

Article

A Novel Hybrid Model (EMD-TI-LSTM) for Enhanced Financial Forecasting with Machine Learning

Olcay Ozupek ¹, Reyat Yilmaz ², Bita Ghasemkhani ^{1,*}, Derya Birant ³ and Recep Alp Kut ³

¹ Graduate School of Natural and Applied Sciences, Dokuz Eylul University, Izmir 35390, Turkey; olcay.ozupek@ogr.deu.edu.tr

² Department of Electrical and Electronics Engineering, Dokuz Eylul University, Izmir 35390, Turkey; reyad.yilmaz@deu.edu.tr

³ Department of Computer Engineering, Dokuz Eylul University, Izmir 35390, Turkey; derya@cs.deu.edu.tr (D.B.); alp@cs.deu.edu.tr (R.A.K.)

* Correspondence: bita.ghasemkhani@ogr.deu.edu.tr

Abstract: Financial forecasting involves predicting the future financial states and performance of companies and investors. Recent technological advancements have demonstrated that machine learning-based models can outperform traditional financial forecasting techniques. In particular, hybrid approaches that integrate diverse methods to leverage their strengths have yielded superior results in financial prediction. This study introduces a novel hybrid model, entitled EMD-TI-LSTM, consisting of empirical mode decomposition (EMD), technical indicators (TI), and long short-term memory (LSTM). The proposed model delivered more accurate predictions than those generated by the conventional LSTM approach on the same well-known financial datasets, achieving average enhancements of 39.56%, 36.86%, and 39.90% based on the MAPE, RMSE, and MAE metrics, respectively. Furthermore, the results show that the proposed model has a lower average MAPE rate of 42.91% compared to its state-of-the-art counterparts. These findings highlight the potential of hybrid models and mathematical innovations to advance the field of financial forecasting.



Citation: Ozupek, O.; Yilmaz, R.; Ghasemkhani, B.; Birant, D.; Kut, R.A. A Novel Hybrid Model (EMD-TI-LSTM) for Enhanced Financial Forecasting with Machine Learning. *Mathematics* **2024**, *12*, 2794. <https://doi.org/10.3390/math12172794>

Academic Editors: Luca Di Persio and Matteo Garbelli

Received: 23 August 2024

Revised: 28 August 2024

Accepted: 5 September 2024

Published: 9 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Financial forecasting aims to empower decision makers to anticipate market movements and asset prices. Price forecasts are crucial in economic decisions and serve as a cornerstone for portfolio construction, risk analysis, and investment strategy development. Accurate price predictions enhance the efficiency and transparency of the financial markets. Methods for price prediction encompass quantitative financial models, statistical analyses, and technical analyses, each with varying degrees of success across different asset classes and market conditions. However, the inherent volatility of markets and the variability of economic indicators make precise price forecasting a formidable challenge. Consequently, the continuous development and refinement of forecasting models are imperative for adaptation and progress. Recent advancements in artificial intelligence (AI) have significantly improved price prediction capabilities. These technologies can extract intelligent patterns from large datasets, yielding more accurate predictions of future price movements. This progress is poised to drive further innovation and revolutionize financial forecasting operations [1].

Machine learning (ML) is a branch of AI dedicated to developing algorithms and mathematical models that enable computers to perform tasks through inductive inference,

mimicking human learning processes. Unlike traditional programming, where explicit instructions dictate behavior, ML models learn from data, identifying patterns, and making decisions based on their acquired knowledge. This process involves training models on vast datasets to recognize complex relationships and generalize from seen examples to unseen scenarios. Such models have led to significant advancements, enabling machines to perform complex tasks with remarkable accuracy and efficiency using data, rather than relying on explicit programming [2]. ML spans various fields, including healthcare [3,4], finance [5–8], retail [9,10], computer vision [11,12], autonomous vehicles [13,14], manufacturing [15,16], entertainment [17,18], and media [19,20]. The continuous evolution of ML solutions is propelled by the development of sophisticated algorithms, advancements in computational power, availability of big data, and ongoing research in the field.

Recent advancements in the application of ML to finance have significantly broadened the scope of financial analysis and prediction, reflecting a growing trend toward integrating these seminal methodologies into mainstream financial practices. For instance, the use of various ML algorithms is explored in [5] to assess financial inclusion, highlighting the potential of these techniques to complement traditional models in evaluating sociodemographic factors influencing financial access. Expanding on this, a comprehensive overview of ML applications is provided in [6], including supervised learning for both cross-sectional and time series data, with advanced material on Gaussian processes and reinforcement learning, illustrating their use in investment management, trading strategies, and derivative modeling. Additionally, a metadata-based systematic review by another study [7] offers a meta-analysis of over 5000 documents, mapping the evolution and current state of ML in finance and revealing key trends that have shaped the field over the past two decades. In addition, a decade-long survey on stock market prediction [8] underscores the increasing accuracy of predictions through advanced ML approaches, such as text data analytics and ensemble methods, while also noting the ongoing challenges posed by the dynamic and erratic nature of market data. Collectively, these works underscore the pivotal role of ML in driving innovation and enhancing decision-making across various financial domains.

Rapid advancements in machine learning have laid the foundation for innovative signal processing techniques, such as empirical mode decomposition (EMD). EMD is a powerful method for analyzing non-linear and non-stationary signals by decomposing them into intrinsic mode functions (IMF). This adaptive technique breaks down complex signals into simpler components, allowing for a detailed examination of the data's underlying structure and dynamics [21]. EMD's ability to operate without predetermined basis functions makes it particularly effective for handling real-world data with inherent variability, a characteristic that complements the adaptive nature of machine learning (ML) techniques. Each IMF captures distinct oscillatory modes, providing insights into the signal's local characteristics and temporal variations. This decomposition process is highly advantageous in financial forecasting, where market data often exhibit complex, non-linear patterns. Financial time series, such as exchange rates and stock prices, can be challenging to model due to their volatility. By applying EMD, significant trends, cyclic behaviors, and fluctuations within a financial time series can be isolated, thereby enhancing the predictive power of the subsequent modeling techniques [22].

In addition to advanced signal processing techniques like EMD, financial forecasting also relies heavily on technical indicators (TI) [23]. TIs are quantitative tools derived from historical price, volume, or open interest data and are essential for analyzing market behavior and making informed trading decisions. Among widely used technical indicators (TIs) are moving averages (MA) [24], which normalize price data to highlight trends; the relative strength index (RSI) [25], which assesses the speed and magnitude of price changes to identify oversold or overbought conditions; and Bollinger bands (BB) [26], which use standard deviations to measure market volatility and potential price levels. TIs are designed to provide an actionable vision by highlighting patterns and trends that might not be immediately apparent from raw data alone. They help analysts understand market momentum, volatility, and trend direction, making it easier to identify potential trading

opportunities and manage risk. For instance, moving averages can signal trend reversals or confirm ongoing trends, while the RSI can indicate when a market may be due for a correction. On the other hand, BBs reveal periods of high or low volatility, which can be crucial for timing trades [27,28].

While technical indicators offer a straightforward and interpretable approach to market analysis, they are often used in conjunction with advanced machine learning models, such as long short-term memory (LSTM), to enhance predictive accuracy. This hybrid approach combines the practical insights of technical indicators with the sophisticated pattern recognition capabilities of LSTMs, resulting in more reliable and precise financial forecasts [29]. LSTM, a variety of recurrent neural networks (RNNs) [30] within the deep learning (DL) framework, excels in capturing temporal sequential interdependencies, making it highly effective for time series prediction. Unlike traditional RNNs, LSTMs address issues such as vanishing and exploding gradients through their architecture, including memory cells and gating mechanisms. These features allow LSTMs to retain and utilize information over long periods and effectively manage the long-term dependencies that are crucial for financial forecasting [31].

LSTM models are ideally suited for sequence prediction tasks where the order and context of elements are important—as in natural language processing (NLP) [32], time series analysis [33], signal processing [34], bioinformatics [35], video processing [36], gaming [37], and healthcare [38]. Their ability to remember past data points makes them valuable for forecasting applications like financial markets [39], energy load prediction [40], and weather forecasting [41], where predictions often depend significantly on historical patterns.

In the current study, by integrating the EMD, TI, and LSTM concepts, we capitalize on the strengths of all three methodologies, combining sophisticated signal decomposition, practical interpretability, and advanced predictive power to achieve superior financial forecasting performance. The key contributions of this research are as follows:

Introduction of a novel hybrid model: This paper proposes a new machine learning method, named EMD-TI-LSTM, which combines empirical mode decomposition (EMD), technical indicators (TI), and long short-term memory (LSTM) for financial forecasting. This hybrid approach is proposed for the first time in academic research.

Enhanced forecasting accuracy: Our proposed model significantly outperformed the conventional LSTM model on widely used financial datasets including BTC, BIST, NASDAQ, and GOLD. It achieved average improvements of 39.56%, 36.86%, and 39.90% in the prediction accuracy, as measured by the MAPE, RMSE, and MAE metrics, respectively. These enhancements are realized through advanced mathematical modeling and algorithmic refinement.

Better results than state-of-the-art studies: The EMD-TI-LSTM method achieved a lower mean absolute percentage error (MAPE) rate of 42.91% compared to state-of-the-art methods, showcasing its outstanding predictive performance. This reduction highlights the mathematical effectiveness of the EMD-TI-LSTM model in improving forecasting precision.

Effective use of EMD: This paper demonstrates the effective application of EMD in the context of financial forecasting, emphasizing its effectiveness in decomposing time series data for improved prediction accuracy.

Technical indicator integration: By incorporating technical indicators in time series analysis, the model enhances its capability to capture market trends and patterns, contributing to more accurate financial forecasts. The integration involves mathematical analysis to improve the model's predictive power.

Comprehensive evaluation: This study conducts thorough evaluations to measure the effectiveness of the EMD-TI-LSTM method, providing a tough comparison against traditional LSTM and state-of-the-art techniques. The evaluation process employs mathematical metrics to ensure accuracy.

Advancement in AI-based financial forecasting: This research underscores the potential of AI-based hybrid models to outperform traditional financial forecasting techniques.

Mathematical advancements in AI are laying the groundwork for future advancements in this field.

The structure of this paper is as follows: Section 2 reviews related work. Section 3 details the proposed EMD-TI-LSTM method. Section 4 presents the experimental studies. Finally, Section 6 concludes the paper with a summary and discusses future work.

2. Related Works

Research on price prediction has explored the use of various different methods. Some studies have been conducted to examine the effectiveness of these methods in financial forecasting. In the following, several notable related studies [42–57] are discussed to provide an in-depth review of progress in the field.

In [42], Gandhmal and Kumar provide a comprehensive review of various approaches used for stock market prediction. Their review covers techniques such as support vector machines (SVMs), artificial neural networks (ANNs), Bayesian models, and fuzzy classifiers. This paper distinguishes between prediction-based and clustering-based methods, showcasing the challenges and limitations of current techniques. It concludes that despite significant progress, forecasting stock markets remains a complex and multifaceted task, which requires a multifaceted approach to achieve higher reliability and efficiency. In [43], a systematic review critically examines the application of artificial intelligence in stock market trading. This research encompasses topics such as portfolio optimization, financial sentiment analysis, stock market prediction, and hybrid models. Their study explores the use of AI approaches, including deep learning and machine learning, to enhance trading strategies, forecast market behavior, and analyze financial sentiment. It also points out the advanced and specialized applications of AI in the stock market.

In [44], the paper reported the significant advancements occurring in foreign exchange (Forex) and stock price prediction through the application of deep learning techniques. It reviews various systematic models based on deep learning, such as convolutional neural networks (CNN), LSTM, deep neural networks (DNN), RNN, and reinforcement learning. Their study evaluates the efficacy of these models in predicting market movements using mathematical metrics such as RMSE, MAPE, and accuracy. The findings indicate an emerging trend in which LSTM is frequently used in conjunction with other models to achieve high prediction accuracy, underscoring the evolving landscape of deep learning in financial market forecasting. Singh et al. [45] developed an enhanced LSTM model that incorporates technical indicators, namely RSI and simple moving average (SMA), for stock price forecasting. Their study highlights the model's superior accuracy and efficiency in predicting stock prices by leveraging these indicators, demonstrating significant improvements over traditional forecasting methods in responding to market trends and patterns more effectively.

In recent studies, several advancements in stock market predictions have been described. For instance, Mittal and Chauhan [46] proposed a model that integrates a range of technical indicators with advanced machine learning techniques, resulting in a significant reduction in error values and an enhancement in forecasting accuracy. Babu and Sathyanarayana [47] established a forecasting model that utilizes technical analysis tools encompassing Bollinger bands, moving averages, and other relevant indicators to boost the reliability of stock price estimates. Kaur et al. [48] demonstrated the effectiveness of combining various parameters and technical indicators within their model, showing a substantial improvement in prediction performance. Venikar et al. [49] investigated the application of a stacked model that takes advantage of extensive historical data and advanced computational techniques to achieve more accurate predictions. These studies collectively address the growing trend of employing sophisticated hybrid approaches and integrated methodologies to significantly improve forecasting accuracy in the stock market.

Yang et al. [50] presented an integrated approach for stock price prediction that combines LSTM with ensemble EMD, indicating its superior effectiveness and accuracy compared to other techniques. Their comprehensive study improves the predictive perfor-

mance of this hybrid model and emphasizes its robustness in handling complex financial data. By utilizing the strengths of both LSTM and EMD, this joint methodology offers significant advantages in precisely predicting stock prices, making it a valuable tool for financial forecasting. Similarly, Ali et al. [51] developed an advanced hybrid model using a novel EMD model and LSTM networks, incorporating Akima spline interpolation for the improved treatment of non-stationary and non-linear financial time series. The decomposed signals are filtered and used as inputs to an RNN, enhancing the modeling of long-term dependencies and improving predictions. Their model, tested on the Karachi Stock Exchange (KSE)-100 index of the Pakistan Stock Exchange (PSX), outperformed pure LSTM and alternative ensemble methods, emphasizing the potential of elaborate data decomposition approaches in deep learning to reinforce stock market predictions.

Xuan et al. [52] represented a novel method for short-term stock price prediction by integrating EMD, LSTM neural networks, and cubic spline Interpolation (CSI). The model aims to enhance both the efficiency and accuracy of predicting short-term trends. It decomposes stock price data into intrinsic mode functions (IMF) and a residual component, classified based on gradient magnitude. High-gradient components use an LSTM model, while the rest employ a CSI model, combining their forecasts for the final prediction. Their approach outperformed conventional models, including the standalone LSTM, EMD-LSTM, and SVM models. Jin et al. [53] examined anticipating stock closing rates using sentiment analysis and LSTM, highlighting the crucial role of investor sentiment in enhancing model predictability. They developed a deep learning model that incorporated sentiment analysis, voltage series analysis, and an LSTM neural network with an improved attention mechanism. This methodology, particularly using EMD for decomposing complex sequences, effectively addresses stock market volatility and noise. Their study underscores the productivity of combining sentiment analysis with advanced machine learning approaches in financial prediction.

Jiang et al. [54] introduced a distinctive two-phase ensemble method for forecasting stock prices, combining extreme learning machine (ELM), empirical or variational mode decomposition (VMD), and the improved harmony search (IHS) technique. In the first stage, stock data are segmented into different frequency elements using EMD or VMD. After that, ELM is applied to each component for future price prediction, with IHS optimizing the ELM parameters to enhance accuracy. The performance of the EMD-ELM-IHS and VMD-ELM-IHS models was compared with the autoregressive integrated moving average (ARIMA), multilayer perceptron (MLP), support vector regression (SVR), ELM, and LSTM models, showcasing superior accuracy and stability. Shu and Gao [55] developed a hybrid model integrating CNN, EMD, and LSTM for stock price forecasting. Their approach involves decomposing stock prices into IMFs using EMD, filtering each IMF with a CNN for feature extraction, and analyzing these features with an LSTM network to model temporal dependencies. The model, tested on the Shanghai Stock Exchange (SSE) composite index for one- and seven-day forecasts, demonstrated improved performance in capturing multifrequency trading patterns compared to counterpart models.

Cao et al. [56] reported an advanced hybrid forecasting model that combines the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), and LSTM approaches to boost the accuracy of stock market price anticipations. The CEEMDAN technique decomposes financial time series data into multiple IMFs and a residual component, capturing various scale-time features of the market. Each IMF, along with the residual, serves as an input to individual LSTM models, which then forecast future stock prices. This hybrid approach leverages CEEMDAN's ability to handle non-linear and non-stationary characteristics of financial data and LSTM's strength in modeling long-term dependencies. Validation using global stock market indices proved that this hybrid model outperforms traditional forecasting methods, including the single SVM, MLP, and LSTM algorithms, by delivering more accurate predictions. Another significant LSTM-based model is the one proposed by Hochreiter and Schmidhuber [57], which addresses the vanishing gradient problem in standard RNNs. This issue complicates RNNs' ability to

learn and retain information over long sequences. LSTMs solve this by incorporating memory cells, enabling the network to maintain information over extended periods.

Unlike previous studies, this paper proposes a new approach that consists of empirical mode decomposition, technical indicators, and long short-term memory to benefit from their capabilities, which have yielded superior results in financial prediction. We capitalized on the strengths of all three methodologies to achieve successful results.

3. Materials and Methods

This study introduces a novel hybrid approach, EMD-TI-LSTM, which integrates empirical mode decomposition (EMD), technical indicators (TI), and long short-term memory (LSTM) to boost prediction precision over traditional methods. EMD is utilized to decompose complex market signals into more interpretable components, which are then processed by LSTM to detect long-range dependencies and patterns. Additionally, the incorporation of technical indicators such as the relative strength index (RSI), exponential moving average (EMA), and Bollinger bands (BB) enriches the model's forecasting capabilities by integrating key market insights. This combination not only improves the model's accuracy but also provides a more nuanced analysis of market behavior, making it a robust tool for financial forecasting. In this section, we will detail the methodologies that form the basis of EMD-TI-LSTM, and explain their theoretical foundations and practical applications to achieve our research objectives.

3.1. Methodologies

3.1.1. Long Short-Term Memory (LSTM)

LSTM networks are specialized architectures within a deep learning framework, classified as a type of recurrent neural network (RNN). In contrast to conventional feedforward neural networks, LSTMs incorporate feedback connections, making them particularly adept at handling sequential data. This architecture excels in applications such as speech recognition, time series forecasting, and natural language processing, where capturing temporal dependencies and sequential patterns is essential. LSTMs address several limitations of standard RNNs, notably the vanishing gradient problem, which can impede the learning and retention of information over long sequences. To overcome this challenge, LSTMs use memory cells, which are specialized units that preserve information over extended periods.

The LSTM architecture features diverse mathematical key elements, comprising the cell state, input gate, output gate, and forget gate, as illustrated in Figure 1. The cell state serves as a long-term memory unit that retains information over time. The input gate controls the addition of new information to the cell state, while the output gate regulates the transfer of information from the cell state to the network's output. The forget gate determines the information that should be removed from the cell state. These components work in concert to dynamically manage information, allowing LSTMs to retain relevant data while discarding irrelevant information, thus enhancing their ability to effectively process and predict sequential data.

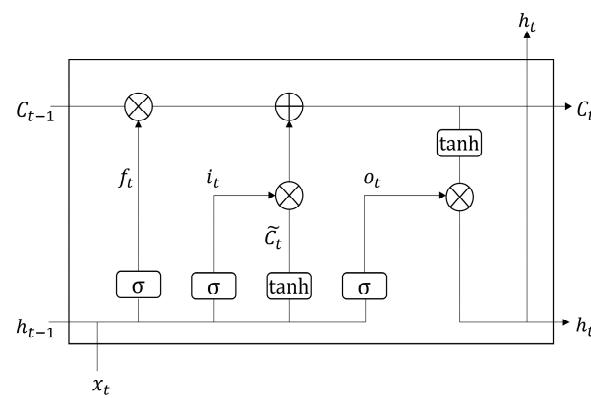


Figure 1. The architecture of LSTM.

The operation of an LSTM cell can be understood through the interaction of its components, as defined by the following Equations (1)–(6). These mathematical equations describe how information is retained, discarded, and output at each time step:

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

where

- f_t : Forget gate;
- W_f : Weight matrix for the forget gate;
- h_{t-1} : Hidden state from the previous time step;
- x_t : Input at the current time step;
- b_f : Bias term for the forget gate.

The forget gate f_t identifies which information from the cell state C_{t-1} should be discarded or retained for the current time step. It uses a sigmoid activation function σ that outputs values between 0 and 1. A value of 0 indicates that the related information in the cell state is entirely forgotten, while a value of 1 means it is completely retained. The output f_t is then used to scale the previous cell state C_{t-1} , regulating the extent to which the old cell state is carried over to the new cell state C_t .

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

where

- i_t : Input gate;
- W_i : Weight matrix for the input gate;
- h_{t-1} : Hidden state from the previous time step;
- x_t : Input at the current time step;
- b_i : Bias term for the input gate.

The input gate i_t regulates the quantity of new information to be incorporated into the cell state C_t . It uses a sigmoid activation function σ to output values between 0 and 1, where 0 indicates that no new information is added, and 1 indicates that the entire new input is stored in the cell state. This gating mechanism ensures that only the relevant information is updated in the cell state, boosting the network's capability to capture and utilize important features over time.

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

where

- \tilde{C}_t : Cell state candidate;
- W_C : Weight matrix for the cell state candidate;
- h_{t-1} : Hidden state from the previous timestep;
- x_t : Input at the current timestep;
- b_C : Bias term for the cell state candidate.

The cell state candidate \tilde{C}_t represents a potential update to the cell state C_t . It is computed by applying a linear transformation to the concatenation of the prior hidden state h_{t-1} and the current input x_t , followed by a bias term b_C . The result is passed through the hyperbolic tangent function \tanh , which squashes the produced values to fall within the range of -1 to 1 . The \tanh function allows the model to propose new information to be incorporated into the cell state, ensuring that this information is normalized. The range of -1 to 1 helps maintain the stability of the learning process and mitigate issues such as vanishing gradients. The proposed cell state candidate \tilde{C}_t is then used, together with the

input gate, to determine the final cell state modification, integrating new information with the existing cell state in a controlled manner.

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

where

- C_t : Updated cell state;
- f_t : Forget gate output;
- C_{t-1} : Previous cell state;
- i_t : Input gate output;
- \tilde{C}_t : Cell state candidate.

The cell state is updated by first multiplying the previous cell state C_{t-1} by the forget gate output f_t , which determines how much of the old information should be retained or discarded. This process ensures that irrelevant data are filtered out. Next, the model incorporates new, relevant information by adding the product of the input gate output i_t and the cell state candidate \tilde{C}_t . This addition updates the cell state with fresh information, which is regarded as crucial, allowing the LSTM to dynamically adjust its memory and improve its ability to process sequential data effectively.

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

where

- o_t : Output gate;
- W_o : Weight matrix for the output gate;
- h_{t-1} : Hidden state from the previous time step;
- x_t : Input at the current timestep;
- b_o : Bias term for the output gate.

The output gate o_t specifies the amount of cell state C_t should be exposed as the hidden state h_t for the current time step. This gate is computed by applying a linear transformation to the concatenated earlier hidden state h_{t-1} and the current input x_t , accompanied by a bias term b_o . The result of this linear combination is then passed through the sigmoid activation function σ , which outputs values between 0 and 1. The sigmoid function's output controls the extent to which the cell state C_t is considered for the hidden state. A value of 0 means that the cell state has minimal influence, while a value of 1 means it has full influence.

$$h_t = o_t \times \tanh(C_t) \quad (6)$$

where

- h_t : Hidden state at the current time step;
- o_t : Output gate;
- C_t : Cell state at the current time step.

The hidden state h_t is derived from multiplying the output gate o_t with the normalized cell state $\tanh(C_t)$. The cell state C_t is first normalized using the hyperbolic tangent function, which scales its values between -1 and 1 for stability. The output gate o_t then scales this normalized cell state to produce h_t , which contains the relevant information from the cell state and is used in the next timestep or as the network output. The result h_t represents the output hidden state at time t , which contains information based on the input sequence up to that point.

With the detailed architecture and equations of the LSTM network established, we can now explore how these components contribute to its performance and the practical considerations involved in configuring an LSTM model effectively. An LSTM cell effectively manages which information to pass, store, and forget, addressing challenges like the vanishing gradient problem commonly associated with traditional RNNs. This capability allows

LSTM networks to effectively handle long-term patterns within sequences, addressing challenges that typically affect standard RNNs, such as difficulties in retaining information over extended sequences.

The quality of the LSTM models is predominantly determined by the chosen hyperparameters. Hyperparameters are model settings established before training begins and remain constant throughout the learning process. The key hyperparameters of LSTM include the following:

- Number of units in an LSTM layer: This defines the dimension of the memory cell and influences the network's ability to recognize complex patterns. More units can enable the network to obtain more information about intricate features, but also increase computational complexity and the potential of overfitting. The ideal number of units depends on the task's complexity and the available training data.
- Learning rate: This controls the magnitude of the updates made by the optimizer during the gradient descent. An elevated learning rate might lead the model toward converging to a suboptimal solution or even diverging, while a reduced learning rate can result in prolonged training times or getting stuck in local minima. Adaptive methods like ADAM, RMSprop, and ADAGRAD tune the learning rate depending on the update history of the weights, helping to mitigate these issues.
- Batch size: This involves the number of training examples utilized in each iteration. A smaller batch size results in more frequent updates per epoch and can lead to faster convergence, but may introduce more noise in gradient estimates. Conversely, a larger batch size ensures more steady gradient estimations and uses more memory, potentially slowing convergence. The determination of the batch size also alters how well the model generalizes from the training data.
- Depth of the network: This is determined by the number of LSTM layers and influences the network's capacity to recognize intricate patterns in the data. Deeper networks can model more intricate relationships but may encounter training difficulties due to issues like vanishing gradients. Techniques such as gradient clipping and the inherent gating mechanisms of LSTM help address these problems. Additionally, deeper networks require more computational resources and are more susceptible to overfitting, necessitating regularization methods like dropout.
- Dropout rate: Dropout is a regularization method employed to avoid overfitting by randomly eliminating units and their connections during training. In LSTMs, dropout is typically applied between layers rather than within recurrent connections to avoid disrupting the flow of memory information. The dropout rate specifies the proportion of units to be removed, balancing the need for regularization with the risk of underfitting.
- Length of input sequences: This hyperparameter affects the LSTM's ability to learn long-term dependencies. Longer sequences allow the model to capture extended input patterns but increase computational complexity and memory demands. Techniques such as sequence shortening or attention mechanisms can help manage long sequences and enhance model performance.

In conclusion, selecting the appropriate hyperparameters for an LSTM model is crucial for optimizing its performance and training efficiency. The best hyperparameter settings are typically task specific and can be determined through a combination of expert knowledge, empirical testing, and automatic optimization methods, such as grid search or Bayesian optimization. Understanding the role and impact of each hyperparameter will assist practitioners in fine-tuning their LSTM models for diverse applications, ensuring optimal results.

3.1.2. Technical Indicators (TI)

Technical indicators are numerical computations that depend on the price, volume, and open interest of a security or contract and are widely used in technical analysis. These indicators help analyze past and present market behaviors to predict future price movements. Their primary applications include identifying trade opportunities by generating

buy and sell signals, determining market trends, and evaluating the strength or weakness of security. Technical indicators can be categorized into the following groups:

- Trend indicators: These indicators identify the direction and magnitude of a trend. Examples include the moving average (MA), exponential moving average (EMA), moving average convergence divergence (MACD), and directional movement index (DMI).
- Momentum indicators: Used to identify the velocity of price movements, these indicators help recognize overbought or oversold conditions. Examples include relative strength index (RSI), stochastics, and commodity channel index (CCI).
- Volatility indicators: These indicators measure the extent of price changes, regardless of direction, and are used for risk assessment and gauging market sentiment. Examples include the Bollinger bands (BB), average true range (ATR), and volatility index (VIX).
- Volume indicators: These indicators analyze trading volumes to verify trends or predict trend reversals. Examples include the volume oscillator, on-balance volume (OBV), and Chaikin money flow (CMF).

Investors and traders use technical indicators to recognize optimal entry and exit points, as these indicators can signal the best times to buy or sell security. They also confirm price movements by validating the strength and consistency of price trends using multiple indicators. Additionally, technical indicators are crucial for risk management, helping to place stop-loss orders or adjust portfolio risk exposure based on market conditions. Understanding the assumptions and mathematical calculations behind these indicators is essential for their effective use in both academic and practical applications. By employing varied signals and incorporating other types of analysis, traders can develop a more sophisticated approach to market analysis. To delve deeper into specific technical indicators, we will explore EMA, RSI, and BB, examining how each can be utilized to enhance market analysis and trading strategies.

Exponential Moving Average (EMA)

EMA is a financial market technical indicator that helps transform price data over a certain period, emphasizing the importance of recent price movements. This weighting method makes the EMA more sensitive to recent price changes relative to the simple moving average (SMA), which assigns equal weights to all values within the period. EMA is crucial for identifying trend directions and potential reversals and is an important component of other technical indicators. Mathematically, EMA can be expressed through Equations (7) and (8).

$$EMA_{today} = (Price_{today} \times K) + (EMA_{yesterday} \times (1 - K)) \quad (7)$$

$$K = \frac{2}{Days + 1} \quad (8)$$

where

- EMA_{today} : EMA value for the current period;
- $Price_{today}$: Closing price for the current period;
- $Days$: Number of periods in EMA;
- K : Smoothing factor in EMA;
- $EMA_{yesterday}$: EMA value from the previous period.

EMA utilizes a smoothing factor K to adjust the weight given to recent prices, with higher values of K increasing sensitivity to recent price changes. This dynamic weighting allows the EMA to reflect current market trends more accurately by influencing recent data while still considering historical prices. Updated daily using the closing price and the previous EMA value, the EMA adapts quickly to new information. Its ability to identify trends and potential reversals makes it a valuable tool when used along with other technical indicators to enhance financial strategies and gain deeper insights into market conditions.

Relative Strength Index (RSI)

RSI is a highly regarded momentum oscillator that is commonly exploited in technical analysis to evaluate the velocity and magnitude of recent price movements. By comparing the magnitude of recent gains to recent losses, the RSI serves as a crucial tool for identifying overbought or oversold conditions in a given asset. This comparative analysis allows market participants to gauge the underlying momentum-driving price fluctuations, thereby providing a more subtle comprehension of an asset's price performance over a specified period. The RSI is mathematically derived using Equations (9)–(12), and is designed to quantify momentum in a way that reduces the impact of transient price anomalies. This smoothing effect enhances the reliability of the RSI, making it a more robust indicator for making informed trading and investment decisions.

$$\text{RSI} = 100 - \frac{100}{1 + \text{RS}} \quad (9)$$

$$\text{RS} = \frac{\text{Average Gain over } N \text{ periods}}{\text{Average Loss over } N \text{ periods}} \quad (10)$$

$$\text{Average Gain} = \frac{\text{Previous Average Gain} \times (N - 1) + \text{Current Gain}}{N} \quad (11)$$

$$\text{Average Loss} = \frac{\text{Previous Average Loss} \times (N - 1) + \text{Current Loss}}{N} \quad (12)$$

where

- RSI: Relative strength index as a momentum indicator with a value ranging between 0 and 100
- RS: Relative strength representing the ratio of average gains to average losses over the specific period N
- Average gain: Mean of all positive price changes over the period N
- Previous average gain: Average gain calculated during the previous period
- Current gain: Gain in the current period representing a positive price change, or zero if the price did not increase
- Average loss: Mean of all negative price changes over the period N
- Previous average loss: Average loss calculated during the previous period
- Current loss: Loss in the current period representing a negative price change, or zero if the price did not decrease
- N : Number of periods, commonly set to 14 days

It is worth noting that an RSI value of 70 or above typically suggests that security may be overbought, which could lead to trend reversal or price pullback. Conversely, an RSI value of 30 or less indicates that the security might be oversold or undervalued, potentially setting the stage for a price rebound. A critical signal provided by the RSI is the divergence between the RSI and price action, which often precedes significant trend changes. A bullish divergence occurs when the price hits a new low, but the RSI forms a higher low, suggesting that despite declining prices, the underlying momentum is strengthening. This divergence signals a potential upward reversal. On the other hand, a bearish divergence occurs in scenarios where the price reaches a new high, while the RSI shows a lower high, indicating weakening momentum and a possible downward shift. The RSI is an indispensable tool for traders and analysts, as it helps not only identify overbought and oversold conditions, but also detect these crucial divergences. To maximize its effectiveness, RSI should be integrated with other technical analysis tools to enhance the robustness of trading and investment decisions.

Bollinger Bands (BB)

BB is a technical analysis tool comprising three lines plotted around a security price to assess volatility. The middle band is calculated as the simple moving average (SMA)

of the closing prices over a specific period N . The upper band is determined by adding a multiple K of the standard deviation of the closing prices over the same period to the middle band, while the lower band is determined by subtracting this value from the middle band. Bollinger bands adapt to market conditions, widening during times of high volatility and narrowing during times of low volatility. The mathematical formulas for these bands are presented in Equations (13)–(15).

$$\text{UpperBand} = \text{MiddleBand} + K \times \sigma(N) \quad (13)$$

$$\text{LowerBand} = \text{MiddleBand} - K \times \sigma(N) \quad (14)$$

$$\text{MiddleBand} = \text{SMA}(N) \quad (15)$$

where

- K : Number of standard deviations, typically set to 2;
- $\sigma(N)$: Standard deviation of the closing prices over the specific period N .

Wider Bollinger bands indicate higher market volatility, while narrower bands suggest lower volatility. The middle band's direction shows a trend—an upward tilt signals an uptrend and a downward tilt indicates a downtrend. The upper and lower bands function as adjustable support and resistance levels, with price movements often touching or breaching these bands to signal overbought or oversold conditions. Combining BB with other indicators like EMA and RSI can improve financial analysis, although it may increase the complexity and risk of overfitting. Effective feature selection and regularization are crucial for mitigating this risk. The indicators' responsiveness varies with market conditions, so their use should be guided by empirical evidence and expertise. Ultimately, integrating these indicators with advanced models like LSTM can enhance predictive accuracy by leveraging detailed market data.

3.1.3. Empirical Mode Decomposition (EMD)

EMD is a method intended to deconstruct a signal into intrinsic mode functions (IMF), which help reveal underlying trends and cycles. Unlike traditional methods that assume linearity and stationarity, EMD is particularly effective for analyzing non-linear and fluctuating data, making it valuable in fields such as finance, geophysics, mechanics, and biomedical engineering. This method relies on recognizing inherent oscillatory modes within a complex dataset and separating them according to the local attributes of the data. The decomposition process is entirely empirical, with IMFs derived directly from the data without any predetermined basis. The EMD process iteratively removes IMFs from the signal, where each IMF must satisfy two key conditions: the number of local maxima and zero crossings should be equal or vary by at most one, and the mean of the envelope between the maxima and minima must be zero at every location. The decomposition process involves the following steps through Equations (16)–(23):

1. Initialize the residue from the original signal $x(t)$:

$$r_0(t) = x(t) \quad (16)$$

where

- $x(t)$: Original signal as a function of time;
- $r_0(t)$: Initial residue at the start of the decomposition process.

2. Iteratively extract each IMF $c_i(t)$:

- Identify and interpolate the local extrema to form the lower and upper envelopes by incorporating local minima and maxima:

$$e_{lower,i}(t) = \text{interp}\{(t_i, r(t_i)), | local\ minima\} \quad (17)$$

$$e_{upper,i}(t) = \text{interp}\{(t_i, r(t_i)), | local\ maxima\} \quad (18)$$

where

- $e_{lower,i}(t)$: Lower envelope at iteration i , formed by interpolating through the local minima of the signal;
- $e_{upper,i}(t)$: Upper envelope at iteration i , formed by interpolating through the local maxima of the signal;
- t : Time point to evaluate the variables;
- $\text{interp}\{(t_i, r(t_i))\}$: Interpolation function to create a smooth curve connecting the given points;
- t_i : Time instances corresponding to the local extrema of maxima or minima;
- $r(t_i)$: Residue value at time t_i , where the local extrema occur.
- Compute the mean envelope as follows:

$$m_i(t) = \frac{e_{upper,i}(t) + e_{lower,i}(t)}{2} \quad (19)$$

where

- $m_i(t)$: Mean envelope at iteration i representing the average of the upper and lower envelopes at time point t .
- Detrend the residue by subtracting the mean envelope to find a candidate IMF:

$$d_i(t) = r_{i-1}(t) - m_i(t) \quad (20)$$

where

- $d_i(t)$: IMF at iteration i to represent the oscillatory component extracted from the residue after removing the trend;
- $r_{i-1}(t)$: Residue from the previous iteration including both the trend and the oscillatory components.
- Define the extracted IMF if it meets the following criteria:

$$c_i(t) = d_i(t) \quad (21)$$

where

- $c_i(t)$: Extracted IMF at iteration i meeting the criteria for being an IMF after the detrending step.
- Update the residue for the next iteration:

$$r_i(t) = r_{i-1}(t) - c_i(t) \quad (22)$$

where

- $r_i(t)$: Residue at iteration i , the remaining signal after removing the IMF $c_i(t)$ at the current iteration;
- $r_{i-1}(t)$: Residue from the previous iteration $i - 1$, the signal before removing the current IMF $c_i(t)$.
- 3. The process concludes when the residue $r_n(t)$ evolves into a monotonic function or cannot be further decomposed. The original signal is represented as the sum of all extracted IMFs and the final residue:

$$x(t) = \sum_{i=1}^n c_i(t) + r_n(t) \quad (23)$$

where

- $x(t)$: Final residue signal after completing the EMD process;
- n : Total number of extracted IMFs;
- $\sum_{i=1}^n c_i(t)$: Sum of extracted IMFs during the EMD process to capture oscillatory components of the signal;
- $r_n(t)$: Residue signal after completing the EMD process, which is either a monotonic function or a fixed constant, representing the portion of the original signal that cannot be further decomposed.

3.2. Proposed Model

In the current study, we developed and evaluated two distinct models to compare their performance using identical hyperparameters and datasets. The first model employs a standard long short-term memory (LSTM) network integrated with technical indicators (TI). The second model, introduced in this study as EMD-TI-LSTM, incorporates an innovative approach by integrating empirical mode decomposition (EMD) with technical indicators, which are then processed by an LSTM network. The workflows of both models are depicted in Figure 2, which illustrates the differences in their design and implementation.

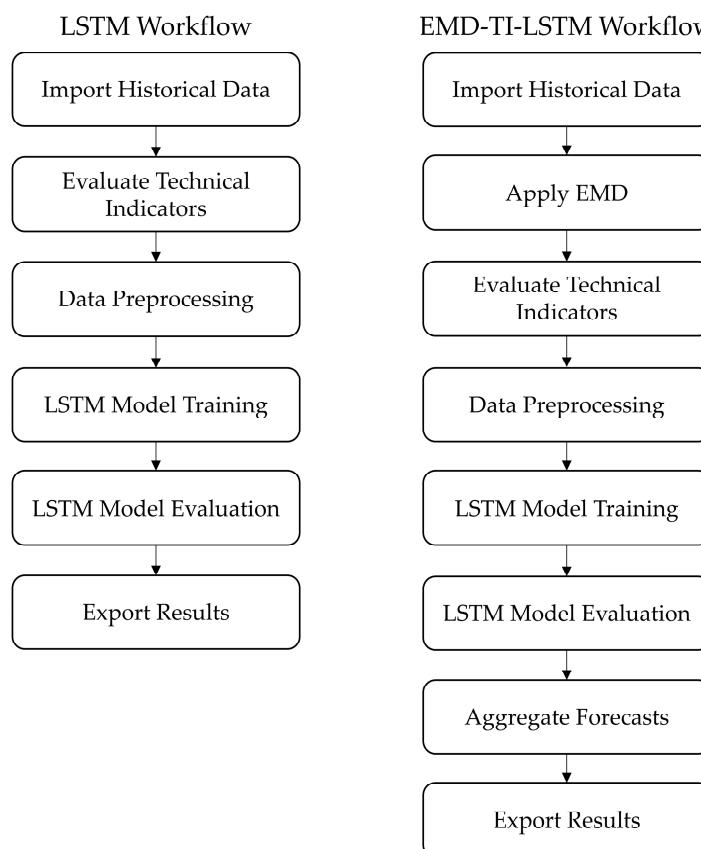


Figure 2. Workflows of LSTM and EMD-TI-LSTM models.

The first model follows a conventional approach by utilizing the capabilities of the LSTM network to detect enduring patterns in time series data. Historical data, specifically past asset prices, are first imported. Technical indicators such as the exponential moving average (EMA), relative strength index (RSI), and Bollinger bands (BB) are then computed and used as additional inputs to strengthen the model's prediction accuracy. After calculating the technical indicators, the data are normalized, and sequences are prepared for the LSTM model. The network of LSTM, implemented using TensorFlow 2.2 framework (Google Inc., Mountain View, CA, USA), Keras 3.4 open-source library (Google Inc., Mountain View, CA, USA), consists of two layers: a dropout layer to mitigate overfitting and a fully connected output layer. The first model is configured with an appropriate loss function and optimizer.

It is trained on the training set using early stopping to avoid overfitting. Following training, the model is applied to predict the test dataset, and these predictions are rescaled to the original one. The effectiveness of the model is measured based on metrics such as the mean absolute percentage error (MAPE), root mean square error (RMSE), and mean absolute error (MAE) to determine the accuracy and reliability of the forecasts.

The EMD-TI-LSTM model introduces EMD into the prediction process. After importing historical data, EMD breaks down the initial signal into intrinsic mode functions (IMF), capturing various frequency components of the signal. These IMFs provide a richer representation of the data by breaking them down into multiple scales. The decomposed signals are then combined with technical indicators and processed through the LSTM network, following a training procedure similar to the first model. However, EMD-TI-LSTM includes an additional step of aggregating the forecasts derived from multiple IMFs before exporting the final results. In summary, while the first model processes technical indicators directly through the LSTM network, EMD-TI-LSTM first applies EMD to the signal, creating a more detailed feature set by combining IMFs with technical indicators before feeding them into the LSTM. A comparison between these models seeks to evaluate the effect of incorporating EMD on the predictive accuracy of the LSTM network to determine whether the enhanced feature representation provided by EMD offers a significant advantage over the standard LSTM approach.

As detailed in Figure 3, the EMD-TI-LSTM model employs an advanced multi-step methodology for time series forecasting, specifically designed to enhance predictive accuracy by integrating EMD with technical indicators and an LSTM network. The process unfolds through the following steps:

- Import historical data: The model begins by importing historical asset data, which serve as the foundational input for the forecasting process. These data typically consist of time series information, such as past asset prices.
- Apply EMD: The historical data undergo the EMD process, which breaks down the original signal into multiple IMFs. EMD is a powerful method that splits a signal into its underlying components, each of which represents various frequency bands. This decomposition allows the model to capture various oscillatory behaviors within the data.
- Calculate TI for each IMF: For each IMF extracted through the EMD process, the model calculates a set of technical indicators, including EMA, BB, and RSI. These indicators are crucial for capturing different market dynamics and trends. The IMFs, enriched with their respective technical indicators, are then used as input features for the subsequent LSTM model. This combination of IMFs and technical indicators provides a more elaborate and complete presentation of the underlying data.
- Apply LSTM: The model then constructs a sequential LSTM model for each IMF-TI combination, with carefully defined hyperparameters and dropout layers for regularization to prevent overfitting. LSTM networks are designed to capture long-term dependencies in the time series by learning from the enriched feature sets provided by each IMF. Each LSTM model is trained independently on its respective IMF and associated technical indicators.
- Obtaining forecasts: Once the training is complete, each LSTM network generates forecasts for its corresponding IMF. These forecasts reflect the model predictions for each decomposed component of the original signal.
- Calculate the forecast: The final step involves aggregating the individual forecasts produced by each LSTM network. This aggregation synthesizes the predictions from all IMFs into a single unified forecast of the asset's future price. By combining the insights derived from each IMF, the EMD-TI-LSTM model delivers a more reliable and accurate prediction than a single LSTM model.
- Model performance evaluation: The performance of the EMD-TI-LSTM model is evaluated using metrics such as MAPE, RMSE, and MAE. These metrics are calculated

by comparing the aggregated forecasts against the actual asset prices, providing a clear measure of the model's accuracy and reliability.

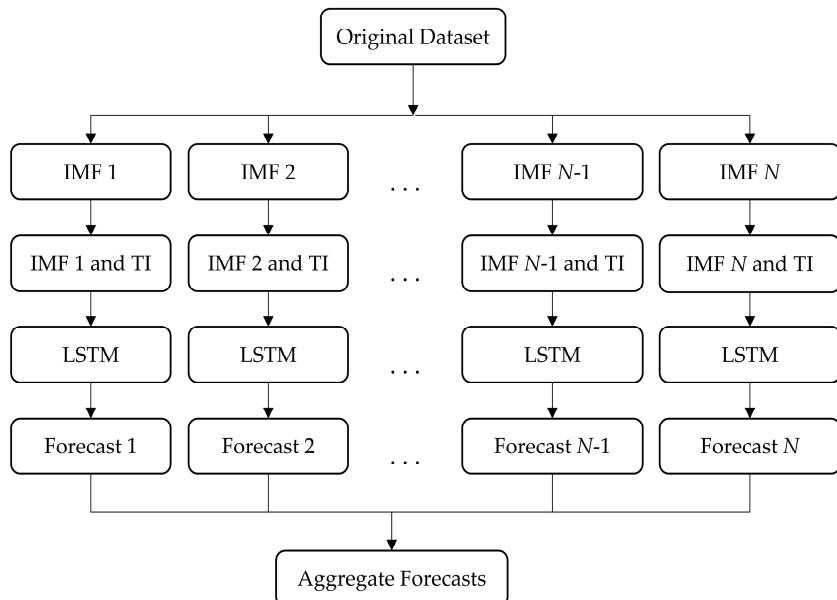


Figure 3. Detailed workflow of the EMD-TI-LSTM model.

The hyperparameters for the EMD-TI-LSTM model are summarized in Table 1. The model configuration includes a window size of seven for technical indicators, including EMA, RSI, and BB, meaning each indicator is calculated using the most recent seven time steps of time series data, which helps capture short-term trends. Mathematically, this approach integrates recent historical data into forecasts. The model employs two LSTM layers, each with 512 units, leveraging a deep learning framework to effectively identify and remember sophisticated patterns and temporal relationships in the time series data. The training process is set for 500 epochs, where each epoch represents a full pass through the entire training dataset and a learning rate of 0.0001, ensuring a stable learning curve. The sequence length and batch size are configured at 60 and 32, respectively. The sequence length defines the number of time steps used in each input segment, enabling the model to be trained on a predetermined window of historical data. The batch size specifies the number of samples processed before updating the model hyperparameters, thus influencing the data processing efficiency and memory usage. A dropout rate of 0.1 is applied to reduce overfitting, and a train/test ratio of 0.95 is chosen to allocate the majority of the dataset for training, consequently refining the model's forecasting ability in time series analysis.

Table 1. Hyperparameters of the EMD-TI-LSTM model.

Name	Value
Number of assets	4
Dataset time interval	daily
Number of LSTM layer	2
Number of neurons or units in each layer	512
Activation	tanh
Epoch	500
Learning rate	0.0001
Dropout rate for each layer	0.1
Sequence length	60
Train/test ratio	0.95
Batch size	32
Loss function	mean squared error
Size of technical indicators window	7

Table 2 presents the comprehensive algorithm for the EMD-TI-LSTM model, detailing each step from the initial setup to model evaluation. The process begins by recording the start time and ensuring that all necessary packages are installed and libraries are imported. Google Drive is mounted to facilitate file access. Configurations are defined, including assets, time intervals, and hyperparameter ranges. The algorithm then iterates over each asset and hyperparameter combination. For each configuration, it prints the current setup, sets the Adam optimizer with the specified learning rate, and loads the data from an Excel file. The data are preprocessed by converting dates, sorting, and extracting close values. The EMD technique is applied to these values, followed by the definition of functions for data preparation and LSTM model creation. Lists are initialized to collect predictions and actual values. Each IMF is processed sequentially, where indicators like EMA, BB, and RSI are calculated and combined into a single data frame. Data are then prepared for LSTM training, split into training and testing sets, and used to create and train the LSTM model. Predictions are made, inverse transformed, and stored. Forecasts are aggregated by summing them and aligning them with test close prices and dates, and then performance metrics, such as MAPE, RMSE, and MAE, are calculated. Results are printed, saved in an Excel file, and plotted. The process is concluded by printing a completion message and recording the end time to calculate the total execution duration.

Table 2. Algorithm of the EMD-TI-LSTM model.

Step	Action	Explanation
1	Start timer	Record the start time for execution
2	Install packages	Ensure necessary packages are installed
3	Import libraries	Import required libraries
4	Mount Google Drive	Mount Google Drive to access files
5	Define configurations	Define assets, time intervals, hyperparameter ranges, and configurations
6	For each asset	Loop through each asset in the defined list
6.1	For each hyperparameter	Loop through each combination of hyperparameters
6.1.1	Print configuration	Print the current training configuration
6.1.2	Set optimizer	Set the Adam optimizer with the specified learning rate
6.1.3	Load data	Load data from the Excel file for the current asset and time interval
6.1.4	Preprocess data	Convert date to date-time type, sort data, and extract close values
6.1.5	Apply EMD	Apply EMD to the close values
6.1.6	Define functions	Define functions for data preparation and LSTM model creation
6.1.7	Initialize lists	Initialize lists to collect predictions and actual values
6.1.8	For each IMF	Loop through each IMF
6.1.8.1	Print IMF	Print the current IMF being processed
6.1.8.2	Calculate indicators	Calculate EMA, BB, and RSI for the IMF
6.1.8.3	Combine data	Combine IMF and indicators into a single data frame
6.1.8.4	Prepare data	Prepare data for LSTM model training
6.1.8.5	Split data	Split data into training and testing sets
6.1.8.6	Create model	Create the LSTM model with the given input shape
6.1.8.7	Train model	Train the LSTM model with training data
6.1.8.8	Make predictions	Make predictions with the trained model
6.1.8.9	Inverse transform	Inverse transform the predictions
6.1.8.10	Store forecasts	Store the forecasts in the list
6.1.9	Aggregate forecasts	Aggregate forecasts by summing them
6.1.10	Align data	Align test close prices and dates with aggregated forecasts
6.1.11	Calculate metrics	Calculate performance metrics, including MAPE, RMSE, and MAE
6.1.12	Print results	Print the performance results
6.1.13	Save results	Save results to an Excel file and copy them to Google Drive
6.1.14	Plot results	Plot the actual and forecasted values
7	Print completion	Print a completion message and execution duration
8	End timer	Record the end time for execution and calculate the duration

The proposed model (EMD-TI-LSTM) has a number of advantages that can be summarized as follows. First, it tends to enhance financial forecasting since it benefits from

the strengths of three useful concepts (EMD, TI, and LSTM). The unique combination of these methodologies can contribute to both the theoretical and practical aspects of financial forecasting and can have broader implications for machine learning applications in the field. The relative advantage of this model is its ability to use EMD to break down complex market signals into more interpretable components, which can then be processed using LSTM. Furthermore, the incorporation of technical indicators (EMA, RSI, etc.) enhances the model's forecasting capabilities by incorporating meaningful market information into the analysis. An important advantage of the EMD-TI-LSTM approach is that it can be applied to any historical asset data without any prior information about the given dataset. It is entirely unaware of the asset type; in fact, it simply learns from the time series data samples. One of the main advantages of EMD-TI-LSTM is that it can be easily implemented using the existing pyEMD 1.0.0 library and LSTM networks by lightly setting understandable parameters. Another advantage is that it can be easily facilitated for further research and can be adapted for advanced forecasting problems in various financial domains within the academic community and industry. Thus, the EMD-TI-LSTM method not only pushes the boundaries of conventional forecasting approaches but also fosters innovation in AI-driven financial analysis. The subsequent sections will substantiate these advantages by applying the EMD-TI-LSTM model to well-known real-world datasets, presenting experimental results, and conducting comprehensive comparisons to validate the model's practical efficacy.

3.3. Dataset Description

In this study, four distinct financial assets were analyzed: the close price of BTC/USD, the close index of the BIST 100 Index, the close index of the NASDAQ-100 Index, and the close price of GOLD/USD. To clarify, the datasets will be referred to using the tickers BTC, BIST, NASDAQ, and GOLD, respectively. All datasets were sourced from the TradingView website [58] and are briefly presented in Table 3. Additionally, each dataset is described in detail in this table to provide an inclusive view of its characteristics and significance.

Table 3. A brief overview of utilized datasets.

Ticker Name	Pair/Index	Link (accessed on 16 November 2023)
BTC	BTC/USD	https://www.tradingview.com/symbols/BTCUSD/?exchange=INDEX
BIST	BIST 100 Index	https://www.tradingview.com/symbols/BIST-XU100/
NASDAQ	NASDAQ 100 Index	https://www.tradingview.com/symbols/NASDAQ-NDX/
GOLD	GOLD/USD	https://www.tradingview.com/symbols/XAUUSD/?exchange=OANDA

3.3.1. BTC/USD

BTC/USD represents the exchange rate between Bitcoin (BTC), the world's most traded and widely adopted cryptocurrency, and the United States Dollar (USD). It indicates how much one Bitcoin is worth in U.S. dollars. Bitcoin, as the first digital currency, pioneered the cryptocurrency market and remains a dominant asset within this emerging class. The BTC/USD trading pair not only reflects Bitcoin's value in terms of U.S. dollars but also captures the high volatility and dynamic nature of cryptocurrency markets, making it a critical indicator for analyzing market trends and investor behavior.

3.3.2. BIST 100 Index

The BIST 100 Index is the primary stock market index of Turkey, representing the outcomes of the leading and highly liquid enterprises listed on the Borsa Istanbul (BIST) stock exchange. This capitalization-weighted index comprises 100 companies with the highest trading volumes and market values, excluding investment trusts. The selection of constituents for the BIST 100 is based on predetermined criteria designed to ensure the inclusion of the most significant and stable companies in the Turkish market. As a key indicator of Turkey's economic health, the BIST 100 Index provides meaningful perspectives on overall performance and trends within the Turkish equity market.

3.3.3. NASDAQ-100 Index

The NASDAQ-100 Index, representing the national association of securities dealers' automated quotations, encompasses 100 of the top companies listed on the NASDAQ stock market, the world's second-largest by market capitalization. This modified capitalization-weighted index includes major technology and high-growth companies across various industries, such as technology, telecommunications, biotechnology, media, and services, while excluding financial services companies. As a key indicator, NASDAQ-100 offers a broad view of the U.S. technology sector and serves as a valuable benchmark for investors assessing the performance of the stock market, particularly in sectors characterized by innovation and rapid growth.

3.3.4. GOLD/USD

GOLD/USD refers to the trading pair that represents the exchange rate between gold, measured in troy ounces, and the United States Dollar (USD). This trading pair indicates the value of one troy ounce of gold in terms of US dollars. Gold has been a fundamental asset in financial markets and was the basis of economic capitalism until the repeal of the Gold Standard, which led to the adoption of a fiat currency system. Gold continues to be a widely followed and critical asset in global finance. The GOLD/USD exchange rate serves as a key metric for evaluating gold's value and trends within a broader economic context.

Daily closing values for these assets were obtained from the TradingView website (TradingView Inc., New York, NY, USA), covering a ten-year period from 15 November 2013 to 15 November 2023. The statistical information of these datasets, including the count, minimum, mean, maximum, and standard deviation (SD), is presented in Table 4. This table reveals distinct volatility patterns among the four financial assets: BTC, BIST, NASDAQ, and GOLD. BTC displays the highest volatility, reflecting the unpredictable nature and significant price fluctuations typical of cryptocurrency markets. Conversely, GOLD exhibits the lowest volatility, underscoring its role as a stable asset. The BIST and NASDAQ indices demonstrate moderate volatility, highlighting variations in risk and stability across different market sectors. This comparison underscores the diverse risk profiles of these assets, with BTC being the most volatile and GOLD being the most stable.

Table 4. Statistical information of datasets.

Ticker	Count	Min	Mean	Max	SD
BTC	3653	182.00	13,071.57	67,556.46	15,768.55
BIST	2510	611.89	1642.71	8513.54	1620.14
NASDAQ	2517	3367.17	8332.16	16,573.34	3966.32
GOLD	2585	1051.71	1481.54	2063.31	289.33

Additionally, the daily close values of BTC, BIST, NASDAQ, and GOLD over time from 1 January 2014 to 1 January 2024 are illustrated through Figure 4. The figure provides a clear visualization of the trends and fluctuations in the closing prices of these assets over the ten years. BTC shows a pattern of extreme volatility with sharp peaks and troughs, reflecting the high-risk nature of the cryptocurrency market. BIST exhibits a steady upward trend, particularly in later years, indicating growth in the stock market. In addition, NASDAQ shows a strong upward trajectory, consistent with the expansion of the sector. GOLD demonstrates relatively lower volatility compared to BTC, with gradual increases and a few significant dips, underscoring its stability.



Figure 4. Close values of BTC, BIST, NASDAQ, and GOLD.

The actual close values and their corresponding IMFs for BTC, BIST, NASDAQ, and GOLD are shown in Figures 5–8, respectively. These figures highlight the underlying trends and cycles at various frequencies, which are crucial for forecasting and understanding market behavior. By examining these IMFs, we can identify the significant patterns that make them valuable features for more accurate market predictions. For example, Figure 5 illustrates the original close values of BTC along with their respective IMFs. The top subplot shows the BTC close value time series, which reflects the overall market trend. The subsequent plots showcase the decomposed IMFs, ranging from IMF 1 to IMF 9, each representing different frequency components of the original signal. The high-frequency intrinsic modes, including IMF 1 to IMF 3, capture rapid fluctuations and noise, providing visions for short-term market volatility. The mid-frequency IMFs, namely IMF 4 to IMF 6, reveal medium-term cyclical patterns that may correspond to periodic market behaviors, while the lower-frequency IMFs, involving IMF 7 to IMF 9, reveal long-term trends and slow-moving cycles, which are critical for understanding the broader market trajectory. This decomposition allows us to isolate and analyze the various components of the BTC time series, enabling a more thorough grasp of its fundamental dynamics and improving the precision of forecasting models by focusing on specific IMFs relevant to the forecasting horizon.

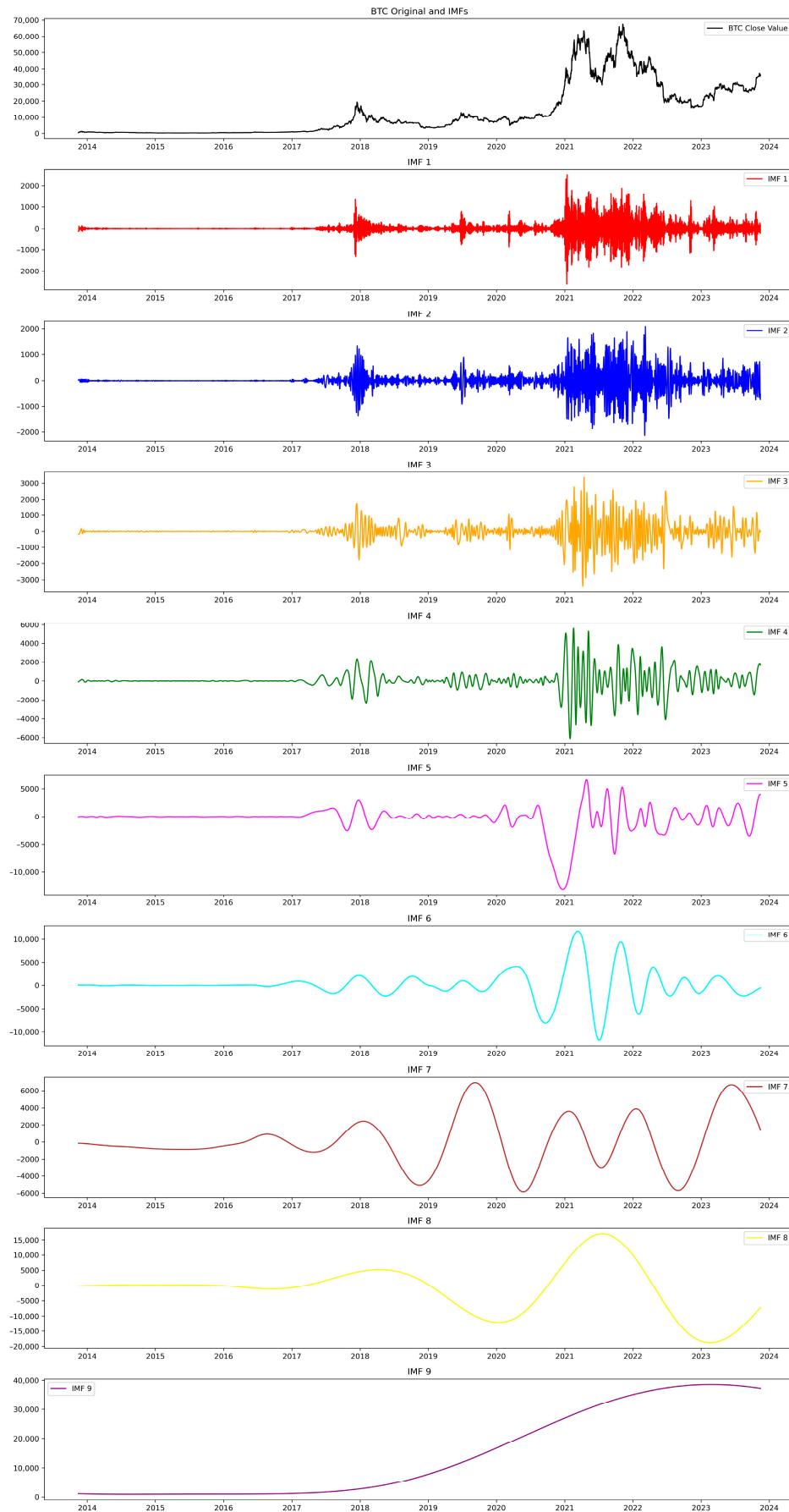


Figure 5. BTC close actual values and IMFs.

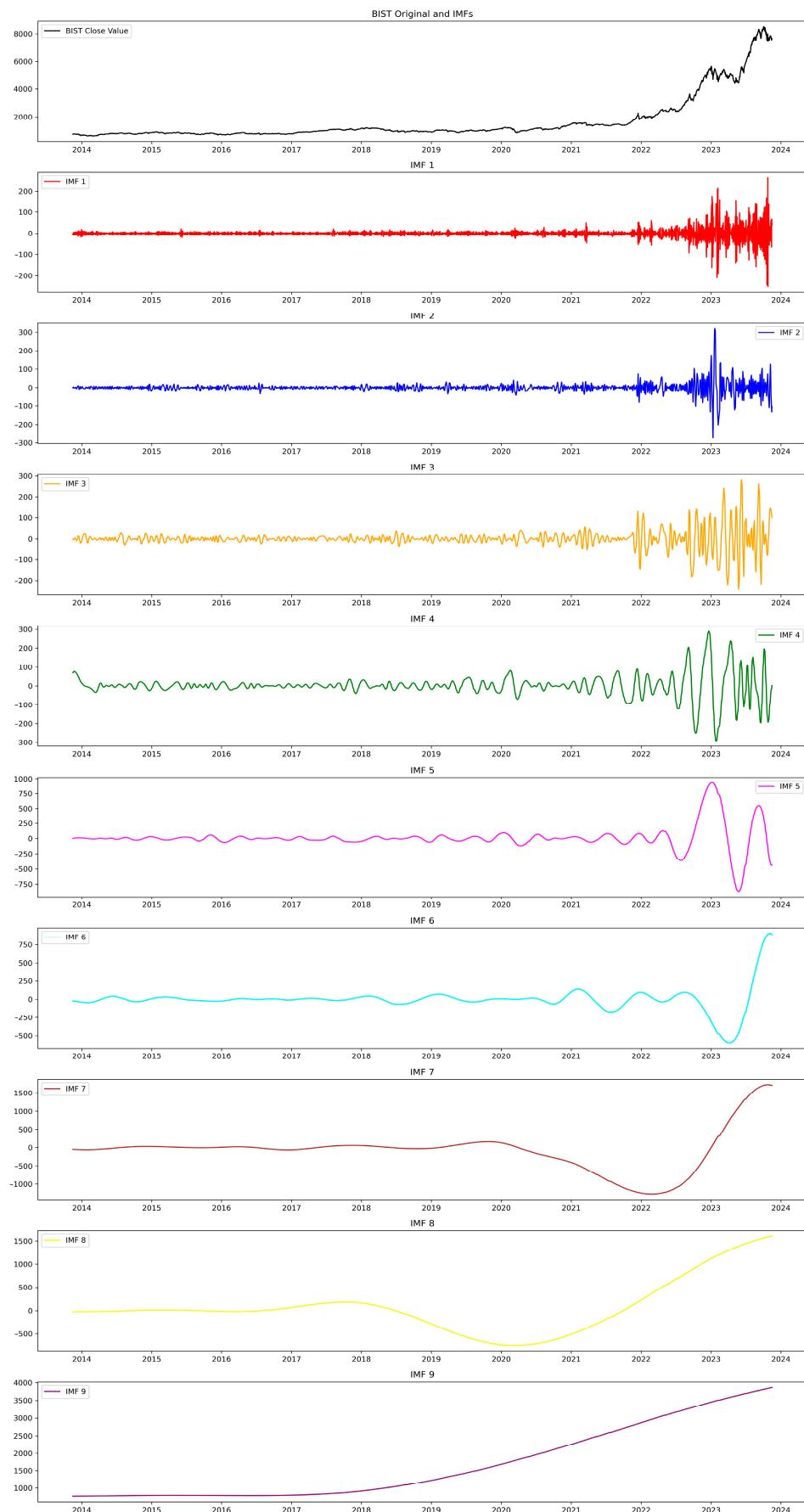


Figure 6. BIST close actual values and IMFs.

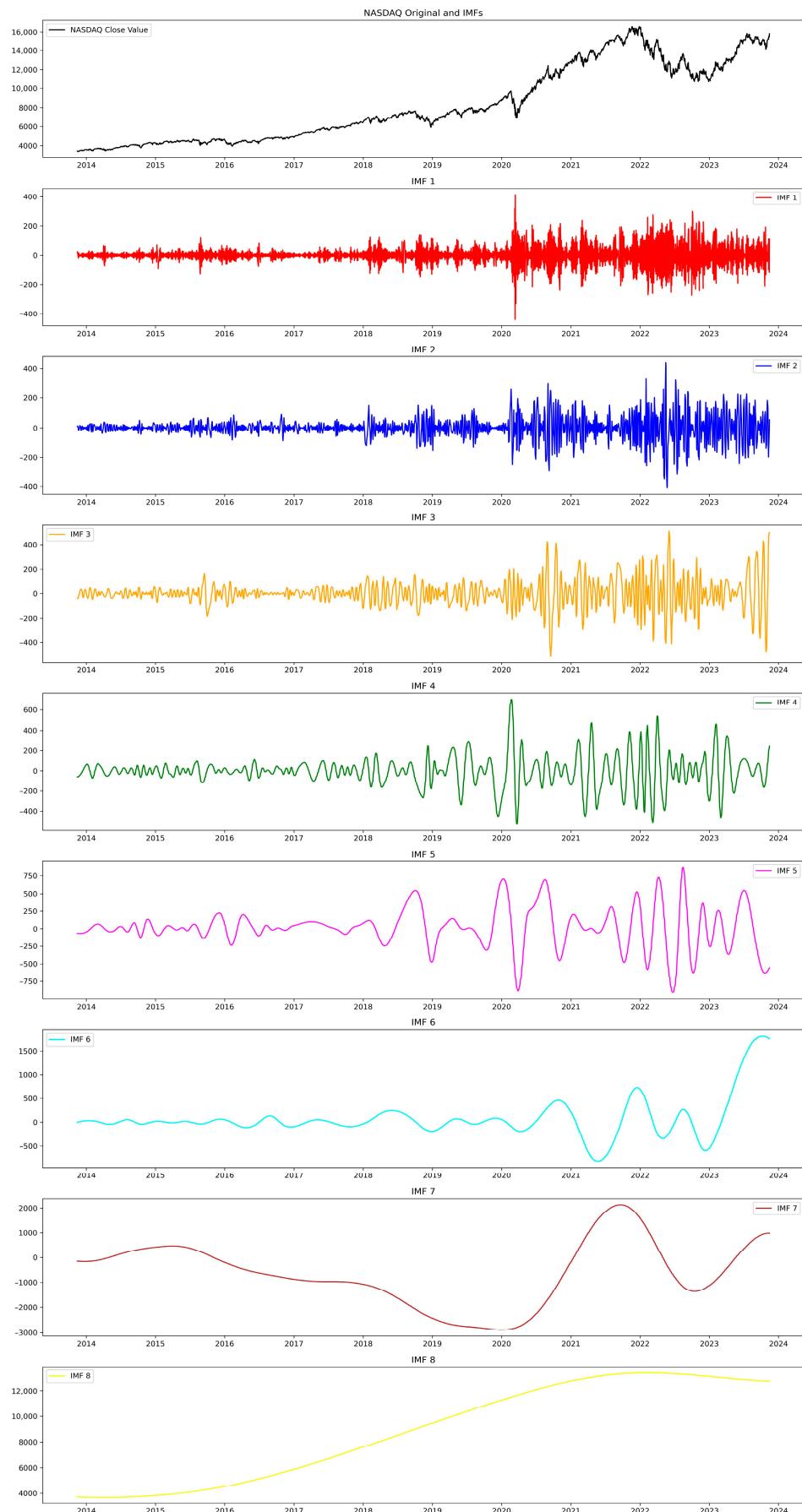


Figure 7. NASDAQ close actual values and IMFs.

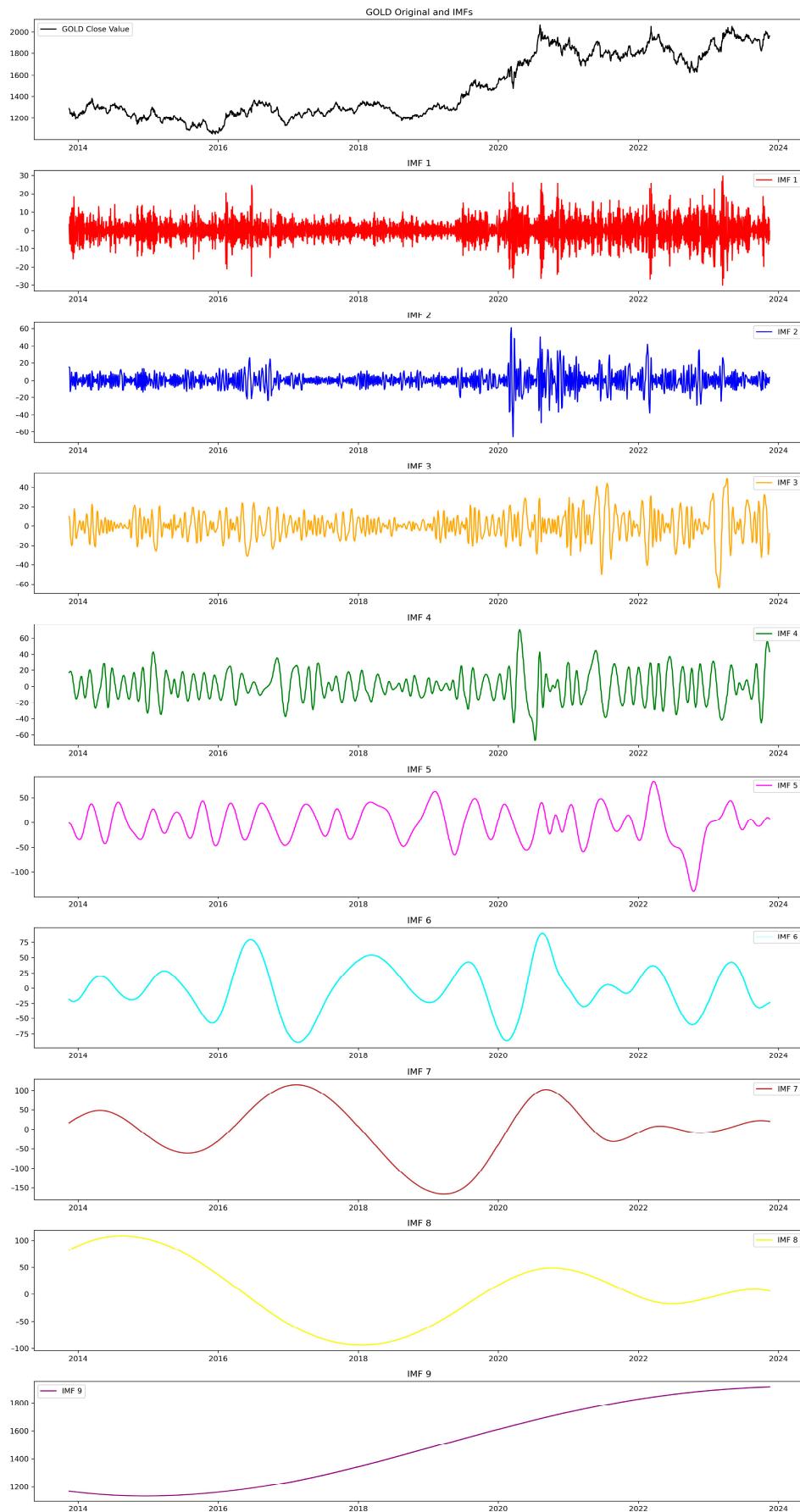


Figure 8. GOLD close actual values and IMFs.

3.4. Tools and Software

The code used in this study was developed using Python 3.12 programming language and Google Colab tool. We used several key libraries and tools to support the data analysis, model development, and visualization tasks. Python was the primary programming language used due to its versatility, simplicity, and extensive library ecosystem, which supports a wide variety of applications from basic data manipulation to complex simulations. For deep learning model development and validation, we employed TensorFlow 2.2 framework (Google Inc., Mountain View, CA, USA), a powerful open-source library for numerical computation and machine learning, along with Keras 3.4 open-source library (Google Inc., Mountain View, CA, USA), which provides a high-level interface for building and training neural networks, simplifying the creation of intricate models. We also utilized PyEMD 1.0.0 library for empirical mode decomposition and scikit-learn 1.5.1 Python library (sklearn) for machine learning tasks, namely, model training and evaluation. For financial analysis and charting, TradingView was used to provide advanced charting tools and comprehensive market data, aiding in visualizing and interpreting financial trends efficiently. Google Colab served as the cloud-based environment for coding and running the machine learning experiments. It offered access to powerful computational resources, including NVIDIA GPUs and Google TPUs, which significantly enhanced the efficiency of our model training and evaluation processes. Additionally, Google Colab's seamless integration with Google Drive facilitated the management of datasets and storage of code results. These tools and resources collectively enhanced the effectiveness of our work, enabling robust data analysis, accurate modeling, and clear representation of results.

To ensure the reusability of the method, the source codes are publicly available in the GitHub repository (<https://github.com/ojaayojaay/Financial-Forecasting-with-EMD-TI-LSM> (accessed on 28 August 2024)). This repository contains all relevant codes and datasets, as well as instructions for replicating the experiments.

4. Experimental Studies

4.1. Evaluation Metrics

In this study, we employed the MAPE, RMSE, and MAE as key evaluation metrics to assess the accuracy of our forecasting models. These measures provide a quantitative evaluation of the deviations between the forecasted and observed values. The mathematical formulas for the metrics are presented individually in Equations (24)–(26) in the following subsections.

4.1.1. Mean Absolute Percentage Error (MAPE)

MAPE measures the magnitude of the error relative to the actual values and is expressed as a percentage. It is calculated by dividing the mean of the absolute errors by the actual values, with the result multiplied by 100 to obtain a percentage. This metric is particularly useful for understanding the error in relation to the scale of the predicted values. However, its applicability can be limited in certain situations, as it may produce undefined or excessively large values due to division by zero or extremely small actual values.

$$\text{MAPE} = \left(\frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \right) \times 100\% \quad (24)$$

where

- n : Number of observations;
- y_i : Actual value;
- \hat{y}_i : Predicted value.

4.1.2. Root Mean Square Error (RMSE)

The RMSE is a quadratic evaluation metric that indicates the mean size of the error. It is computed as the square root of the arithmetic mean of the squared differences between

the predicted and actual values. Because the errors are squared, the RMSE places a higher weight on larger errors compared to smaller errors, making it particularly sensitive to outliers. This characteristic makes the RMSE a valuable metric when large errors are especially unacceptable. However, the scale of the RMSE is dependent on the dataset, making it challenging to compare the RMSE values across different datasets or with other evaluation metrics.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (25)$$

where

- n : Number of observations;
- y_i : Actual value;
- \hat{y}_i : Predicted value.

4.1.3. Mean Absolute Error (MAE)

MAE measures the average size of errors in a prediction set, disregarding their direction. It is determined by averaging the absolute deviations between the actual and predicted values. Unlike RMSE, which penalizes larger errors more heavily, MAE treats all errors equally, producing a linear score that reflects the average deviation of the forecasts. MAE is particularly suitable for applications where the significance of errors is uniform, regardless of their direction.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (26)$$

where

- n : Number of observations;
- y_i : Actual value;
- \hat{y}_i : Predicted value.

Each method has its strengths and weaknesses, and the selection of a metric may hinge on the particular use case and the goals of the analysis. For example, while the MAPE is intuitive and easy to understand, it might create difficulties when the actual values are zero or around zero. The RMSE emphasizes larger errors, which can be both beneficial and problematic depending on the context, as it may amplify the impact of outliers. The MAE provides a straightforward measure of the average error magnitude but does not distinguish between systematic and random errors. Therefore, choosing the correct metric should match the objectives of the analysis and the characteristics of the data being mathematically evaluated.

4.2. Results

The performance evaluation of the forecasting models reveals significant differences in accuracy across the tested financial assets. In particular, the proposed EMD-TI-LSTM model consistently outperforms the traditional LSTM model across key metrics, as detailed in Tables 5 and 6. The results showcase the EMD-TI-LSTM model's enhanced capability in predicting the future prices of BTC, BIST, NASDAQ, and GOLD, with notable improvements across all metrics, including MAPE, RMSE, and MAE.

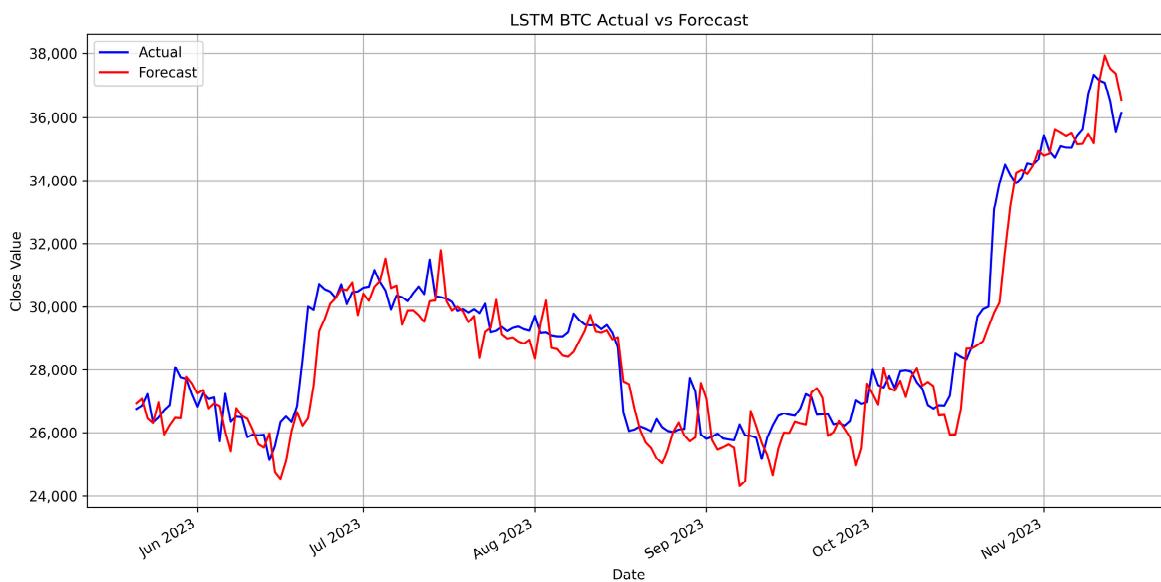
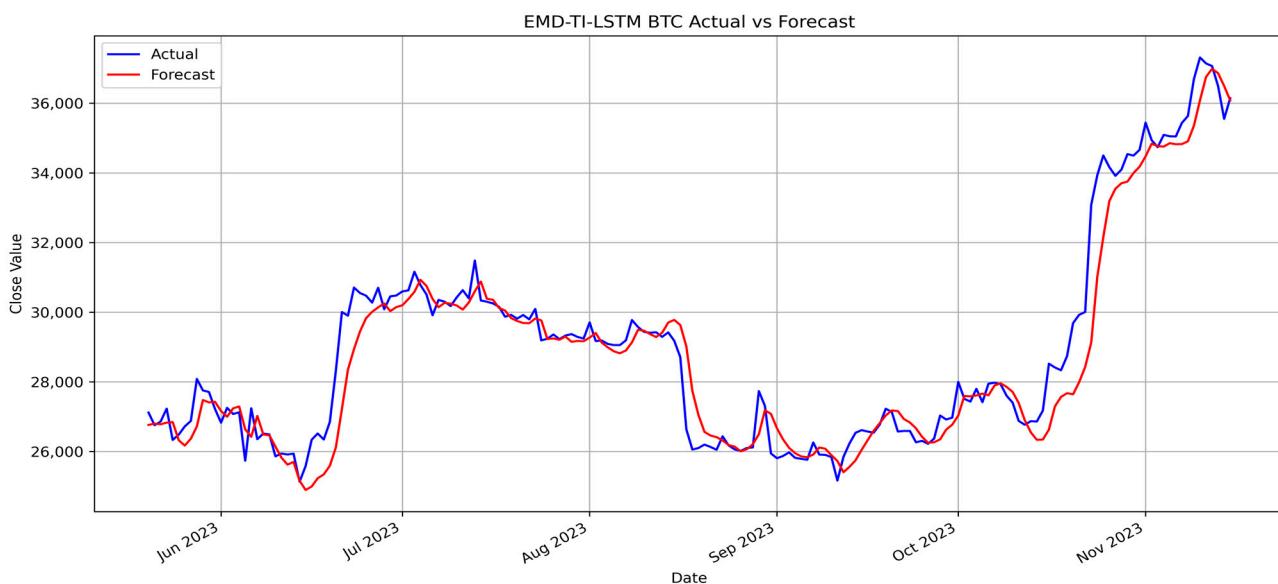
Table 5. Results of LSTM model for various metrics.

Metrics	BTC	BIST	NASDAQ	GOLD
MAPE	2.41	5.38	5.36	0.92
RMSE	941	424	861	23
MAE	698	370	809	18

Table 6. Results of the EMD-TI-LSTM model for various metrics.

Metrics	BTC	BIST	NASDAQ	GOLD
MAPE	1.69	3.47	1.14	0.79
RMSE	767	273	207	19
MAE	493	242	170	15

The prediction effectiveness of the models is further illustrated through the analysis of forecast plots for BTC, NASDAQ, BIST, and GOLD. These plots, depicted in Figures 9–16, compare the traditional LSTM model with the proposed EMD-TI-LSTM model. The time range for the plots extends from June to November 2023, with the vertical axis displaying the close values of the assets.

**Figure 9.** LSTM model prediction for close values of BTC.**Figure 10.** EMD-TI-LSTM model prediction for close values of BTC.

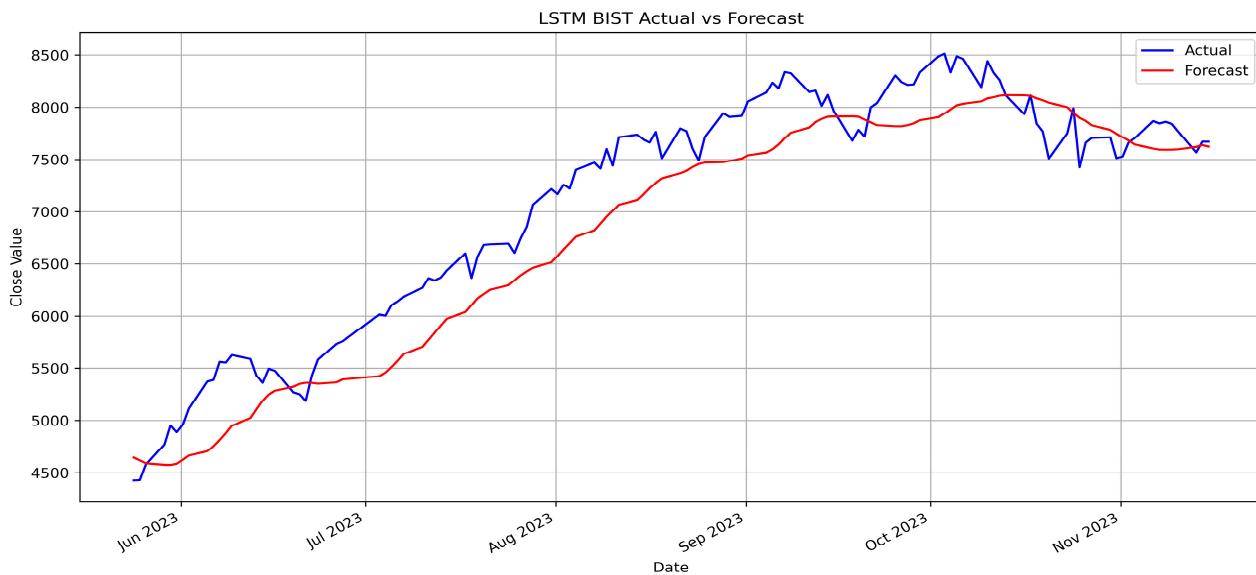


Figure 11. LSTM model prediction for close values of BIST.

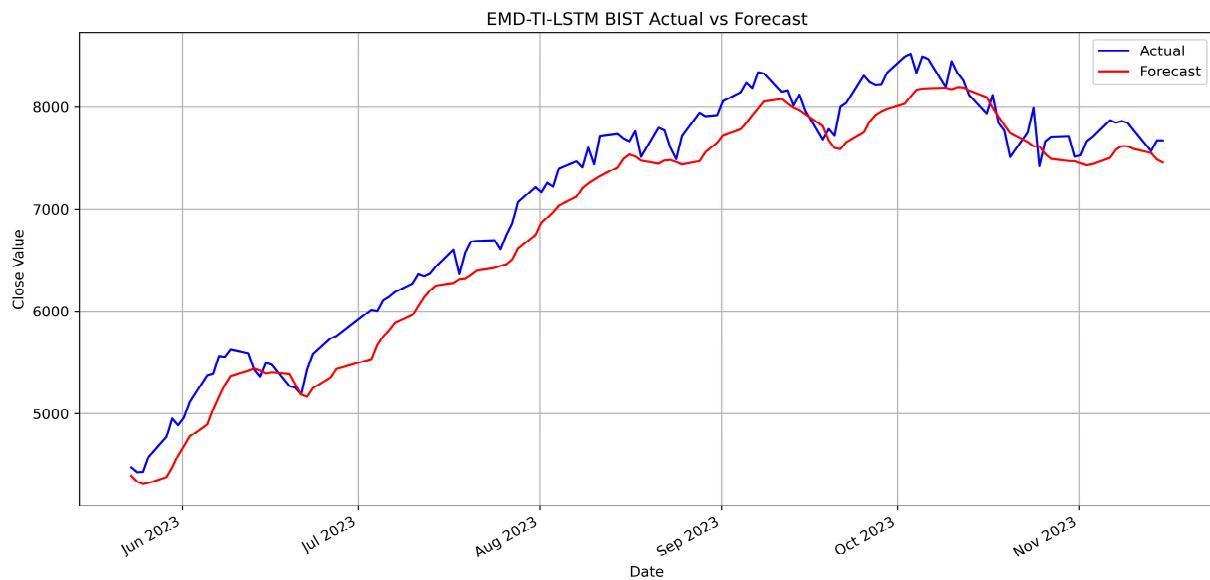


Figure 12. EMD-TI-LSTM model prediction for close values of BIST.

Figure 9 illustrates the performance of the LSTM model in predicting BTC, highlighting the discrepancy between the model's predictions and actual market values. This gap arises because the LSTM model, which relies on sequential price data, often struggles to accurately track BTC's sharp volatility, leading to greater discrepancies and reduced precision. Conversely, Figure 10 demonstrates the performance of the EMD-TI-LSTM model, which outperforms the standard LSTM model in predicting BTC. The enhanced model provides a more accurate representation of actual prices, offering improved forecasts with minimized error margins. This increased accuracy underscores the model's capability to effectively handle the complex, non-linear, and non-stationary nature of BTC's market data.

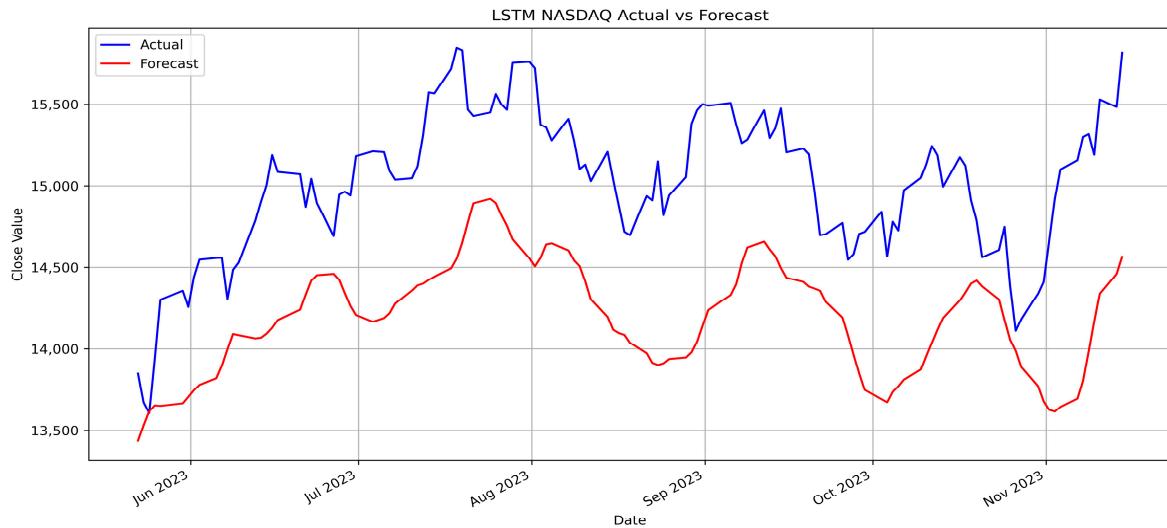


Figure 13. LSTM model prediction for the close values of NASDAQ.

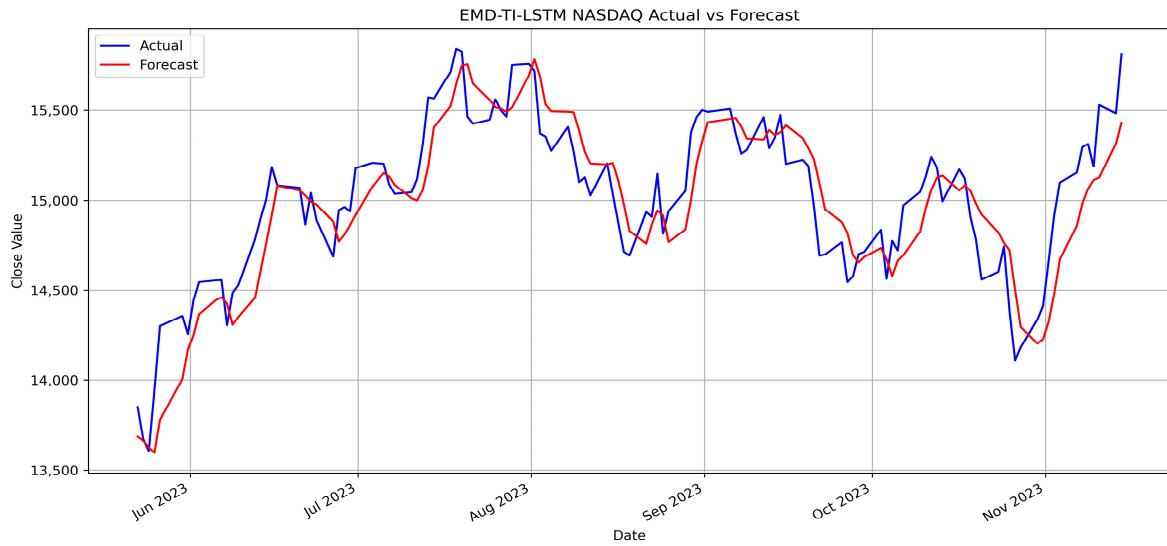


Figure 14. EMD-TI-LSTM model prediction for close values of NASDAQ.

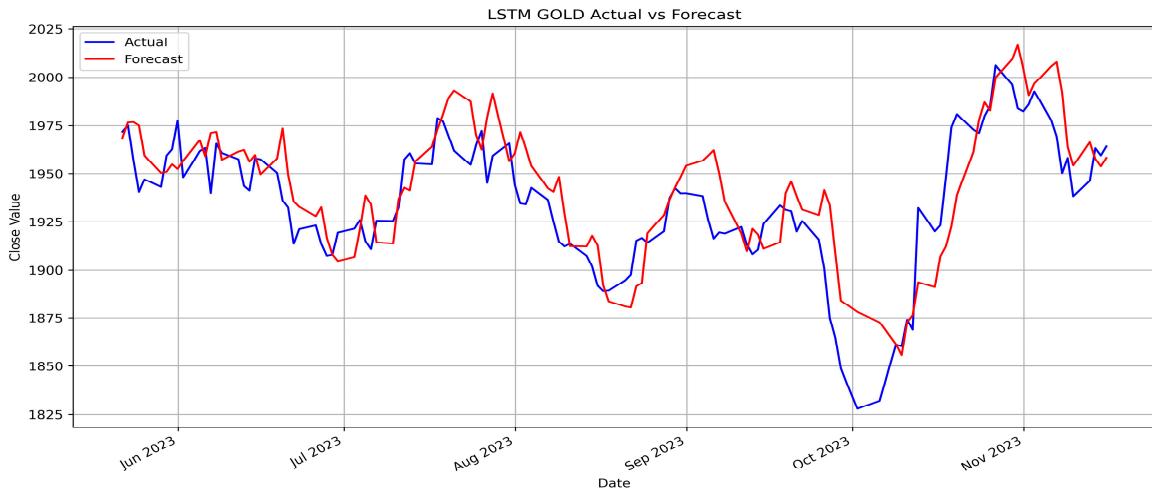


Figure 15. LSTM model prediction for close values of GOLD.

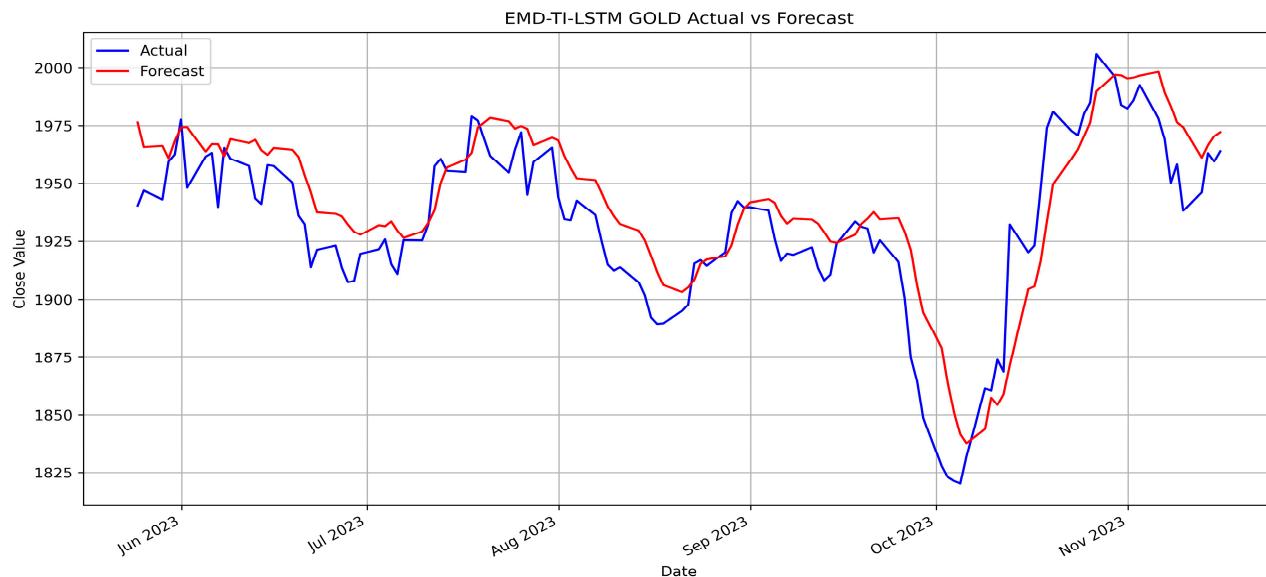


Figure 16. EMD-TI-LSTM model prediction for close values of GOLD.

Figures 11 and 12 show the actual and forecasted values of the BIST index obtained using the LSTM and EMD-TI-LSTM models, respectively. The LSTM model, similar to its performance with BTC, exhibited substantial errors with the BIST data, often failing to capture sharp price movements accurately. In contrast, the EMD-TI-LSTM model demonstrates a significant improvement in forecast accuracy. Overall, this model enhances market awareness and improves the prediction accuracy for the BIST data.

The predictions for the NASDAQ index are revealed in Figures 13 and 14 for the LSTM and EMD-TI-LSTM models, respectively. It is evident that the LSTM model, as shown in Figure 13, exhibits a lag and less accurate pattern compared to the actual closing prices. The forecast errors are significant, reflecting the model's difficulty in capturing the complexity of NASDAQ price movements. In contrast, the EMD-TI-LSTM model yields a more exact prediction, aligning closer to the actual data. By decomposing the signal into intrinsic mode functions and incorporating technical indicators, this model offers a more reliable forecast that better reflects real market conditions.

The predictions for GOLD are illustrated in Figures 15 and 16 for the LSTM model and the EMD-TI-LSTM model, respectively. Specifically, Figure 15 shows that while the LSTM model performs relatively well compared to other assets, it still exhibits forecast errors and delays, particularly during significant value swings. In contrast, the EMD-TI-LSTM model delivers more convincing results, with forecast values aligning closely with the observed gold prices. A comprehensive analysis of these figures highlights the superior performance achieved by the EMD-TI-LSTM model across a diverse range of financial assets. By integrating EMD and technical analysis within its architecture, the model is exceptionally well-suited to identifying underlying market trends and patterns, leading to more precise and dependable forecasts. This comparative study underscores the significant potential of hybrid models in financial forecasting, providing crucial guidance for investors in financial markets.

5. Discussion

In analyzing the forecast accuracy of both models, Table 7 illustrates the clear performance advantage of the EMD-TI-LSTM model over the standard LSTM model across four different financial assets: BTC, BIST, NASDAQ, and GOLD. The results demonstrate that the EMD-TI-LSTM model consistently delivers lower MAPE values, reflecting its superior predictive capability. The most significant improvement is observed in the NASDAQ asset, where the MAPE is reduced by 78.73%, while the smallest improvement is seen in the GOLD asset, with a reduction of 14.13%. These observations emphasize the effectiveness of

the EMD-TI-LSTM model, which achieved a substantial improvement of 39.56% in forecast accuracy across all examined financial assets, on average, based on the MAPE metric.

Table 7. Comparative analysis of MAPE values for LSTM and EMD-TI-LSTM models across various financial assets.

Model	BTC	BIST	NASDAQ	GOLD
LSTM	2.41	5.38	5.36	0.92
EMD-TI-LSTM	1.69	3.47	1.14	0.79
MAPE Improvement	29.88%	35.50%	78.73%	14.13%

Table 8 showcases the comparison of RMSE values for the LSTM model versus the EMD-TI-LSTM model for forecasting the same set of financial assets. The results demonstrate that the EMD-TI-LSTM model consistently delivers lower RMSE values, indicating enhanced forecasting precision. Similar to the MAPE metric, the most substantial improvement is observed in the NASDAQ asset, where the RMSE decreases by 75.96%, while the smallest improvement is noted in the GOLD asset, with a reduction of 17.39%. These findings reinforce the EMD-TI-LSTM model's ability to significantly reduce the average magnitude of forecast errors across various financial assets. The results demonstrate the strong performance of the EMD-TI-LSTM model, showing an average 36.86% enhancement in forecast accuracy for all analyzed financial assets, as indicated by the RMSE metric.

Table 8. Comparative analysis of RMSE values for LSTM and EMD-TI-LSTM models across various financial assets.

Model	BTC	BIST	NASDAQ	GOLD
LSTM	941	424	861	23
EMD-TI-LSTM	767	273	207	19
RMSE Improvement	18.49%	35.61%	75.96%	17.39%

Table 9 presents a comparison of the MAE values between the LSTM and EMD-TI-LSTM models when forecasting the same financial assets. The EMD-TI-LSTM model consistently outperforms the LSTM model, achieving lower MAE values across all assets. Notably, the most significant reduction in MAE is observed for the NASDAQ asset, with a decrease of 78.99%, while the smallest improvement is seen in the GOLD asset, with a reduction of 16.67%. These reductions in MAE values further support the efficacy of the EMD-TI-LSTM model in delivering more accurate and reliable financial forecasts compared to the traditional LSTM model. The analysis reveals that the EMD-TI-LSTM model significantly outperforms the conventional LSTM, achieving an average 39.90% improvement in forecast accuracy across all financial assets, as indicated by the MAE metric.

Table 9. Comparative analysis of MAE values for LSTM and EMD-TI-LSTM models across various financial assets.

Model	BTC	BIST	NASDAQ	GOLD
LSTM	698	370	809	18
EMD-TI-LSTM	493	242	170	15
MAE Improvement	29.37%	34.59%	78.99%	16.67%

Table 10 presents an in-depth comparison of various state-of-the-art methods [59–67] developed between 2018 and 2023 for predicting BTC prices measured using the MAPE metric. The methods exhibit considerable variability in prediction accuracy, with MAPE values ranging from 1.73 to 3.55, and an average MAPE of 2.96 across these approaches.

Notably, our proposed EMD-TI-LSTM model significantly outperforms these methods, achieving a lower MAPE of 1.69. This represents a 42.91% improvement relative to the average MAPE of the other techniques listed. The enhanced performance of the EMD-TI-LSTM model is due to the successful integration of empirical mode decomposition (EMD) and technical indicators (TI) with LSTM, which boosts the model's proficiency in accurately understanding and forecasting the complex behaviors of BTC prices. This improvement underscores the robustness and precision of the EMD-TI-LSTM approach compared to existing state-of-the-art methods, further validating its effectiveness in financial forecasting.

Table 10. Performance comparisons between EMD-TI-LSTM and state-of-the-art methods for BTC assets according to the MAPE metric.

Reference	Year	Method	MAPE
[59]	2023	GRU	1.73
[60]	2023	BiGRU	3.41
[61]	2023	Random Forest Regression	3.29
[62]	2023	HMSA-ARIMA	3.04
[63]	2022	Holt-Winter	2.61
[64]	2019	ARIMA (4,1,4)	2.92
[65]	2019	LSTM with Random Data Split	3.52
[66]	2019	EWT-LSTM-CS	3.55
[67]	2018	LSTM with AR(2)	2.55
Average			2.96
Proposed Model		EMD-TI-LSTM	1.69

6. Conclusions and Future Works

In this paper, we introduced a new hybrid model, named EMD-TI-LSTM, designed to advance financial forecasting by integrating empirical mode decomposition (EMD), technical indicators (TI), and long short-term memory (LSTM). The consistent performance across various metrics and datasets highlights the scalability and applicability of EMD-TI-LSTM in different financial forecasting scenarios, making it a versatile tool for both researchers and practitioners. Our results clearly demonstrate that EMD-TI-LSTM expressively outperformed traditional LSTM and other state-of-the-art methods in predicting financial asset prices. The model achieves consistently lower MAPE, RMSE, and MAE values, indicating a substantial enhancement in prediction accuracy. Mathematically, the EMD-TI-LSTM model improved the accuracy by 39.56%, 36.86%, and 39.90% over the conventional LSTM model on the BTC, BIST, NASDAQ, and GOLD datasets, as measured by the MAPE, RMSE, and MAE metrics, respectively. Notably, the model achieved a MAPE of 1.69 for the BTC dataset, reflecting a remarkable 42.91% improvement compared to the average MAPE of 2.96 from other state-of-the-art methods. Additionally, the innovative integration of EMD and TI not only simplifies complex market signals but also enriches the model with valuable market insights, leading to superior predictive performance. The mathematical methodology and findings of this study offer a reliable framework for future research.

Future inquiries could proceed with using the EMD-TI-LSTM method by exploring several promising avenues. First, a web/mobile application can be implemented to offer a user-friendly interface for the EMD-TI-LSTM model, facilitating performance analyses. Second, expanding the applications of the presented model to a diverse range of financial assets will help evaluate its generalizability and reveal its strengths and limitations in several marketplace conditions. Another promising area for research is applying the EMD-TI-LSTM model to real-world challenges across various fields. Domain-specific studies can uncover the practical benefits of this method in different areas. To conclude, the EMD-TI-LSTM method signifies meaningful progress in AI-powered financial forecasting, offering a potent alternative to traditional approaches and facilitating further innovative strides in the sector.

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

Funding: This research received no external funding.

Data Availability Statement: All data are publically available on the TradingView website (<https://www.tradingview.com>), accessed on 16 November 2023), including the “BTC/USD” dataset (BTC) (<https://www.tradingview.com/symbols/BTCUSD/?exchange=INDEX>, accessed on 16 November 2023), the “BIST 100 Index” dataset (BIST) (<https://www.tradingview.com/symbols/BIST-XU100/>, accessed on 16 November 2023), the “NASDAQ 100 Index” dataset (NASDAQ) (<https://www.tradingview.com/symbols/NASDAQ-NDX/>, accessed on 16 November 2023), and finally the “GOLD/USD” dataset (GOLD) (<https://www.tradingview.com/symbols/XAUUSD/?exchange=OANDA>, accessed on 16 November 2023).

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

Abbreviations

AI	Artificial intelligence
ANN	Artificial neural networks
ARIMA	Autoregressive integrated moving average
ATR	Average true range
BB	Bollinger bands
BiGRU	Bidirectional gated recurrent unit
BIST	Borsa Istanbul
BTC	Bitcoin
CCI	Commodity channel index
CEEMDAN	Complete ensemble empirical mode decomposition with adaptive noise
CMF	Chaikin money flow
CNN	Convolutional neural networks
CSI	Cubic spline interpolation
DL	Deep learning
DMI	Directional movement index
DNN	Deep neural networks
ELM	Extreme learning machine
EMA	Exponential moving average
EMD	Empirical mode decomposition
EMD-TI-LSTM	Empirical mode decomposition-technical indicators-long short-term memory
EWT-LSTM-CS	Empirical wavelet transforms long short-term memory cuckoo search
Forex	Foreign exchange
GRU	Gated recurrent unit
HMSA- ARIMA	Hybrid model sentiment analysis autoregressive integrated moving average
IHS	Improved harmony search
IMF	Intrinsic mode functions
LSTM	Long short-term memory
MA	Moving averages
MACD	Moving average convergence divergence
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine learning
MLP	Multilayer perceptron
NLP	Natural language processing
OBV	On-balance volume
RMSE	Root mean square error
RNN	Recurrent neural networks

RSI	Relative strength index
SMA	Simple moving average
SVM	Support vector machines
TI	Technical indicators
VIX	Volatility index
VMD	Variational mode decomposition

References

1. Sahu, S.K.; Mokhade, A.; Bokde, N.D. An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges. *Appl. Sci.* **2023**, *13*, 1956. [[CrossRef](#)]
2. Mukhamediev, R.I.; Popova, Y.; Kuchin, Y.; Zaitseva, E.; Kalimoldayev, A.; Symagulov, A.; Levashenko, V.; Abdoldina, F.; Gopejenko, V.; Yakunin, K.; et al. Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics* **2022**, *10*, 2552. [[CrossRef](#)]
3. Borna, S.; Maniaci, M.J.; Haider, C.R.; Gomez-Cabello, C.A.; Pressman, S.M.; Haider, S.A.; Demaerschalk, B.M.; Cowart, J.B.; Forte, A.J. Artificial Intelligence Support for Informal Patient Caregivers: A Systematic Review. *Bioengineering* **2024**, *11*, 483. [[CrossRef](#)] [[PubMed](#)]
4. Onim, M.S.H.; Thapliyal, H.; Rhodus, E.K. Utilizing Machine Learning for Context-Aware Digital Biomarker of Stress in Older Adults. *Information* **2024**, *15*, 274. [[CrossRef](#)]
5. Maehara, R.; Benites, L.; Talavera, A.; Aybar-Flores, A.; Muñoz, M. Predicting Financial Inclusion in Peru: Application of Machine Learning Algorithms. *J. Risk Financ. Manag.* **2024**, *17*, 34. [[CrossRef](#)]
6. Dixon, M.F.; Halperin, I.; Bilokon, P. *Machine Learning in Finance*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1406.
7. Warin, T.; Stojkov, A. Machine Learning in Finance: A Metadata-Based Systematic Review of the Literature. *J. Risk Financ. Manag.* **2021**, *14*, 302. [[CrossRef](#)]
8. Rouf, N.; Malik, M.B.; Arif, T.; Sharma, S.; Singh, S.; Aich, S.; Kim, H.-C. Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions. *Electronics* **2021**, *10*, 2717. [[CrossRef](#)]
9. Chapman, J.T.E.; Desai, A. Macroeconomic Predictions Using Payments Data and Machine Learning. *Forecasting* **2023**, *5*, 652–683. [[CrossRef](#)]
10. Modreanu, A.; Toma, S.-G.; Burcea, M.; Grădinaru, C. Perceptions and Attitudes of SMEs and MNCs Managers Regarding CSR Implementation: Insights from Companies Operating in the Retail Sector. *Sustainability* **2024**, *16*, 3963. [[CrossRef](#)]
11. Montazerian, M.; Leymarie, F.F. Simple Hybrid Camera-Based System Using Two Views for Three-Dimensional Body Measurements. *Symmetry* **2024**, *16*, 49. [[CrossRef](#)]
12. Iqbal, U.; Davies, T.; Perez, P. A Review of Recent Hardware and Software Advances in GPU-Accelerated Edge-Computing Single-Board Computers (SBCs) for Computer Vision. *Sensors* **2024**, *24*, 4830. [[CrossRef](#)] [[PubMed](#)]
13. Mahdi, A.E.; Azouz, A.; Noureldin, A.; Abosekeen, A. A Novel Machine Learning-Based ANFIS Calibrated RISS/GNSS Integration for Improved Navigation in Urban Environments. *Sensors* **2024**, *24*, 1985. [[CrossRef](#)] [[PubMed](#)]
14. Yun, K.; Yun, H.; Lee, S.; Oh, J.; Kim, M.; Lim, M.; Lee, J.; Kim, C.; Seo, J.; Choi, J. A Study on Machine Learning-Enhanced Roadside Unit-Based Detection of Abnormal Driving in Autonomous Vehicles. *Electronics* **2024**, *13*, 288. [[CrossRef](#)]
15. Ghasemkhani, B.; Yilmaz, R.; Birant, D.; Kut, R.A. Logistic Model Tree Forest for Steel Plates Faults Prediction. *Machines* **2023**, *11*, 679. [[CrossRef](#)]
16. Karimzadeh, M.; Basvoju, D.; Vakanski, A.; Charit, I.; Xu, F.; Zhang, X. Machine Learning for Additive Manufacturing of Functionally Graded Materials. *Materials* **2024**, *17*, 3673. [[CrossRef](#)]
17. Jaiswal, P.; Setia, H.; Raghuwanshi, P.; Randhwawa, P. A Natural Language Processing Model for Predicting Five-Star Ratings of Video Games on Short-Text Reviews. *Eng. Proc.* **2023**, *59*, 58. [[CrossRef](#)]
18. Guste, R.R.A.; Ong, A.K.S. Machine Learning Decision System on the Empirical Analysis of the Actual Usage of Interactive Entertainment: A Perspective of Sustainable Innovative Technology. *Computers* **2024**, *13*, 128. [[CrossRef](#)]
19. Al-Buenain, A.; Haouari, M.; Jacob, J.R. Predicting Fan Attendance at Mega Sports Events—A Machine Learning Approach: A Case Study of the FIFA World Cup Qatar 2022. *Mathematics* **2024**, *12*, 926. [[CrossRef](#)]
20. Suh, J.H. Multi-Label Prediction-Based Fuzzy Age Difference Analysis for Social Profiling of Anonymous Social Media. *Appl. Sci.* **2024**, *14*, 790. [[CrossRef](#)]
21. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
22. Van Jaarsveldt, C.; Peters, G.W.; Ames, M.; Chantler, M. Tutorial on empirical mode decomposition: Basis decomposition and frequency adaptive graduation in non-stationary time series. *IEEE Access* **2023**, *11*, 94442–94478. [[CrossRef](#)]
23. Padhi, D.K.; Padhy, N.; Bhoi, A.K.; Shafi, J.; Ijaz, M.F. A Fusion Framework for Forecasting Financial Market Direction Using Enhanced Ensemble Models and Technical Indicators. *Mathematics* **2021**, *9*, 2646. [[CrossRef](#)]

24. Zhou, Y.; Wang, L.; Qian, J. Application of Combined Models Based on Empirical Mode Decomposition, Deep Learning, and Autoregressive Integrated Moving Average Model for Short-Term Heating Load Predictions. *Sustainability* **2022**, *14*, 7349. [[CrossRef](#)]
25. Wilder, J.W. *New Concepts in Technical Trading Systems*; Trend Research: Greensboro, NC, USA, 1978.
26. Bollinger, J. Using Bollinger Bands. *Stock. Comod.* **1992**, *10*, 47–51.
27. Almeida, L.; Vieira, E. Technical Analysis, Fundamental Analysis, and Ichimoku Dynamics: A Bibliometric Analysis. *Risks* **2023**, *11*, 142. [[CrossRef](#)]
28. Frattini, A.; Bianchini, I.; Garzonio, A.; Mercuri, L. Financial Technical Indicator and Algorithmic Trading Strategy Based on Machine Learning and Alternative Data. *Risks* **2022**, *10*, 225. [[CrossRef](#)]
29. Van Houdt, G.; Mosquera, C.; Nápoles, G. A review on the long short-term memory model. *Artif. Intell. Rev.* **2020**, *53*, 5929–5955. [[CrossRef](#)]
30. Lazcano, A.; Herrera, P.J.; Monge, M. A Combined Model Based on Recurrent Neural Networks and Graph Convolutional Networks for Financial Time Series Forecasting. *Mathematics* **2023**, *11*, 224. [[CrossRef](#)]
31. Lindemann, B.; Müller, T.; Vietz, H.; Jazdi, N.; Weyrich, M. A survey on long short-term memory networks for time series prediction. *Procedia CIRP* **2021**, *99*, 650–655. [[CrossRef](#)]
32. Adewumi, T.; Sabry, S.S.; Abid, N.; Liwicki, F.; Liwicki, M. T5 for Hate Speech, Augmented Data, and Ensemble. *Science* **2023**, *5*, 37. [[CrossRef](#)]
33. Lu, P.; Liu, Z.; Zhang, T. A Machine Learning Model to Predict the Seismic Lifecycle Behavior of a Cross-Sea Cable-Stayed Bridge. *Buildings* **2024**, *14*, 1190. [[CrossRef](#)]
34. Wang, Z.; Yan, B.; Wang, H. Application of Deep Learning in Predicting Particle Concentration of Gas–Solid Two-Phase Flow. *Fluids* **2024**, *9*, 59. [[CrossRef](#)]
35. Gao, Y.; Zhao, Y.; Ma, Y.; Liu, Y. Prediction of Protein Secondary Structure Based on WS-BiLSTM Model. *Symmetry* **2022**, *14*, 89. [[CrossRef](#)]
36. Al-Dulaimi, O.A.H.H.; Kurnaz, S. A Hybrid CNN-LSTM Approach for Precision Deepfake Image Detection Based on Transfer Learning. *Electronics* **2024**, *13*, 1662. [[CrossRef](#)]
37. Li, X.; Lei, Y.; Ji, S. BERT- and BiLSTM-Based Sentiment Analysis of Online Chinese Buzzwords. *Future Internet* **2022**, *14*, 332. [[CrossRef](#)]
38. Lalapura, V.S.; Bhimavarapu, V.R.; Amudha, J.; Satheesh, H.S. A Systematic Evaluation of Recurrent Neural Network Models for Edge Intelligence and Human Activity Recognition Applications. *Algorithms* **2024**, *17*, 104. [[CrossRef](#)]
39. Kim, J.; Kim, H.-S.; Choi, S.-Y. Forecasting the S&P 500 Index Using Mathematical-Based Sentiment Analysis and Deep Learning Models: A FinBERT Transformer Model and LSTM. *Axioms* **2023**, *12*, 835. [[CrossRef](#)]
40. Shering, T.; Alonso, E.; Apostolopoulou, D. Investigation of Load, Solar and Wind Generation as Target Variables in LSTM Time Series Forecasting, Using Exogenous Weather Variables. *Energies* **2024**, *17*, 1827. [[CrossRef](#)]
41. Ma, Y.; Han, H.; Tang, X.; Chan, P.-W. Research on Short-Term Prediction Methods for Small-Scale Three-Dimensional Wind Fields. *Appl. Sci.* **2024**, *14*, 1871. [[CrossRef](#)]
42. Gandhamal, D.P.; Kumar, K. Systematic analysis and review of stock market prediction techniques. *Comput. Sci. Rev.* **2019**, *34*, 100190. [[CrossRef](#)]
43. Ferreira, F.G.D.C.; Gandomi, A.H.; Cardoso, R.T.N. Artificial Intelligence Applied to Stock Market Trading: A Review. *IEEE Access* **2021**, *9*, 30898–30917. [[CrossRef](#)]
44. Hu, Z.; Zhao, Y.; Khushi, M. A Survey of Forex and Stock Price Prediction Using Deep Learning. *Appl. Syst. Innov.* **2021**, *4*, 9. [[CrossRef](#)]
45. Singh, P.R.; Manohar, N.; Mahesh, R. Stock Price Prediction System with Improved LSTM. In Proceedings of the IEEE North Karnataka Subsection Flagship International Conference, Vijaypur, India, 20–21 November 2022; pp. 1–6. [[CrossRef](#)]
46. Mittal, S.; Chauhan, A. A RNN-LSTM-Based Predictive Modelling Framework for Stock Market Prediction Using Technical Indicators. *Int. J. Rough Sets Data Anal.* **2021**, *7*, 1–13. [[CrossRef](#)]
47. Babu, D.R.; Sathyanarayana, B. Design and Implementation of Technical Analysis Based LSTM Model for Stock Price Prediction. *Int. J. Recent Innov. Trends Comput. Commun.* **2023**, *11*, 1–7. [[CrossRef](#)]
48. Kaur, A.; Bhadauria, M.; Monika, M. Heuristic Approach for Forecasting Stock Price Using LSTM and Technical Indicators. In Proceedings of the 4th International Conference on Artificial Intelligence and Speech Technology, Delhi, India, 9–10 December 2022; pp. 1–6. [[CrossRef](#)]
49. Venikar, I.; Joshi, J.; Jalnekar, H.; Raut, S. Stock Market Prediction Using LSTM. *Int. J. Res. Appl. Sci. Eng. Technol.* **2022**, *10*, 920–924. [[CrossRef](#)]
50. Yang, Y.; Yang, Y.; Xiao, J. A Hybrid Prediction Method for Stock Price Using LSTM and Ensemble EMD. *Complexity* **2002**, *2020*, 6431712. [[CrossRef](#)]
51. Ali, M.; Khan, D.M.; Alshanbari, H.M.; El-Bagoury, A.A.-A.H. Prediction of Complex Stock Market Data Using an Improved Hybrid EMD-LSTM Model. *Appl. Sci.* **2023**, *13*, 1429. [[CrossRef](#)]
52. Xuan, Y.; Yu, Y.; Wu, K. Prediction of Short-term Stock Prices Based on EMD-LSTM-CSI Neural Network Method. In Proceedings of the 5th IEEE International Conference on Big Data Analytics (ICBDA), Xiamen, China, 8–11 May 2020; pp. 135–139. [[CrossRef](#)]

53. Jin, Z.; Yang, Y.; Liu, Y. Stock closing price prediction based on sentiment analysis and LSTM. *Neural Comput. Appl.* **2020**, *32*, 9713–9729. [[CrossRef](#)]
54. Jiang, M.; Jia, L.; Chen, Z.; Chen, W. The two-stage machine learning ensemble models for stock price prediction by combining mode decomposition, extreme learning machine and improved harmony search algorithm. *Ann. Oper. Res.* **2020**, *309*, 1–33. [[CrossRef](#)]
55. Shu, W.W.; Gao, Q. Forecasting Stock Price Based on Frequency Components by EMD and Neural Networks. *IEEE Access* **2020**, *8*, 206388–206395. [[CrossRef](#)]
56. Cao, J.; Li, Z.; Li, J. Financial time series forecasting model based on CEEMDAN and LSTM. *Phys. Stat. Mech. Appl.* **2019**, *519*, 127–139. [[CrossRef](#)]
57. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
58. TradingView—Track All Markets. Available online: <https://www.tradingview.com/> (accessed on 16 November 2023).
59. Khan, M.S.; Bazai, S.U.; Ghafour, M.I.; Marjan, S.; Ameen, M.; Shah, S.A.A. Forecasting Cryptocurrency Prices Using a Gated Recurrent Unit Neural Network. In Proceedings of the International Conference on Energy, Power, Environment, Control, and Computing, Gujrat, Pakistan, 8–9 March 2023; pp. 1–6. [[CrossRef](#)]
60. Singathala, H.; Malla, J.; Jayashree, J.; Vijayashree, J. A Deep Learning Based Model for Predicting the Future Prices of Bitcoin. In Proceedings of the 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 5–6 May 2023; pp. 1–4. [[CrossRef](#)]
61. Chen, J. Analysis of Bitcoin Price Prediction Using Machine Learning. *J. Risk Financial Manag.* **2023**, *16*, 51. [[CrossRef](#)]
62. Feizian, F.; Amiri, B. Cryptocurrency Price Prediction Model Based on Sentiment Analysis and Social Influence. *IEEE Access* **2023**, *11*, 142177–142195. [[CrossRef](#)]
63. Kurniawan, K.; Madelan, S. Forecasting Using Time Series Analysis Method in Cryptocurrency Period 2015–2022. *Int. J. Innov. Sci. Res. Technol.* **2022**, *7*, 1454–1459. [[CrossRef](#)]
64. Wirawan, I.M.; Widyaningtyas, T.; Hasan, M.M. Short Term Prediction on Bitcoin Price Using ARIMA Method. In Proceedings of the International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 21–22 September 2019; pp. 260–265. [[CrossRef](#)]
65. Ji, S.; Kim, J.; Im, H. A Comparative Study of Bitcoin Price Prediction Using Deep Learning. *Mathematics* **2019**, *7*, 898. [[CrossRef](#)]
66. Altan, A.; Karasu, S.; Bekiros, S. Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. *Chaos Solitons Fractals* **2019**, *126*, 325–336. [[CrossRef](#)]
67. Wu, C.-H.; Lu, C.-C.; Ma, Y.-F.; Lu, R.-S. A New Forecasting Framework for Bitcoin Price with LSTM. In Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 168–175. [[CrossRef](#)]

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