

Article

LSTM–Transformer-Based Robust Hybrid Deep Learning Model for Financial Time Series Forecasting

Md R. Kabir ^{1,*}, Dipayan Bhadra ², Moinul Ridoy ² and Mariofanna Milanova ^{1,*}

¹ Department of Computer Science, University of Arkansas, Little Rock, AR 72204, USA

² Applied Statistics and Data Science, Jahangirnagar University, Dhaka 1342, Bangladesh; dipu_dipayan@outlook.com (D.B.); moinulridoy30@gmail.com (M.R.)

* Correspondence: mrkabir@ualr.edu (M.R.K.); mgmilanova@ualr.edu (M.M.)

Abstract: The inherent challenges of financial time series forecasting demand advanced modeling techniques for reliable predictions. Effective financial time series forecasting is crucial for financial risk management and the formulation of investment decisions. The accurate prediction of stock prices is a subject of study in the domains of investing and national policy. This problem appears to be challenging due to the presence of multi-noise, nonlinearity, volatility, and the chaotic nature of stocks. This paper proposes a novel financial time series forecasting model based on the deep learning ensemble model LSTM-mTrans-MLP, which integrates the long short-term memory (LSTM) network, a modified Transformer network, and a multilayered perception (MLP). By integrating LSTM, the modified Transformer, and the MLP, the suggested model demonstrates exceptional performance in terms of forecasting capabilities, robustness, and enhanced sensitivity. Extensive experiments are conducted on multiple financial datasets, such as Bitcoin, the Shanghai Composite Index, China Unicom, CSI 300, Google, and the Amazon Stock Market. The experimental results verify the effectiveness and robustness of the proposed LSTM-mTrans-MLP network model compared with the benchmark and SOTA models, providing important inferences for investors and decision-makers.

Keywords: deep learning; financial time series; LSTM; Transformer; MLP; stock market; hybrid neural network



Academic Editor: Francesco Tajani

Received: 3 November 2024

Revised: 23 December 2024

Accepted: 7 January 2025

Published: 10 January 2025

Citation: Kabir, M.R.; Bhadra, D.; Ridoy, M.; Milanova, M. LSTM–Transformer-Based Robust Hybrid Deep Learning Model for Financial Time Series Forecasting. *Sci* **2025**, *7*, 7. <https://doi.org/10.3390/sci7010007>

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

1. Introduction

Universal fiscal sectors are becoming progressively interconnected and interdependent along with the rapid expansion of technology and communication. With the application of conventional equity and energy markets to the newly developed cryptocurrency industry, financial markets are continually growing and expanding. Predicting stock values remains a complex challenge due to their long-term unpredictability and chaotic nature [1]. Earlier market hypotheses suggested that stock prices move randomly, making predictions impossible. However, recent technical analyses indicate that historical records often influence stock prices, making trend analysis a key component of effective forecasting [2]. The study of financial time series has gained significant attention in both academic and professional domains because of its inherent complexity and importance in financial forecasting. Stock markets play a pivotal role in recent economies, exhibiting nonlinear behavior impacted by several factors, including political events, economic policies, and market sentiments. Due to this instability, stock price prediction in financial time series analysis is a difficult but crucial endeavor. Political, geographical, and socioeconomic concerns and a variety of factors influence the viability of the stock and cryptocurrency markets. The substantial

variability across these various aspects contributes to fluctuations in stock market trends. Confidently predicting financial market movements is vital for important research areas like derivative pricing, risk management, and financial time series analysis, empowering us to make informed decisions and drive success [3].

This study focuses on the stock market forecasting precision by overcoming the constraints of previous models. Stock prices are affected by diverse and often chaotic dynamics. These factors make it challenging to anticipate them using standard approaches. Existing research and models struggle in capturing the nonlinearity, multi-noise, and volatility inherent in financial markets. Previously, various models have been developed that focus on regional and specific stock markets. However, no single model can be seen in the literature that performs efficiently in both the diverging stock and cryptocurrency markets. This research is inspired by the strong need for a more robust single model that is capable of dealing with these complexities and thus providing reliable predictions in diverse financial markets. By utilizing recent discoveries in deep learning, notably in hybrid models incorporating several architectures, this study tries to increase the prediction accuracy for time series data.

Some studies focus on predicting stock prices using regression, while others use classification to predict trends. Investors and organizations, however, are mainly interested in future stock trends [4]. Random walk theory holds that stock prices can only be accurately forecasted by 50% of researchers and that stock prices cannot be anticipated using previous data. According to this idea, policies and news have the greatest impacts on stock prices. However, according to other academics, experimental results support the idea that historical data can offer valuable foresight [5]. The movement of equity trends is unpredictable, making these ventures risky. Additionally, it is often useful for governments to determine the market conditions. This is primarily because stock values are typically dynamic, nonparametric, and nonlinear in nature. As a result, these characteristics can lead to inadequate performance in statistical models, making it difficult to predict values and movements accurately [6,7]. RNNs are ideal for economic forecasts because they can use networks to maintain the recent memory occurrences and establish connections between individual network units [8,9]. An enhanced version of the RNN technique is identified as LSTM. To accuracy eliminate the shortcomings of RNNs, LSTM features three distinct gates. It can handle individual data points or entire data sequences.

This research presents a time series prediction model titled LSTM-mTrans-MLP, which combines an LSTM network, a modified Transformer network, and an MLP network to forecast the ending value of stock data. Initially, the proposed model implements LSTM to capture sequences of context for a prolonged duration and with higher robustness in terms of noise and missing data. Second, the feature vector produced by the LSTM is regarded as the input of the modified Transformer network, which utilizes a self-attention mechanism to simultaneously consider the relationship between a specific part of the feature vector and all other parts. The self-attention mechanism also allows our modified Transformer network to concentrate on the most important aspects of the feature sequence, boosting the model's capabilities to recognize the context and linkages within the feature vector. Third, an MLP further discovers and emulates the complex interconnections among the attributes, such as the output of the modified Transformer network and the model output, which enables the precise estimation of the next-day market valuation. Finally, we assess the suggested LSTM-mTrans-MLP model with the previously reported state-of-the-art (SOTA) models using Bitcoin, the Shanghai Composite Index, China Unicom, CSI 300, CSI 100, Google, and Amazon stock prices to verify the model's efficacy. The major achievement of this study is the evolution of an innovative hybrid forecasting model. It has a unique network size, shows superior capabilities, and gains diverse stock data from miscellaneous

market features by improving the combination of SOTA models. Unlike most existing works, in this work, the model architecture, including the layers and activation tasks, model size, parameter number, and hyperparameters, are the same for all cases considered in the investigation, which suggests the resilience and effectiveness of the proposed model.

2. Literature Review

2.1. Statistical Linear Models for Financial Time Series

Statistical models like ARMA have historically been used to examine linear properties—for instance, autocorrelation in financial time series—often serving as benchmarks. As an illustration, Ibrahim et al. [10] utilized ARMA, alongside other techniques like random forest and MLP, to predict Bitcoin's price movement, highlighting the advantages and limitations of such models. Similarly, Chevallier [11] proposed a nonparametric model for the estimation of carbon spot and futures prices, achieving lower prediction errors compared to traditional linear approaches. Zhao et al. [12] introduced a combination-MIDAS model, demonstrating its superior forecasting capabilities compared to AR, MA, and TGARCH models. While these methods effectively capture linear characteristics, they struggle with the nonlinear dynamics inherent in financial time series.

2.2. Nonlinear Models for Financial Time Series

Several attempts have been made to demonstrate the characteristics of nonlinear data, but the intricate nature of nonlinear dynamics frequently undermines the foundational assumptions of the model. To address the limitations of linear models, researchers have turned to nonlinear approaches. Artificial intelligence techniques, particularly neural networks, have gained interest for their capability to model complex nonlinear relationships [13–15]. These methods are frequently linked with traditional econometric and time series approaches to enhance the precision and reliability of predictions. For example, a hybrid prediction model has been introduced by Fenghua et al. [16] that incorporates singular spectrum analysis (SSA) with support vector machines (SVM), demonstrating a significant improvement in accuracy over standalone SVM and EEMD-SVM models. Similarly, Shen et al. [17] outlined a GRU-SVM hybrid framework, which leverages a GRU alongside SVM. Their comparative analysis highlighted the GRU-SVM model as superior to the individual GRU, SVM, and DNN models in terms of predictive performance.

In the context of cryptocurrency, Atsalakis et al. [18] established a hybrid neuro-fuzzy system named PATSOS, specifically designed for the forecasting of daily directional changes in Bitcoin prices. Another notable contribution by Nagula and Alexakis [19] involved creating a hybrid model capable of performing both classification and regression analyses. This model effectively identified critical factors influencing Bitcoin price fluctuations and provided accurate forecasts for its future trends. Zhu and Wei [20] tackled the challenge of carbon price prediction utilizing a hybrid methodology that combined ARIMA with a least-squares SVM. Their conclusions demonstrated enhanced prediction precision compared to conventional models.

Sun et al. [21] presented an innovative forecasting framework that integrated variational mode decomposition (VMD) with spiking neural networks (SNNs), proving highly effective for carbon futures price predictions. Similarly, Fan et al. [14] refined a forecasting method for carbon pricing using a multilayer perceptron (MLP), which excelled in capturing nonlinear patterns in the data. Furthermore, Atsalakis [22] employed computational intelligence to design three distinct algorithms for the prediction of carbon futures prices. Among these, the PATSOS model stood out, delivering the highest prediction accuracy in terms of carbon futures pricing.

2.3. Hybrid Deep Learning Models for Financial Time Series

The integration and advancement of deep learning, such as CNNs and LSTM networks, have significantly contributed to advancements in financial forecasting across various domains, including stock trading, exchange rates, electricity pricing, and crude oil valuation [23–27]. For instance, Ni et al. [23] introduced a C-RNN model, which combined the strengths of CNNs and RNNs to enhance the precision of exchange rate anticipation. Similarly, Long et al. [24] established a multi-filter neural network architecture that integrated CNN and RNN layers, effectively predicting stock price movements with improved precision, effectiveness, and constancy. The empirical research of Gonçalves et al. [25] was conducted using three distinct deep learning models, including a deep neural network classifier (DNNC), CNN, and LSTM, to evaluate their capabilities in predicting price movement in exchange markets. In addition, Peng et al. [26] employed a hybrid prediction architecture that combined LSTM networks with a differential evolution (DE) algorithm. This model was validated through electricity price forecasting experiments conducted in regions such as New South Wales, Germany, Austria, and France, and it exhibited remarkable predictive accuracy. Cen and Wang [27] utilized LSTM to construct a model capable of forecasting crude oil prices effectively. These studies highlight the growing reliance on hybrid approaches to tackle the complexities of financial time series. Several researchers have explored hybrid methodologies to address both linear and nonlinear characteristics in financial data. Zhang [28] merged ARIMA with neural networks (ANN) to leverage the strengths of both linear and nonlinear modeling, achieving improved forecasting accuracy. A hybrid model proposed by Pai and Lin [29] integrated ARIMA with SVM to forecast stock prices, outperforming benchmark models. An innovative forecasting model was developed by Shafie-khah and his team [30] for power market prices; it integrated wavelet transformation, the Autoregressive Integrated Moving Average (ARIMA) methodology, and radial basis function neural networks (RBFNs). This comprehensive approach effectively captured and analyzed linear and nonlinear behaviors present in the data, enhancing the accuracy of price predictions in a complex market environment. Jeong et al. [31] utilized SARIMA in conjunction with an ANN to develop a new method for the forecasting of South Korea's yearly energy expenditure, demonstrating improved accuracy in power consumption predictions. Leonardo Ranaldi et al. [32] built a "CryptoNet" system combined with an autoregressive multilayer ARNN simulator and achieved superior accuracy on Bitcoin and Ether time series. Despite these advancements, there is still a lack of clarity on how different modeling techniques can be optimally integrated or ensembled to address the intricate properties of financial time series data.

In more recent developments, a hybrid model known as CNN-BiLSTM-ECA was developed by Chen et al. [33] for the forecasting of stock prices, which was evaluated on datasets like China Unicom, the Shanghai Composite Index, and CSI 300, showcasing the model's strength and efficiency.

Kaijian et al. [34] developed an ensemble model that integrated ARMA and CNN-LSTM. This ARMA-CNN-LSTM approach was tested on three key financial datasets—Bitcoin, EU-ETS, and the Shanghai Composite Index—demonstrating superior performance compared to the baseline models.

Wang et al. [35] presented an enhanced model using interval-valued decomposition (FIVMD) with optimized deep learning techniques for stock price prediction. Their framework outperformed other comparative models across four datasets, including the Shenzhen Stock Exchange Index, Shanghai Stock Exchange Index, China Securities 100, and GEM.

Omoware et al. [36] focused on LSTM to predict stock series for Apple and Google. They evaluated their model by applying metrics such as the R2 score, MAE, RMSE, and MSE, and the results indicated a notable improvement over the SOTA machine learning algorithms.

From the literature, it can be seen hybrid models perform better than the base models. However, no reported work has combined LSTM and the Transformer as an encoder-decoder-based model to predict diverging financial markets. Moreover, most of the works suggest tuning the network size and the number of parameters for different datasets. Therefore, developing a single model architecture with a unique parameter number and model size that can efficiently forecast multiple financial markets from multiple datasets is a challenging task.

3. Methodology

3.1. LSTM

The LSTM network represents a refined adaptation of the RNN, distinguished by its recurrent connections within the hidden layer architecture. This unique design incorporates a feedback mechanism that spans multiple layers, making it particularly effective in modeling the nonlinear temporal dependencies found in time series. The LSTM model has been specifically designed to overcome the limitations of conventional RNNs, particularly the issues of vanishing and exploding gradients. It accomplishes this through an external feedback mechanism that repeatedly introduces the concealed condition from the preceding time step into the network's inputs, thus shaping future predictions [37]. The memory cell is fundamental to the LSTM architecture and a crucial component of the internal feedback system. This memory cell acts as an autonomous referencing component, enabling the retention of temporal information across prolonged durations. This capability enables the model to effectively address challenges related to exploding or vanishing gradients, which often impair traditional RNNs [38].

The architecture of a basic LSTM unit, as depicted in Figure 1, comprises a memory storage unit and three key control gates: the input gate, output gate, and forget gate. The input and hidden state are denoted by variables x_t , h_t at time t . The forget, input, and output gates are denoted by f_t , i_t , and o_t , while \tilde{C}_t represents the candidate information intended for storage. The input gate controls the volume of retained metrics. The specific calculations governing each gate, along with the input candidates, cell state, and hidden state, are defined in the following equations:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ h_t &= o_t \cdot \tanh(C_t). \end{aligned}$$

3.2. Transformer

The Transformer model, introduced by Google's team in 2017 [39], marked a substantial innovation within the domain of natural language processing (NLP). In contrast to traditional architectures based on recurrent neural networks (RNNs), the Transformer leverages a self-attention mechanism that eliminates the need for sequential data processing. This innovation enables the model to analyze input data simultaneously, significantly improving its efficiency while also allowing it to capture global dependencies within the dataset. Multiple encoder and decoder components are linked to the Transformer component. An encoder composed of multiple stacked layers transforms the raw input data into a

structured representation. The decoder then takes this encoded information and generates the desired output based on it. The encoder's multi-head self-attention mechanism is essential, enabling the model to identify and utilize dependencies across both short-term and long-term metrics. By concentrating on various parts of the input sequence simultaneously, the model extracts and emphasizes key features, enhancing its capability to understand intricate structures, as shown in Figure 2. The self-attention mechanism relies on three core matrices, namely Q (query), K (key), and V (value). They interact to compute the relationships between the elements in the sequence. The dimensionality of the key vector is denoted as d_K , and the relationships are mathematically represented as illustrated within Equation (1):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QV^T}{\sqrt{d_K}}\right)V \quad (1)$$

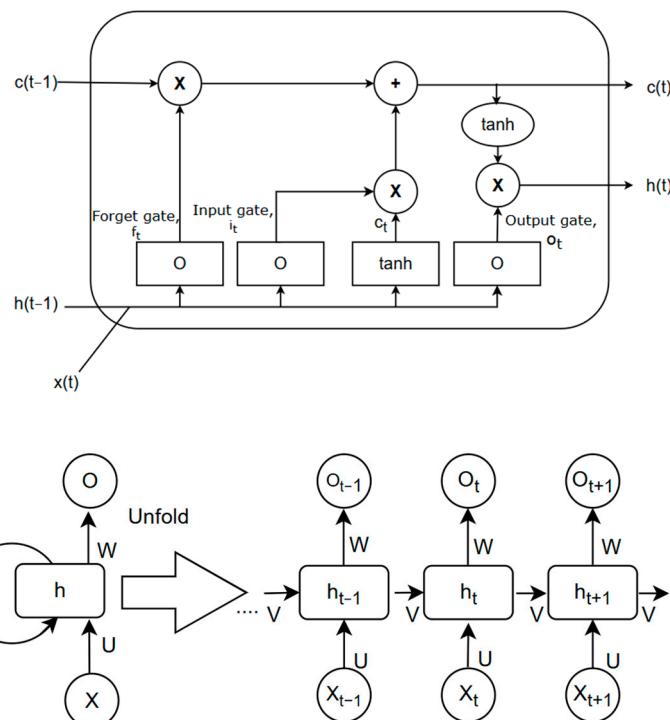


Figure 1. LSTM architecture.

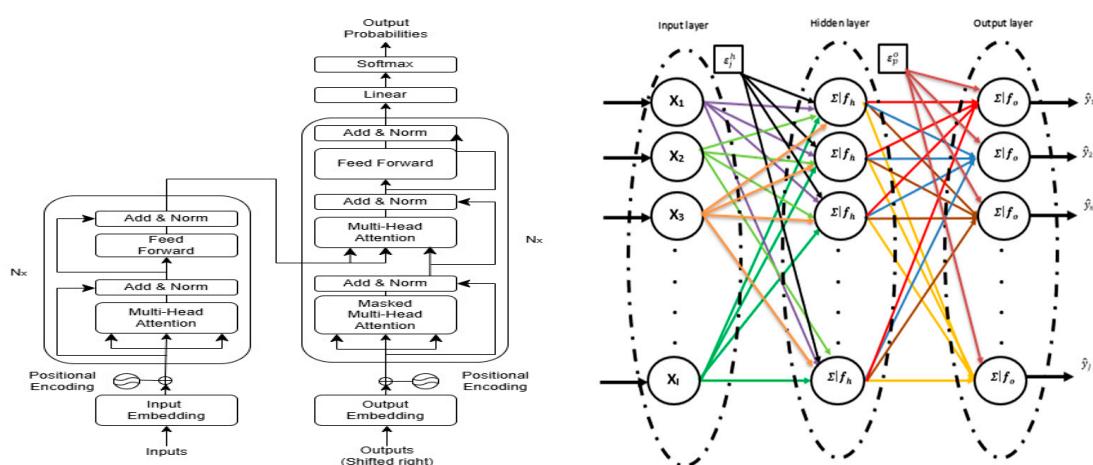


Figure 2. Transformer architecture model (left) and an MLP (3 layers) (right).

The Transformer excels in understanding the context, giving it unique capabilities in addressing temporal data forecasting challenges. For sequential financial data forecasting, a modified encoder component in the Transformer is used, which is employed as the core of the model.

3.3. Multilayer Perceptron (MLP)

A simple MLP incorporates multiple layers, including an input layer, a hidden layer, and finally an output layer. Each layer is composed of distinct neurons. This method involves multiplying the previous layer's output by a weighted matrix and later adding a bias term to adjust the results. Activation functions are applied at each layer to introduce nonlinearity and enable the model to learn complex relationships. The mathematical formulation of an MLP with this three-layer structure is depicted in Equation (2), emphasizing the transformation process between the layers.

$$\text{Output}_{\text{MLP}} = f^o \left[\sum_{j=1}^J W_{pj} \times f^h \left(\sum_{i=1}^I W_{ji} X_i + \xi_j^h \right) + \xi_p^o \right] \quad (2)$$

3.4. Proposed LSTM-mTrans-MLP Model

In this research, an efficient and robust model named LSTM-mTrans-MLP is developed for financial time series prediction by integrating an LSTM network, a modified Transformer network, and an MLP network as shown in Figure 3.

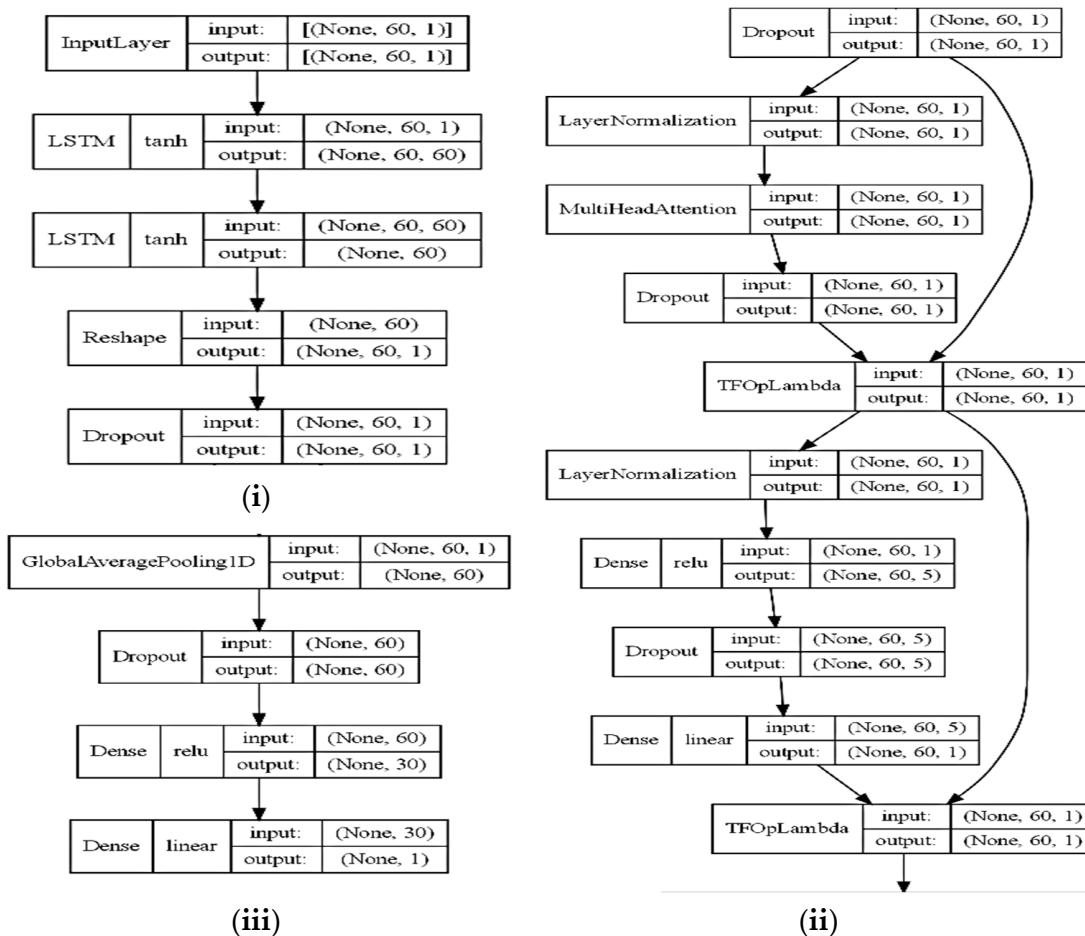


Figure 3. (i) LSTM network, (ii) mTransformer, and (iii) MLP block of the proposed LSTM-mTrans-MLP model.

First, our model utilizes LSTM to capture the sequence of contexts for a prolonged duration, with higher robustness in terms of noise and missing data. The LSTM network (Figure 3i) structure and parameters are established and outlined below: Input layer (60,1) -> LSTM layer (units = 60, activation = Tanh, recurrent_activation = Sigmoid, return_sequences = true) -> LSTM layer (units = 60, activation = Tanh, return_sequences = false) -> reshape layer (60,1) -> dropout layer (0.1). The following figure describes the LSTM network and the input–output size of the model. The first LSTM layer consists of 14,880 trainable parameters, with an additional 29,040 trainable parameters in the second LSTM layer. The reshape and dropout layers do not contain any trainable parameters, but, as the dropout hyperparameter, 0.1 (10%) has been used in this model. The dropout layer helps the model to become more generalized, thus enabling the utilization of the same architecture and size for multiple datasets.

After taking the output of the dropout layer of the LSTM (feature shape = [60, 1]) as an input, the network structure and parameters for the modified Transformer network, as shown in Figure 3ii, are set as follows: normalization layer ($\text{epsilon} = 1 \times 10^{-6}$, scale = true, gamma_normalization = “ones”) -> MultiHeadAttention Layer (attention_head_size = 120, number_heads = 5, dropout = 0.15) with input = (normalized feature (60,1); normalized feature (60,1)) -> dropout layer (0.15) -> residual layer = output of MultiHeadAttention + output of LSTM network -> normalization layer ($\text{epsilon} = 1 \times 10^{-6}$, scale = true, gamma_normalization = “ones”) -> dense layer (units = 5, activation = ReLU) -> dropout layer (0.15) -> dense layer (units = 60, activation = “linear”) -> residual layer = output of the last dense layer + output of the previous residual layer. Residual connections enable the more efficient training of deep neural networks by addressing the vanishing gradient challenge. They help to prevent degradation problems by allowing the direct flow of gradients through the network. They were first popularized by the introduction of ResNet (residual networks) by Kaiming He et al. in 2015 [40].

There are 4221 trainable parameters in the modified Transformer network. In the standard Transformer model by Google, normalization and addition are performed after the attention layer or the dense layer. In our work, we have normalized the inputs before the attention or dropout layer in the modified Transformer section. Input embedding (Figure 4) in the first model has also been deleted. The purpose of the input embedding module is to convert language and text into vector representations; however, stock price data do not require such text vectorization. The Transformer decoder is replaced with the MLP network described in the following paragraph. Furthermore, the decoder’s additional inputs are eliminated, retaining only the encoder’s output as its sole input. These modifications enable us to achieve consistently high performance across all key evaluation metrics, including the RMSE, MAE, MAPE, and R^2 score. The self-attention mechanism enables our modified Transformer network to concentrate on the most significant sections of the feature sequence, enhancing the model’s capability to recognize the context and linkages within the feature vector.

The modified Transformer network’s output is subsequently passed to the MLP network which is shown in Figure 3iii. The system configuration and parameters of the modified Transformer network are set as follows: GlobalAveragePooling layer (data_format = ‘channels_first’) -> dropout layer (0.10) -> dense layer (units = 30, activation = “ReLU”) -> dense layer (units = 1, activation = “linear”). The first dense layer has 1830 trainable and the last dense layer includes 31 trainable parameters of the MLP network. The MLP network further discovers and emulates the complex nonlinear relationships between the mTrans network’s output features and the model’s output, presenting the forecast value of the following day’s stock price. The model is trained incorporating a learning rate of 0.001. The Adam optimizer and the mean square error metric are applied as the loss function

to compile the model. The unit batch size is considered with epoch values from 12 to 30, based on the type and size of the datasets.

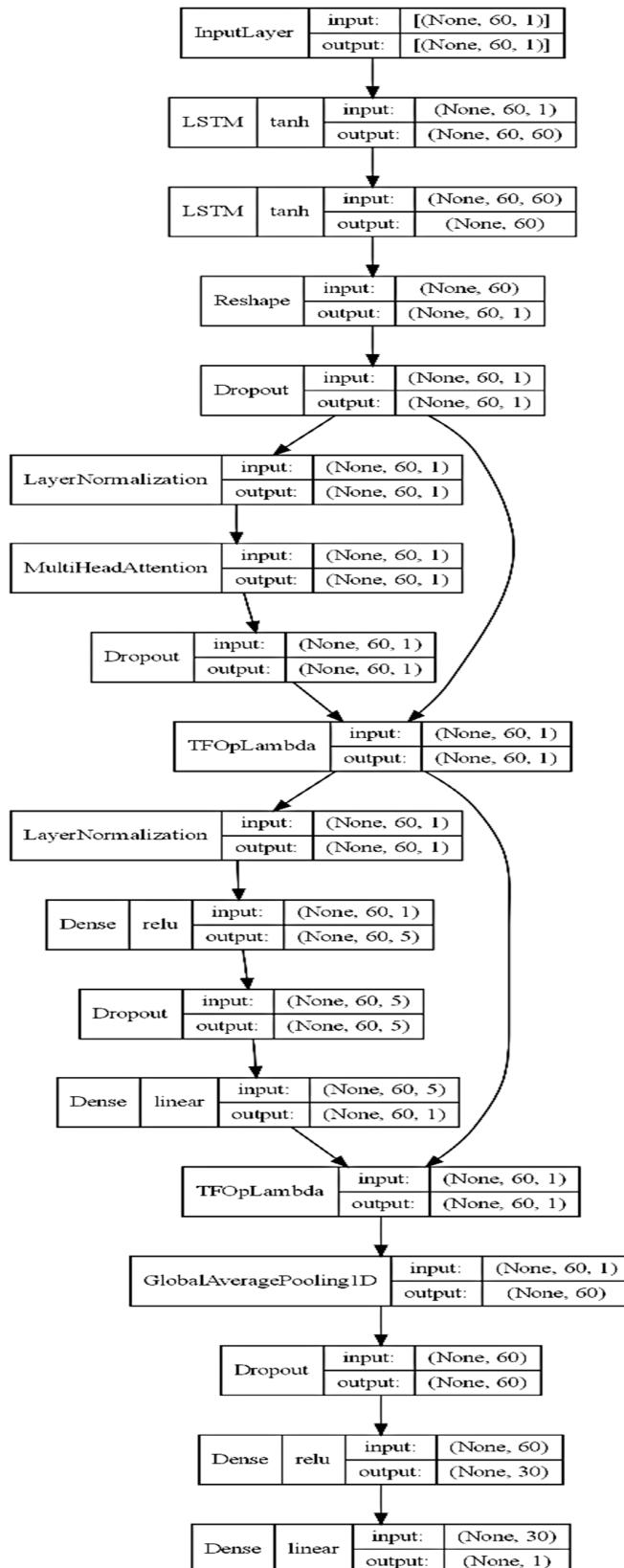


Figure 4. Block diagram of the proposed model: LSTM-mTrans-MLP.

One of the reasons for the model's forecasting performance on multiple datasets from diverse financial markets is the dropout layers and regularization techniques. The model avoids overfitting due to careful design choices, like the dropout layers, residual connections, and regularization techniques, even when trained on datasets with different characteristics and sizes.

4. Results

4.1. Dataset

In this research, the advanced LSTM-mTrans-MLP model was tested on seven different financial datasets from different stock markets and countries. The seven stock price datasets are the Bitcoin stock price, China Unicom (China United Network Communication Limited, Beijing, China), CSI 300 (China Securities Index Company, Shanghai, China, Shanghai and Shenzhen Stock Exchange, CSI 300), Shanghai Stock Market Composite Index (SCI/SSEC), CSI 100, Amazon (Seattle, WA, USA), and Alphabet Inc. (Google, Mountainview, CA, USA) stock prices. Among them, CSI 300, CSI 100, China Unicom, and the SCI are the most researched Chinese stock prices in CNY. Meanwhile, Bitcoin represents cryptocurrency, and the stock price is in USD. Amazon and Google are the most renowned American companies with diversified products and services, and the stock price is also in USD.

In our study, we conducted a comparative analysis of various forecasting models published in recent papers, ensuring that they were evaluated using the same training and test dataset sizes, with identical start and end dates. The main data sources were Yahoo Finance, stock market quotes, and Investing.com, encompassing the mentioned datasets. Each stock dataset provided details including the opening and closing prices, daily highs and lows, the preceding day's concluding price, and the trading volume, along with appropriate time series data.

Table 1 provides detailed information about the datasets used in this study, including their start and end dates, the total number of observations for each stock, the training-to-testing data ratio, and the corresponding training and testing data counts based on this ratio. For example, the Bitcoin stock price dataset spans from 6 January 2012 to 23 January 2023, with a total of 2939 observations. The training-to-testing data ratio for Bitcoin is 80:20, resulting in specific data counts such as 2351 training points and 588 testing points. Similarly, the CSI 300 dataset includes 3170 training points and 699 testing points, following the same structured outline for all other datasets.

Table 1. Dataset information and training sizes.

Stock	Related Work to Match Dataset	Start and End Time	Training–Test Dataset	Test Data Duration	Important Global Events During Test Period
Bitcoin	[34]	2012.01.06–2020.01.23	80%:20% 2292, 587	2018.06.15–2020.01.23	The Great Crypto Crash
China Unicom	[33]	2002.10.09–2021.03.17	80%:20% 3496, 888	2017.03.02–2021.03.17	COVID-19 pandemic
CSI 300	[33]	2005.01.18–2021.03.17	82%:18% 3170, 699	2018.05.02–2021.03.17	COVID-19 pandemic
SCI	[35]	2014.04.10–2023.04.20	80%:20% 1699, 440	2021.06.29–2023.04.20	Russia–Ukraine war
CSI 100	[35]	2014.04.10–2023.04.20	81%:19% 1813, 440	2021.08.11–2023.04.20	Russia–Ukraine war

Table 1. Cont.

Stock	Related Work to Match Dataset	Start and End Time	Training–Test Dataset	Test Data Duration	Important Global Events During Test Period
AMZN	[36]	2011.01.05–2019.12.31	70%:30% 1523, 678	2017.04.21–2019.12.31	COVID-19 pandemic
GOOGL	[36]	2013.01.02–2017.12.29	80%:20% 948, 251	2016.12.30–2017.12.29	COVID-19 pandemic

Table 2 illustrates the descriptive study of the datasets in this research. The data statistics comprise the mean, max, and min values; the standard deviation of the data; the data variability; and the distribution indication through skewness and kurtosis. Moreover, the *p*-values associated with the stationarity tests, KPSS test, and ADF test are mentioned in the table. Normality tests and Shapiro tests were conducted on the datasets, and the *p*-value is presented in the table.

Table 2. Summary metrics and analytical evaluations of the financial datasets.

Dataset Name	Mean Value	Min Value	Max Value	Standard Deviation	Skewness	Kurtosis	P _{KPSS}	P _{ADF}	P _{Shapiro}
Bitcoin	2604.16	4.2	19,345.5	3632.54	1.488	1.443	0.01	0.558	0
China Unicom	4.959	2.2	13.08	1.821	1.043	1.747	0.01	0.0345	6.6×10^{-42}
CSI 300	3049.74	818.03	5877.2	1068.02	−0.073	−0.113	0.01	0.283	6.15×10^{-23}
SCI	3242.81	2003.49	5166.35	437.66	0.303	2.893	0.033	0.012	6.59×10^{-29}
CSI 100	2108.96	1285.16	2983.26	305.46	0.136	−0.484	0.01	0.231	2.56×10^{-10}
Google	33.31	17.59	54.25	9.03	0.387	−0.799	0.01	0.909	3.84×10^{-19}
Amazon	36.80	8.05	101.98	29.08	0.900	−0.664	0.01	0.986	1.21×10^{-44}

4.2. Preprocessing

The original datasets are chaotic and nonlinear, which means that they cannot be used for good model performance; thus, there is a need for data preprocessing. First, missing or disordered attribute values were handled. Next, data normalization was applied to address inconsistencies in the data magnitude. In this experiment, data assessments were normalized to a range between 0 and 1, enhancing the model's efficiency and precision. The following function in Equation (3) represents the transformation process:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3)$$

where x_{\max} and x_{\min} represent the max value and min value of the test dataset, respectively.

A key advantage of this model is its minimal preprocessing requirements. In this work, preprocessing is limited to handling missing or disordered values and applying scaling, which are computationally inexpensive and efficient. This simplicity reduces the time and resources needed to prepare the data, making the model highly practical and energy-efficient.

4.3. Model Comparison and Results

In this section, we provide an overview of the performance metrics for each model and several datasets. The outcome of the comparison is an evaluation of how efficiently each model performs in several market conditions—ranging from the high volatility seen in cryptocurrency markets to the relatively stable growth in traditional equity markets.

Major key metrics are examined here: the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean average percentage error (MAPE), and R^2 score. This research mainly aims to evaluate the precision of each model and its robustness across diverse financial datasets. The Results section summarizes each table with a detailed evaluation that emphasizes the strengths and gaps in each attempt and focuses on the stable and erratic nature of these stock market behaviors. This research also aims to provide information about which model performs best in predicting trends under various levels of market volatility and complexity.

Table 3 presents a comparison of the suggested LSTM-mTrans-MLP model's performance with the latest SOTA works, ARMA-CNN-LSTM [34], and the traditional statistical and deep learning models on the closing prices of Bitcoin. The model performance and result comparison are determined by the RMSE, MAPE, and MAE. In the comparison with the other benchmark and hybrid models (ARMA-CNN-LSTM), the proposed LSTM-mTrans-MLP model outperformed them on all metrics. With the lowest RMSE = 288.3428, MAPE = 0.0268, and MAE = 186.2394, the suggested model significantly improves the prediction accuracy.

Table 3. Model comparison on Bitcoin closing price.

Model	RMSE	MAPE	MAE
Random walk [34]	323.8311	0.0257	199.1424
ARMA [34]	324.6788	0.0258	199.5287
MLP [34]	341.0648	0.028	217.3472
LSTM [34]	476.8439	0.0423	327.0795
CNN [34]	378.66	0.0315	243.013
ARMA-CNN-LSTM [34]	323.7705	0.0254	197.04
Proposed LSTM-mTrans-MLP model	288.3428	0.0268	186.2394

Table 4 presents a comparison of the suggested LSTM-mTrans-MLP model's performance with the latest SOTA works, CNN-BiLSTM-ECA [33], and the traditional statistical and deep learning models on the China Unicorn and CSI 300 datasets. The model comparison and performance evaluation were achieved using the RMSE, MSE, and MAE metrics. In the comparison with the other benchmark and hybrid models (ARMA-CNN-LSTM), the suggested LSTM-mTrans-MLP model outperformed them on all metrics. With the lowest MSE = 0.018, RMSE = 0.0133, and MAE = 0.092 for China Unicorn and MSE = 4161.203, RMSE = 64.507, and MAE = 46.453 for CSI 300, respectively, the suggested model significantly improves the prediction accuracy.

Table 4. Model comparison on China Unicorn and CSI 300 datasets.

Model	China Unicorn			CSI 300		
	MSE	RMSE	MAE	MSE	RMSE	MAE
CNN [4]	0.037	0.193	0.134	6218.092	78.855	63.981
LSTM [41]	0.036	0.189	0.128	5809.153	76.218	58.679
BiLSTM [42,43]	0.035	0.187	0.132	5091.610	71.356	52.119
CNN-LSTM [44]	0.030	0.174	0.110	4905.472	70.039	52.457
CNN-BiLSTM [33]	0.029	0.170	0.110	4643.541	68.144	51.143
BiLSTM-ECA [33]	0.039	0.198	0.142	4161.203	64.507	46.453
CNN-LSTM-ECA [33]	0.032	0.180	0.127	4568.808	67.593	51.061
CNN-BiLSTM-ECA [33]	0.028	0.167	0.103	3434.408	58.604	39.111
Proposed Model	0.018	0.133	0.092	3331.691	57.720	42.070

An assessment of the suggested LSTM-mTrans-MLP model's performance compared with the latest SOTA works, FIVMD-MFA-WOA-LSTM [35], and the traditional statistical and deep learning models on the Shanghai Composite Index and CSI 100 datasets is presented in Table 5. The RMSE, MAPE, and MAE have been applied to compare the models' efficacy. Compared to the other benchmark and hybrid models (FIVMD-MFA-WOA-LSTM), the suggested LSTM-mTrans-MLP model outperformed them on all metrics. With the lowest MAPE = 0.9674, RMSE = 41.2808, and MAE = 31.9298 for the Shanghai Composite Index and MAPE = 2.0506, RMSE = 50.9529, and MAE = 37.4231 on CSI 100, respectively, the suggested model significantly improves the prediction accuracy.

Table 5. Model comparison on Shanghai Composite Index and CSI 100 datasets.

Model	SSEC			CSI100		
	MAPE	RMSE	MAE	MAPE	RMSE	MAE
SVR [35]	8.4563	323.9135	269.5802	10.9925	469.6172	393.2344
GRU [35]	5.9736	249.6411	205.2211	7.8870	415.6537	353.2755
LSTM [35]	5.8748	246.9161	201.9285	7.0536	382.5686	313.8541
FIVMD-LSTM [35]	2.2257	91.4471	75.5311	2.772	154.6032	116.5628
GWO-LSTM [45]	4.2884	174.7373	145.6766	6.0828	326.4571	265.3836
BO-LightGBM [46]	4.547	187.6829	154.3968	6.1401	334.3083	268.6137
CEEMDAN-LSTM [47]	3.6285	139.3788	116.0111	5.2485	296.3123	233.4450
SSA-BIGRU [48]	6.4242	281.892	217.7316	13.6878	545.2107	501.1394
VMD-SE-GRU [49]	2.5404	110.3014	86.5217	3.316	192.8174	148.9184
FIVMD-MFA-WOA-LSTM [35]	1.1244	50.5778	37.1922	1.9001	93.5436	78.2486
Proposed Model	0.9674	41.2808	31.9298	2.0506	50.9529	37.4231

An evaluation comparing the implementation of the LSTM-mTrans-MLP model and the latest SOTA works, LSTM, linear regression, exponential smoothing, and the traditional statistical and deep learning models on the Amazon dataset is presented in Table 6. The RMSE, MSE, and MAE have been employed as model comparison and performance parameters. Compared with the other benchmark and hybrid models using LSTM, the suggested LSTM-mTrans-MLP model outperformed them on all metrics. With the lowest MAE = 1.122, RMSE = 1.541, and MSE = 2.375 for Amazon, the suggested model significantly improves the prediction accuracy.

Table 6. Model comparison on Amazon dataset.

Model	AMAZON			
	R ² Score	MAE	MSE	RMSE
Linear regression [50]	0.7163	72.47	7231.59	85.04
MA (3mo) [51]	0.6938	21.08	609.22	24.68
Exponential Smoothing [52]	0.6938	16.62	363.83	19.074
LSTM [36]	0.9961	14.971	418.97	20.468
CNN-LSTM	0.8375	6.045	47.366	6.882
CNN-BiLSTM	0.9023	4.518	28.478	5.336
CNN-LSTM-ECA [33]	0.9211	3.945	22.981	4.794
CNN-BiLSTM-ECA [33]	0.9710	2.378	8.447	2.906
Proposed LSTM-mTrans-MLP Model	0.9918	1.122	2.375	1.541

CNN-BiLSTM-ECA model has been replicated from work [33].

Table 7 presents a comparison of the suggested LSTM-mTrans-MLP model's performance with that of the latest SOTA works using LSTM [36] and the traditional statistical and deep learning models on the Google dataset. The RMSE, MSE, and MAE have been

employed as model comparison and performance parameters. Compared with the other benchmark and hybrid models using LSTM, the suggested LSTM-mTrans-MLP model outperformed them on all metrics. With the lowest MAE = 0.642, RMSE = 0.815, and MSE = 0.664 for Google, the suggested model significantly improves the prediction accuracy.

Table 7. Model comparison on Google stock dataset.

Model	Google			
	R ² Score	MAE	MSE	RMSE
LSTM [33]	0.9421	13.139	316.53	17.791
CNN-LSTM	0.8757	1.324	2.765	1.663
CNN-BiLSTM	0.8779	1.330	2.715	1.648
CNN-LSTM-ECA [33]	0.9327	0.934	1.497	1.224
CNN-BiLSTM-ECA [33]	0.9511	0.774	1.087	1.043
Proposed LSTM-mTrans-MLP Model	0.9533	0.642	0.664	0.815

CNN-BiLSTM-ECA model has been replicated from work [33].

As mentioned in Section 4.1, the datasets used in this study were selected to facilitate benchmark comparisons. The sampling periods were aligned with those in previous studies to ensure the comparability of the results. The sampling periods were also constrained by the data availability from different sources. To prove the robustness, consistency, and generalizability of the proposed model, experiments have been conducted on a normalized timeframe as well. Table 8 shows the evaluation metrics across the normalized timeframe for the comparison (Tables 4–7). It can be seen that the model performs consistently during the larger normalized test period of January 2020 to June 2024, containing significant external events like the COVID-19 pandemic (2020–2022) and the Russia–Ukraine war (2022–2023).

Table 8. Comparison of the model's performance across the normalized timeframe and the timeframe for the comparative study.

Stock Name	Related Work	Normalized Time (Start and End Time)	Training and Test Data Size	Test Dataset Duration	Evaluation Metrics On	
					Normalized Test Dataset	Original Dataset (Tables 3–7)
China Unicom	[33]	2011-01-05– 2024-07-01	2133, 1080 (67%:33%)	2020-01-13– 2024-07-01	RMSE: 0.133 MSE: 0.018 MAPE: 2.4556% MAE: 0.1055 R ² : 0.9595	RMSE: 0.133 MSE: 0.018 MAPE: 1.5688% MAE: 0.092 R ² : 0.9784
CSI 300	[33]	2011-01-05– 2024-07-01	2136, 1081 (67%:33%)	2020-01-10– 2024-07-01	RMSE: 61.6587 MSE: 3801.793 MAPE: 1.0397% MAE: 45.7935 R ² : 0.9893	RMSE: 57.720 MSE: 3331.691 MAPE: 1.049% MAE: 42.070 R ² : 0.9919
SCI	[35]	2011-01-05– 2024-07-01	2197, 1082 (67%:33%)	2020-01-14– 2024-07-01	RMSE: 38.2837 MSE: 1465.6407 MAPE: 0.8853% MAE: 28.6112 R ² : 0.9726	RMSE: 41.2808 MSE: 1704.101 MAPE: 0.967% MAE: 31.9298 R ² : 0.9552

Table 8. Cont.

Stock Name	Related Work	Normalized Time (Start and End Time)	Training and Test Data Size	Test Dataset Duration	Evaluation Metrics On	
					Normalized Test Dataset	Original Dataset (Tables 3–7)
CSI 100	[35]	2011-08-29–2024-07-01	2131, 1079 (67%:33%)	2020-05-05–2024-07-01	RMSE: 47.5384 MSE: 2259.899 MAPE: 1.866% MAE: 33.4194 R^2 : 0.9875	RMSE: 50.9529 MSE: 2596.198 MAPE: 2.051% MAE: 37.4231 R^2 : 0.96199
AMZN	[36]	2011-01-05–2024-07-01	2213, 1119 (67%:33%)	2020-01-17–2024-07-01	RMSE: 7.4205 MSE: 55.0635 MAPE: 4.3972% MAE: 6.4952 R^2 : 0.9324	RMSE: 1.541 MSE: 2.375 MAPE: 1.417% MAE: 1.122 R^2 : 0.9919
GOOGL	[36]	2011-01-05–2024-06-28	2213, 1119 (67%:33%)	2020-01-16–2024-06-28	RMSE: 2.7205 MSE: 7.4012 MAPE: 1.8143% MAE: 2.0260 R^2 : 0.9907	RMSE: 0.815 MSE: 0.664 MAPE: 1.318% MAE: 0.642 R^2 : 0.9533

Figures 5–13 show the prediction results. The forecasted metrics are nearly identical to the definite metrics. In the interim, the x-axis denotes the time, and the y-axis illustrates the stock price.

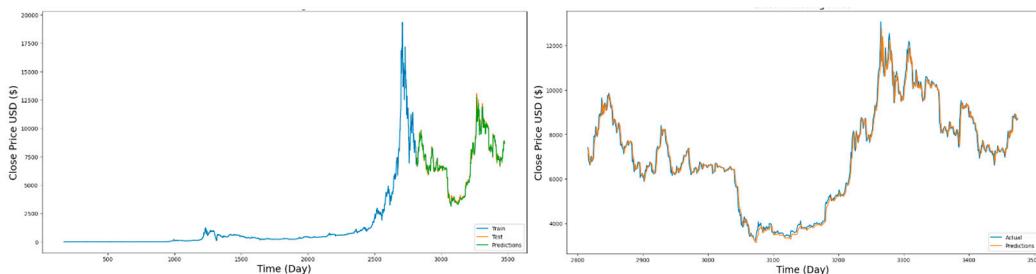


Figure 5. Training, test, and predicted price (left) and test vs. predicted price (right) for Bitcoin stock.

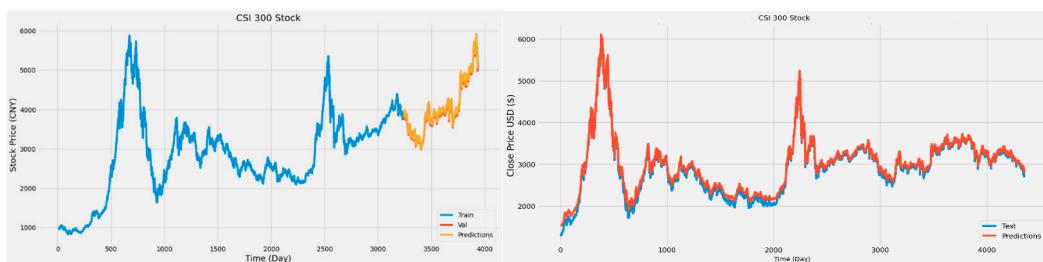


Figure 6. Training, test, and predicted price (left) and test vs. predicted price (right) for CSI 300.

The financial data analysis in Figures 5–13 reveals distinct correlations and performance differences across the cryptocurrency (Bitcoin), US stocks (Amazon, Google), and Chinese markets (China Unicom, Shanghai Stock Exchange, CSI 300, CSI 100). Despite Bitcoin's reputation for being unpredictable and risky, the model used in this analysis shows reliable performance in predicting its price trends. Despite its high volatility and irregular cycles, Bitcoin shows patterns that the model captures effectively, indicating that, with appropriate adjustments, cryptocurrency prices can be anticipated with a reasonable

degree of accuracy. The model also performs well for these stocks, accurately tracking long-term trends with minor deviations, particularly during stable market periods. In the case of the Chinese markets, particularly China Unicom and the Shanghai Stock Exchange, the investment shows significant peaks and troughs due to both global market events and domestic policy changes. While the model generally performs well, the Shanghai Stock Exchange's erratic behavior poses challenges in capturing rapid shifts influenced by local economic conditions.

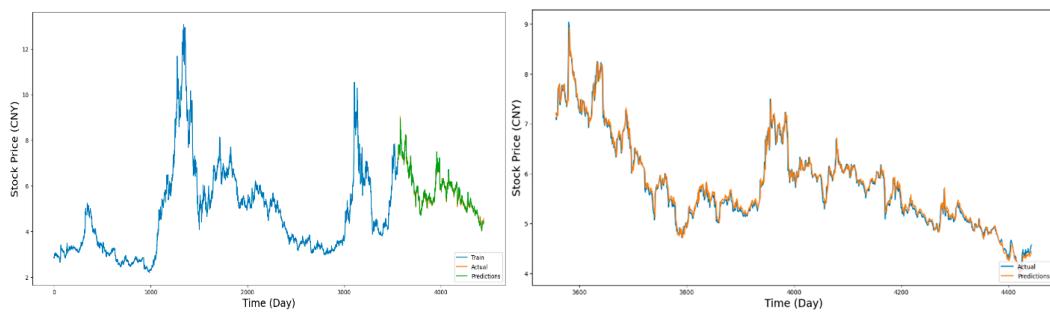


Figure 7. Training, test, and predicted price (left) and test vs. predicted price (right) for China Unicom.

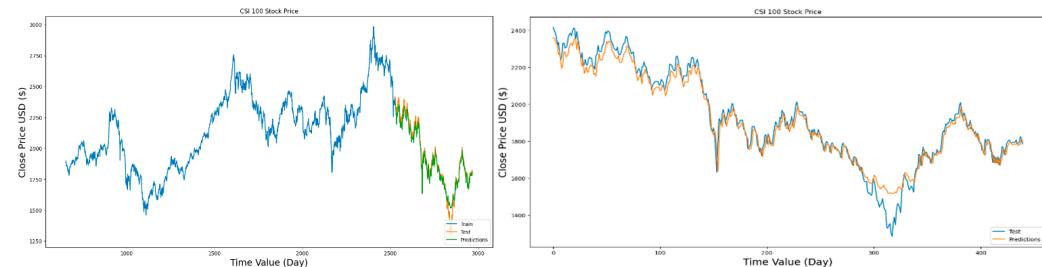


Figure 8. Training, test, and predicted price (left) and test vs. predicted price (right) for CSI 100.

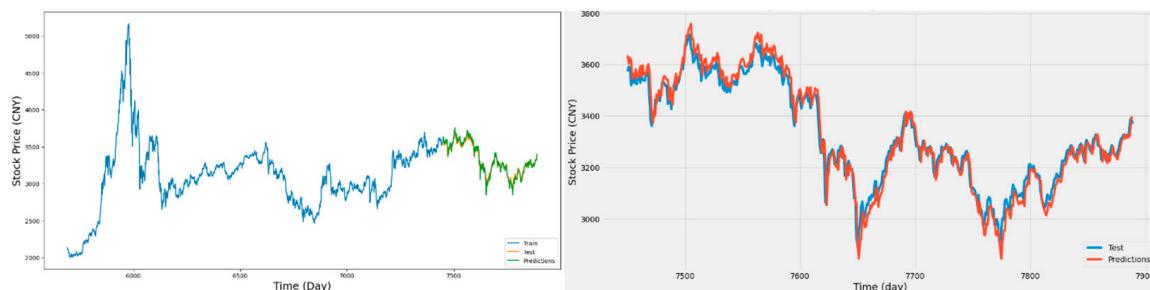


Figure 9. Training, test, and predicted price (left) and test vs. predicted price (right) for Shanghai Composite Stock Exchange.

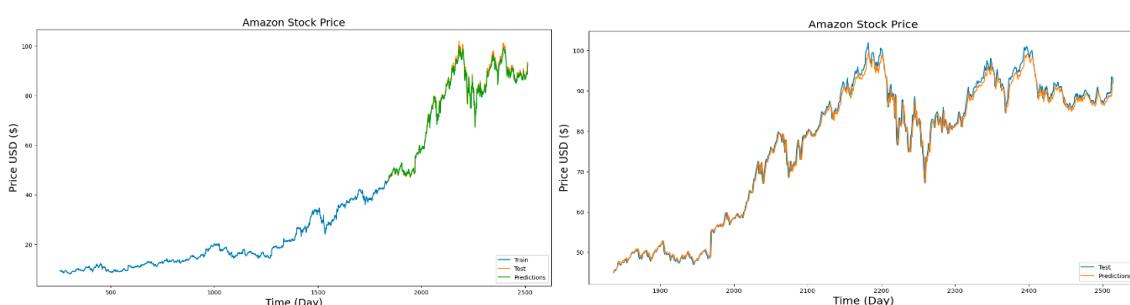


Figure 10. Training, test, and predicted price (left) and test vs. predicted price (right) for Amazon.

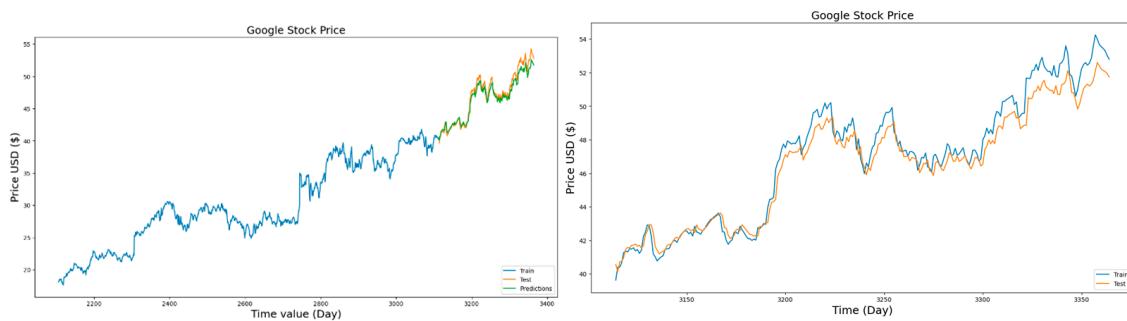


Figure 11. Training, test, and predicted price (left) and test vs. predicted price (right) for Google.

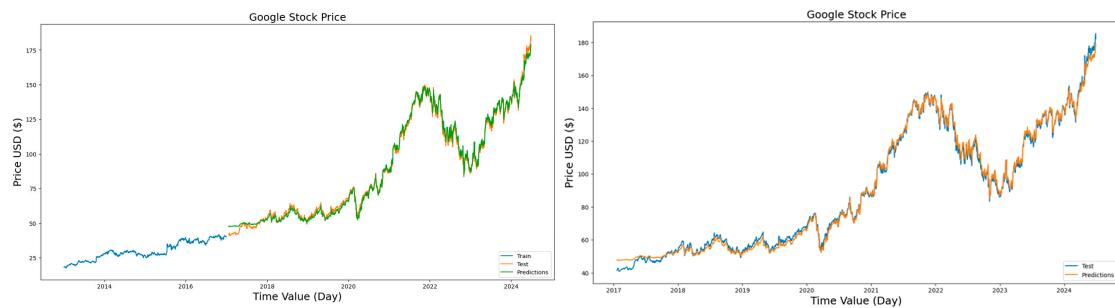


Figure 12. Training, test, and predicted price (left) and test vs. predicted price (right) of Google stock for extended test duration (2017–2024).

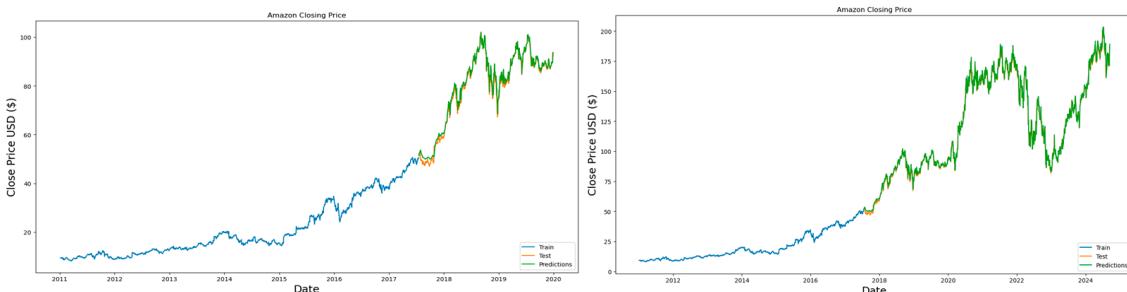


Figure 13. Training, test, and predicted price (left) and test vs. predicted price (right) of Amazon stock for extended test duration (2017–2024).

Despite relying only on historical data, the proposed model demonstrates robust performance during periods of significant external events. For example, 2017–2018 is recognized as the ‘Great Crypto Crash’ due to the boom and subsequent crash of Bitcoin. Meanwhile, 2020–2022 exerted a huge impact on the stock markets of software and communication companies, e-commerce companies, and banks due to the COVID-19 pandemic. During periods of high volatility, such as the cryptocurrency crash of 2017–2018, the COVID-19 pandemic (2020–2022), and the Russia–Ukraine war (2022–2023), the model maintains competitive performance, capturing critical price fluctuations and trends in Bitcoin, Google stocks, China Unicom (telecom operator company), and CSI 300 with reliability, as shown in Figures 6 and 7. This resilience highlights the model’s ability to extract meaningful patterns from historical data, even in challenging market conditions.

For the Amazon stock price, the model was initially trained on data from January 2011 to April 2017 and tested using the test dataset from April 2017 to December 2019 to match the benchmark work in [36]. To check the effectiveness of the model, the trained model was then further evaluated on an enhanced testing period from April 2017 to 2024. Figure 13 shows that it yielded accurate predictions even during the COVID-19 pandemic

and captured the subsequent price drops caused by the Russia–Ukraine war despite the very small number of training data (2011–2017).

Similarly, for Google, the model was trained with data from January 2013 to December 2016. To enable the comparison with the benchmark model, the trained model was tested from December 2016 to December 2017. Additionally, the same trained model was tested for a longer period of time covering COVID-19. The new test data from December 2016 to 2024 show that the model’s predictions are very close to the test data, demonstrating the effectiveness of its predictions. The model can effectively forecast the stock price rise during the COVID-19 period (2020–22) and the subsequent price drop during the Russia–Ukraine war (2022–2023).

The proposed model demonstrates superior prediction performance compared to other models observed in various research studies.

One important aspect of the proposed model is that the architecture is optimized for efficiency, containing a relatively small number of parameters compared to larger and more complex models. This design ensures that the time, memory, and energy required for training and inference are significantly lower than those for more complex models, such as large-scale language models, making it practical for real-time applications in resource-constrained environments, such as mobile financial advisory tools or on-device predictions. Table 9 illustrates the average time required to train the model and to test its performance for different datasets.

Table 9. Time requirement for training and testing of the model for different datasets.

Serial No.	Dataset Name	Training and Test Dataset Size	Training Time per Epoch (sec)	No. of Epochs for Training	Total Training Time	Prediction Time on Test Dataset (s)
1	Bitcoin	2292, 587	29.34	12	5 min 52.14 s	1.027
2	China Unicom	3496, 888	19.38	22	7 min, 6.28 s	1.022
3	CSI 300	3170, 699	40.51	12	8 min 6.15 s	0.022
4	SCI	1699, 440	23.19	17	6 min 34.23 s	0.034
5	CSI 100	1813, 440	22.01	14	5 min 8.17 s	0.021
6	AMZN	1523, 678	19.01	25	7 min 55.31 s	1.026
7	GOOGL	948, 251	13.86	22	5 min 5.32 s	0.021

Note: Hardware environment: Intel Core i5 11400 processor, 16 GB DDR4 RAM machine. Software environment: Operating System: Windows 11 pro; Software platform: Anaconda 2021.05 (2.0.3), Major Libraries: Jupyter 1.0.0, Python 3.8.8, TensorFlow: 2.13.0, Keras: 2.13.1, Pandas: 1.3.2.

5. Conclusions

Within this research, we present a resilient and groundbreaking hybrid model, LSTM-mTrans-MLP, aimed at improving the precision of financial time series predictions. The LSTM-mTrans-MLP model is built on a hybrid architecture, blending the strengths of various deep learning paradigms. This design makes it both robust and state-of-the-art (SOTA). It has demonstrated superior predictive performance in financial time series forecasting compared to other SOTA architectures mentioned earlier.

The LSTM layers identify and retain both long-term and short-term dependencies and fluctuations in stock prices, the modified Transformer improves the model’s abilities through its attenuation mechanism, and, finally, the MLP is capable of learning complex nonlinear relationships, further improving the model’s generalization capabilities.

This combination allows the projected model to outperform traditional and SOTA methods in terms of both stability and prediction accuracy across seven different financial datasets (Bitcoin, China Unicom, CSI 300, Shanghai Composite Index, CSI 100, Amazon, and Google stock datasets). The proposed model, having a single model architecture with a unique parameter number and model size, is compared with ARMA, CNN, LSTM, and

ARMA-CNN-LSTM [34] on the Bitcoin dataset; LSTM, CNN, CNN-LSTM, Bi-LSTM, CNN-Bi-LSTM, CNN-LSTM-ECA, CNN-Bi-LSTM-ECA, and Bi-LSTM-ECA [33] on the China Unicom and CSI 300 datasets; the GRU, LSTM, FIV-LSTM, GWO-LSTM, BO-LightGBM, CEEMDAN-LSTM, VMD-SE-GRU, SSA-BIGRU, and FIVMD-MFA-WOA-LSTM [35] on the SSEC and CSI 100 datasets; regression random forest, DNN-LSTM, linear regression, MA, exponential smoothing, and LSTM [36] on the Amazon stock price dataset; and finally an RNN, ANN, and LSTM [36] on the Google dataset. The comparison shows that the forecasting capabilities of the projected model are superior in terms of different evaluation metrics: the MSE, RMSE, MAE, MAPE, and R2 score. In contrast, our model realistically predicts stock prices and offers valuable insights for investors to optimize their returns.

Proposed Future Work:

- (a) The model's effectiveness across diverse volatility levels demonstrates its generalizability and resilience, making it a versatile tool for financial forecasting. Future research may explore the integration of diffusion models or the use of external textual data, such as financial news, to further refine the model's forecasting abilities. The diffusion model shows prospects in time series prediction and forecasting applications. Diffusion models can be explored as an extension of or ensemble with the hybrid model architecture to enhance both the efficacy and efficiency of its forecasting.
- (b) The impact of textual information, along with historical stock data, can be investigated for the further enhancement of the model's performance. Textual information such as financial news, company earnings reports, social media statuses, and stock bar comments may significantly impact stock price movement.

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

Funding: This publication was funded by the UA Little Rock Open Access Article Publishing Support Fund.

Data Availability Statement: Data are collected from <https://www.finance.yahoo.com/>, <https://www.investing.com/>, accessed on 3 November 2024.

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

Abbreviations

The following abbreviations are used in this manuscript.

LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DNN	Deep Neural Network
SSN	Sparkling Neural Network
RBFNN	Radial Basis Function Neural Network
SOTA	State of the Art
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
SARIMA	Seasonal Autoregressive Integrated Moving Average
MIDAS	Multiple Instance Detection and Segmentation

CEEMDAN	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
CNN-BiLSTM-ECA	Convolutional Neural Network- Bidirectional LSTM and Efficient Channel Attention
SSA	Singular Spectrum Analysis
SVM	Support Vector Machine
GWO	Grey Wolf Optimizer
ECA	Exponential Component Attention
SGD	Stochastic Gradient Descent
WOA	Whale Optimization Algorithm
KPSS	Kwiatkowski–Phillips–Schmidt–Shin (Test)
ADF	Augmented Dickey–Fuller (Test)
EEMD	Ensemble Empirical Mode Decomposition
GRU	Gated Recurrent Unit
VMD	Variational Mode Decomposition
FIVMD	Interval Variational Modal Decomposition
RMSE	Root Mean Square Error
MSE	Mean Square Error
MAPE	Mean Average Percentage Error
MAE	Mean Average Error
CSI	China Security Index
SCI	Shanghai Composite Index

References

- Asadi, S.; Hadavandi, E.; Mehmanpazir, F.; Nakhostin, M.M. Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. *Knowl. Based Syst.* **2012**, *35*, 245–258. [[CrossRef](#)]
- Akhter, S.; Misir, M.A. Capital Markets Efficiency: Evidence from the Emerging Capital Market with Particular Reference to Dhaka Stock Exchange. *South Asian J. Manag. New Delhi* **2005**, *12*, 35–51.
- Kim, H.Y.; Won, C.H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst. Appl.* **2018**, *103*, 25–37. [[CrossRef](#)]
- Chen, W.; Jiang, M.; Zhang, W.-G.; Chen, Z. A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Inf. Sci.* **2021**, *556*, 67–94. [[CrossRef](#)]
- Chen, Q.; Zhang, W.; Lou, Y. Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism, Multi-Layer Perceptron, and Bidirectional Long-Short Term Memory Neural Network. *IEEE Access* **2020**, *8*, 117365–117376. [[CrossRef](#)]
- Naeini, M.P.; Taremiān, H.; Hashemi, H.B. Stock market value prediction using neural networks. In Proceedings of the 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), Krackow, Poland, 8–10 October 2010; IEEE: New York City, NY, USA, 2010; pp. 132–136. [[CrossRef](#)]
- Qian, B.; Rasheed, K. Stock market prediction with multiple classifiers. *Appl. Intell.* **2007**, *26*, 25–33. [[CrossRef](#)]
- Guo, T.; Xu, Z.; Yao, X.; Chen, H.; Aberer, K.; Funaya, K. Robust Online Time Series Prediction with Recurrent Neural Networks. In Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, 17–19 October 2016; IEEE: New York City, NY, USA, 2016; pp. 816–825. [[CrossRef](#)]
- Chen, P.-A.; Chang, L.-C.; Chang, F.-J. Reinforced recurrent neural networks for multi-step-ahead flood forecasts. *J. Hydrol.* **2013**, *497*, 71–79. [[CrossRef](#)]
- Ibrahim, A.; Kashef, R.; Corrigan, L. Predicting market movement direction for bitcoin: A comparison of time series modeling methods. *Comput. Electr. Eng.* **2021**, *89*, 106905. [[CrossRef](#)]
- Chevallier, J. Nonparametric modeling of carbon prices. *Energy Econ.* **2011**, *33*, 1267–1282. [[CrossRef](#)]
- Zhao, X.; Han, M.; Ding, L.; Kang, W. Usefulness of economic and energy data at different frequencies for carbon price forecasting in the EU ETS. *Appl. Energy* **2018**, *216*, 132–141. [[CrossRef](#)]
- Fan, X.; Li, S.; Tian, L. Chaotic characteristic identification for carbon price and an multi-layer perceptron network prediction model. *Expert Syst. Appl.* **2015**, *42*, 3945–3952. [[CrossRef](#)]
- Bhadra, D.; Tarique, T.A.; Ahmed, S.U.; Shahjahan; Murase, K. An encoding technique for design and optimization of combinational logic circuit. In Proceedings of the 2010 13th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 23–25 December 2010; IEEE: New York City, NY, USA, 2010; pp. 232–236. [[CrossRef](#)]

15. Bhadra, D.; Hossain, M.; Alam, F. Speaker Independent Bangla Isolated Speech Recognition Using Deep Neural Network. In Proceedings of the International Conference on Technology, Business, and Justice Towards Smart Bangladesh | ICTBJ-2023, Mymensingh, Bangladesh, 5–6 June 2023; pp. 41–42.
16. Fenghua, W.; Jihong, X.; Zhifang, H.; Xu, G. Stock Price Prediction Based on SSA and SVM. *Procedia Comput. Sci.* **2014**, *31*, 625–631. [[CrossRef](#)]
17. Shen, G.; Tan, Q.; Zhang, H.; Zeng, P.; Xu, J. Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions. *Procedia Comput. Sci.* **2018**, *131*, 895–903. [[CrossRef](#)]
18. Atsalakis, G.S.; Atsalaki, I.G.; Pasiouras, F.; Zopounidis, C. Bitcoin price forecasting with neuro-fuzzy techniques. *Eur. J. Oper. Res.* **2019**, *276*, 770–780. [[CrossRef](#)]
19. Nagula, P.K.; Alexakis, C. A new hybrid machine learning model for predicting the bitcoin (BTC-USD) price. *J. Behav. Exp. Financ.* **2022**, *36*, 100741. [[CrossRef](#)]
20. Zhu, B.; Wei, Y. Carbon price forecasting with a novel hybrid ARIMA and least squares support vector machines methodology. *Omega* **2013**, *41*, 517–524. [[CrossRef](#)]
21. Sun, G.; Chen, T.; Wei, Z.; Sun, Y.; Zang, H.; Chen, S. A Carbon Price Forecasting Model Based on Variational Mode Decomposition and Spiking Neural Networks. *Energies* **2016**, *9*, 54. [[CrossRef](#)]
22. Atsalakis, G.S. Using computational intelligence to forecast carbon prices. *Appl. Soft Comput.* **2016**, *43*, 107–116. [[CrossRef](#)]
23. Ni, L.; Li, Y.; Wang, X.; Zhang, J.; Yu, J.; Qi, C. Forecasting of Forex Time Series Data Based on Deep Learning. *Procedia Comput. Sci.* **2019**, *147*, 647–652. [[CrossRef](#)]
24. Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. *Knowl. Based Syst.* **2018**, *164*, 163–173. [[CrossRef](#)]
25. Gonçalves, R.; Ribeiro, V.M.; Pereira, F.L.; Rocha, A.P. Deep learning in exchange markets. *Inf. Econ. Policy* **2019**, *47*, 38–51. [[CrossRef](#)]
26. Peng, L.; Liu, S.; Liu, R.; Wang, L. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy* **2018**, *162*, 1301–1314. [[CrossRef](#)]
27. Cen, Z.; Wang, J. Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. *Energy* **2018**, *169*, 160–171. [[CrossRef](#)]
28. Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [[CrossRef](#)]
29. Pai, P.-F.; Lin, C.-S. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* **2004**, *33*, 497–505. [[CrossRef](#)]
30. Shafie-Khah, M.; Moghaddam, M.P.; Sheikh-El-Eslami, M. Price forecasting of day-ahead electricity markets using a hybrid forecast method. *Energy Convers. Manag.* **2011**, *52*, 2165–2169. [[CrossRef](#)]
31. Jeong, K.; Koo, C.; Hong, T. An estimation model for determining the annual energy cost budget in educational facilities using SARIMA (seasonal autoregressive integrated moving average) and ANN (artificial neural network). *Energy* **2014**, *71*, 71–79. [[CrossRef](#)]
32. Ranaldi, L.; Gerardi, M.; Fallucchi, F. Fallucchi, CryptoNet: Using Auto-Regressive Multi-Layer Artificial Neural Networks to Predict Financial Time Series. *Information* **2022**, *13*, 524. [[CrossRef](#)]
33. Chen, Y.; Fang, R.; Liang, T.; Sha, Z.; Li, S.; Yi, Y.; Zhou, W.; Song, H. Stock Price Forecast Based on CNN-BiLSTM-ECA Model. *Sci. Program.* **2021**, *2021*, 2446543. [[CrossRef](#)]
34. He, K.; Yang, Q.; Ji, L.; Pan, J.; Zou, Y. Financial Time Series Forecasting with the Deep Learning Ensemble Model. *Mathematics* **2023**, *11*, 1054. [[CrossRef](#)]
35. Wang, J.; Liu, J.; Jiang, W. An enhanced interval-valued decomposition integration model for stock price prediction based on comprehensive feature extraction and optimized deep learning. *Expert Syst. Appl.* **2023**, *243*, 122891. [[CrossRef](#)]
36. Omoware, J.M.; Abiodun, O.J.; Wreford, A.I. Predicting Stock Series of Amazon and Google Using Long Short-Term Memory (LSTM). *Asian Res. J. Curr. Sci.* **2023**, *5*, 205–217.
37. Patterson, J.; Gibson, A. *Deep Learning: A Practitioner’s Approach*, 1st ed.; O’Reilly: Sebastopol, CA, USA, 2017.
38. Cao, J.; Li, Z.; Li, J. Financial time series forecasting model based on CEEMDAN and LSTM. *Stat. Mech. Its Appl.* **2019**, *519*, 127–139. [[CrossRef](#)]
39. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. [[CrossRef](#)]
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York City, NY, USA, 2016; pp. 770–778. [[CrossRef](#)]
41. Connor, J.; Martin, R.; Atlas, L. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254. [[CrossRef](#)]

42. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [[CrossRef](#)]
43. Xu, Y.; Chhim, L.; Zheng, B.; Nojima, Y. Stacked Deep Learning Structure with Bidirectional Long-Short Term Memory for Stock Market Prediction. In *Neural Computing for Advanced Applications*; Zhang, H., Zhang, Z., Wu, Z., Hao, T., Eds.; Communications in Computer and Information Science; Springer: Singapore, 2020; Volume 1265, pp. 447–460. [[CrossRef](#)]
44. Nelson, D.M.Q.; Pereira, A.C.M.; de Oliveira, R.A. Stock market’s price movement prediction with LSTM neural networks. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: New York City, NY, USA, 2017; pp. 1419–1426. [[CrossRef](#)]
45. Mahmoodzadeh, A.; Nejati, H.R.; Mohammadi, M.; Ibrahim, H.H.; Rashidi, S.; Rashid, T.A. Forecasting tunnel boring machine penetration rate using LSTM deep neural network optimized by grey wolf optimization algorithm. *Expert Syst. Appl.* **2022**, *209*, 118303. [[CrossRef](#)]
46. Shen, B.; Yang, S.; Gao, X.; Li, S.; Ren, S.; Chen, H. A Novel Co2-Eor Potential Evaluation Method Based on Bo-Lightgbm Algorithms Using Hybrid Feature Mining. *SSRN Electron. J.* **2022**, *222*, 211427. [[CrossRef](#)]
47. Lin, Y.; Lin, Z.; Liao, Y.; Li, Y.; Xu, J.; Yan, Y. Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM. *Expert Syst. Appl.* **2022**, *206*, 117736. [[CrossRef](#)]
48. Li, X.; Ma, X.; Xiao, F.; Xiao, C.; Wang, F.; Zhang, S. Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA). *J. Pet. Sci. Eng.* **2022**, *208*, 109309. [[CrossRef](#)]
49. Zhang, S.; Luo, J.; Wang, S.; Liu, F. Oil price forecasting: A hybrid GRU neural network based on decomposition–reconstruction methods. *Expert Syst. Appl.* **2023**, *218*, 119617. [[CrossRef](#)]
50. Umer, M.; Awais, M.; Muzammul, M. Stock Market Prediction Using Machine Learning(ML) Algorithms. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.* **2019**, *8*, 97–116. [[CrossRef](#)]
51. Ullah, K.; Qasim, M. Google Stock Prices Prediction Using Deep Learning. In Proceedings of the 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 9 November 2020; IEEE: New York City, NY, USA, 2020; pp. 108–113. [[CrossRef](#)]
52. Xu, Y.; Cohen, S.B. Stock Movement Prediction from Tweets and Historical Prices. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 1970–1979. [[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.