



STL-ELM: A computationally efficient hybrid approach for predicting high volatility stock market

Temitope Olubanjo Kehinde^{a,*}, Oluyinka J. Adedokun^b, Morenikeji Kabirat Kareem^c, Joseph Akpan^d, Oludolapo A. Olanrewaju^e

^a Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

^b Department of Industrial & Systems Engineering and Engineering Management, University of Alabama in Huntsville, USA

^c Department of Computer Science, Federal University of Agriculture, Abeokuta, Nigeria

^d Department of Industrial Engineering, Durban University of Technology, South Africa

^e Institute of Systems Science, Durban University of Technology, South Africa

ARTICLE INFO

JEL Classification:

C45
C53
G11
G12
G17
G32

Keywords:

Machine learning
Deep learning
Stock price forecasting
Time series
Extreme learning machine
Neural network

ABSTRACT

Accurate forecasting of high-volatility stock markets is critical for investors and policymakers, yet existing models struggle with computational inefficiency and noise sensitivity. This study introduces STL-ELM, a novel hybrid model combining Seasonal-Trend decomposition using LOESS (STL) and Extreme Learning Machine (ELM), to deliver unparalleled accuracy and speed. By decomposing stock data into trend, seasonal, and residual components, STL-ELM isolates multiscale features, while ELM's lightweight architecture ensures rapid training and robust generalization, outperforming advanced techniques such as LSTM, GRU, and transformer variants in both prediction and trading simulations. With faster runtimes and minimal memory usage, STL-ELM is tailored for real-time trading applications and high-frequency financial forecasting, offering institutional investors, traders, and financial analysts a competitive edge in volatile markets. The hybrid nature of STL-ELM, which combines STL's multiscale decomposition with ELM's rapid learning, enhances its adaptability to various financial domains, including stocks, commodities, foreign exchange, and cryptocurrencies, by efficiently capturing domain-specific volatility patterns. This work not only sets a new standard for predictive accuracy in stock market modelling but also presents an invaluable tool for those navigating the complexities of modern financial markets.

1. Introduction

The stock market has long been a cornerstone of modern finance and economics, serving as a barometer for economic health (Olorunnimbe & Viktor, 2023; Shah et al., 2022; Zhao et al., 2025), a platform for investment, and a critical mechanism for capital allocation. Accurate prediction of stock market behavior is not only crucial for investors seeking to maximize returns but also for policymakers and economic analysts aiming to stabilize markets and promote economic growth (Oyewola et al., 2025). However, the inherent complexity of stock markets, amplified by factors such as high volatility, noise, and uncertainty, presents significant challenges for accurate forecasting (Abolmakarem et al., 2024). In this context, the development of robust and innovative models for stock market prediction becomes a pressing need. Stock market prediction holds immense importance in finance and

economics for several reasons. First, it enables investors to make informed decisions, optimize portfolio performance, and minimize risks. Second, accurate forecasting supports financial institutions and regulators in anticipating market disruptions, thereby enhancing market stability. Third, from a macroeconomic perspective, stock market trends often reflect broader economic, political, and social conditions, making them vital indicators for policy formulation (T.O. Kehinde et al., 2025; Wang, 2025; Wei et al., 2025).

Traditional methods of stock market prediction relied heavily on statistical and econometric models. Techniques such as Exponential Smoothing (ES), Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) have been widely used to model time series data and volatility (Kehinde, Chan et al., 2023; Zhu et al., 2024). Among the available statistical or econometric methods, SARIMA

* Corresponding author.

E-mail address: temitope.kehinde@connect.polyu.hk (T.O. Kehinde).

<https://doi.org/10.1016/j.iswa.2025.200564>

Received 26 June 2025; Received in revised form 23 July 2025; Accepted 27 July 2025

Available online 4 August 2025

2667-3053/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

appears to be the most prominent technique for predicting seasonal time series (Xiong et al., 2018); however, the prediction performance may be low due to the linear assumption of the SARIMA model. While these methods offer valuable insights, they are often limited in their ability to capture the non-linear and dynamic nature of stock markets, particularly during periods of high volatility. High-volatility markets pose unique challenges for predictive modeling. These markets are characterized by rapid price fluctuations driven by factors such as geopolitical events, economic announcements, and market sentiment (Pham et al., 2025). This unpredictability introduces noise into the data, obscuring underlying trends and patterns. Conventional statistical models, despite their mathematical rigor, often struggle to adapt to these rapidly changing conditions, resulting in suboptimal performance.

Machine learning (ML) models have emerged as a powerful alternative to conventional methods, leveraging data-driven approaches to uncover complex patterns in financial data. Algorithms such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forests (RF), and Gradient Boosting Machines (GBM) have demonstrated considerable success in improving prediction accuracy. However, these models also have limitations, particularly in their sensitivity to feature selection and the risk of overfitting. Deep learning (DL) models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNN) and their variants, Long Short-Term Memory (LSTM) networks (Kehinde, Chung et al., 2023), and Gated Recurrent Units (GRU), have further advanced the field of stock market prediction (Jiang, 2021). These models excel in capturing temporal dependencies and learning hierarchical representations from large datasets. Nevertheless, their reliance on extensive computational resources and vulnerability to overfitting in the presence of noise limit their applicability in high-volatility scenarios.

Stock market data exhibits multiscale properties, characterized by features such as trend, seasonality, and residual components. These properties become evident when a decomposition plot is performed on stock data, revealing its noisy, chaotic, non-stationary, and non-linear nature. Most existing ML and DL algorithms are not equipped to handle such scenarios in their default form, limiting their performance. Conventional statistical methods often fail to capture non-linear dependencies, while ML models are prone to overfitting and require extensive feature engineering. DL, although powerful, is computationally intensive and may struggle with noisy data and high computational requirements, as these models are not naturally built on understanding the multiscale properties of the stock market. To address this challenge, it is essential to develop robust hybrid models capable of decomposing the time series data first to expose its multiscale properties and then modeling these properties to facilitate easier processing by ML/DL algorithms.

In recent years, hybrid ML models have emerged as a promising approach to address the limitations of individual methodologies. By combining the strengths of different techniques, hybrid models aim to enhance predictive performance while mitigating weaknesses. For instance, models that integrate statistical decomposition methods with ML algorithms have shown significant potential in handling the complexities of financial time series data. These models include combinations of two or more ML algorithms, two or more DL algorithms, statistical + DL algorithms, econometric models + ML algorithms, statistical + ML algorithms, econometric models + DL algorithms, and decomposition algorithms combined with ML or DL algorithms. Common decomposition methods, such as Empirical Mode Decomposition (EMD), Ensemble EMD (EEMD), Complete Ensemble EMD with Adaptive Noise (CEEMDAN), wavelet decomposition, and Variational Mode Decomposition (VMD), have been used alongside ML/DL algorithms (Asl et al., 2024; Kumar et al., 2025; Zhou et al., 2019). However, these models often lack robustness, producing significant errors and yielding less profit.

In this regard, we propose a new hybrid model combining Seasonal-Trend decomposition using LOESS (STL) and Extreme Learning Machine

(ELM), termed the STL-ELM model, which has not been explored in the existing literature on stock forecasting. STL, using Loess smoothing, is particularly suitable for stock data with multiscale features, as it decomposes data into trend, seasonal, and remainder components while allowing the original data to be reconstructed by summing these individual components. When combined with robust machine learning algorithms such as ELM, the resulting hybrid model can leverage the strengths of both techniques to deliver superior predictive performance. Among the various existing ML/DL models, we chose ELM due to its versatility, minimal computational requirements, and reduced parameter tuning compared to other models like SVM, KNN, MLP, RF, RNN, GRU, LSTM, and CNN, which are computationally intensive and often non-feedforward (Van Thieu et al., 2025). By integrating STL and ELM, our proposed STL-ELM model offers a robust solution for handling the multiscale properties of stock market data, addressing limitations in existing methods.

Unlike existing methods, the proposed STL-ELM model is specifically designed to decompose and analyze highly volatile time series data, leveraging ELM's fast training capability and generalization performance to enhance predictive accuracy. The primary objective of this study is to develop and validate a hybrid STL-ELM model for robust and accurate prediction of high-volatility stock market indices. To achieve this, the research aims to design a systematic framework for integrating STL and ELM to address the challenges of volatility and noise in stock market data, evaluate the performance of the STL-ELM model against other high-performing ML/DL approaches, apply the model to real-world stock market data to demonstrate its practical applicability and relevance, and highlight its potential to enhance decision-making for investors, financial analysts, and policymakers. This study makes several key contributions to the field of stock market prediction:

1. **Novel Integration of STL and ELM:** The research introduces a unique hybrid model that combines STL's ability to decompose and analyze time series data with ELM's rapid training and generalization capabilities. This integration is specifically tailored to address the challenges of high-volatility markets.
2. **Application to Real-World Data:** The proposed model is applied to real-world stock market datasets, showcasing its effectiveness in handling practical challenges such as noise and uncertainty.
3. **Enhanced Predictive Performance:** By leveraging the strengths of STL and ELM, the hybrid model demonstrates superior performance in terms of accuracy, robustness, and computational efficiency compared to existing methods.
4. **Profitability Assessment through Trading Strategy:** The proposed model is also tested for its profitability by subjecting it to a trading strategy. This ensures that the predictions are not only accurate but also actionable in a real-world trading environment.
5. **Implications for Decision-Making:** The study highlights the potential of the STL-ELM model to enhance decision-making for stakeholders by providing reliable predictions that inform investment strategies, risk management, and policy formulation.

The subsequent sections of this work are systematically structured as follows: [Section 2](#) reviews related studies. [Section 3](#) discusses models and methods, while [Section 4](#) provides empirical analysis and results. [Section 5](#) serves as the final part of the work, summarizing the acquired insights and outlining a direction for future work.

2. Literature review

Stock market evaluation is divided into two main approaches: fundamental analysis and technical analysis (Lohrmann & Luukka, 2019). Fundamental analysis evaluates a company's financial well-being, management, industry standing, and economic indicators to determine its inherent value. In contrast, technical analysis examines price fluctuations, chart formations, and trading volume to anticipate

future patterns. Although there are supporters for both sides, the question of whether they can continuously outperform the market is still a subject of debate. Traditional financial time series analysis was based on the assumptions of linearity and stationarity, often utilizing the linear regression model. However, non-linear connections and other complications in financial time series have rendered classic linear models insufficient. Accurate stock market prediction is arduous due to the immense amount of financial data and the complex dynamics of stock markets. The complexity is additionally expanded by variables such as market volatility, economic data, investor mood, and global events, all contributing to the unpredictability of stock movements. Stock price prediction is essentially a task of forecasting time-series data, aiming to estimate future values based on past data. Over three decades, a diverse range of techniques, as discussed in the subsequent sub-sections, have been developed to address the inherent complexity and uncertainty of stock markets.

2.1. Econometric models for stock forecasting

Econometric models such as ARIMA, SARIMA, and GARCH have been foundational in time series analysis. SARIMA is widely used for modeling linear relationships in time series data that has seasonality, while GARCH effectively captures volatility clustering, a common feature in financial markets. GARCH extends the ARCH model by incorporating lagged values of the conditional variance. ARIMA is a popular statistical method for time series forecasting. It combines autoregressive (AR) and moving average (MA) models, along with differencing to make the time series stationary. ARIMA models are effective for short-term forecasting and can capture linear relationships in the data. Some studies have utilized these models to model their financial data. For instance, [Mondal et al. \(2014\)](#) employ the Akaike information criterion (AIC) to pick the optimal ARIMA model for individual stock and quantify prediction accuracy using MAE. In addition, they assess the predictive accuracy across various training data set sizes and employ *t*-tests to evaluate the statistical significance of any observed differences. The ARIMA model used in their study demonstrates high predictive accuracy for some industries, particularly FMCG and IT, while exhibiting lower effectiveness in the banking and automobile sectors. In a separate study, [Singh et al. \(2021\)](#) put forth a novel method that integrates wavelet decomposition, wavelet denoising, ARMA, and ARIMA models to produce precise forecasts using historical data from the BSE 100 S&P index. The authors compare their hybrid models against the baseline models, demonstrating that their models exhibit reduced predicting errors and improved accuracy. The proposed model by the authors effectively mitigates risk and uncertainty for investors in stock market investments.

However, econometric models struggle with non-linear patterns and sudden market changes, which are common in high-volatility markets. The model's reliance on historical data means it may not always adapt quickly to new market conditions, making it less effective during periods of rapid change. Additionally, ARIMA requires careful parameter tuning and model selection, which can be time-consuming and complex. While GARCH models are particularly useful in financial markets where volatility clustering is common, they may not perform well in highly volatile markets with frequent sudden changes. The model's ability to predict volatility is valuable for risk management and derivative pricing, but the assumption of a constant mean return can limit its performance. Moreover, GARCH models can be sensitive to the choice of parameters and may require extensive historical data to produce accurate forecasts. The existence of this gap has resulted in an increasing trend towards the use of more advanced methods that can effectively handle these limitations.

2.2. Machine learning (ML) models for stock forecasting

With the advent of ML, new models have been developed to improve

the accuracy of stock market predictions. These models can capture complex patterns and non-linear relationships in the data, making them suitable for high volatility markets. ML models can reveal complex patterns in data that may not be immediately obvious, providing a more sophisticated comprehension of market dynamics compared to conventional statistical models. In this realization, some researchers already employed ML approaches to study stock market prediction. For instance, [Yunneng \(2020\)](#) proposed an improved KNN algorithm that integrates the historical stock price data from the past *N* days with the overall pattern observed over the initial *N* days to predict the price trend for the following day. The study evaluates the effectiveness of the suggested approach by benchmarking it with the conventional KNN algorithm and the regression prediction method. The results demonstrate that the proposed method achieves superior accuracy and exhibits a reduced standard error. [Hindrayani et al. \(2020\)](#) utilize historical fundamental data from four firms (TLKM, EXCL, FREN, and ISAT) to train and evaluate four regression models (Multiple Linear Regression, Support Vector Regression, Decision Tree Regression, and K-Nearest Regression). The results indicate that the Decision Tree Regression method yields the most favourable outcomes in terms of correlation coefficient and MAPE.

SVM is a supervised learning algorithm used for classification and regression tasks. In stock market prediction, SVM can classify stock price movements based on historical data. The SVM model finds the hyperplane that best separates the data into different classes ([Bao et al., 2024](#)). SVM can handle non-linear data by using kernel functions, such as the radial basis function (RBF). However, it may struggle with large datasets and high-dimensional data. The model's ability to find the optimal hyperplane makes it effective for classification tasks, but its performance can be affected by the choice of kernel and regularization parameters. Due to promising results, SVM has emerged as the most widely used machine learning algorithm for stock prediction. SVM provides more adaptability to acquire knowledge from data, resulting in greater forecast accuracy. Some researchers have used SVM to predict the stock market. For instance, [Lin et al. \(2013\)](#) present an SVM-based market trend prediction. The method consists of a feature selection and a prediction model. The authors use a correlation-based SVM filter to rank and select financial indicators that are highly correlated with the stock market trend. The prediction model uses a quasi-linear SVM to avoid over-fitting and improve generalization.

The article by [Khaidem et al. \(2016\)](#) suggests a method for predicting stock price trends using RF. The authors use historical trading data from four US-listed companies and apply exponential smoothing to reduce noise and volatility. The authors calculate ten technical indicators as features and use random search to optimize parameters. The optimized model achieves higher accuracy, precision, recall, and F1 score, making it more stable and reliable. Similarly, [Basak et al. \(2019\)](#) applied RF and Gradient Boosted Decision Tree (GBDT) algorithms to classify the direction of stock prices throughout a trading period spanning 3 to 90 days. The study's results suggest that the models are suitable for trading over longer durations, as the accuracy improved as the trading period rose.

While these aforementioned approaches may yield satisfactory results in many cases of stock predictions, their application to extensive datasets is hindered by their limited capacity to extract relevant features ([Vadlamudi, 2017](#)). To fill this gap, [Yun et al. \(2021\)](#) developed a three-stage predictive model utilising a hybrid GA-XGBoost, which specifically aimed to improve feature expansion, data preprocessing, and optimal feature selection. Similarly, [Mohanty et al. \(2021\)](#) devised a hybrid model that integrates Auto Encoder (AE) and Kernel ELM (KELM) to boost the robustness of stock predictions. Although ML methods show proficiency in modelling non-linear connections and extracting insights from complex datasets, these models suffer from high computational costs and high sensitivity to parameter selection. Also, these models do not have the capacity to extract the multiscale properties of stock data, such as trend, seasonality, and residuals. Consequently, investors and researchers increasingly emphasize new approaches, such as DL models.

2.3. Deep learning (DL) models for stock forecasting

DL models utilize intricate structures to acquire knowledge from extensive datasets, capturing hidden and intricate patterns that conventional machine learning models may overlook. The ability of deep learning to analyze sequential data and its proficiency in managing extensive datasets make it very suitable for evaluating the sequential characteristics of stock prices.

2.3.1. Application of CNN in time series forecasting

CNNs, RNNs, LSTMs, and GRU play a novel role in enhancing stock prediction approaches. For example, Gunduz et al. (2017) proposed a CNN-based stock market prediction using diverse variables. The work explores CNN architectures, including convolutional, pooling, and fully connected layers, and introduces two models, 2D-CNNpred and 3D-CNNpred, for improved prediction accuracy. Nevertheless, the proposed CNN performance was surpassed by the one developed by Hoseinzade and Haratizadeh (2019). The authors present a CNN structure to forecast the intraday movement of Borsa Istanbul 100 equities. The distinguishing characteristic of this CNN model is that it utilizes a meticulously arranged feature set obtained by using diverse indicators, pricing data, and temporal information. This approach is contrasted with a CNN that employs randomly ordered features and Logistic Regression. The results indicate that using ordered features, the suggested classifier surpasses the randomly ordered CNN and Logistic Regression. This emphasizes the effectiveness of feature selection in improving prediction accuracy, as well as lowering model complexity and training time. More recently, Kim et al. (2025) proposed an advanced algorithmic trading model for index futures prediction, integrating TimeGAN for time-series data augmentation and 3D-CNN for capturing multidimensional patterns. Unlike conventional stock prediction methods, this approach focuses on the futures market, which is more volatile and lacks sufficient historical data. Empirical testing on KOSPI 200, S&P 500, and NASDAQ 100 futures markets demonstrates significant performance improvements, with the model achieving up to 1.35x higher risk-adjusted returns and 6,390x greater computational efficiency compared to baseline methods.

2.3.2. Application of RNN and its variants in time series forecasting

RNNs, renowned for their effectiveness in managing time-series data using past information, have also found extensive use in stock prediction. For instance, in Kaczmarek et al. (2022), the authors adopt RNN volatility forecasts to determine market conditions and develop an investment approach that adaptively blends stocks, cash, and low-risk assets. Unfortunately, RNNs have a notable drawback: their vulnerability to the vanishing and ballooning gradient problem, especially when dealing with long sequence dependencies (Kehinde, Khan et al., 2023; Wang et al., 2022). The emergence of LSTMs directly responded to RNN limitations, as they incorporated gate logic units into RNNs to tackle these problems effectively. LSTMs have been prominent in stock forecasting. For instance, Rokhsatyazdi et al. (2020) introduced a neural network model using LSTM and differential evolution (DE) to predict future stock prices. More recently, Yang et al. (2024) designed a model called MEEMD-LSTM-MLP to improve stock price index prediction, particularly in volatile markets like those affected by COVID-19. The model integrates Modified Ensemble EMD (MEEMD) to decompose stock price indices into components, followed by Multilayer Perceptron (MLP) for high-frequency fluctuations and LSTM for long-term trends. The hybrid approach significantly outperforms traditional models and advanced ML techniques (such as CNN-LSTM and GRU-CEEMDAN-wavelet) across multiple stock market datasets, including Shenzhen Component Index (SZI) and Dow Jones Industrial Average (DJIA).

Sun et al. (2024) develop a novel hybrid approach for stock price prediction, integrating CEEMDAN, LSTM, Simulated Annealing (SA), and LightGBM to improve prediction accuracy and robustness in

financial markets. CEEMDAN decomposes stock data to filter out noise, LSTM captures long-term dependencies in time series, LightGBM enhances predictive stability through ensemble learning, and SA optimizes model parameters to prevent overfitting. The proposed CEEMDAN-LSTM-SA-LightGBM model outperforms traditional models, including standalone LSTM and other hybrid methods, in forecasting six stock datasets with higher accuracy and lower error metrics (RMSE and MAE). More recently, Kumar et al. (2025) present a cutting-edge ML framework that enhances volatility forecasting across financial markets using high-frequency data from SSE, INFY, and NIFTY. The authors introduce a hybrid model that integrates VMD with DL architectures (ANN, LSTM, GRU) and a Q-learning ensemble mechanism, forming the Q-VMD-ANN-LSTM-GRU model. This hybrid approach excels in capturing complex, nonlinear, and time-dependent patterns in realized volatility, outperforming traditional models (e.g., GARCH) and standalone ML models in predictive accuracy.

GRU is well noted for its straightforward architecture with a lesser number of gates (Bao et al., 2024), and in some cases, it has often outperformed LSTM or exhibited identical performances in making accurate stock predictions. Leng et al. (2022) combine Gated Orthogonal Recurrent Units (GORU) with a Variational Auto-Encoder (VAE) to capture both temporal dependencies and market sentiment. More recently, Gkonis and Tsakalos (2025) developed a novel hybrid DL, GJO-GRU|LSTM, optimized using the Golden Jackal Optimizer (GJO), a nature-inspired metaheuristic algorithm for accurate stock price forecasting. Compared against 18 benchmark neural network models, the GJO-GRU|LSTM consistently delivers superior prediction accuracy based on RMSE, MAE, and MAPE metrics.

While GRU is specifically designed to address the issue of vanishing gradients, it may still face difficulties in capturing dependencies that extend over a substantial number of time steps. This limitation can impact the model's robustness when handling unusually lengthy sequences over prolonged durations. It is worth noting that despite the immense potential of AI models, ML and DL models are computationally intensive and require large amounts of memory and runtime for the successful training of high-volatility stock markets.

2.4. Beyond DL models in time series forecasting

Away from traditional DL models, recent advancements have shifted towards attention-based DL architectures for time series prediction (T. Kehinde et al., 2025). Vaswani et al. (2017) introduced the Transformer model, marking a significant breakthrough in modelling and prediction tasks, particularly in natural language processing (NLP). Transformers excel in capturing long-term dependencies through their multi-head self-attention mechanism, outperforming conventional feedforward and feedback networks. While substantial research has explored the application of Transformers in time series forecasting, their implementation in structured data scenarios remains limited. This limitation arises from the quadratic complexity of self-attention computation in terms of both memory and processing time (Hao & Liu, 2024).

To address this challenge, various Transformer-based variants have been proposed to enhance efficiency and adapt the architecture to time series forecasting, particularly in stock price prediction. For instance, LogTrans introduced a novel LogSparse self-attention mechanism where queries and keys for self-attention are generated through causal convolution, reducing computational demands (Li et al., 2019). Another variant, Reformer, replaces the standard self-attention mechanism with locality-sensitive hashing and employs reversible residual connections to optimize memory usage (Kitaev et al., 2020). Informer improves efficiency by utilizing a ProbSparse self-attention mechanism that selectively focuses on extracting significant queries (Lu et al., 2023; Zhou et al., 2021), while Autoformer integrates an autocorrelation-based approach with sequence decomposition to enhance predictive performance (Wu et al., 2021). Similarly, Fedformer leverages the Fourier transform to improve time series forecasting accuracy (Zhou et al.,

2022). More recently, T. Kehinde et al. (2025) introduced a novel attention-based model called Helformer, which displays a stellar performance in predicting highly volatile time series data.

Despite the introduction of numerous other Transformer variants, such as Helformer, longformer, ETSformer, Pyraformer, Crossformer, Quatformer, Galformer, and more (Ji et al., 2024; T. Kehinde et al., 2025; Lin et al., 2022; Tay et al., 2022), empirical studies have demonstrated that their performance can often be surpassed by simpler linear models or hybrid approaches. While Transformers have undoubtedly revolutionized prediction tasks, their default architecture struggles when applied to time series forecasting, particularly in high-volatility stock markets like the Hang Seng Index, S&P 500, FTSE 100 and DJIA. The computational burden, coupled with suboptimal performance in rapidly fluctuating market conditions, highlights the necessity of exploring alternative models that balance efficiency and predictive accuracy.

2.5. Research gaps

Despite significant advancements in stock market prediction, existing methods still face notable limitations. Traditional statistical and econometric models, such as ARIMA, SARIMA, and GARCH, struggle to capture the complex, nonlinear, and dynamic nature of financial markets, especially under high-volatility conditions. While ML and DL models, including LSTM, GRU, and CNN, have improved predictive accuracy, they are often computationally expensive, sensitive to hyperparameter tuning, and prone to overfitting, particularly when dealing with noisy and multiscale financial data. Hybrid models that integrate statistical decomposition techniques with ML/DL algorithms have emerged as a promising approach to address these challenges. However, many existing hybrid frameworks lack a systematic integration process, resulting in suboptimal performance. Additionally, recent attention-based DL models, such as Transformer variants, have shown promise in time series forecasting. However, their application to high-volatility stock markets remains limited due to their computational inefficiency, quadratic complexity, and challenges in handling structured financial data. While various Transformer modifications, such as Helformer, LogTrans, Reformer, Informer, Autoformer, and FedFormer, have attempted to optimize time series forecasting, empirical findings suggest that their performance may be outmatched by simpler, more efficient models (Das et al., 2023; Hao & Liu, 2024; Zeng et al., 2023).

To bridge these gaps, this study introduces a novel hybrid STL-ELM framework that leverages STL for effective time series decomposition and ELM for rapid and robust forecasting. Unlike conventional models, STL-ELM efficiently captures the multiscale features of financial time series data while maintaining high predictive accuracy and computational efficiency, making it particularly suitable for real-time and high-frequency trading scenarios. This research aims to fill the gap by providing a computationally lightweight yet highly effective model for stock market forecasting, demonstrating superior performance over existing ML, DL, and hybrid approaches. STL's ability to decompose time series data into trend, seasonal, and residual components provides a robust foundation for analyzing multiscale properties, while ELM's rapid training and generalization capabilities make it well-suited for handling complex relationships. By leveraging these complementary strengths, the STL-ELM model offers a practical and efficient solution for predicting high-volatility stock markets, with the potential to significantly improve predictive accuracy and profitability at a lesser computational requirement, resource usage, and hyperparameter tuning.

3. Data and methods

The proposed STL-ELM model integrates Seasonal-Trend Decomposition using Loess (STL) with Extreme Learning Machine (ELM) to create a robust framework for predicting high-volatility stock market data. The methodology involves decomposing the stock price data into trend,

seasonal, and residual components using STL. Each component is modeled and predicted individually using ELM, and the final predictions are obtained by summing the outputs of the individual components.

3.1. Seasonal-Trend decomposition using LOESS (STL)

STL is a robust and flexible method that was developed by Cleveland et al. (1990) for decomposing time series data into three distinct components: trend, seasonal, and remainder. STL works by first applying LOESS (Locally Estimated Scatterplot Smoothing) to estimate the seasonal component, which captures repeating patterns within a specified seasonal window. The seasonal component is then subtracted from the original data to obtain a deseasonalized series. Next, LOESS is applied again to the deseasonalized series to estimate the trend component, which represents the long-term progression of the data. The residual component, which captures the irregular fluctuations, is obtained by subtracting both the trend and seasonal components from the original series. STL, as a filtering procedure decomposes a given stock series X_t into three additive components: trend (T_t), seasonal (S_t), and remainder (R_t). This decomposition can be expressed mathematically as:

$$X_t = T_t + S_t + R_t \quad (1)$$

STL is an iterative method that involves two recursive methods: inner and outer loops. Each pass through the inner loop includes a seasonal smoothing step to update the seasonal component, followed by a trend smoothing step to update the trend component. After completing the inner loop, robustness weights are calculated in the outer loop to reduce the influence of outliers on the seasonal and trend components in subsequent inner loops. The inner loop consists of six steps as follows:

1. Detrending: The time series is detrended by removing the estimated trend component:

$$X_{detrend} = X_t - T_t \quad (2)$$

2. Seasonal Smoothing: A Loess smoother is applied to $X_{detrend}$ to extract the preliminary seasonal component $\bar{S}_t$.

$$\bar{S}_t = \text{Loess}(X_{detrend}) \quad (3)$$

3. Low-Pass Filtering: The preliminary seasonal component is processed to remove remaining trends, producing $\bar{T}_t$.

4. Seasonal Component Extraction: The final seasonal component is computed as:

$$S_t = \bar{S}_t - \bar{T}_t \quad (4)$$

5. Deseasonalizing: The original series is adjusted by removing the seasonal component:

$$X_{deseason} = X_t - S_t \quad (5)$$

6. Trend Smoothing: The deseasonalized series is smoothed to extract the final trend component:

$$T_t = \text{Loess}(X_{deseason}) \quad (6)$$

This iterative process ensures robust decomposition, isolating seasonality and trends effectively while handling outliers. For the outer loop, the residual components are analyzed, and any extreme deviations among them are classified as outliers. These outliers are detected, and corresponding weights are calculated. The computed weights are

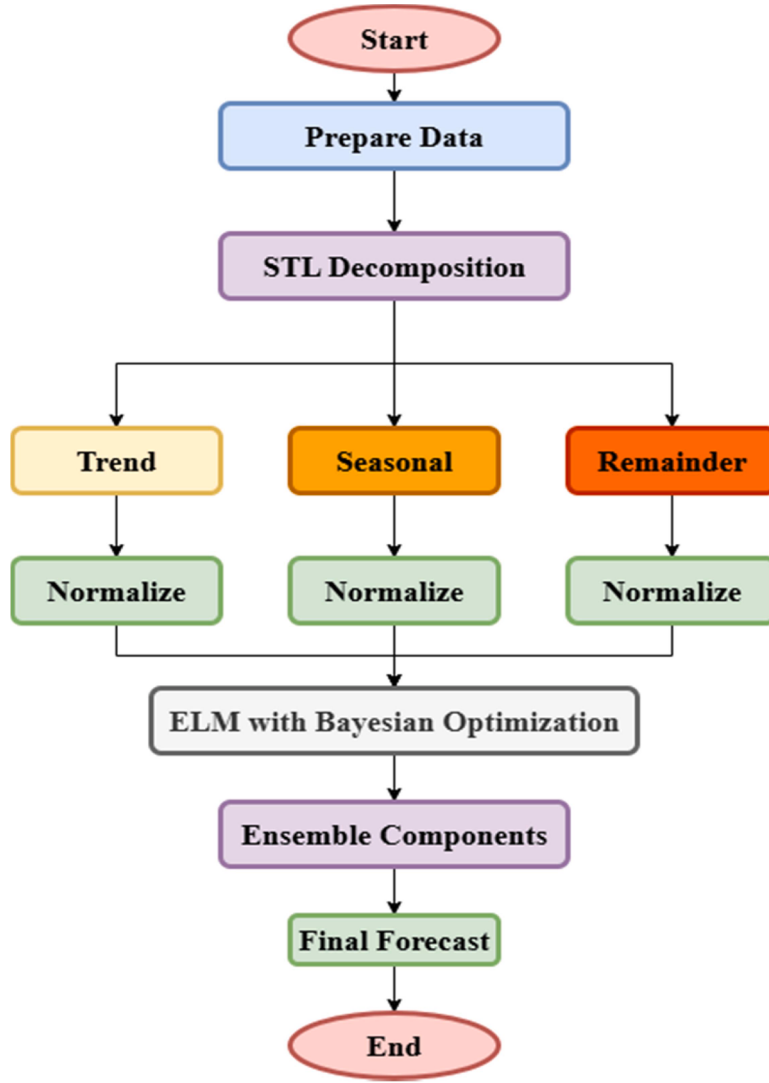


Fig. 1. STL-ELM methodology flow chart.

utilized in subsequent iterations of the inner loop to reduce the influence of the identified outliers from the previous iteration in the outer loop, ensuring improved robustness and accuracy in the modelling process.

3.2. Extreme learning machine (ELM)

ELM was first developed by Huang et al. (2006) as an innovative and efficient method for training Single-Layer Feedforward Neural Networks (SLFNs). ELM differentiates itself from traditional ANNs by randomly assigning input weights and biases instead of iteratively optimizing them through backpropagation (Van Thieu et al., 2025). This significantly reduces computational complexity, as only the output weights are computed by solving a linear system, ensuring a global optimum and eliminating the risk of getting stuck in local minima. ELM offers exceptional generalization performance, often surpassing traditional models like SVM. Its ability to handle large-scale datasets efficiently and adapt across diverse applications, such as regression, classification, and time-series forecasting, makes it a unique and highly effective tool. The lightweight architecture and computational speed of ELM, coupled with its ability to deliver high accuracy, make it an excellent choice for this work, particularly in high-frequency stock forecasting, where rapid and precise predictions are crucial. Given N training samples (x_j, t_j) where x_j represents inputs and t_j represents outputs, ELM is defined by:

$$\sum_{i=1}^{\bar{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, \text{ where } j = 1, 2, \dots, N \quad (7)$$

Where w_i is the input weight vector, b_i is the bias, β_i is the output weight vector, $g(\cdot)$ is the activation function, and $\bar{N}$ is the number of hidden neurons. The training involves:

1. Random initialization: input weights (w_i) and biases (b_i) are randomly assigned.
2. The hidden layer output calculation is expressed as follows:

$$H = \begin{pmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\bar{N}} \cdot x_1 + b_{\bar{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\bar{N}} \cdot x_N + b_{\bar{N}}) \end{pmatrix} \quad (8)$$

preprocessing involved several steps to ensure data quality and improve model performance.

3. Output Weight Optimization: Solve $H\beta = T$. $\bar{\beta} = H^\dagger T$ where $H^\dagger$ is the Moore-Penrose (MP) generalized inverse of matrix H .

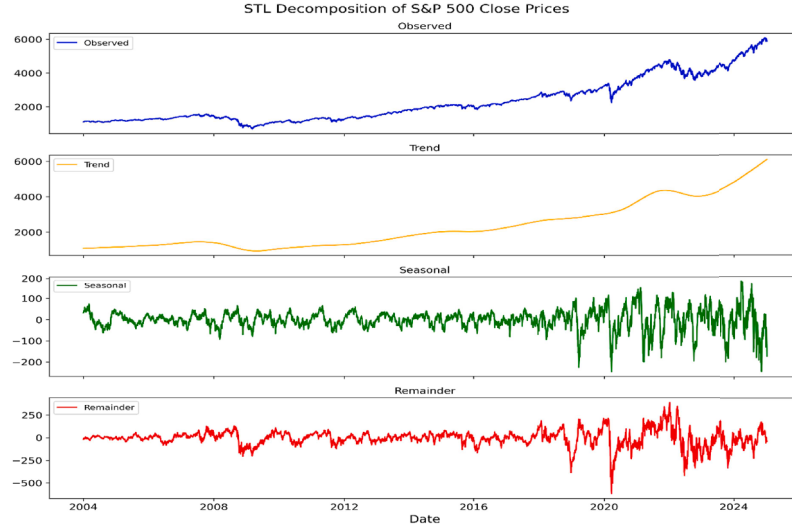
3.3. Proposed STL-ELM method

The STL-ELM method combines STL decomposition with ELM to forecast high-volatility stock prices. The framework of the proposed

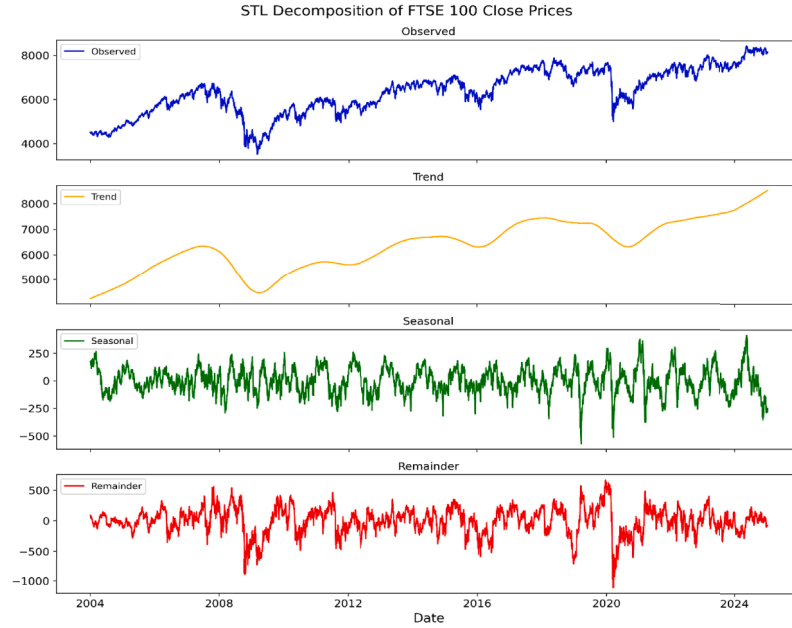
Table 1

Descriptive statistics of selected stock indices.

Index	Country	Samples	Start date (dd/mm/yy)	End date (dd/mm/yy)	Mean	Std. Dev.
S&P 500	USA	5285	01-01-2004	31-12-2024	2295.73	1272.85
FTSE 100	UK	5304	01-01-2004	31-12-2024	6351.58	1024.23
HSI	Hong Kong	5175	01-01-2004	31-12-2024	21,600.58	4584.05
DJIA	USA	5285	01-01-2004	31-12-2024	19,516.68	9427.17



(a) S&P 500



(b) FTSE 100

Fig. 2. Decomposition plot of stock indices.

model is illustrated in Fig. 1. The methodology involves the following steps:

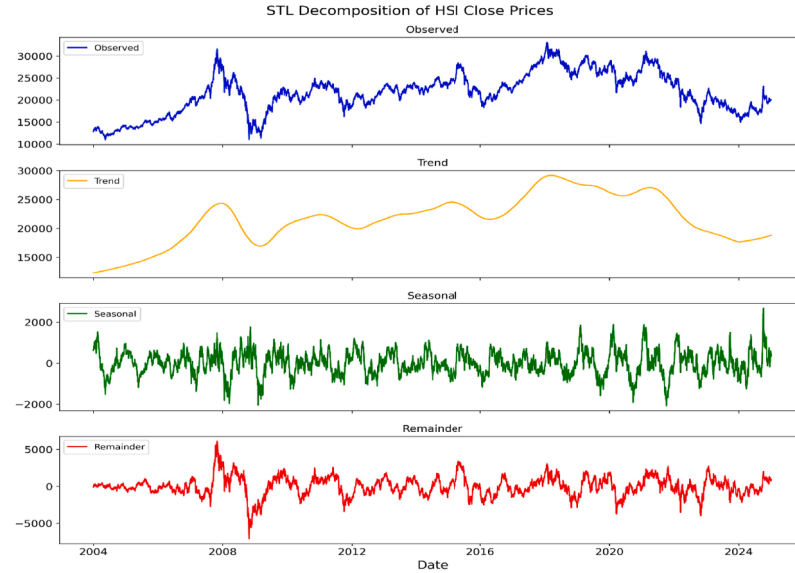
1. Decomposition: Decompose the original time series X_t into three components using STL:

$$X_t = T_t + S_t + R_t \quad (9)$$

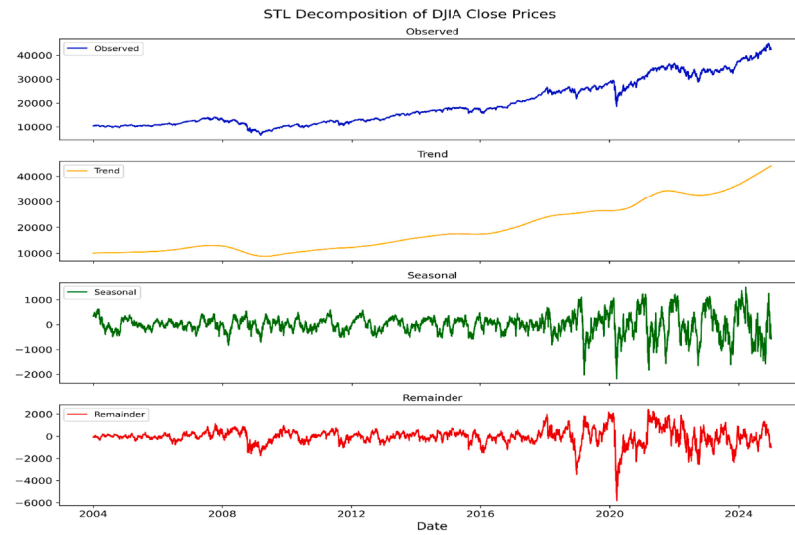
2. Independent Component Prediction: Use ELM to model and predict each component (S_t , T_t , R_t) separately as follows:

$$\bar{S}_t = ELM(S_t), \bar{T}_t = ELM(T_t), \bar{R}_t = ELM(R_t) \quad (10)$$

3. Reconstruction: Combine the predicted components to generate the final forecast as follows:



(c) HSI



(d) DJIA

Fig. 2. (continued).

$$\bar{X}_t = \bar{S}_t + \bar{T}_t + \bar{R}_t \quad (11)$$

This hybrid approach ensures that each decomposed component is accurately modelled and predicted, leveraging the strengths of STL for decomposition and ELM for fast and efficient learning. The final predictions are obtained by summing the individual components, offering a robust and computationally efficient solution for high-volatility stock forecasting.

3.4. Data preprocessing

Data preprocessing plays a crucial role in ensuring the quality and reliability of input data for ML models, especially in the context of financial forecasting. For this study, daily historical stock data were collected from investing.com for four major global indices: the S&P 500 (USA), FTSE 100 (UK), Hang Seng Index (HSI – Hong Kong), and Dow Jones Industrial Average (DJIA – USA). These indices were carefully selected for their geographical coverage, economic significance, and

inherent volatility. The S&P 500 and DJIA are leading benchmarks for the US market, representing a mix of large-cap corporations across various sectors. The FTSE 100 captures market dynamics within the UK and broader European context, while the HSI represents a critical window into the Asian financial markets, particularly China-linked equities. The inclusion of these four indices ensures that the proposed model is tested on datasets exhibiting diverse volatility profiles, economic conditions, and trading behaviours. Each dataset spans from January 1, 2004, to December 31, 2024, with a summary provided in Table 1, encompassing multiple market cycles, including the 2008 global financial crisis, the COVID-19 pandemic, and subsequent recoveries, and the ongoing war between Russia and Ukraine, as well as Israel and Hamas. This long-term horizon supports robust temporal analysis and enhances the generalizability of the proposed STL-ELM model. The proposed model is designed to predict the next day's value using a look-back period of 10 days. The choice of market indices over individual stock data offers broader market representativeness, reduced idiosyncratic noise, and greater relevance for institutional investors.

Table 1 provides descriptive statistics for four major stock indices, highlighting their sample sizes, time spans, mean values, and standard

Table 2

TPE hyperparameters for various model configurations.

Model	#P	#SR	#TP S&P 500	#TP FTSE 100	#TP HSI	#TP DJIA
STL-ELM	Neurons (trend)	(30, 35, step = 1)	33	34	35	35
	Neurons (seasonal)	(30, 35, step = 1)	35	35	35	35
	Neurons (remainder)	(30, 35, step = 1)	35	32	35	35
LSTM	Neurons	(25, 40, step = 5)	40	40	40	40
	Epochs	(70, 130, step = 10)	120	120	120	130
	Dropout	(0, 0.2)	0.0252	0.0307	0.0003	0.0170
	Learning rate	(0.0001, 0.01)	0.0080	0.0016	0.0075	0.0046
	Batch size	[16, 32, 64]	16	16	16	16
	Layers	(1,4)	1	2	1	1
GRU	Neurons	(25, 40, step = 5)	35	35	40	40
	Epochs	(70, 130, step = 10)	110	110	120	120
	Dropout	(0, 0.2)	0.0384	0.0384	0.0040	0.0348
	Learning rate	(0.0001, 0.01)	0.0048	0.0048	0.0037	0.0043
	Batch size	[16, 32, 64]	32	32	16	32
	Layers	(1,4)	1	1	1	1

#P = Parameter, #SR = Search Range, #TP = Tuned Parameter.

Table 3

Evaluation metrics on out-of-sample period (Base model configuration).

Model	Metric	S&P 500	FTSE 100	HSI	DJIA
LSTM	RMSE	55.5021	82.6004	356.9658	383.9929
	MAE	43.9355	68.5327	276.9732	307.1941
	MAPE (%)	0.98	0.91	1.35	0.88
	R ²	0.9922	0.9753	0.9923	0.9882
GRU	RMSE	74.9982	75.3061	342.5812	382.6947
	MAE	58.5996	58.5328	254.0223	307.3048
	MAPE (%)	1.25	0.80	1.21	0.87
	R ²	0.9857	0.9795	0.9929	0.9883
STL-ELM	RMSE	49.2262	61.1117	327.7894	323.0286
	MAE	38.6157	44.4965	244.2430	245.5001
	MAPE (%)	0.87	0.61	1.16	0.71
	R ²	0.9938	0.9865	0.9935	0.9917

Table 4

Evaluation metrics on out-of-sample period (Tuning using Optuna).

Model	Metric	S&P 500	FTSE 100	HSI	DJIA
LSTM	RMSE	53.1989	61.2743	334.7227	339.6334
	MAE	40.4696	44.6358	248.9309	259.4398
	MAPE (%)	0.92	0.61	1.19	0.75
	R ²	0.9928	0.9864	0.9932	0.9908
GRU	RMSE	56.2221	61.6468	333.2739	371.2612
	MAE	44.5258	45.4073	245.1551	289.1579
	MAPE (%)	0.99	0.62	1.17	0.83
	R ²	0.9920	0.9863	0.9933	0.9890
STL-ELM	RMSE	46.5275	60.9782	324.8360	317.7204
	MAE	35.4788	44.4916	242.5197	242.2876
	MAPE (%)	0.81	0.61	1.16	0.70
	R ²	0.9945	0.9866	0.9936	0.9919

deviations. The HSI and DJIA exhibit the highest average prices and volatility, indicating greater market fluctuations. In contrast, the FTSE 100 shows lower average values and variability. These differences emphasize the diverse behavior of global markets and the need for adaptable forecasting models like STL-ELM.

Preliminary data checks confirmed the absence of missing values, and outlier analysis was conducted to validate data consistency. A correlation analysis identified the closing price as the most informative and representative feature, prompting the removal of redundant variables. Focusing on the close price simplifies the model input while retaining critical information about market behaviour. To ensure uniformity and improve model training efficiency, a Min-Max [0, 1] scaler was fitted on the training data to transform both the training and test sets. This normalization facilitates faster convergence during training and ensures comparability across different indices with varying price magnitudes. The Min-Max scaling transformation is defined as follows:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (12)$$

Where X is the original closing price, and X_{min} and X_{max} are the minimum and maximum values of the price series, respectively.

By adopting this uniform preprocessing approach, the study ensures that each index retains its intrinsic temporal characteristics while being compatible with the same hybrid modelling architecture. These steps collectively establish a solid foundation for fair model comparison, effective training, and credible evaluation of stock forecasting performance across global markets.

3.5. Decomposition analysis

To uncover the multiscale properties inherent in financial time series, STL decomposition was applied to the daily prices of four major global stock indices: S&P 500 (USA), FTSE 100 (UK), HSI (Hong Kong), and DJIA (USA). This method effectively disaggregates each time series into trend, seasonal, and residual components, enabling more precise insights into long-term market movements, recurring cyclical patterns, and irregular fluctuations. As illustrated in Fig. 2, each index displays distinctive multiscale behavior, characterized by non-linear trends and varying degrees of seasonality. The trend component captures the persistent directional shifts over time, the seasonal component reveals cyclical movements potentially linked to institutional or macroeconomic cycles, and the remainder isolates unpredictable volatility. The clarity of these decomposed elements provides a strong foundation for improved forecasting by reducing noise and isolating meaningful structure within each index.

To further validate the presence of trends and assess stationarity, the Augmented Dickey-Fuller (ADF) test was conducted on the daily price series of all four indices. The results revealed non-stationary characteristics across the datasets, with S&P 500 (ADF = 2.2711, $p = 0.9989$), DJIA (ADF = 1.3579, $p = 0.9969$), and FTSE 100 (ADF = -2.1377, $p = 0.2296$) all failing to reject the null hypothesis of non-stationarity at the 5 % level. Although the HSI presented a relatively lower ADF statistic (-2.7713) with a p-value of 0.0625, it still did not meet the standard threshold for stationarity. These findings reinforce the rationale for adopting STL, as the presence of significant non-stationary components across all indices necessitates decomposition to enhance model interpretability and performance. By isolating the structural components of stock market data, STL serves as a critical preprocessing step in enabling robust, data-driven forecasting within the proposed STL-ELM framework.

3.6. Evaluation metrics and trading strategy

This study employed four key evaluation metrics, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R²), to assess the performance of the proposed model. These metrics provide a comprehensive evaluation of prediction accuracy, error magnitude, and model reliability. The evaluation metrics are expressed as follows:

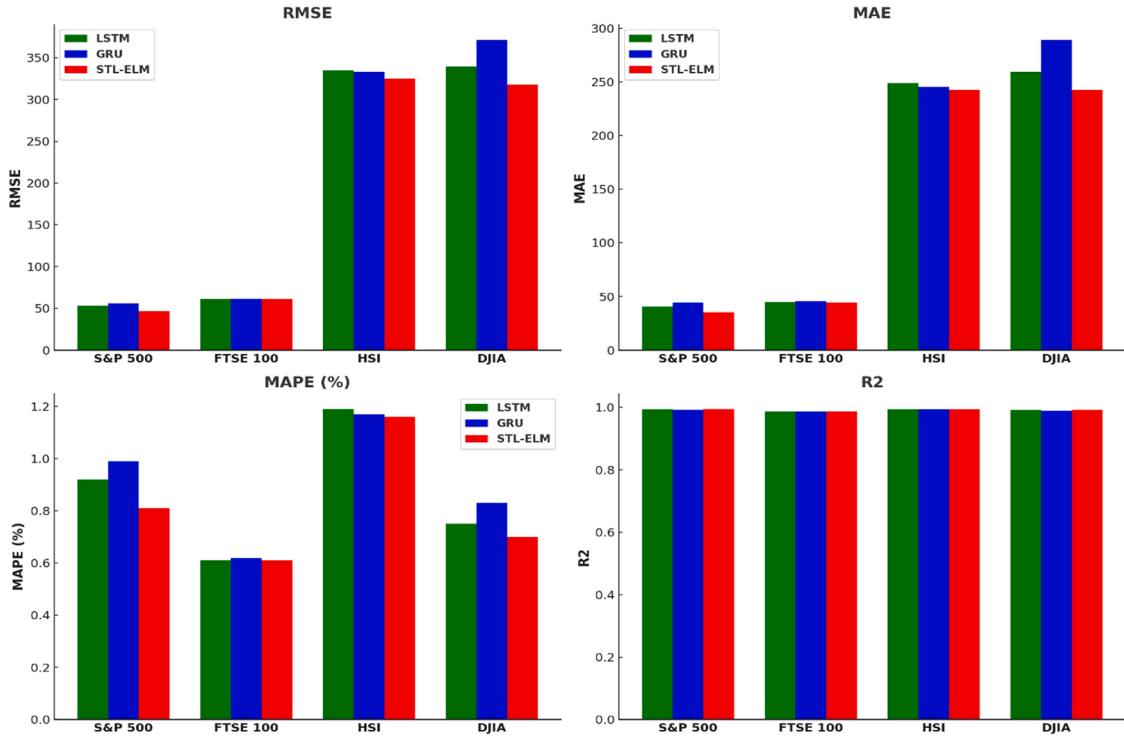


Fig. 3. Error plot of experiments.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}_i)^2} \quad (13)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{x_i - \bar{x}_i}{x_i} \right| * 100\% \quad (14)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{x}_i| \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (x_i - \bar{x}_i)^2}{\sum_{i=1}^N (x_i - \hat{x}_i)^2} \quad (16)$$

Where, x_i is the actual value, $\bar{x}_i$ is the predicted value, $\hat{x}_i$ is the mean value, and N is the length of the dataset.

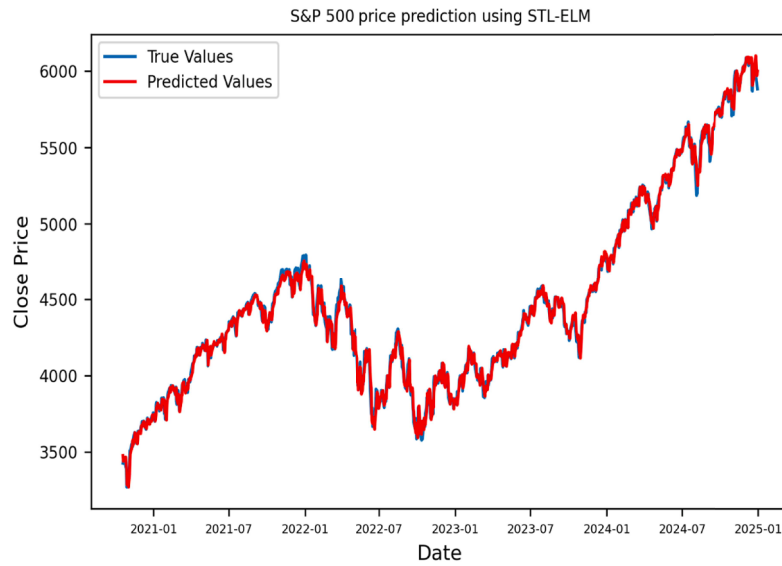
To evaluate the practical applicability of the models in financial trading, the proposed model, STL-ELM, and other baseline models were subjected to a trading strategy to estimate net portfolio values and evaluate their monetary implications. The strategy leverages the model's ability to predict future (next day) stock prices accurately, enabling practical trading decisions aimed at generating monetary rewards. This approach evaluates the effectiveness of translating predictive accuracy into real-world portfolio investments. The trading strategy uses key metrics, including total return, volatility, maximum drawdown, and the Sharpe ratio, to assess performance. Total return quantifies the overall profit or loss generated by the portfolio over the trading period, providing an essential indicator of the strategy's financial performance. Volatility measures the degree of price variation over time, serving as a critical risk assessment tool for investors and traders. Maximum drawdown evaluates the largest decline in portfolio value from its peak to its lowest point, reflecting the strategy's risk of significant losses. Finally, the Sharpe ratio assesses the risk-adjusted performance by calculating the additional return per unit of risk taken. In this study, the risk-free interest rate was assumed to be 1 %, and the number of trading days

per year was set at 252, which is in line with standard market conventions.

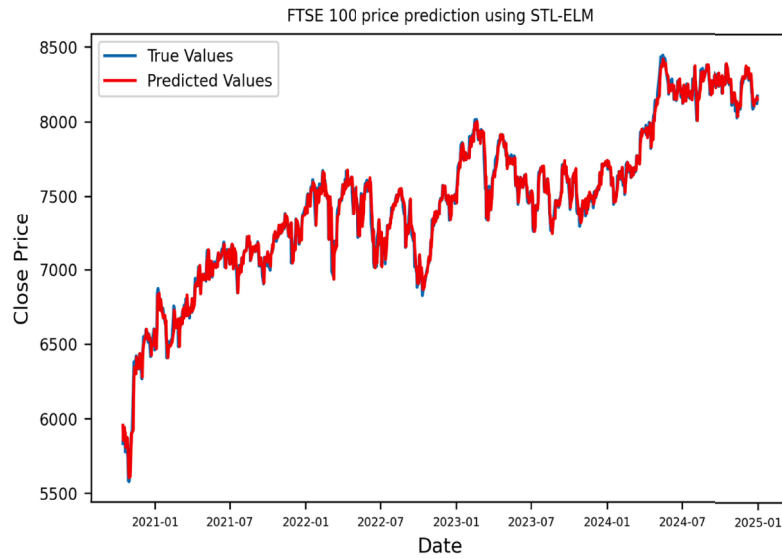
Therefore, a dynamic signal-based trading strategy was implemented. This strategy leverages the directional prediction generated by STL-ELM and other models to inform long and short positions. Specifically, a trading signal is triggered based on the sign of the difference between the predicted price and the previous actual price: a positive signal indicates a long position, while a negative signal initiates a short position. The model incorporates trading frictions by applying a 0.05 % transaction cost on both entry and exit, totaling 0.10 % transaction cost. The capital is updated iteratively based on the holding position and actual market returns, enabling the computation of performance metrics such as total return, volatility, Sharpe ratio, and maximum drawdown. This strategy aligns with the directional forecasting approach in quantitative trading and allows for assessing the predictive model's profitability and risk profile under realistic market conditions. By integrating prediction accuracy with trade execution logic, the framework bridges the gap between financial forecasting and actionable investment decisions.

3.7. Hyperparameter settings

Hyperparameter tuning is a crucial step in optimizing model performance, as it helps identify the best set of parameters for achieving superior accuracy and efficiency. In this study, Optuna, an efficient and flexible optimization framework, is employed to fine-tune the hyperparameters of the proposed STL-ELM model and the baseline models (LSTM and GRU). Optuna's ability to automate the hyperparameter search process through techniques such as Tree-structured Parzen Estimators (TPE) allowed for efficient exploration of the parameter space while minimizing computational overhead (T. Kehinde et al., 2025). For the STL-ELM model, Optuna was used to optimize the number of neurons assigned to the trend, seasonal, and remainder components independently. The search range for each parameter was carefully defined based on initial experiments, ensuring a balance between model complexity and computational efficiency. Similarly, for the baseline



(a) S&P 500



(b) FSTE100

Fig. 4. STL-ELM - Predicted vs True curve (Out of sample data).

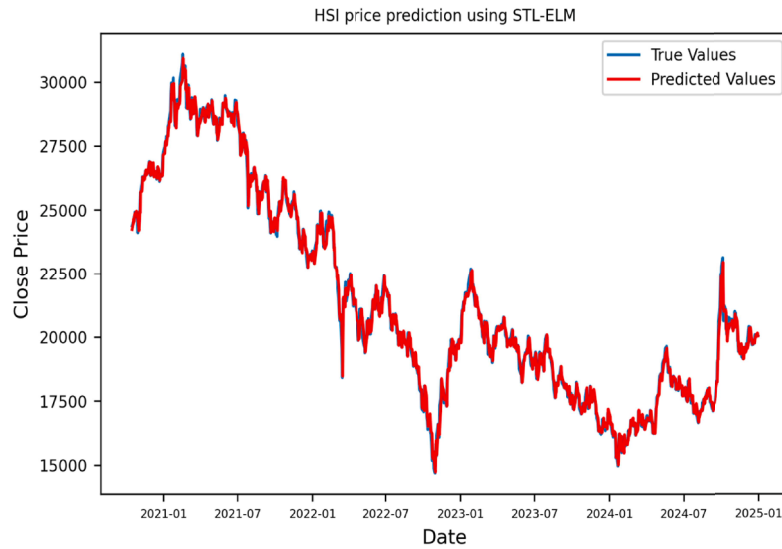
models (LSTM and GRU), key hyperparameters such as the number of neurons, batch size, learning rate, dropout rate, number of layers, and epochs were optimized. The search ranges for these parameters were tailored to the specific characteristics of each model to ensure fairness in the comparison. The tuning process for all models began with a defined objective function that minimized the validation error, such as the MSE, over a predefined number of trials. Optuna's framework dynamically adjusts the search space based on previous results, making the tuning process both efficient and effective. Once the best hyperparameter combinations were identified, these values were applied to the models to generate optimized results.

4. Experiments, results, and discussion

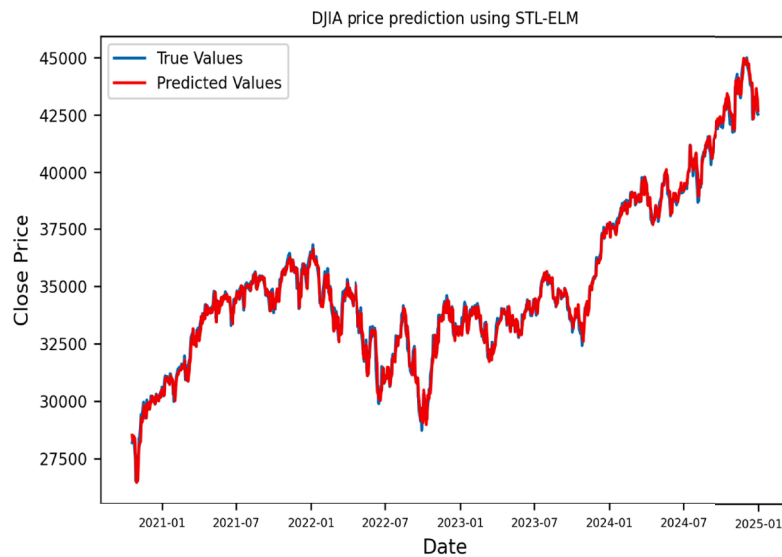
4.1. Hardware and software requirements

All experiments were conducted on a personal computer equipped

with an Intel® Core™ Ultra 7 processor, 32 GB of RAM, and 1 TB of solid-state storage (SSD). The implementation and evaluation of the proposed models were carried out using Python version 3.12.7, distributed via Anaconda on a 64-bit Windows environment (MSC v.1929 AMD64). The modeling and statistical computations were performed using key scientific libraries and frameworks including NumPy (v1.26.4), Pandas (v2.2.2), Seaborn (v0.13.2), Matplotlib (v3.9.2), and Statsmodels (v0.14.2). Model training and hyperparameter optimization leveraged Scikit-learn (v1.5.1), Optuna (v4.2.1), and Hyperopt (v0.2.7). Deep learning components utilized TensorFlow (v2.19.0) and Keras (v3.9.0). System-level monitoring and performance evaluation were supported by psutil (v5.9.0) and GPUtil (v1.4.0). This configuration ensured efficient execution of training routines, performance benchmarking, and visual analytics throughout the experimental workflow.



(c) HSI



(d) DJIA

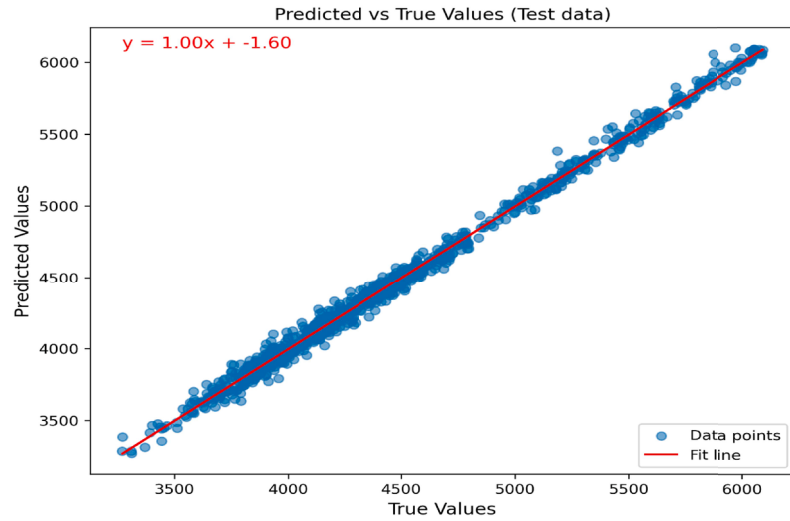
Fig. 4. (continued).

4.2. Experimental set-up

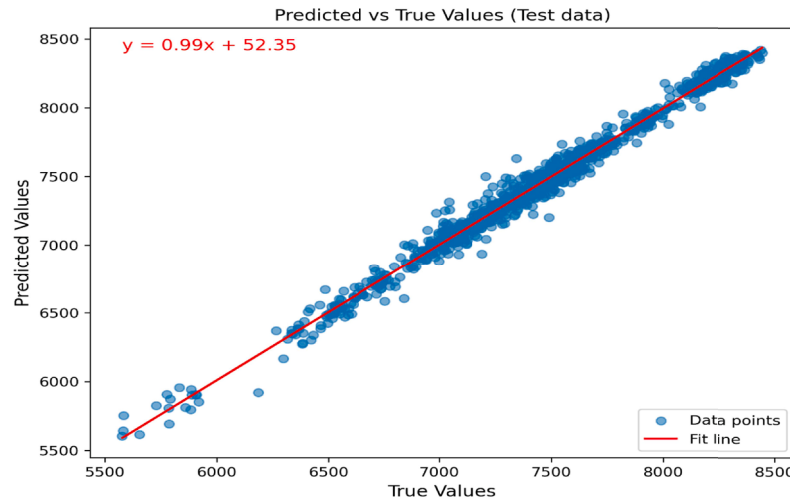
To evaluate the effectiveness of the STL-ELM hybrid model for high-volatility stock forecasting, all selected stock index datasets are partitioned into training and testing sets, with 80 % of the data used for rigorous training and 20 % as an out-of-sample period. Additionally, a validation split of 0.2 was applied during training to assess model performance during the training process to prevent overfitting. For all models, the loss function used is MSE. These settings formed the foundation for the experiments, which undergo 50 trials during tuning to ensure optimal performance. All models, including the baseline models, LSTM and GRU were configured with the following initial parameters: a look-back window of 10 days, batch size of 32, then 30 neurons were applied, one layer was selected, a dropout rate of 0.1, a learning rate of 0.001, Adam optimizer, tanh activation function, and 100 epochs. To enhance the performance of all models, hyperparameter tuning was conducted using the Optuna framework.. The tuning parameters, their respective search ranges, and the final optimized values for each model are summarized in Table 2.

4.3. Experimental results

Tables 3 and 4 present a comparative evaluation of the proposed STL-ELM model alongside LSTM and GRU across four major stock indices: S&P 500, FTSE 100, HSI, and DJIA, under two experimental configurations: without hyperparameter tuning (Table 3) and after tuning with Optuna (Table 4). This two-tiered analysis enables a comprehensive understanding of both the models' inherent forecasting capabilities and the performance enhancements brought by optimization. In the base model configuration (Table 3), STL-ELM consistently outperforms LSTM and GRU across all evaluation metrics. Specifically, STL-ELM achieves the lowest RMSE, MAE, and MAPE in all four indices. For instance, on the DJIA dataset, STL-ELM records an RMSE of 323.03 and MAPE of 0.71 %, outperforming LSTM (RMSE = 383.99, MAPE = 0.88 %) and GRU (RMSE = 382.69, MAPE = 0.87 %). A similar trend is observed in the FTSE 100, where STL-ELM achieves a significantly lower RMSE (61.11) compared to LSTM (82.60) and GRU (75.31). These results demonstrate the robustness of STL-ELM even without parameter tuning, reflecting the strength of its hybrid architecture in modeling multiscale and volatile



(a) S&P 500.



(b) FSTE 100.

Fig. 5. Scatter plots of stock index's predicted vs. true values (STL-ELM).

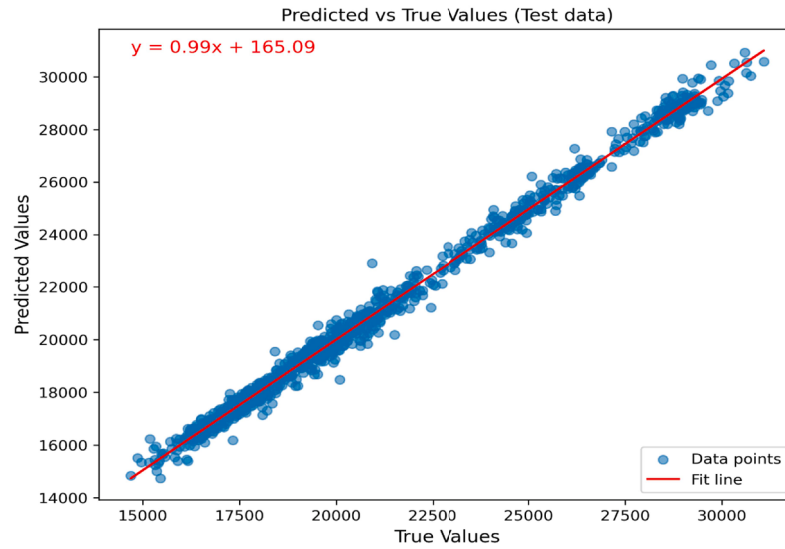
stock data.

Following hyperparameter tuning using Optuna (Table 4), all models exhibit improvements in performance, but STL-ELM maintains its leading position across all indices. The gains are particularly notable in LSTM and GRU, which benefit from better dropout rates, neuron counts, and learning rate configurations. For example, LSTM's RMSE on the S&P 500 improves from 55.50 to 53.20, while GRU's RMSE on the HSI drops from 342.58 to 333.27. Despite these improvements, STL-ELM continues to yield the lowest RMSE and MAE in all four datasets, reaffirming its computational and predictive superiority. On the HSI, STL-ELM achieves an RMSE of 324.84 and MAPE of 1.16 %, slightly better than GRU (RMSE = 333.27, MAPE = 1.17 %) and LSTM (RMSE = 334.72, MAPE = 1.19 %). Additionally, STL-ELM achieves the highest R^2 values across all datasets post-tuning, peaking at 0.9945 for the S&P 500. What is particularly compelling is STL-ELM's minimal error reduction after tuning, suggesting that its architecture is naturally robust and less sensitive to hyperparameter settings. In contrast, LSTM and GRU exhibit more significant changes in performance after tuning, highlighting their dependence on careful parameter configuration. STL-ELM's lightweight feedforward design, combined with STL's decomposition of the time series, appears to provide a strong structural advantage in capturing

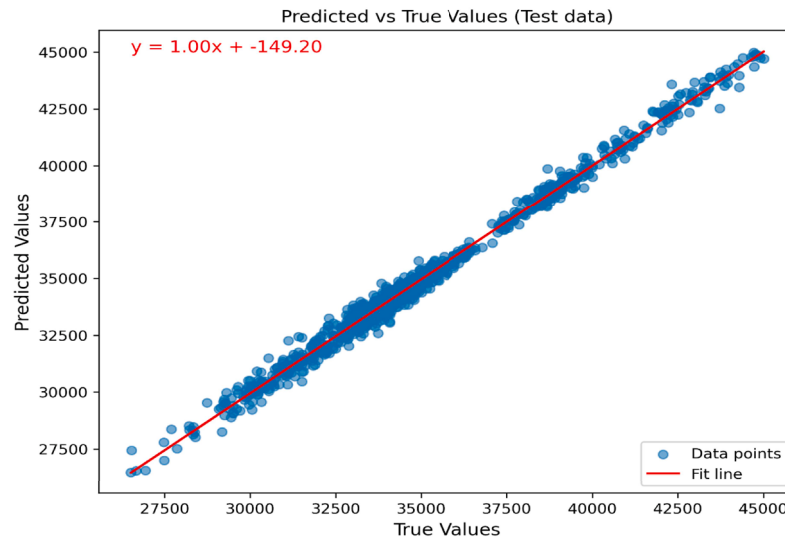
complex stock market dynamics with fewer adjustments.

Fig. 3 illustrates the comparative out-of-sample performance of LSTM, GRU, and STL-ELM models across four major stock indices using RMSE, MAE, MAPE, and R^2 metrics.

In Fig. 4, the out-of-sample prediction plots for the DJIA and S&P 500 indices clearly illustrate the STL-ELM model's strong alignment between predicted and actual closing prices over time. For DJIA, the model captures both long-term upward momentum and intermediate market corrections with impressive accuracy. Similarly, the S&P 500 plot showcases a tight overlap between the true and predicted curves across bullish trends, dips, and rebounds. These results indicate the model's capacity to learn from historical trends and generalize effectively, even in the presence of non-linear fluctuations. The consistency observed throughout these U.S.-based indices highlights STL-ELM's robustness in modelling complex dynamics in high-cap, developed markets. For the HSI and FTSE 100, the predicted curves also closely follow the actual price trajectories, though these markets exhibit more pronounced volatility and trend shifts. The HSI plot reflects significant drawdowns and short-term spikes, which the model captures well, reinforcing STL-ELM's adaptability to unstable and noisy environments like the Hong Kong market. Meanwhile, the FTSE 100 prediction aligns with the



(c) HSI.



(d) DJIA.

Fig. 5. (continued).

index's moderate growth path, capturing recurring fluctuations and stabilization periods effectively. Across all indices, the minimal visual deviation between true and predicted values validates the model's high predictive precision and confirms its capability to operate reliably across different global financial markets and market regimes.

Fig. 5 illustrates the scatter plots of predicted versus actual values for all four indices, highlighting the regression lines that quantify the STL-ELM model's predictive performance. For the DJIA, the regression equation is $y = 1.00x - 149.20$, indicating a perfect slope of 1.00, which signifies that the model accurately captures the scale of the actual values. The negative intercept reflects a slight underestimation across the prediction range. Nevertheless, the tight clustering of points around the regression line affirms the model's precision in tracking high-cap U. S. stock index movements. In the case of the HSI, the regression line $y = 0.99x + 165.09$ reveals a strong linear relationship, with a slope just under 1.00, suggesting minor underprediction at higher value ranges. The positive intercept introduces a slight upward bias, which can be attributed to the inherent volatility in the Hong Kong market. Still, the model maintains consistent accuracy across the spectrum.

For the FTSE 100, the equation $y = 0.99x + 52.35$ also implies high

fidelity, with a slope close to unity and a modest positive intercept. This result signifies that the STL-ELM model performs stably in moderately volatile environments such as the UK market, accurately reproducing both the trend and scale of the original series. Finally, the S&P 500 exhibits a regression line of $y = 1.00x - 1.60$, indicating near-perfect scale alignment with minimal bias. The slope of 1.00 and the negligible negative intercept confirm the model's capability to generalize effectively and maintain predictive accuracy across all value ranges. Together, these regression plots validate STL-ELM's robustness and generalization strength across diverse market behaviors, highlighting its suitability for both stable and volatile financial forecasting scenarios.

4.4. Trading results

The trading performance of the proposed STL-ELM model, its baseline counterparts (LSTM and GRU), and the Buy and Hold (B&H) strategy is benchmarked using four key metrics: total return, volatility, maximum drawdown, and Sharpe ratio, as presented in Table 5. The results clearly demonstrate that STL-ELM substantially outperforms the baseline models and B&H strategy across all indices, especially in terms

Table 5
Trading strategy on out of sample period.

Model	Metric	S&P 500	FTSE 100	HSI	DJIA
LSTM	Total Return	−60.77 %	56.11 %	−32.11 %	−45.60 %
	Volatility	0.0104	0.0083	0.0158	0.0090
	Maximum Drawdown	−0.6569	−0.1078	−0.5998	−0.5334
	Sharpe Ratio	−1.3331	0.7929	−0.2899	−1.0090
	Total Return	−39.01 %	17.02 %	−46.89 %	11.34 %
GRU	Volatility	0.0104	0.0084	0.0158	0.0090
	Maximum Drawdown	−0.4562	0.1698	−0.5017	−0.1960
	Sharpe Ratio	−0.6931	0.2725	−0.5299	0.1808
	Total Return	81.84 %	59.12 %	256.12 %	68.05 %
	Volatility	0.0104	0.0083	0.0158	0.0090
STL-ELM	Maximum Drawdown	−0.1718	−0.1040	−0.2277	−0.1260
	Sharpe Ratio	0.8854	0.8266	1.3165	0.8657
	Total Return	71.54 %	40.06 %	−17.78 %	50.82 %
B&H	Volatility	0.0104	0.0083	0.0158	0.0090
	Maximum Drawdown	−0.2543	−0.1103	−0.5275	−0.2194
	Sharpe Ratio	0.8044	0.5970	−0.1044	0.6862
	Sharpe Ratio				

of total return, drawdown control, and risk-adjusted returns. In the high-volatility HSI market, STL-ELM achieves an exceptional total return of 256.12 %, significantly outperforming LSTM (−32.11 %) and GRU (−46.89 %), both of which result in substantial losses. The B&H approach also fails under these conditions, returning −17.78 %. Notably, STL-ELM attains the highest Sharpe ratio (1.3165) among all models and indices, underscoring its superior ability to generate consistent risk-adjusted returns even in turbulent markets. Additionally, it records a relatively modest maximum drawdown of −0.2277, significantly lower than the B&H (−0.5275) and GRU (−0.5017) strategies.

In the S&P 500, STL-ELM yields the highest total return of 81.84 %, outperforming B&H (71.54 %), while LSTM and GRU perform poorly with −60.77 % and −39.01 % returns, respectively. The STL-ELM strategy also posts the highest Sharpe ratio (0.8854) and the lowest drawdown (−0.1718) in this market, indicating better profitability and stability during adverse movements. For the FTSE 100, STL-ELM again leads with a return of 59.12 %, exceeding both B&H (40.06 %) and GRU (17.02 %), while LSTM still underperforms with a 56.11 % return. Although LSTM's return is slightly competitive, its Sharpe ratio (0.7929) and drawdown (−0.1078) trail those of STL-ELM, which maintains a stronger Sharpe ratio of 0.8266 and a lower drawdown (−0.1040). In the DJIA, STL-ELM secures a solid 68.05 % return, clearly outperforming B&H (50.82 %) and vastly outpacing LSTM (−45.60 %) and GRU (11.34 %). The Sharpe ratio of STL-ELM (0.8657) again leads all models, reflecting strong risk-adjusted performance, while its drawdown (−0.1260) remains the smallest, reinforcing its capital preservation capabilities. These results affirm STL-ELM's superior robustness and adaptability to diverse market conditions, positioning it as a practical and reliable approach for algorithmic trading across global equity markets.

Notably, STL-ELM demonstrates a clear advantage over the B&H strategy during periods of heightened volatility, such as sharp corrections, crisis-induced drawdowns, or rapid rebounds. While B&H suffers in such regimes due to its static exposure, STL-ELM's signal-based strategy enables timely exits during downturns and swift re-entry during recoveries. This dynamic responsiveness allows STL-ELM to reduce exposure during adverse market movements, limiting losses, and re-engage during upward trends, ultimately achieving superior risk-adjusted returns. Such performance is particularly evident in indices like the HSI and DJIA, which experienced extended periods of instability during the test horizon.

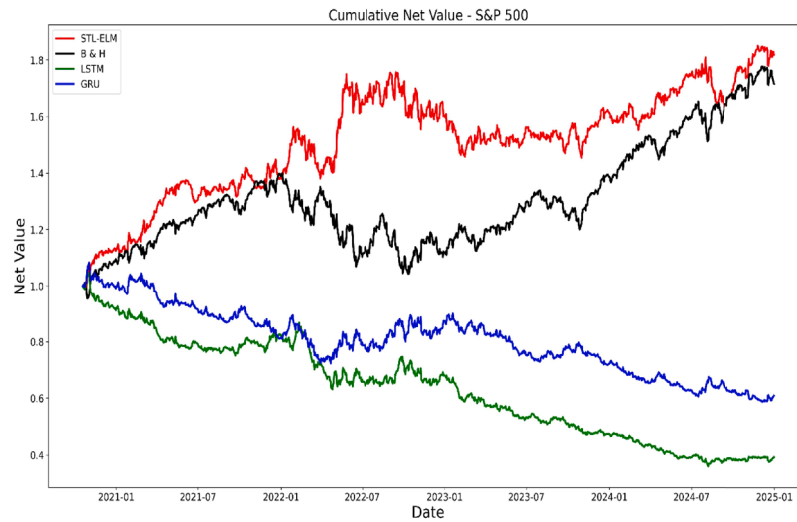
Fig. 6 presents the cumulative net value curves for the STL-ELM model, its baseline models (LSTM and GRU), and B&H strategy across the DJIA, HSI, FTSE 100, and S&P 500 indices. These curves depict portfolio growth over time, normalized to an initial value of 1, and provide critical insights into the models' trading effectiveness and stability. In the DJIA plot, the STL-ELM model (red curve) exhibits the most consistent and highest performance, ending with the greatest cumulative return and displaying smooth upward growth with minimal drawdowns. The B&H strategy (black curve) performs respectably but is marked by more volatility. GRU (blue curve) yields moderate gains with less stability, while LSTM (green curve) performs the worst, reflecting a steady and significant decline in value, highlighting its poor adaptability to the DJIA's market behavior. The HSI plot further amplifies the dominance of STL-ELM. The red curve shows a steep and sustained rise, ultimately delivering the highest net value among all strategies, even in the face of HSI's disreputable volatility. In contrast, the B&H (black), GRU (blue), and LSTM (green) curves all trend downward or remain flat over time, indicating their inability to capitalize on market fluctuations or manage downside risk effectively in a turbulent environment.

In the FTSE 100, STL-ELM once again demonstrates a strong and relatively stable upward trajectory, outperforming its counterparts by the end of the testing period. LSTM closely trails STL-ELM early on and even overlaps for some duration, but its final performance is marginally lower. B&H remains competitive, while GRU lags with visibly weaker and more erratic performance throughout. For the S&P 500, the STL-ELM model consistently maintains superior performance, culminating in the highest net value. The B&H strategy follows closely, demonstrating strong long-term returns in this relatively stable market. However, GRU and LSTM underperform significantly, especially LSTM, which suffers from a prolonged and deep decline. STL-ELM's ability to navigate and outperform even in upward-trending markets showcases its robustness and effectiveness as a tactical model. Across all indices, the STL-ELM model delivers not only the highest cumulative returns but also the most stable equity growth, affirming its resilience to market volatility and its suitability for algorithmic trading across diverse global stock markets. These plots substantiate STL-ELM's superiority in both return maximization and risk mitigation, reinforcing its value as a practical, high-performance trading strategy.

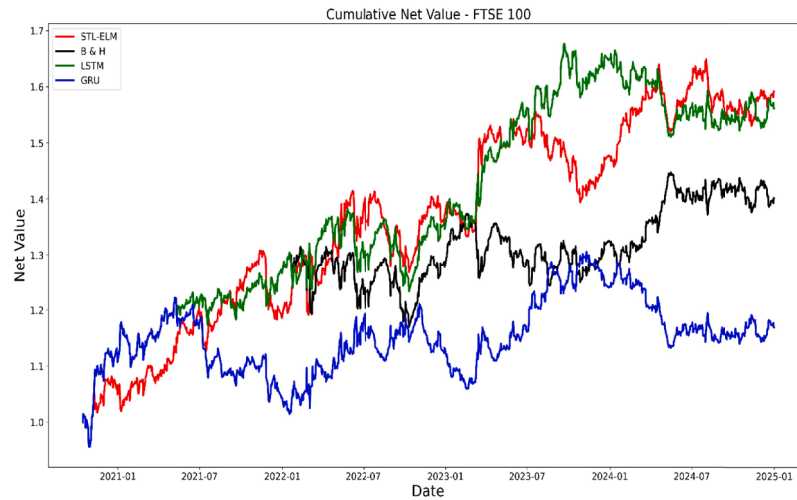
4.5. Computational efficiency

Table 6 and Fig. 7 present the computational performance metrics of the evaluated models, including runtime and memory usage. These aspects are critical for evaluating the models' feasibility in real-time trading and high-frequency financial applications where computational efficiency is essential. Table 6 indicates that the proposed STL-ELM model outperforms both LSTM and GRU by a substantial margin in terms of computational efficiency. In terms of runtime, STL-ELM completes training in just 4 to 5 s, compared to LSTM and GRU models, which require between 2120 and 4634 s. Specifically, while LSTM takes as long as 4634 s on the FTSE 100 and GRU reaches 4611 s on the DJIA, STL-ELM maintains a consistently fast runtime across all indices, highlighting its remarkable training speed.

This rapid execution is attributed to its feedforward structure and the simplicity of the ELM algorithm, which eliminates iterative back-propagation. Memory usage shows an equally striking advantage. STL-ELM consumes only a fraction of the memory used by LSTM and GRU models, ranging from just 0.70 MB on the HSI to 2.20 MB on the S&P 500. In contrast, LSTM and GRU demand hundreds to thousands of megabytes, with GRU peaking at 2191 MB on the DJIA dataset. This lightweight memory footprint underscores STL-ELM's superior scalability and suitability for deployment in constrained environments such as low-latency trading systems, edge computing platforms, or mobile applications. This high level of efficiency makes STL-ELM well-suited for deployment in high-frequency trading (HFT) systems, where real-time data processing and rapid decision-making are critical. The model's



(a) S&P 500



(b) FTSE 100

Fig. 6. Cumulative net value curves on the out-of-sample period of selected indices.

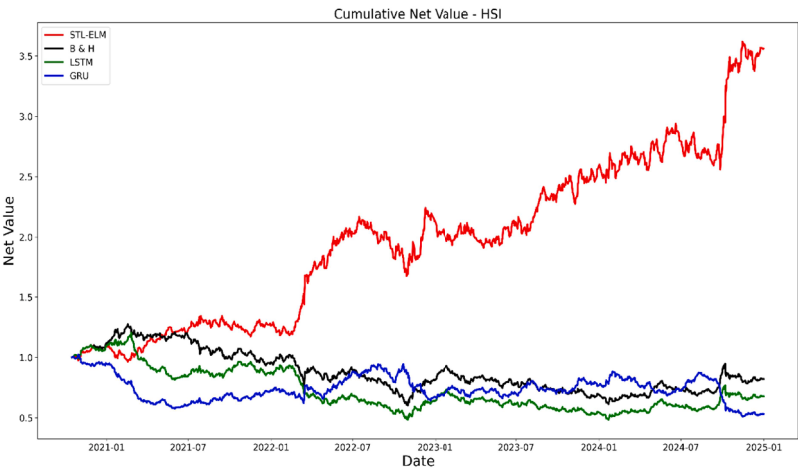
lightweight architecture allows for fast retraining and real-time inference on incoming data streams, enabling traders to execute strategies based on up-to-the-moment forecasts. Furthermore, its minimal resource consumption facilitates deployment in low-latency environments, such as edge servers or co-located trading infrastructure, making STL-ELM a practical choice for time-sensitive algorithmic trading operations.

4.6. Comparison with existing studies

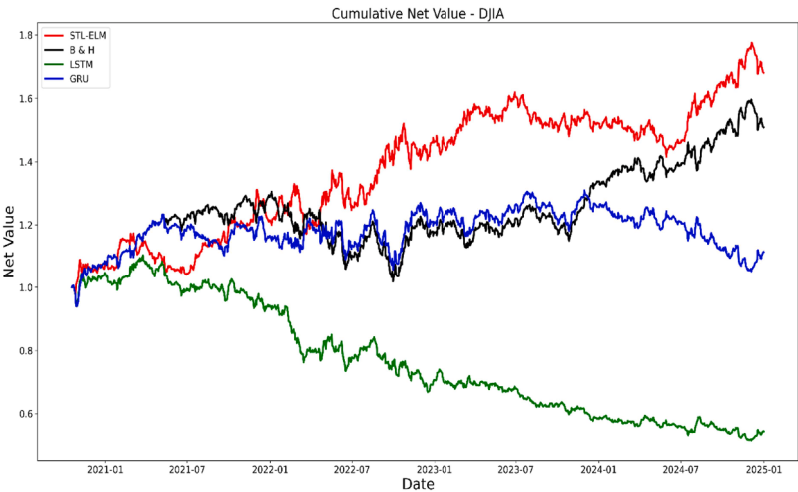
To establish the robustness and superiority of the proposed STL-ELM model, a comprehensive comparative analysis was conducted against state-of-the-art (SOTA) models from existing literature. This comparison was carefully aligned in terms of datasets, stock indices, and experimental setups to ensure fairness and reproducibility. STL-ELM was benchmarked against a range of baseline and advanced models, including traditional DL architectures (MLP, CNN, LSTM), attention-based models, e.g., the Uncertainty-Aware (UA) model by Gao et al. (2020), and sophisticated hybrid architectures like SDTP (Series Decomposition Transformer with Periodic Correlation) by Tao et al. (2024) and EMD2FNN by Zhou et al. (2019), which combines Empirical

Mode Decomposition with Factorization Machine-based neural networks. As depicted in Table 7, on the HSI index, STL-ELM significantly outperformed SDTP in all key metrics: it achieved a lower MAE (233.164 vs. 256.024), lower RMSE (315.404 vs. 345.411), lower MAPE (0.8997 vs. 0.9920), and a higher R^2 (0.9866 vs. 0.9788), indicating both superior prediction accuracy and a better fit. In the S&P 500 comparison with models reported by Gao et al. (2020), STL-ELM demonstrated the lowest RMSE (19.2237) among all contenders, including the UA model (25.4851), CNN (25.7888), LSTM (35.4955), and MLP (44.5137). While UA achieved a slightly lower MAPE (0.67 %) compared to STL-ELM (0.71 %), STL-ELM's substantially lower RMSE reinforces its robustness in minimizing absolute prediction errors.

Further comparison with EMD2FNN models from Zhou et al. (2019) across varying window sizes also affirmed STL-ELM's superiority. At all tested window sizes (3, 4, and 5), STL-ELM achieved lower RMSE and comparable or better MAE and MAPE. For instance, with window size 3, STL-ELM slightly outperformed EMD2FNN in RMSE (17.5691 vs. 17.6591) and MAE (13.0201 vs. 13.0396), while maintaining identical MAPE (1.05 %). This trend persisted for other window sizes, underscoring the STL-ELM model's consistent edge in both precision and



(c) HSI



(d) DJIA

Fig. 6. (continued).

Table 6
Computational performance metrics on training datasets.

Model	Computational Efficiency	S&P 500	FTSE 100	HSI	DJIA
LSTM	Runtime (seconds)	2916	4634	2120	2534
	Memory Usage (MB)	1228	1046	1378	1346
GRU	Runtime (seconds)	3326	2440	2401	4611
	Memory Usage (MB)	1250	645	850	2191
STL-ELM	Runtime (seconds)	4	5	5	4
	Memory Usage (MB) (* 10 ⁻²)	220	196	70	190

generalization. Based on these results, the STL-ELM model not only achieves top-tier performance compared to existing singular, attention-based, and hybrid models but also does so with fewer parameters and reduced computational complexity. These findings further reinforce STL-ELM’s potential as a lightweight, accurate, and scalable forecasting solution for high-volatility financial markets.

4.7. Managerial implications

The findings of this study offer significant managerial insights for practitioners and decision-makers in financial services, portfolio

management, and algorithmic trading. The superior forecasting accuracy and trading performance of the proposed STL-ELM model suggest that financial institutions can benefit from adopting hybrid machine learning frameworks that integrate signal decomposition with lightweight neural networks. By achieving higher returns and lower draw-downs than traditional deep learning models and benchmark strategies like B&H, STL-ELM demonstrates its practical viability in real-world trading environments. Managers can leverage this model to enhance decision-making under uncertainty, particularly in volatile markets such as the HSI and DJIA. Moreover, its exceptional computational efficiency, in both runtime and memory usage, makes STL-ELM an ideal solution for firms operating under hardware constraints or seeking scalable models for high-frequency trading. The interpretability of the STL decomposition also aids risk analysts and compliance teams in understanding the model’s behaviour, addressing regulatory transparency requirements. All things considered, this research equips managers with a robust and efficient forecasting tool that supports dynamic portfolio optimization and risk management strategies, ultimately leading to improved profitability and operational agility in fast-moving capital markets.

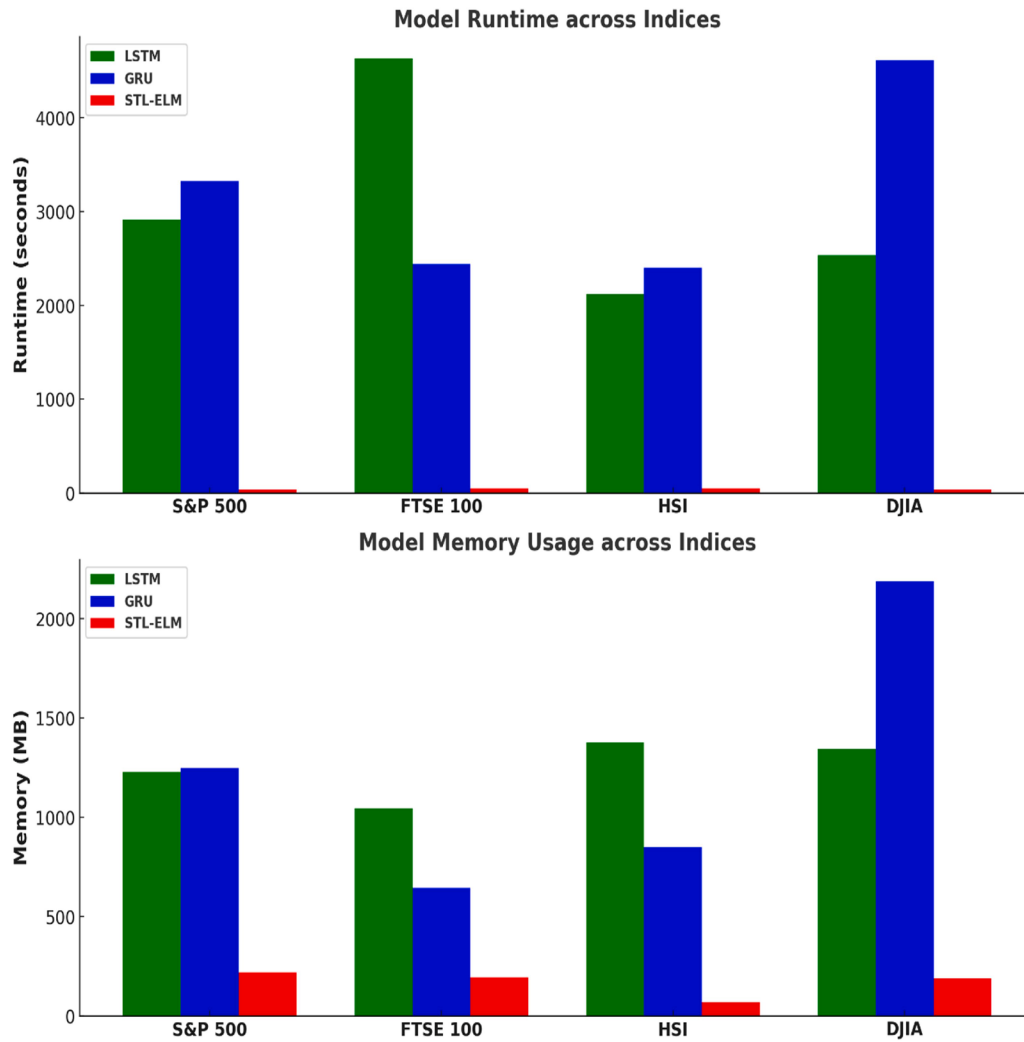


Fig. 7. Computational measures during training.

5. Conclusion and future study

5.1. Conclusion

This study introduced a computationally efficient hybrid model, STL-ELM, for forecasting high-volatility stock markets. By integrating STL with ELM, STL-ELM overcomes the limitations of conventional statistical, ML, and DL models in handling the complexities of stock price prediction. STL's decomposition capabilities allow the model to extract multiscale features from volatile stock data, such as trends, seasonality, and residual components, which are often overlooked by traditional methods. ELM complements this by providing rapid training, robust generalization, and minimal computational overhead, ensuring that the model remains efficient even in high-frequency trading environments. The proposed model not only outperforms state-of-the-art models, including notable singular, hybrid, or attention-based models, but also demonstrates superior computational efficiency, making it ideal for real-time forecasting applications.

The novelty of STL-ELM lies in its ability to combine two complementary techniques, STL and ELM, into a seamless hybrid model that excels in predictive accuracy while maintaining a lightweight and scalable architecture. Unlike other hybrid approaches that struggle with high volatility and noise in stock market data, STL-ELM demonstrates exceptional robustness and profitability, outperforming competitors across various global stock indices, including the S&P 500, FTSE 100,

HSI, and DJIA. The model's ability to handle multiscale time series features and its superior generalization capability position it as a powerful tool for investors, analysts, and policymakers seeking to navigate dynamic financial markets. Additionally, the STL-ELM model provides accurate predictions and ensures that these predictions are actionable in real-world trading, as evidenced by its success in a trading strategy that maximizes returns while minimizing risks.

From a practical standpoint, the STL-ELM model represents a significant advancement in stock market forecasting, particularly in high-volatility markets, where traditional methods often fall short. The model's efficiency, combined with its ability to produce reliable, high-quality forecasts, makes it an attractive solution for institutional investors, financial analysts, and policy decision-makers. Furthermore, due to its modular architecture, the STL-ELM model is easily transferable across financial time series with differing characteristics. The decomposition step enables the extraction of structural patterns unique to each domain, while ELM's lightweight configuration facilitates retraining with minimal overhead. This makes STL-ELM suitable for broader applications such as commodities forecasting, FX rate prediction, and high-frequency trading scenarios.

5.2. Future study

Future research can explore several avenues to build upon this work. First, enhancing ELM by integrating optimization algorithms like the Bat

Table 7
Comparison of the STL-ELM model with other models in the existing studies.

Tao et al. (2024)					
Stock Index	Models	MAE	RMSE	MAPE	R ²
HSI	SDTP	256.024	345.411	0.9920	0.9788
	STL-ELM	233.164	315.404	0.8997	0.9866
Gao et al. (2020)					
Stock Index	Models	MAE	RMSE	MAPE	R ²
S&P 500	MLP	–	44.5137	1.18 %	0.9745
	LSTM	–	35.4955	0.96 %	0.9803
	CNN	–	25.7888	0.68 %	0.9888
	Uncertainty-aware Attention (UA)	–	25.4851	0.67 %	0.9891
	STL-ELM	–	19.2237	0.71 %	0.9558
Zhou et al. (2019)					
Stock Index	Models	MAE	RMSE	MAPE	R ²
S&P 500	EMD2FNN (window size = 3)	13.0396	17.6591	1.05 %	–
	STL-ELM (window size = 3)	13.0201	17.5691	1.05 %	–
	EMD2FNN (window size = 4)	15.1386	20.3978	1.22 %	–
	STL-ELM (window size = 4)	13.0474	17.9865	1.05 %	–
	EMD2FNN (window size = 5)	16.1700	22.1277	1.30 %	–
	STL-ELM (window size = 5)	13.7026	18.5118	1.11 %	–

algorithm (Saheed et al., 2024), particle swarm optimization (PSO) or genetic algorithms (GA) for weight initialization may address variability issues and improve predictive stability. Second, expanding the evaluation to include diverse datasets across different markets and sectors would test the model’s generalizability and robustness. Additionally, future studies can investigate more sophisticated trading strategies that incorporate portfolio diversification, adaptive risk management, and market sentiment analysis to better simulate real-world scenarios. Finally, integrating explainability techniques to interpret the model’s predictions can provide deeper insights for users and stakeholders, enhancing its practical utility in decision-making.

Declarations

CRedit authorship contribution statement

Temitope Olubanjo Kehinde: Conceptualization, Methodology, Writing – original draft, Validation, Software. **Oluyinka J. Adedokun:** Writing – review and editing, Investigation, Validation. **Morenikeji Kabirat Kareem:** Visualization, Data Curation. **Joseph Akpan:** Writing – review and editing, Formal Analysis, Resources. **Oludolapo A. Olanrewaju:** Supervision, Project Administration, Funding Acquisition.

Funding

The authors are grateful to the Durban University of Technology for supporting the APC.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Abolmakarem, S., Abdi, F., Khalili-Damghani, K., & Didehkhani, H. (2024). A multi-stage machine learning approach for stock price prediction: Engineered and derivative indices. *Intelligent Systems with Applications*, 24, Article 200449.

Asl, M. G., Adekoya, O. B., Rashidi, M. M., Oliyide, J. A., & Rajab, S. (2024). A new approach to forecasting Islamic and conventional oil and gas stock prices. *International Review of Economics & Finance*, 96, Article 103513.

Bao, W., Cao, Y., Yang, Y., Che, H., Huang, J., & Wen, S. (2024). Data-driven stock forecasting models based on neural networks: A review. *Information Fusion*, Article 102616.

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, 47, 552–567.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. *J. off. Stat*, 6(1), 3–73.

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint. arXiv:2304.08424*.

Gao, P., Zhang, R., & Yang, X. (2020). The application of stock index price prediction with neural network. *Mathematical and Computational Applications*, 25(3), 53.

Gkonis, V., & Tsakalos, I. (2025). A hybrid optimized deep learning model via the Golden Jackal Optimizer for accurate stock price forecasting. *Expert Systems with Applications*, 278, Article 127287.

Gunduz, H., Yaslan, Y., & Cataltepe, Z. (2017). Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems*, 137, 138–148.

Hao, J., & Liu, F. (2024). Improving long-term multivariate time series forecasting with a seasonal-trend decomposition-based 2-dimensional temporal convolution dense network. *Scientific Reports*, 14(1), 1689.

Hindrayani, K. M., Fahrudin, T. M., Aji, R. P., & Safitri, E. M. (2020). Indonesian stock price prediction including covid19 era using decision tree regression. In *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*.

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129, 273–285.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), 489–501.

Ji, Y., Luo, Y., Lu, A., Xia, D., Yang, L., & Wee-Chung Liew, A. (2024). Galfomer: A transformer with generative decoding and a hybrid loss function for multi-step stock market index prediction. *Scientific Reports*, 14(1), Article 23762.

Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, 184, Article 115537.

Kaczmarek, T., Będowska-Sójka, B., Grobelny, P., & Perez, K. (2022). False safe haven assets: Evidence from the target volatility strategy based on recurrent neural network. *Research in International Business and Finance*, 60, Article 101610.

Kehinde, T., Adedokun, O. J., Joseph, A., Kabirat, K. M., Akano, H. A., & Olanrewaju, O. A. (2025). Helformer: An attention-based deep learning model for cryptocurrency price forecasting. *Journal of Big Data*, 12(1), 1–39.

Kehinde, T., Chan, F. T., & Chung, S. (2023). Scientometric review and analysis of recent approaches to stock market forecasting: Two decades survey. *Expert Systems with Applications*, 213, Article 119299.

Kehinde, T., Chung, S., & Chan, F. T. (2023). Benchmarking TPU and GPU for Stock Price Forecasting Using LSTM Model Development. *Science and Information Conference*.

Kehinde, T., Khan, W. A., & Chung, S.-H. (2023). Financial Market Forecasting Using RNN, LSTM, BiLSTM, GRU and Transformer-Based Deep Learning Algorithms. In *Proceedings of the IEOM International Conference on Smart Mobility and Vehicle Electrification*.

Kehinde, T. O., Chung, S.-H., & Olanrewaju, O. A. (2025). An Integrated Approach to Stock Selection and Ranking: Combining Shannon Entropy Technique, DEA, and Inverse DEA. *IEEE access : Practical innovations, open solutions*.

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv preprint. arXiv:1605.00003*.

Kim, W., Jeon, J., Kim, S., Jang, M., Lee, H., Yoo, S., & Oh, K. J. (2025). Prediction of Index Futures Movement Using TimeGAN and 3D-CNN: Empirical Evidence from Korea and the United States. *Applied Soft Computing*, Article 112748.

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The efficient transformer. *arXiv preprint. arXiv:2001.04451*.

Kumar, S., Rao, A., & Dhochak, M. (2025). Hybrid ML Models for Volatility Prediction in Financial Risk Management. *International Review of Economics & Finance*, Article 103915.

Leng, J., Liu, W., & Guo, Q. (2022). Stock movement prediction model based on gated orthogonal recurrent units. *Intelligent Systems with Applications*, 16, Article 200156.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*.

Lin, Y., Guo, H., & Hu, J. (2013). An SVM-based approach for stock market trend prediction. In *The 2013 international joint conference on neural networks (IJCNN)*.

Lohrmann, C., & Luukka, P. (2019). Classification of intraday S&P500 returns with a Random Forest. *International Journal of Forecasting*, 35(1), 390–407.

- Lu, Y., Zhang, H., & Guo, Q. (2023). Stock and market index prediction using Informer network. *arXiv preprint. arXiv:2305.14382*.
- Mohanty, D., Parida, A. K., & Khuntia, S. S. (2021). Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine. *Applied Soft Computing*, 99, Article 106898.
- Mondal, P., Shit, L., & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2), 13.
- Olorunnimbe, K., & Viktor, H. (2023). Deep learning in the stock market—A systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review*, 56(3), 2057–2109.
- Oyewola, D. O., Kehinde, T. O., Akinwunmi, S. A., & Abdulrahim, A.-m. (2025). Stock Market Prediction with Optimized PLSTM-AL In Smart Urban Cities. *Finance Research Open*, Article 100019.
- Pham, Q., Pham, H., Pham, T., & Tiwari, A. (2025). Revisiting the Role of Investor Sentiment in the Stock Market. *International Review of Economics & Finance*, Article 104089.
- Rokhsatyazdi, E., Rahnamayan, S., Amirinia, H., & Ahmed, S. (2020). Optimizing LSTM based network for forecasting stock market. In *2020 IEEE congress on evolutionary computation (CEC)*.
- Saheed, Y. K., Kehinde, T. O., Ayobami Raji, M., & Baba, U. A. (2024). Feature selection in intrusion detection systems: A new hybrid fusion of Bat algorithm and Residue Number System. *Journal of Information and Telecommunication*, 8(2), 189–207.
- Shah, J., Vaidya, D., & Shah, M. (2022). A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Intelligent Systems with Applications*, 16, Article 200111.
- Singh, S., Parmar, K. S., & Kumar, J. (2021). Soft computing model coupled with statistical models to estimate future of stock market. *Neural Computing and Applications*, 33, 7629–7647.
- Sun, Y., Mutalib, S., Omar, N., & Tian, L. (2024). A novel integrated approach for stock prediction based on modal decomposition technology and machine learning. *IEEE access : Practical innovations, open solutions*.
- Tao, Z., Wu, W., & Wang, J. (2024). Series decomposition Transformer with period-correlation for stock market index prediction. *Expert Systems with Applications*, 237, Article 121424.
- Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys*, 55(6), 1–28.
- Vadlamudi, S. (2017). Stock Market Prediction using Machine Learning: A Systematic Literature Review. *American Journal of Trade and Policy*, 4(3), 123–128.
- Van Thieu, N., Houssein, E. H., Oliva, D., & Hung, N. D. (2025). IntelELM: A python framework for intelligent metaheuristic-based extreme learning machine. *Neurocomputing*, 618, Article 129062.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, C., Chen, Y., Zhang, S., & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. *Expert Systems with Applications*, 208, Article 118128.
- Wang, K. (2025). Multifactor prediction model for stock market analysis based on deep learning techniques. *Scientific Reports*, 15(1), 5121.
- Wei, D., Wang, Z., Qiu, M., Yu, J., Jin, Y., Sha, X., & Ouyang, K. (2025). Multiple objectives escaping bird search optimization and its application in stock market prediction based on transformer model. *Scientific Reports*, 15(1), 5730.
- Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34, 22419–22430.
- Xiong, T., Li, C., & Bao, Y. (2018). Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China. *Neurocomputing*, 275, 2831–2844.
- Yang, Q., Yu, Y., Dai, D., He, Q., & Lin, Y. (2024). Can hybrid model improve the forecasting performance of stock price index amid COVID-19? Contextual evidence from the MEEMD-LSTM-MLP approach. *The North American Journal of Economics and Finance*, 74, Article 102252.
- Yun, K. K., Yoon, S. W., & Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications*, 186, Article 115716.
- Yunneng, Q. (2020). A new stock price prediction model based on improved KNN. In *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*.
- Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*.
- Zhao, C., Cai, J., & Yang, S. (2025). A hybrid stock prediction method based on periodic/non-periodic features analyses. *EPJ Data Science*, 14(1), 1.
- Zhou, F., Zhou, H.-m., Yang, Z., & Yang, L. (2019). EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, 115, 136–151.
- Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*.
- Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*.
- Zhu, W., Dai, W., Tang, C., Zhou, G., Liu, Z., & Zhao, Y. (2024). PMANet: A time series forecasting model for Chinese stock price prediction. *Scientific Reports*, 14(1), Article 18351.