \right) dt \#(3)$$

Because of the orthogonality between $\{x_i(t), i = 1,2, \ldots, n\}$, the expression simplifies or can be rewritten as:

$$\int_{-\infty}^{\infty} x_i(t) x_j(t) \, dt \approx 0, \quad i \neq j \# \quad (4)$$

and then signal $x(t)$ has total energy as

$$E_x = \int_{-\infty}^{\infty} \left(\sum_{i=1}^{n} x_i(t) \right)^2 dt = \int_{-\infty}^{\infty} x_1^2(t) \, dt + \int_{-\infty}^{\infty} x_2^2(t) \, dt + \cdots + \int_{-\infty}^{\infty} x_n^2(t) \, dt = E_1 + E_2 + \cdots + E_n \#(5)$$

Where E_i represents the energy of the i-th component $x_{i(t)}$ of $x(t)$. If Empirical Mode Decomposition (EMD) is used to decompose the signal and it is assumed that the component $c_{1(t)}$ is precisely the orthogonal component $x_{1(t)}$ of $x(t)$, then after $c_{1(t)}$ has been separated from $x(t)$, the residual signal energy is calculated as follows:

$$E_{2,\ldots,n} = \int_{-\infty}^{\infty} \left(\sum_{i=2}^{n} X_i(t) \right)^2 dt = \int_{-\infty}^{\infty} \left(\sum_{i=2}^{n} X_i^2(t) \right) dt \#(6)$$

Thus, the total energy of the signal $c_1(t)$, which has been isolated, along with the energy of the remaining signal, referred to as E_{total}, equals the energy of the original signal E_x.

$$E_{\text{total}} = E_1 + E_{2,\ldots,n} = E_1 + E_2 + \ldots + E_n = E_x \#(7)$$

If a certain component $c_1(t)$ is not orthogonal to x(t), then after $c_1(t)$ is separated from x(t), the total energy of $c_1(t)$ and the remaining signal, referred to as E_{total} , would be

$$E_{\text{total}} = \int_{-\infty}^{\infty} c_1^2(t) \, dt + \int_{-\infty}^{\infty} [x(t) - c_1(t)]^2 \, dt = E_{c_1} + \int_{-\infty}^{\infty} [x^2(t) - 2x(t)c_1(t) + c_1^2(t)] \, dt = 2E_{c_1} + E_x - 2\int_{-\infty}^{\infty} x(t)c_1(t) \, dt \#(8)$$

In a broad sense, we can assume that

$$c_1(t) = Ax_i(t) + e(t) \# \qquad (9)$$

In this context, A represents a constant, and e(t) denotes the error component of $c_1(t)$, which is completely independent from $x_i(t)$ for all $i = 1, 2, \ldots, n$. This independence implies that e(t) is orthogonal to $x_i(t)$ across all dimensions iii.

In this context, C_j represents an Intrinsic Mode Function (IMF). Therefore, the signal can be decomposed into n empirical modes and a residual r_n, which represents the mean

trend of $x(t)$. The IMFs $c_1, c_2, \ldots, c_n$ span different frequency bands, from high to low frequencies. Each frequency band contains distinct frequency components that vary with changes in the signal.

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) was developed to address the limitations of traditional EMD by introducing adaptive noise in a systematic manner. Building on Ensemble Empirical Mode Decomposition (EEMD), which reduces mode mixing through the addition of white noise, CEEMDAN further enhances the decomposition process by scaling noise based on the signal's characteristics at each step. This method ensures a more accurate, complete, and data-driven decomposition, effectively isolating distinct frequency components and minimizing noise-related distortions. CEEMDAN's ability to handle non-stationary and nonlinear data, along with its reduced sensitivity to noise, makes it particularly valuable for analyzing and forecasting financial time series.

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) is an advanced ensemble learning method designed to overcome the limitations of traditional Empirical Mode Decomposition (EMD). CEEMDAN introduces adaptive noise systematically during the decomposition process, which enhances the separation of different frequency components within the time series, reducing mode mixing and ensuring a more accurate and data-driven decomposition (Torres et al., 2011). This makes it particularly effective in handling the complexities of non-stationary and nonlinear data, such as exchange rates.

To evaluate the decomposition's effectiveness, CEEMDAN follows these steps:

1. Initial Noise Addition:

A small amount of white noise is added to the original signal, and EMD is performed to obtain the first Intrinsic Mode Function (IMF).

The noise is then removed, and the process is repeated with different noise realizations. The resulting IMFs are averaged to obtain the first ensemble IMF.

2. Adaptive Noise Addition:

For subsequent IMFs, adaptive noise, which is scaled based on the characteristics of the residual signal, is added to the residual signal (the original signal minus the sum of all previously obtained IMFs).

EMD is applied to the noise-added residual, and the process is iteratively repeated for each IMF.

The mathematical expression for the CEEMDAN process can be represented as:

$$x(t) = \sum_{i=1}^{n} \mathrm{IMF}_i(t) + r(t)\# \qquad (10)$$

Where: $x(t)$ is the original signal, $\mathrm{IMF}_i(t)$are the Intrinsic Mode Functions obtained from CEEMDAN, $r(t)$ is the residual after all IMFs are extracted.

3.2 Models and Estimation

The forecasting process begins with the application of CEEMDAN to the high-frequency EUR/USD exchange rate data, decomposing the original signal into several IMFs. Each IMF captures specific patterns and frequencies within the data, allowing the model to focus on different aspects of the time series separately.

Following the decomposition, each IMF is input into the LSTM network, which is trained to predict future values based on the patterns identified in the IMFs. The LSTM network's architecture is designed to capture the temporal dependencies within each IMF, leveraging its ability to remember long-term sequences of data. By training the LSTM on each IMF individually, the model can generate more accurate forecasts that account for both short-term fluctuations and long-term trends.

The final forecast is obtained by aggregating the predictions from all IMFs, reconstructing the original time series with enhanced accuracy. This hybrid approach combines the strengths of both CEEMDAN and LSTM, leading to a more robust and reliable forecasting model.

3.3 Forecasting

The process begins by preparing time series data using a sliding window approach. The Time Series function segments the dataset into feature-label pairs, where the features are sequences of past data points and the labels are the future data points to be predicted. The get_tain_val_test function then divides this data into training, validation, and testing sets, while standardizing them to maintain consistent scaling across all sets.

Next, an LSTM (Long Short-Term Memory) model is defined and trained. The model consists of an LSTM layer with 30 units that processes the time series data and captures temporal dependencies, followed by a Dense layer for producing the final prediction. The model is compiled with the mean squared error loss function and is trained over 20 epochs using the training data, with validation data used to monitor performance during training.

Finally, the trained model predicts the values in the test set. The actual and predicted values are then plotted together to compare how closely the model's predictions align with the real data. This comparison is crucial for evaluating the model's effectiveness in capturing the underlying patterns in the time series.

The process involves defining four functions to evaluate the performance of the LSTM model using different error metrics. The mean absolute error (MAE) and mean squared error (MSE) functions calculate the average absolute and squared differences between the actual and predicted values, providing a measure of the model's accuracy. The harmonic mean absolute error (HMAE) and harmonic mean squared error (HMSE) functions extend these metrics by normalizing the errors relative to the mean of the predicted values, offering a standardized way to assess the magnitude of errors compared to the predicted data. Finally, the calculated error metrics are printed, allowing for a quick evaluation of the model's performance in predicting the test set.

The process starts by iterating through each column in the decompose_data dataframe, which contains different decomposed components (such as IMFs) of a time series. For each component, the get_tain_val_test function is applied to split the data into training, validation, and testing sets using a sliding window approach. The time series data is then standardized, and an LSTM model is implemented and trained using the implement_LSTM function. The trained model is used to predict the test data, and the predicted values along with the actual test values are stored in respective lists. This process is repeated for each time series component.

The process involves utilizing a Support Vector Regression (SVR) model to predict time series data. Initially, the SVR model is trained using the combined training data consisting of x_train_all and y_train_all. After training, predictions are made on the training, validation, and test datasets separately, resulting in y_pre_all_svr1, y_pre_all_svr2, and y_pre_all_svr3. These predicted values are then concatenated horizontally to create a single array of predictions, referred to as y_pre_all_svr. Similarly, the actual values from the training, validation, and test sets are concatenated to form y_all_label. Finally, a plot is generated to visually compare the actual values with the predicted values across the entire dataset.

The process involves applying an AutoRegressive (AR) model to predict a time series. Initially, the time series data (RVs) is split into training and testing sets, with 90% of the data allocated for training. An AR model of order 2, which uses two lagged values, is then fitted to the training data. After fitting, the model generates predictions in two stages: it first predicts the values for the training set to assess the fit (predict_AR_1), and then it forecasts the future points for the test set (predict_AR_2). These predictions are combined into a single continuous series (predict_all_ar). The actual values from the entire dataset are also combined to create y_all_label, allowing for a comparison between the actual and predicted values.

4. Evaluation of Forecast Methods

4.1 Support Vector Regression

Support Vector Regression (SVR) is the most common regression analysis method based on Support Vector Machines (SVM) (Basak et al., 2007). The core idea of SVR is to find the optimal regression function by introducing the "ε-insensitive loss function," where most data points lie within the "ε-tube" of the regression function, then create a multi-objective function from the loss function and the geometrical properties of the tube (Awad et al., 2015). Meanwhile, it keeps the model complexity as low as possible. SVR excels at handling high-dimensional data and nonlinear relationships. By using different kernel functions, SVR is able to capture nonlinear patterns in data, making it suitable for complex time series forecasting tasks, such as exchange rate prediction.

4.2 Autoregressive Model

Autoregressive Model (AR) is a linear time series model that assumes the current value is a linear combination of past values plus a random error term. The key factor in AR models is the "lag order," which refers to how many past time points' values are used to predict the current value. Simple AR models are effective in capturing linear trends and seasonal components in time series, but they may perform poorly when dealing with nonlinear relationships or structural changes.

4.3 Heterogeneous Autoregressive Model

Heterogeneous Autoregressive Model (HAR) is firstly introduced by Corsi (2009) to realize the goal of presenting the primary observable characteristics of financial returns in a highly manageable and concise way. It is an extended time series model specifically designed for financial data with heterogeneous volatility. As a multi-scale autoregressive model, it considers volatility over different time scales, such as daily, weekly, and monthly, allowing it to better capture the volatility structure in financial markets.

4.4 Functions

Mean Squared Error (MSE) is adopted as one of the primary evaluation metrics to assess the performance of the LSTM-CEEMDAN forecasting technique.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \# \qquad (11)$$

In this context, n stands for the number of forecast samples, y_i represents the actual values of the exchange rate, and$\hat{y}_i$ is the predicted values of the exchange rate. By squaring the error $(y_i - \hat{y}_i)$, the model eliminates the effect of the error's sign and leads all errors contribute positively. The process of squaring also amplifies larger errors, increasing the model's sensitivity to significant deviations.

MSE provides a direct measure of the performance of the model in foreign exchange rate forecasting. The smaller the MSE values are, the closer the model's predictions are to the actual exchange rates values, which indicates a higher prediction accuracy. Since MSE is more sensitive to larger errors, it helps assess how the model performs when dealing with large fluctuations in the market.

To assess the accuracy of the LSTM-CEEMDAN method, Mean Absolute Error (MAE) is also applied. MAE reflects the average absolute difference between the predicted values and the actual values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2 \# \qquad (12)$$

Same as MSE, n represent the number of forecast samples, y is the actual values of exchange rate while $\hat{y}_i$ is the forecast values of exchange rate here. The absolute error, $|y_i - \hat{y}_i|$, sum up values of each time point. This operation ensures that every part of the error contribute to the total error, no matter it is negative or positive. Compared to MSE, MAE is less sensitive to extreme errors. Thus, it can more fairly reflect the performance of the method.

Comparing with MSE, it magnifies the impact of larger errors due to squaring and making it more sensitive to fluctuations in the model's bias. If it is crucial to focus on large prediction mistakes, MSE is typically a more appropriate method. On the other hand, MAE is more suited for evaluating the overall deviation across all forecasts.

Except for MAE and MSE, their variations Harmonic Mean Absolute Error (HMAE) as well as Harmonic Mean Squared Error (HMSE) are used in this study to analyse the reliability of the LSTM-CEEMDAN method.

Compared with MAE, HMAE uses

harmonic mean instead of the arithmetic mean in the function. The harmonic mean is often used in situations that user want to average rates or ratios and gives more weight to smaller values.

$$HMAE = \frac{n}{\Sigma_{\mathrm{i}=1}^{n} \frac{1}{|e_i|}} \# \qquad (13)$$

In this context, n still means the number of forecast samples, e_i represents the error for the ith prediction, calculated as the difference between the predicted value and the actual value.

The HMAE metric tends to penalize large errors more than MAE since the harmonic mean is influenced more by smaller values. It is particularly useful when reducing outliers' impact on the overall error metric.

HMSE is also the variation that uses harmonic mean instead of the arithmetic mean in the function.

$$HMSE = \frac{n}{\sum_{i=1}^{n} \frac{1}{e_i^2}} \# \qquad (14)$$

The meanings of n and e_i are same as in HMAE. Also, similar to HMAE, HMSE emphasizes smaller errors due to the harmonic mean's properties, while still squaring the errors like MSE, which penalizes larger errors more heavily. The squared term ensures that larger errors still have a significant impact on the metric, but the harmonic mean reduces the overall influence of outliers compared to MSE.

5. Results

These IMFs represent the high-frequency components of the signal, typically containing more noise or detailed information. The waveforms in IMF 1 to IMF 4 show rapid fluctuations, indicating that these parts have captured fast changes within the signal. These components are crucial for analyzing high-frequency noise or sudden events in the signal.

Mid-Frequency Component Analysis (IMF 5 to IMF 8):

These IMFs display the mid-frequency components of the signal, often associated with periodic trends or specific patterns. The waveforms in IMF 5 to IMF 8 gradually become smoother, suggesting that these components may have captured cyclical patterns or regular oscillations

within the signal. This makes them useful for periodic analysis or trend detection.

Low-Frequency Component Analysis (IMF 9 to IMF 11):

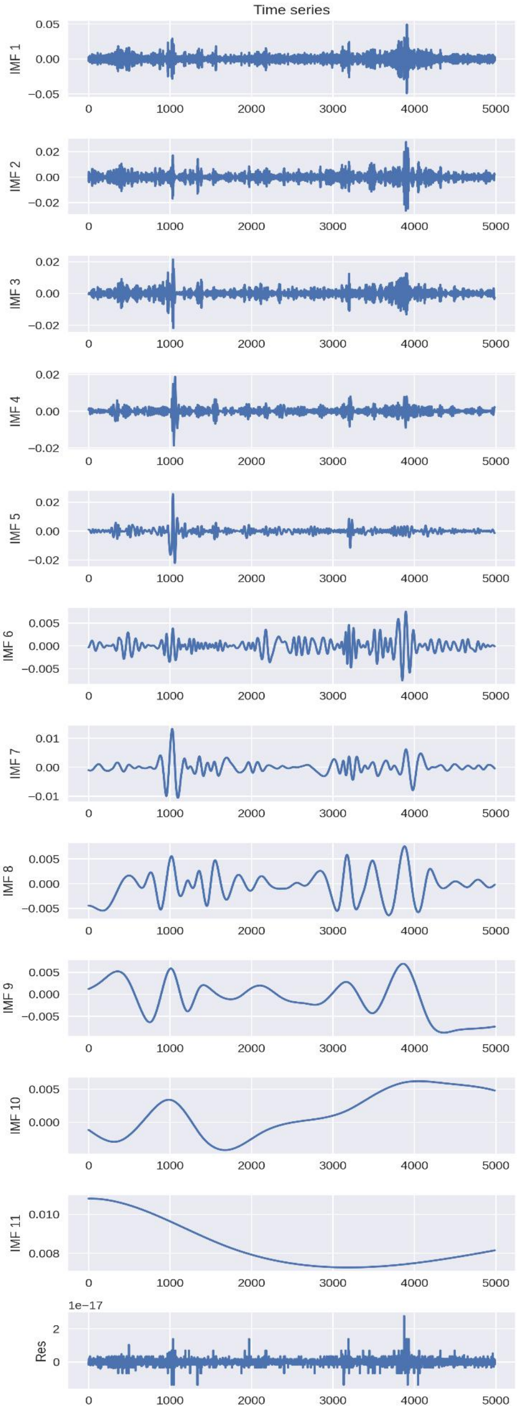


Figure 1. High-Frequency Component Analysis (IMF 1 to IMF 4):

These IMFs represent the low-frequency

Table 1. Analysis of IMF1-IMF11

IMF	count	mean	std	skew	kurtosis	J-B	Q(10)
IMF1	4987	-0.00032	0.005877	-0.0016	6.106947	7749.552	1188.52
IMF2	4987	-1.22E-05	0.003064	0.055941	10.19024	21579.89	4208.493
IMF3	4987	-4.22E-05	0.00265	-0.02754	6.912068	9928.228	9004.275
IMF4	4987	-5.46E-06	0.002144	0.422843	15.76393	51785.11	14028.55
IMF5	4987	-0.00013	0.002698	-0.87629	25.99861	141090.4	27043.33
IMF6	4987	-8.67E-06	0.001556	0.02678	4.050747	3410.157	39659.9
IMF7	4987	-5.15E-05	0.002371	-0.00985	7.69091	12290.98	46714.34
IMF8	4987	-0.00042	0.002752	0.026669	0.065694	1.487907	49135.19
IMF9	4987	-0.00099	0.004116	-0.30251	-0.71474	182.2132	49719.57
IMF10	4987	0.00138	0.003373	0.011835	-1.34242	374.575	49847.47
IMF11	4987	0.008356	0.001165	0.958886	-0.56006	829.4039	49697.13
Res	4987	3.311972e-20	1.835131e-18	1.017122	22.99659	110749	36.30548

components of the signal, typically related to overall trends or long-term fluctuations. The waveforms in IMF 9 to IMF 11 are smooth and slow, reflecting the main trend or long-term patterns in the signal. These parts are important for trend analysis or understanding the fundamental structure of the signal.

Residual (Res) Analysis:

The residual component shows what remains after extracting all the IMFs from the signal, and it should typically be close to zero. The residual in the graph is very small, nearly zero, indicating that the EMD successfully captured the main components of the signal. The near-zero residual suggests that the decomposition was accurate and effectively removed noise and details while retaining the key information in the signal.

According to Table 1, analysis is below:

Mean and Standard Deviation:

The mean values for all IMFs and the residual are very close to zero, which is typical for components extracted by Empirical Mode Decomposition (EMD). This indicates that the decomposition has effectively separated the signal into components without introducing significant bias.

The standard deviations vary, with IMF 1 having the highest standard deviation (0.005877), indicating that it contains the most variability or noise. As the IMF number increases, the standard deviation generally decreases, especially in the higher IMFs, indicating that these components capture less variability and represent smoother, low-frequency trends.

Skewness:

Skewness measures the asymmetry of the distribution. Most IMFs have skewness values close to zero, indicating a fairly symmetric distribution. However, IMF 5 has a relatively high negative skewness (-0.876294), which suggests a longer tail on the left side, possibly indicating the presence of outliers or sudden drops in the signal.

IMF 11 has a positive skewness (0.958886), indicating a longer tail on the right side, suggesting occasional spikes or upward trends in this component.

Kurtosis:

Kurtosis measures the "tailedness" of the distribution. Several IMFs, particularly IMF 2, IMF 4, and IMF 5, have high positive kurtosis values, indicating that these components have heavy tails and are more prone to producing extreme values. This suggests that these IMFs might capture significant events or anomalies within the original signal.

Negative kurtosis in IMF 9 and IMF 10 suggests a distribution flatter than a normal distribution, possibly indicating less pronounced peaks or troughs in these components.

Jarque-Bera (J-B) Statistic:

The Jarque-Bera test statistic is used to test whether the data has the skewness and kurtosis matching a normal distribution. A higher J-B statistic indicates a departure from normality. IMF 5 shows an extremely high J-B statistic (141090.403401), indicating a significant deviation from normality, likely due to its skewness and high kurtosis.

Similarly, IMF 2, IMF 3, IMF 4, and IMF 7 also show high J-B values, indicating that these components do not follow a normal distribution and might contain significant non-Gaussian features.

Q(10) Statistic:

The Q(10) statistic measures autocorrelation within the IMFs over 10 lags. High values indicate significant autocorrelation, suggesting that these components contain patterns or cycles that persist over time.

IMF 10 and IMF 11 have extremely high Q(10) values, suggesting strong autocorrelation and indicating these components likely represent long-term trends or persistent cycles in the original signal.

The lower IMFs (IMF 1 through IMF 6) show lower Q(10) values, suggesting that these components are less correlated and may capture more noise or short-term variations.

The analysis of the IMFs and residual reveals that the Empirical Mode Decomposition (EMD) effectively separated the original signal into distinct components. The lower IMFs (IMF 1 to IMF 5) capture high-frequency variations and noise, characterized by significant non-normality and occasional extreme values. In contrast, the higher IMFs (IMF 6 to IMF 11) reflect lower-frequency trends and persistent patterns, with strong autocorrelation particularly evident in IMF 10 and IMF 11. The residual component shows minimal variance, indicating that the decomposition accurately captured the main features of the signal.

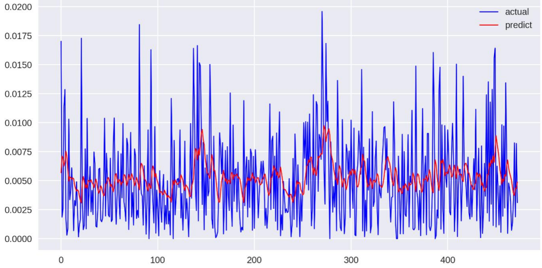


Figure 2 .LSTM Model of the Actual Time Series Data

The Figure 2 shows that the LSTM model effectively follows the general trend of the actual time series data. The predicted values (red line) align reasonably well with the actual values (blue line), indicating that the model has successfully learned the overall pattern in the data.

However, the predictions are noticeably smoother than the actual data, failing to capture the more extreme spikes and fluctuations. This suggests that while the model is good at predicting the central trend, it may not be as effective at handling high-frequency noise or sudden changes in the data.

The analysis of the model's performance, based on the provided metrics, indicates that the LSTM model has achieved reasonable accuracy in predicting the time series data. The Mean Absolute Error (MAE) of 0.0034 suggests that, on average, the model's predictions are very close to the actual values, with an error margin of approximately 0.0034 units. The Mean Squared Error (MSE) of 1.73e-05 further supports this, indicating that the squared deviations between predicted and actual values are minimal, reflecting a good overall fit.

Figure3 Higher-Frequency Components (Top Panels)

However, the Harmonic Mean Absolute Error (HMAE) of 0.6466 and Harmonic Mean Squared Error (HMSE) of 0.6208 suggest that when normalized relative to the mean of the predicted values, the errors are relatively significant. This implies that while the model captures the general trend, there are still considerable discrepancies, especially when considering the scale of the predictions. This aligns with the earlier observation from the plot where the model successfully follows the overall pattern but smooths out sharp fluctuations, leading to underestimation or overestimation in regions with high volatility or spikes.

In summary, the LSTM model is effective in capturing the main trends of the time series but may require further refinement or additional features to better handle extreme variations and reduce the normalized errors.

For the higher-frequency components of CEEMDAD, the predictions closely follow the actual data, though some deviations are evident. The model captures the general oscillatory behavior, but it smooths out some of the sharper peaks and troughs. This is expected, as LSTM models often tend to smooth out noise while capturing underlying trends.

Mid-Frequency Components (Middle Panels):

In the middle-frequency components of CEEMDAD, the model performs well, with predictions almost mirroring the actual values. The alignment between the red and blue lines is strong, indicating that the model is effectively capturing the key patterns in these components. This suggests that the LSTM model is particularly adept at modeling the mid-range frequencies where the signal's structure is more regular and less noisy.

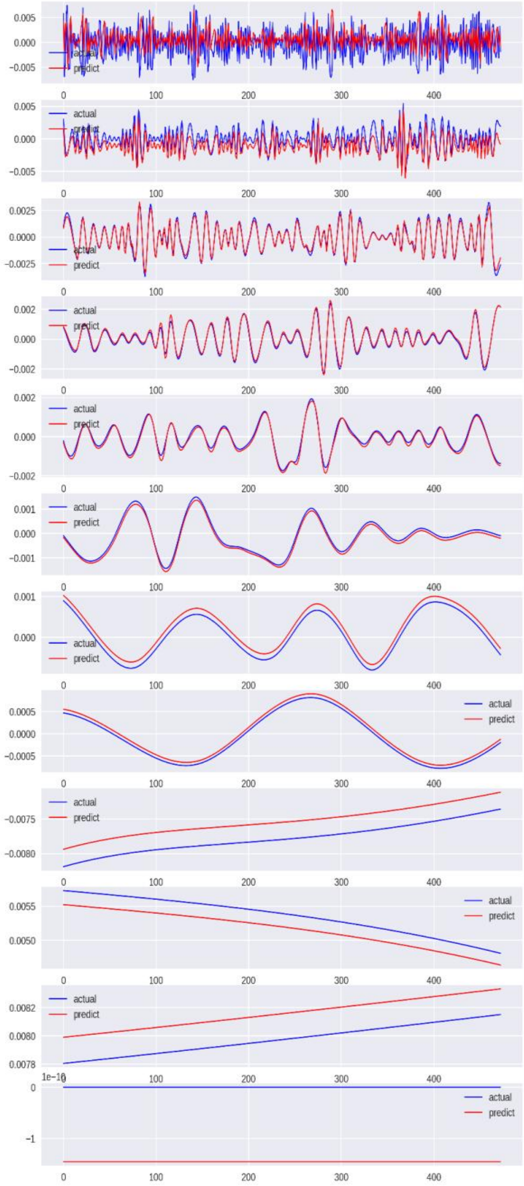


Figure 3. Higher-Frequency Components (Top Panels)

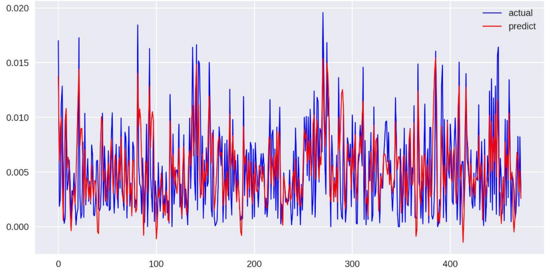


Figure 4. Prediction of IMFs

Low-Frequency Components (Bottom Panels):

For the low-frequency components of CEEMDAD, the predictions are very close to the actual values, with minimal discrepancies. The model effectively captures the long-term trends in the data, as seen by the near-perfect overlap of the predicted and actual lines. The low-frequency trends are generally easier to predict due to their smoother, more predictable nature, which the LSTM model handles well.

Based on Figure 4, the aggregated prediction, which sums the individual predictions from the decomposed components (IMFs), closely follows the actual time series data, indicating that the model effectively captures the overall trends and patterns. However, the predicted values sometimes lag or overshoot the actual values, particularly in highly volatile regions with sharp spikes. This suggests that while the model performs well in approximating the general dynamics of the time series, it struggles with accurately predicting rapid fluctuations, highlighting a common challenge in time series forecasting that may require further refinement for improved accuracy.

The error metrics indicate that the LSTM model's performance is fairly accurate in predicting the time series after aggregating the predictions of all decomposed components. The Mean Absolute Error (MAE) of 0.0024 suggests that, on average, the predictions are very close to the actual values, with a small error margin. The Mean Squared Error (MSE) of 9.98e-06 reflects even smaller squared deviations between the predicted and actual values, indicating a good fit overall. However, the Harmonic Mean Absolute Error (HMAE) of 0.4553 and Harmonic Mean Squared Error (HMSE) of 0.3541 suggest that when the errors are normalized relative to the mean of the predictions, the discrepancies are still somewhat significant. This indicates that while the model captures the overall trend effectively, it may still struggle with certain patterns, particularly in areas with sharp changes or higher volatility, as seen in the earlier analysis.

The figure 5 shows the actual time series values in blue and the predicted values from the SVR model in red. The results indicate that the SVR model produces a relatively constant prediction line that does not capture the variability present in the actual data. This flat prediction suggests that the model is either underfitting or failing to learn the underlying patterns in the data. The SVR's inability to follow the peaks and valleys of the actual time series data points to a significant mismatch between the model's predictions and the true values, especially in regions with high volatility or rapid changes. This outcome implies that the SVR model may not be well-suited for this particular time series, possibly due to the complexity or noise within the data that the model struggles to handle.

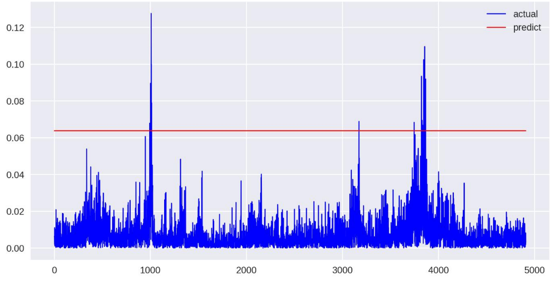


Figure 5. Actual Time Series Values and the Predicted Values from the SVR Model

The error metrics indicate that the SVR model's performance in predicting the time series data is suboptimal. The Mean Absolute Error (MAE) of 0.0588 suggests that the model's predictions, on average, deviate significantly from the actual values. The Mean Squared Error (MSE) of 0.00347 further emphasizes this, indicating larger squared deviations and pointing to substantial prediction errors. The Harmonic Mean Absolute Error (HMAE) of 0.9210 and Harmonic Mean Squared Error (HMSE) of 0.8523 reveal that when these errors are normalized relative to the mean of the predictions, the discrepancies remain notably high. These metrics, combined with the flat prediction line observed in the plot, suggest that the SVR model is underfitting and failing to capture the underlying patterns in the data, leading to poor predictive performance, especially in capturing the time series' volatility and variability.

The figure 6 compares the actual time series values (in blue) with the predicted values from the AR model (in red). The AR model's predictions are relatively smooth and fail to capture the sharp spikes and variability present in the actual data. The predicted values tend to stay within a narrow range, which indicates that the AR model is underfitting the data, especially in periods of high volatility. The model does not effectively track the large fluctuations seen in the actual time series, particularly during peak periods, resulting in significant discrepancies. This suggests that the AR model, while potentially useful for capturing general trends in simpler or more stable time series, may not be well-suited for complex data with rapid changes and high volatility as seen in this case.

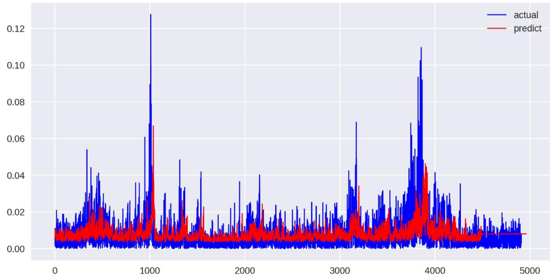
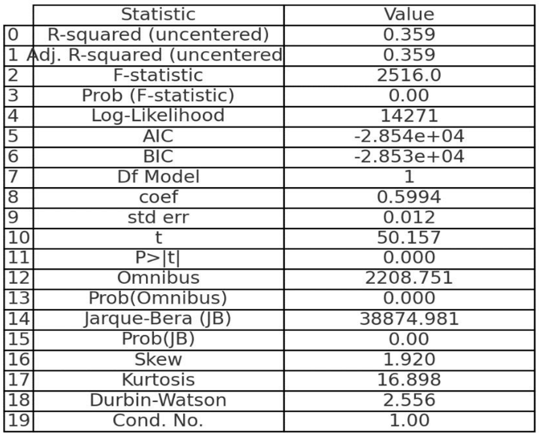


Figure 6. Comparison of Actual Time Series Values with the Predicted Values from the AR Model

The error metrics suggest that the AutoRegressive (AR) model performs moderately well in predicting the time series data. The Mean Absolute Error (MAE) of 0.0044 indicates that the model's predictions are, on average, relatively close to the actual values, though some discrepancies are present. The Mean Squared Error (MSE) of 2.53e-05 further supports this, showing that the squared deviations between the predicted and actual values are relatively small, but not negligible. The Harmonic Mean Absolute Error (HMAE) of 0.5450 and Harmonic Mean Squared Error (HMSE) of 0.3914 reveal that, when normalized relative to the mean of the predictions, the errors remain somewhat significant. This suggests that while the AR model captures the general trend of the time series to some extent, it struggles with accurately predicting rapid changes and high volatility, as indicated by the smoother prediction line seen in the plot compared to the actual data's sharp fluctuations.



	Statistic	Value
0	R-squared (uncentered)	0.359
1	Adj. R-squared (uncentered	0.359
2	F-statistic	2516.0
3	Prob (F-statistic)	0.00
4	Log-Likelihood	14271
5	AIC	-2.854e+04
6	BIC	-2.853e+04
7	Df Model	1
8	coef	0.5994
9	std err	0.012
10	t	50.157
11	P>|t|	0.000
12	Omnibus	2208.751
13	Prob(Omnibus)	0.000
14	Jarque-Bera (JB)	38874.981
15	Prob(JB)	0.00
16	Skew	1.920
17	Kurtosis	16.898
18	Durbin-Watson	2.556
19	Cond. No.	1.00

Figure 7. OLS Regression Results

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The OLS regression results indicate that the model explains approximately 35.9% of the variance in the dependent variable, as reflected by the R-squared and adjusted R-squared values. The coefficient for RV is 0.5994 and is statistically significant, with a t-value of 50.157 and a p-value of 0.000, indicating a strong positive relationship between RV and the dependent variable. The F-statistic of 2516.0, with a corresponding p-value of 0.00, suggests that the model as a whole is statistically significant. The Durbin-Watson statistic of 2.556 suggests there is no significant autocorrelation in the residuals. However, the high values for the Omnibus and Jarque-Bera tests, along with a skewness of 1.920 and kurtosis of 16.898, indicate that the residuals may deviate from normality. While the model is statistically significant, the relatively low R-squared and potential issues with the residuals suggest that the model may not fully capture all the underlying patterns in the data, and further refinement may be necessary.

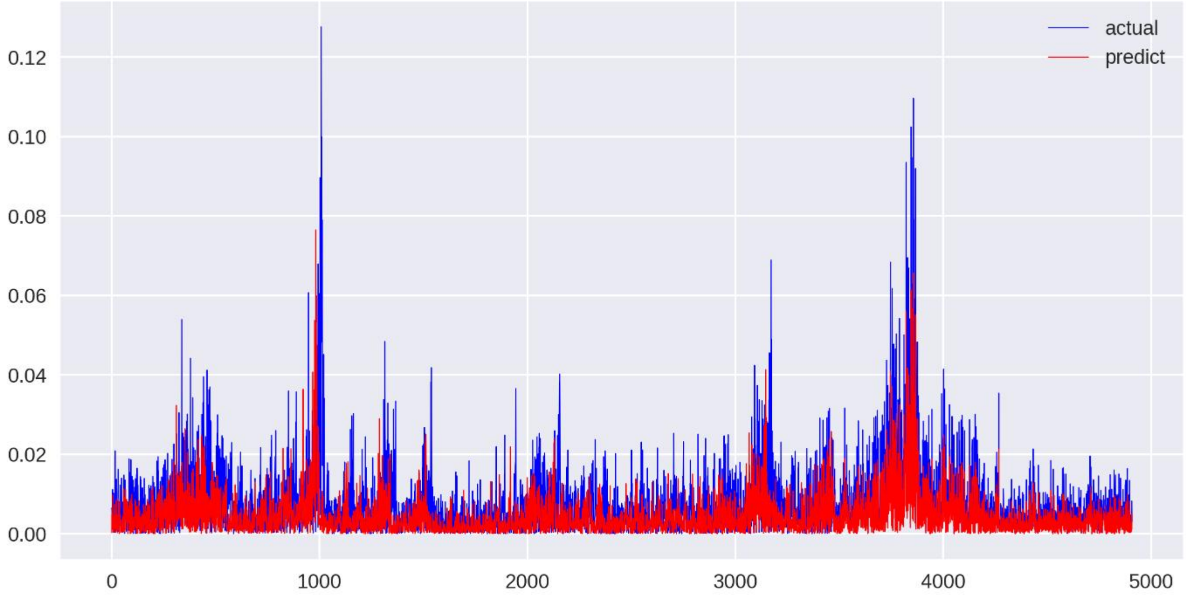


Figure 8. Comparison between the Actual Time Series Values and the Predicted Values

The figure 8 shows the comparison between the actual time series values (in blue) and the predicted values (in red) from the model. The predictions closely follow the general trend of the actual data, particularly in periods of lower volatility. However, the model struggles to capture the sharp spikes and significant fluctuations seen in the actual data, particularly during periods of high volatility. This is reflected in the relatively smooth prediction line compared to the more jagged actual data. The discrepancy suggests that while the model is effective at capturing the broader trends, it may be underfitting, particularly when it comes to modeling extreme values or rapid changes in the time series. This is consistent with the earlier analysis of the model's performance metrics, which indicated that while the model is statistically significant, it does not fully capture all the underlying patterns in the data, leading to some missed dynamics, especially in volatile regions.

The error metrics indicate that the model performs moderately well in predicting the time series but has some significant shortcomings. The Mean Absolute Error (MAE) of 0.0039 suggests that the predictions are relatively close to the actual values on average, which is a positive sign. The Mean Squared Error (MSE) of 2.67e-05, while low, also indicates that the squared deviations are small, though not negligible. However, the Harmonic Mean Absolute Error (HMAE) of 1.2965 and the Harmonic Mean Squared Error (HMSE) of 2.9203 are quite high, indicating that when errors are normalized relative to the mean of the predictions, they become significantly more pronounced. This suggests that while the model captures the general trend, it struggles with larger errors in certain parts of the time series, especially in areas with high volatility or extreme values, as reflected in the plot. The model's difficulty in accurately predicting sharp spikes and rapid fluctuations is further emphasized by these higher normalized error metrics, pointing to a potential underfitting issue where the model fails to fully capture the complexity of the data.

6. Conclusions

This article presents a comprehensive analysis of various machine learning models, including LSTM, CEEMDAD, SVR, AR, and HAR, in forecasting exchange rates, particularly focusing on the EUR/USD pair. The results indicate that the **LSTM model** combined with **CEEMDAD decomposition** effectively captures the general trends in the time series, especially in mid and low-frequency components, but struggles with high-frequency fluctuations, leading to some underfitting. The **SVR model**, however, shows significant limitations, producing overly smooth predictions that fail to capture the volatility and variability of the actual data, making it less suitable for this time series. The **AR model** performs moderately well in simpler, less volatile periods but fails to adequately capture sharp spikes and rapid changes, resulting in a smooth prediction line that misses key fluctuations. The **HAR model** shows similar challenges, with difficulty in predicting extreme values or rapid changes, despite capturing the broader trends. The overall conclusion suggests that while LSTM with CEEMDAD decomposition offers the most promise among the models, there is still a need for further refinement to better handle the complexities and volatility inherent in exchange rate time series data.

References

[1]Apergis, N., Zestos, G. K., & Shaltayev, D. S. (2012). Do market fundamentals determine the Dollar–Euro exchange rate? *Journal of Policy Modeling, 34*(1), 1-15. https://doi.org/10.1016/j.jpolmod.2011.10.003

[2]Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). Support vector regression. In *Efficient learning machines: Theories, concepts, and applications for engineers and system designers* (pp. 67-80).

[3]Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing-Letters and Reviews, 11*(10), 203-224.

[4]Cao, W., Zhu, W., Wang, W., Demazeau, Y., & Zhang, C. (2020). A deep coupled LSTM approach for USD/CNY exchange rate forecasting. *IEEE Intelligent Systems, 35*(2), 43-53. https://doi.org/10.1109/MIS.2020.2977283

[5]Carriero, A., Kapetanios, G., & Marcellino, M. (2009). Forecasting exchange rates with a large Bayesian VAR. *International Journal of Forecasting, 25*(2), 400-417. https://doi.org/10.1016/j.ijforecast.2009.01.007

[6]Chortareas, G., Jiang, Y., & Nankervis, J. C. (2011). Forecasting exchange rate volatility using high-frequency data: Is the euro different? *International Journal of Forecasting, 27*(4), 1089-1107. https://doi.org/10.1016/j.ijforecast.2010.07.003

[7]Clostermann, J., & Schnatz, B. (2000). The determinants of the Euro-Dollar exchange rate - Synthetic fundamentals and a non-existing currency. *Deutsche Bundesbank Working Paper No. 02/00.* http://doi.org/10.2139/ssrn.229472

[8]Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics, 7*(2), 174-196.

[9]Demir, F., & Razmi, A. (2022). The real exchange rate and development theory, evidence, issues and challenges. *Journal of Economic Surveys, 36*(2), 386-428. https://doi.org/10.1111/joes.12418

[10]Gonzalez, J., & Yu, W. (2018). Non-linear system modeling using LSTM neural networks. *IFAC-PapersOnLine, 51*(13), 485-489. https://doi.org/10.1016/j.ifacol.2018.07.326

[11]Granger, C. W., & Poon, S. H. (2001). Forecasting financial market volatility: A review. *Available at SSRN 268866.*

http://dx.doi.org/10.2139/ssrn.268866

[12]Guan, Y. (2022). Financial time series forecasting model based on CEEMDAN-LSTM. *2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, 1-5. https://doi.org/10.1109/CTISC54888.2022.9849780

[13]Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[14]Jung, G., & Choi, S. (2021). Forecasting foreign exchange volatility using deep learning autoencoder-LSTM techniques. *Complexity, 2021*, Article ID 6647534, 16 pages. https://doi.org/10.1155/2021/6647534

[15]Lal, M., Kumar, S., Pandey, D. K., Rai, V. K., & Lim, W. M. (2023). Exchange rate volatility and international trade. *Journal of Business Research, 167*, 114156. https://doi.org/10.1016/j.jbusres.2023.114156

[16]Lubecke, T. H., Nam, K. D., Markland, R. E., & Kwok, C. C. Y. (1998). Combining foreign exchange rate forecasts using neural networks. *Global Finance Journal, 9*(1), 5-27. https://doi.org/10.1016/S1044-0283(98)90012-6

[17]Mancini, L., Ranaldo, A., & Wrampelmeyer, J. (2009). Liquidity in the foreign exchange market: Measurement, commonality, and risk premiums.

[18]Meese, R. A., & Rogoff, K. (1983). Empirical exchange rate models of the seventies. *Journal of International Economics, 14*(1-2), 3-24. https://doi.org/10.1016/0022-1996(83)90017-X

[19]Moosa, I. A., & Vaz, J. J. (2016). Cointegration, error correction and exchange rate forecasting. *Journal of International Financial Markets, Institutions and Money, 44*, 21-34. https://doi.org/10.1016/j.intfin.2016.04.007

[20]Ni, L., Li, Y., Wang, X., Zhang, J., Yu, J., & Qi, C. (2019). Forecasting of Forex time series data based on deep learning. *Procedia Computer Science, 147*, 647-652. https://doi.org/10.1016/j.procs.2019.01.189

[21]Torres, M. E., Colominas, M. A., Schlotthauer, G., & Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4144-4147. https://doi.org/10.1109/ICASSP.2011.5947265

[22]Wang, J., He, M., Xu, W., et al. (2023). A deep learning-based nonlinear ensemble approach with biphasic feature selection for multivariate exchange rate forecasting. *Multimedia Tools and Applications, 82*, 22961-22979. https://doi.org/10.1007/s11042-023-14497-9

[23]Wei, Y., Sun, S., Ma, J., Wang, S., & Lai, K. K. (2019). A decomposition clustering ensemble learning approach for forecasting foreign exchange rates. *Journal of Management Science and Engineering, 4*(1), 45-54. https://doi.org/10.1016/j.jmse.2019.02.001

[24]West, K. D., & Cho, D. (1995). The predictive ability of several models of exchange rate volatility. *Journal of Econometrics, 69*(2), 367-391. https://doi.org/10.1016/0304-4076(94)01654-I

[25]Wijesinghe, S. (2020). Time series forecasting: Analysis of LSTM neural networks to predict exchange rates of currencies. *Instrumentation, 7*, 25. https://www.researchgate.net/publication/351062821_Time_Series_Forecasting_Analysis_of_LSTM_Neural_Networks_to_Predict_Exchange_Rates_of_Currencies