

Does Complexity Improve Forecasting Accuracy? Evidence from Single and Multi-Layer LSTM Models for the BIST 100 Index

Fatih Mangir

Selçuk University

Metehan Ortakarpuz

mortakarpuz@selcuk.edu.tr

Selçuk University

Abdul-Razak Bawa YUSSIF

University of Manitoba

Article

Keywords: BIST 100, LSTM, financial time series forecasting, model complexity, wavelet denoising

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11066715/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Does Complexity Improve Forecasting Accuracy? Evidence from Single and Multi-Layer LSTM Models for the BIST 100 Index

Abstract

Accurate forecasting of the BIST 100 can support informed investment decisions in a volatile emerging market environment. This study compares single layer and multi layer Long Short Term Memory (LSTM) models for one step ahead BIST 100 closing price forecasting using 2,507 daily observations from 4 January 2016 to 31 December 2025. The models combine technical indicators with global and macroeconomic variables, including VIX, USDX, interest rates, unemployment, and consumer sentiment. Following a chronological train-test split, wavelet denoising and Min-Max normalisation were applied before five-day sequence construction, and model performance was evaluated over 30 independent replications using RMSE, MAPE, and Pearson correlation. The results show that single layer models outperform multi layer architectures: the 10-neuron model achieved the lowest mean RMSE among the tested configurations, while the 50 neuron model provides the greatest stability. These findings indicate that greater architectural complexity does not necessarily improve forecasting accuracy and that parsimonious models may enhance generalisation when complexity exceeds the information structure of the data. The results support simpler and more stable model choices for financial forecasting and investment decision making.

Keywords: BIST 100, LSTM, financial time series forecasting, model complexity, wavelet denoising

1. Introduction

Forecasting equity market movements remains a central yet difficult problem in empirical finance. Stock indices are influenced by a combination of historical price dynamics, investor expectations, macroeconomic conditions, monetary policy decisions, and global risk sentiment. This challenge is particularly pronounced in emerging markets, where financial series often display higher volatility, stronger sensitivity to external shocks, and more frequent structural shifts than developed markets. The BIST 100 index, as the main benchmark of Borsa Istanbul, provides a relevant setting for examining this problem because it reflects an emerging equity market exposed to domestic macro financial instability, exchange rate movements, inflationary pressures, and changes in global capital flows. The selection of the BIST 100 is also motivated by its representative role in Türkiye's capital market, as it comprises the largest and most liquid firms listed on Borsa Istanbul and captures the complex interaction between domestic fundamentals and global financial conditions. Its size, sectoral diversity, and sensitivity to both local and external shocks make it an analytically valuable case for testing whether deep learning models can generalise under the uncertainty and structural complexity typical of emerging markets.

Traditional econometric models, including ARIMA type approaches and volatility models, have long been used in financial time series forecasting. The Box–Jenkins framework established a systematic basis for modelling temporal dependence in economic and financial data (Box & Jenkins, 1970). However, financial markets frequently exhibit nonlinear relationships, regime changes, and complex interactions among market, macroeconomic, and global variables. These characteristics may limit the predictive performance of models that rely on restrictive assumptions regarding linearity, stationarity, or prespecified functional forms. Consequently, machine learning and deep learning methods have become increasingly important in financial forecasting.

Among deep learning models, Long Short Term Memory (LSTM) networks have attracted particular attention because of their ability to learn sequential dependencies through gated memory mechanisms. LSTM networks were introduced to overcome the vanishing gradient problem in recurrent neural networks and to retain relevant temporal information over longer sequences (Hochreiter & Schmidhuber, 1997). Prior studies suggest that LSTM based models

can perform effectively in stock market and index forecasting tasks, especially when nonlinear dependencies are present (Fischer & Krauss, 2018; Jiang, 2021; Sezer et al., 2020). Nevertheless, the increasing use of deep learning in finance raises an important methodological question: Does greater architectural complexity necessarily improve forecasting accuracy?

This question is important because a substantial part of the financial machine learning literature tends to associate deeper or more elaborate architectures with superior predictive performance. Yet, in noisy financial environments, more complex models may also become more vulnerable to overfitting. Instead of learning persistent market structure, highly parameterised architectures may capture short lived fluctuations, random noise, or sample specific irregularities. This risk is particularly relevant when the effective sample size is limited by temporal dependence and when the forecast horizon is short. Therefore, the relationship between model complexity and out-of-sample forecasting performance remains an open empirical issue.

This study addresses this issue by comparing single layer and multi-layer LSTM architectures for forecasting the daily closing price of the BIST 100 index. The analysis uses a multivariate dataset covering the period from 4 January 2016 to 31 December 2025, corresponding to 2,507 trading day observations. The dataset includes the BIST 100 closing price as the target variable and combines technical indicators, namely MACD, RSI, and ATR, with global and macroeconomic variables, including VIX, USDX, interest rate, unemployment, and consumer sentiment. This variable structure allows the model to capture price momentum, volatility, global risk perception, dollar strength, and domestic macroeconomic conditions.

The empirical design follows a structured and reproducible forecasting procedure. Missing values in external variables are handled through time-series-consistent imputation. The observations are then divided chronologically into training and test sets, followed by wavelet denoising and Min-Max normalisation. The processed data are converted into supervised learning sequences using a five-day lookback window. Single-layer LSTM models with different neuron sizes and multi-layer LSTM architectures with two and three layer structures are trained and compared. Hyperparameters are selected through grid search, and each model configuration is evaluated over 30 independent replications to account for the stochastic nature of neural network training. Forecasting performance is assessed using RMSE, MAPE, and Pearson correlation, while the statistical significance of model family differences is examined using normality tests and Welch's two sample t-test.

The contribution of this study is threefold. First, it provides a multivariate LSTM based forecasting framework for the BIST 100 index by integrating technical, macroeconomic, and global risk indicators. Second, it shifts the focus from the general usefulness of LSTM models to the more specific issue of architectural choice in financial forecasting. Third, it contributes to the model complexity debate by providing empirical evidence from an emerging market context that deeper LSTM architectures do not necessarily improve predictive performance. In doing so, the study supports a parsimony oriented perspective in financial machine learning, suggesting that model architecture should be aligned with the informational structure of the data rather than expanded mechanically.

The remainder of the paper is organised as follows. Section 2 reviews the literature on financial time series forecasting, LSTM-based stock market prediction, model complexity, and BIST 100 forecasting in the emerging market context. Section 3 describes the dataset, preprocessing procedures, LSTM architectures, hyperparameter tuning strategy, and evaluation framework. Section 4 presents the empirical findings, statistical comparisons, and discussion. The paper concludes with the main implications, limitations, and directions for future research.

2. Literature Review

The literature on financial forecasting has developed from traditional econometric approaches toward machine learning and deep learning methods, while increasingly focusing on the role of data preprocessing, input richness, and model architecture. For the present study, the literature is organised around five interconnected themes: the evolution from econometric to machine learning forecasting, LSTM based financial prediction, hybrid and noise reduction approaches, model complexity and parsimony, and forecasting in the BIST 100 and other emerging markets. Rather than treating these streams as independent bodies of evidence, the review compares their findings to identify where existing evidence converges, where it remains contradictory, and where a specific empirical gap emerges.

2.1. Financial Time Series Forecasting: From Econometric Models to Machine Learning

Financial forecasting has traditionally been grounded in the assumption that historical information contains limited but potentially useful predictive structure. The efficient market hypothesis provides the theoretical starting point by arguing that asset prices incorporate available information, thereby limiting systematic predictability (Fama, 1970). Nevertheless, the persistence of temporal dependence, volatility clustering, and nonlinear behaviour has motivated a substantial forecasting literature. Box and Jenkins (1970) established the classical framework for autoregressive and moving average modelling, while Engle (1982) and Bollerslev (1986) extended econometric forecasting to explicitly account for conditional heteroskedasticity and volatility persistence. Diebold and Mariano (1995) subsequently formalised statistical comparison of competing forecasts, reinforcing the importance of evaluating predictive performance rather than relying solely on model specification.

These traditional approaches remain important because of their interpretability and well established statistical foundations, but their performance may depend on assumptions concerning linearity, stationarity, and the stability of underlying relationships. The development of machine learning has therefore introduced more flexible approaches capable of modelling nonlinear relationships and interactions among multiple predictors. However, empirical comparisons do not provide consistent evidence that machine learning or deep learning models always dominate conventional methods. Kobiela et al. (2022), for example, find that ARIMA can outperform LSTM when historical prices constitute the principal information set and longer forecasting horizons are considered. Kirelli (2024) similarly reports mixed results across evaluation metrics. In contrast, Chowdhury et al. (2024) find stronger performance for deep learning models relative to ARIMA across NSE and NYSE datasets, while Osman and Otaibi (2025) report greater robustness of LSTM and Transformer models during the COVID-19 period. Chen et al. (2025) further demonstrate that predictive performance varies across assets with different volatility profiles and that no single model is universally dominant.

Taken together, these studies suggest that the transition from econometric forecasting to machine learning should not be interpreted as a simple replacement of one methodological paradigm by another. Rather, model suitability appears to depend on the characteristics of the data, forecasting horizon, information set, and market environment. This observation is directly relevant to the present study because it shifts attention from the general question of whether sophisticated models can forecast financial series toward the more specific question of how much model complexity is appropriate for a given information structure.

2.2. Deep Learning and LSTM Models in Stock Market Forecasting

The development of LSTM networks provided an important extension to recurrent neural network approaches for sequential prediction. Hochreiter and Schmidhuber (1997) introduced LSTM to address the vanishing gradient problem and improve the ability of recurrent networks to retain information over extended temporal intervals. Subsequent work established recurrent architectures as useful tools for sequence modelling (Graves, 2012), while deep learning

provided more flexible mechanisms for learning nonlinear and hierarchical representations from complex data (Heaton, 2018). These characteristics are particularly relevant to financial markets, where observations are temporally dependent and relationships among variables may be nonlinear.

Empirical financial studies have generally supported the usefulness of LSTM, although the strength of the evidence varies across markets and model designs. Nelson et al. (2017) report promising short term directional prediction using LSTM with historical prices and technical indicators in the Brazilian market. Bao et al. (2017) combine wavelet transformation, stacked autoencoders, and LSTM and demonstrate that integrating feature extraction and sequential learning can improve financial forecasting performance. Fischer and Krauss (2018), using S&P 500 constituent stocks, find that LSTM outperforms several memory free benchmark models in predicting directional movements. These findings are consistent with the broader reviews of Sezer et al. (2020) and Jiang (2021), which identify recurrent and deep learning architectures as major methodological streams in financial forecasting.

More recent research, however, increasingly compares LSTM with alternative deep learning architectures rather than examining LSTM in isolation. Xiao et al. (2024) compare LSTM, GRU, and Transformer models and report strong LSTM performance in stock trend prediction. Fischer et al. (2024) show that Transformer models can outperform LSTM benchmarks when sufficiently rich multivariate and cross sectional information is available, indicating that architectural advantages depend partly on the information supplied to the model. Bennaton et al. (2025) report favourable performance for LSTM based recurrent models across several major equity indices, while Wang et al. (2026) demonstrate that multimodal fusion of textual and market information can further improve prediction. The literature therefore establishes LSTM as a credible forecasting architecture but simultaneously suggests that predictive performance is conditional on input richness and model design.

The key implication for the present research is that evidence supporting LSTM does not, by itself, establish that deeper or more complex LSTM architectures should perform better than simpler ones. Much of the literature compares different model families, whereas the effect of architectural depth within the LSTM family receives comparatively less direct attention. This distinction motivates the present study's explicit comparison of single layer and multi layer LSTM configurations.

2.3. Hybrid, Noise Reduction and Feature Enhanced Forecasting Models

Financial time series contain both potentially informative structures and substantial noise, which has encouraged researchers to combine forecasting architectures with signal decomposition, feature extraction, and preprocessing techniques. Zhang (2003) provides an early theoretical basis for hybrid forecasting by arguing that linear and nonlinear components of a time series may be modelled more effectively through complementary approaches. Wavelet based methods subsequently became particularly relevant because they allow a signal to be represented across different frequency components. Donoho and Johnstone (1994) established the statistical basis of wavelet shrinkage, while Renaud et al. (2005) demonstrated its potential for filtering and prediction.

Financial applications increasingly combine these techniques with neural networks. Kim and Won (2018), for example, integrate LSTM with GARCH type models for stock index volatility forecasting, illustrating how volatility modelling and sequential learning can complement each other. Bao et al. (2017) similarly combine wavelet transformation with deep feature extraction and LSTM. More recent evidence continues this direction: Yan (2025) finds that CNN-LSTM can outperform standalone LSTM and ARIMA-LSTM models for S&P 500 closing price prediction, while Ziółkowski and Wiecezyński (2025) report that discrete wavelet transform

enhanced LSTM models can reduce out of sample errors for global stock indices. Rehman and Gul (2026) provide evidence from Pakistan's KSE-100 that Wavelet LSTM can outperform linear and standalone artificial intelligence models.

Feature enhancement has also expanded beyond signal denoising. Sarikoc and Celik (2024) combine PCA, ICA, and LSTM to reduce dimensionality and improve predictive performance, while Wang et al. (2026) incorporate textual and market information through multimodal deep fusion. These studies collectively indicate that what information enters a model and how that information is represented may be as important as the model architecture itself. At the same time, the evidence does not establish that increasingly elaborate preprocessing and architecture should always be combined. Rather, it points toward a more conditional relationship between preprocessing, information structure, and predictive performance.

This distinction is important for the present study. Wavelet denoising is employed here as a preprocessing mechanism rather than as a justification for increasing network depth. The study therefore separates two questions that are often combined in hybrid forecasting research: whether preprocessing can improve the quality of the forecasting signal, and whether additional LSTM architectural complexity improves prediction after such preprocessing has already been applied.

2.4. Model Complexity, Overfitting and Parsimony in Financial Forecasting

The question of whether greater model complexity improves prediction is closely related to the statistical learning trade-off between flexibility and generalisation. Hastie et al. (2009) explain that increasing model flexibility can reduce bias but may also increase variance and overfitting when a model begins to capture noise rather than systematic structure. This concern is particularly relevant to financial time series because market observations are noisy, temporally dependent, and exposed to structural changes. Bergmeir and Benítez (2012) consequently emphasise that model evaluation for time series prediction must account for temporal dependence and avoid validation procedures that can produce misleading estimates of predictive performance.

Empirical forecasting evidence reinforces this caution. Makridakis et al. (2018) show that machine learning methods do not automatically outperform traditional statistical models and may require substantially greater computational resources. The M4 Competition further demonstrated that forecasting performance depends on model suitability and robust benchmarking rather than algorithmic sophistication alone (Makridakis et al., 2020). Similar patterns appear in financial forecasting. Kobiela et al. (2022) report that ARIMA can outperform LSTM under particular information and horizon conditions, while Zein et al. (2024) find that N-BEATS can outperform LSTM in Indonesian stock prediction while also providing advantages in interpretability and training efficiency. Fischer et al. (2024) likewise show that more complex Transformer architectures require richer multivariate and cross sectional information to achieve an advantage over LSTM.

Recent evidence further suggests that model superiority can depend on the evaluation criterion. Hartanto et al. (2026) find that attention enhanced GRU improves directional accuracy without necessarily reducing price level prediction error, while Bulus (2025) highlights the importance of balancing predictive accuracy with computational cost when comparing RNN, LSTM, and GRU architectures. Chen et al. (2025) similarly show that model performance varies across assets with different volatility profiles. These findings challenge a common implicit assumption in financial machine learning: that increasing the number of layers, parameters, or architectural mechanisms necessarily produces a better forecast.

The literature therefore provides a strong theoretical and empirical basis for a parsimony oriented approach, but an important limitation remains. Most studies examine complexity indirectly by comparing different algorithms, rather than systematically varying architectural depth within the same LSTM framework while holding the forecasting problem and information set constant. Consequently, the specific relationship between LSTM architectural complexity, forecasting accuracy, and performance stability remains insufficiently established, particularly for emerging market equity indices.

2.5. BIST 100 and Emerging Market Forecasting Context

Emerging equity markets provide an especially relevant environment for examining forecasting models because they are exposed to macroeconomic instability, exchange rate movements, capital flow fluctuations, liquidity conditions, and structural changes. These characteristics can alter the relationships between financial variables and therefore make model generalisation particularly demanding. Within this context, the BIST 100 represents an important case because its dynamics are influenced by both domestic macroeconomic conditions and international financial developments. Demir (2019), for example, identifies significant relationships between BIST 100 fluctuations and economic growth, exchange rates, portfolio investment, foreign direct investment, interest rates, and crude oil prices. Gur (2024) and Ustuner and Cankal (2025) likewise demonstrate the relevance of macro financial variables in BIST 100 forecasting.

The Borsa Istanbul literature has also produced mixed evidence concerning the relative performance of traditional and machine learning approaches. Yigit et al. (2020) compare ARIMA, LSTM, and GRU for BIST 30, BIST 50, and BIST 100 indices and find that ARIMA performs better according to RMSE, demonstrating that deep learning does not automatically dominate conventional models. In contrast, Gur (2024) reports stronger performance for CNN among several machine learning and deep learning algorithms in BIST 100 prediction. Ustuner and Cankal (2025) find that LSTM produces lower errors than Logit and Probit models, particularly around crisis periods. These contrasting findings are consistent with the broader forecasting literature: predictive superiority depends on the model, information set, evaluation criterion, and market conditions.

Other studies have broadened the BIST forecasting problem through alternative information sources and preprocessing strategies. Kilimci and Duvar (2020) combine Turkish financial news, Twitter, and user comments with deep learning models for BIST 100 direction forecasting, while Akyildirim et al. (2023) demonstrate the predictive potential of high frequency Borsa Istanbul data. Sivri and Ustundag (2024) employ feature selection and ensemble learning for daily stock movement prediction, and Akturk et al. (2025) examine real time BIST 100 forecasting under volatility and uncertainty. Oksak et al. (2025) further show that wavelet enhanced configurations can reduce forecast errors across BIST 100 and other major indices. Evidence from other emerging markets points in a similar direction: Mazzouz and Elmoufidi (2025) find LSTM to outperform CNN for Morocco's MASI, while Rehman and Gul (2026) report strong performance from Wavelet LSTM for Pakistan's KSE-100.

These findings establish that the BIST 100 is an appropriate setting for evaluating data rich and nonlinear forecasting models, but they also reveal a fragmented evidence base. Existing studies differ substantially in their target definition, input variables, preprocessing procedures, model families, evaluation metrics, and market periods. More importantly, the existing BIST literature largely asks which forecasting model performs best, rather than systematically examining whether increasing depth within the LSTM family improves forecasting performance. This distinction provides the basis for the research gap addressed in the present study.

2.6. Literature Gap

The preceding literature reveals three related but unresolved issues. First, although machine learning and deep learning models have become increasingly prominent in financial forecasting, comparative evidence does not support the assumption that more sophisticated models consistently outperform traditional or simpler alternatives. Second, LSTM has demonstrated considerable forecasting potential, particularly when combined with technical, macroeconomic, or global information, but much of the literature evaluates LSTM against other model families rather than examining the consequences of increasing architectural depth within LSTM itself. Third, research on the BIST 100 has produced mixed findings across ARIMA, LSTM, CNN, ensemble, and other machine learning approaches, while studies incorporating wavelet denoising and richer information sets have primarily focused on improving individual forecasting architectures rather than isolating the effect of architectural complexity.

The present study addresses these gaps by holding the forecasting problem and multivariate information environment constant while systematically comparing single layer and multi layer LSTM architectures for one step ahead BIST 100 closing price prediction. Unlike studies that primarily ask whether LSTM outperforms another algorithm, the analysis directly examines whether adding layers and increasing architectural complexity produces lower forecasting errors and more stable performance. The study further evaluates this question after wavelet based denoising and multivariate feature integration, thereby allowing architectural complexity to be assessed within a controlled and information rich forecasting framework.

This approach is important because a deeper architecture may increase representational capacity without necessarily improving generalisation. By evaluating multiple single and multi layer configurations across 30 independent stochastic replications and examining RMSE, MAPE, Pearson correlation, and statistical differences between model families, the study contributes evidence on both forecasting accuracy and performance stability. The resulting evidence helps clarify whether the additional complexity of stacked LSTM architectures provides a meaningful predictive advantage for an emerging market equity index or whether a more parsimonious architecture is better aligned with the informational structure of the data.

3. Data and Methodology

This section describes the data, preprocessing procedures, LSTM modelling strategy, hyperparameter selection, and statistical evaluation framework used to forecast the BIST 100 index. The methodological workflow was designed to preserve temporal ordering, minimise information leakage, and provide a consistent basis for comparing single-layer and multi-layer LSTM architectures. Figure 2 summarises the main stages of the empirical procedure.

3.1 Dataset Description and Sources

This study utilises a multivariate daily time series dataset spanning from January 4, 2016, to December 31, 2025, yielding 2,507 trading day observations after accounting for non trading days and minor gaps. The dataset was assembled from multiple publicly accessible financial and economic data sources to capture the major market, macroeconomic, and technical forces driving the BIST 100 index. The target variable for prediction is the daily closing price of the BIST 100 index, denoted Close. A summary of all input features is provided in Table 1.

Table 1. Variables Considered in the BIST 100 Prediction Framework

Feature	Notation	Description
BIST 100 Open Price	Open	Daily opening price of the BIST 100 index
BIST 100 Close Price	Close	Daily closing price (target variable for prediction) BIST 100 index
Volatility Index	VIX	CBOE Volatility Index; proxy for global market fear

Feature	Notation	Description
Effective Interest Rate	TR_INT	Turkish short term interest rate benchmark
Unemployment Rate	TR_UNEMP	Turkish national unemployment rate (%)
Consumer Confidence Index	TR_CCI	Turkish Consumer Confidence (Sentiment) Index
Dollar Index	USDX	US dollar strength relative to a basket of currencies
MACD	MACD	Moving Average Convergence Divergence (momentum indicator)
Relative Strength Index	RSI	Momentum oscillator measuring speed and change of price
Average True Range	ATR	Measure of market volatility derived from price range

Note: MACD, RSI, and ATR are technical indicators derived from BIST 100 price data. All other features are external economic or market variables.

The BIST 100 open and close prices constitute the primary market data. Three widely used technical indicators, namely MACD, RSI, and ATR, are derived from the BIST 100 price series itself. These indicators are commonly employed in quantitative finance for capturing momentum, trend strength, and volatility at the instrument level (Achelis & Achelis, 2001). External macroeconomic and market variables include the CBOE Volatility Index (VIX), which measures implied volatility of S&P 500 options and serves as a proxy for global investor risk sentiment (Whaley, 2000); the Turkish Effective Interest Rate (TR_INT) as a measure of monetary policy in Turkey; the Turkish Unemployment Rate (TR_UNEMP); the Turkish Consumer Sentiment Index (TR_CCI) as a forward looking gauge of economic confidence; and the US Dollar Index (USDX), which reflects the relative strength of the US dollar. These macroeconomic variables were included to capture cross market and systemic economic effects that have been shown to influence emerging market equity indices (Bekiros et al., 2016). Lower-frequency macroeconomic variables were aligned to daily trading observations using the most recent value available as of each trading date. Values were not backfilled from future releases, thereby preserving the information set that would have been available at each forecast origin

The dataset encompasses several significant economic episodes, including the COVID-19 pandemic induced market crash and recovery (2020–2021), periods of extreme Turkish lira depreciation and high inflation, and shifting global interest rate cycles. The breadth of this window, combined with the diversity of features, ensures that models must learn to generalise across a wide range of market regimes. The multivariate structure was selected because BIST 100 movements may reflect not only their own historical dynamics but also changes in global risk sentiment, monetary conditions, Exchange rate conditions and market based technical signals.

3.2 Exploratory Data Analysis

Preliminary inspection of the dataset using descriptive statistics revealed several important characteristics. As shown in Table 2, the Close price exhibits extreme right skewness, with a mean of approximately 3,567 points but a range spanning from 686 to 11,530 points, reflecting the dramatic growth and volatility of the BIST 100 over the study period. The ATR and TR_INT also display wide ranges, consistent with the volatile market and monetary policy environments covered by the dataset.

Table 2. Descriptive Statistics of the Dataset Features (TR_CCI excluded for space)

Statistic	Close	MACD	RSI	ATR	VIX	USDX	TR_INT	TR_UNEMP
Count	2,507	2,507	2,507	2,507	2,448	2,500	2,507	2,507
Mean	3566.74	28.98	56.32	74.58	18.49	98.51	14.58	10.87
Std Dev	3628.29	94.73	18.44	79.80	7.35	5.03	12.65	1.87
Min	685.68	-271.21	6.93	7.09	9.14	88.59	2.79	7.60
25th Pctl	985.03	-8.05	42.81	15.68	13.53	94.72	4.68	9.30
Median	1370.70	9.82	55.65	23.97	16.67	97.82	7.66	10.60
75th Pctl	6123.95	35.98	69.95	144.48	21.49	102.40	26.00	12.50
Max	11529.81	409.71	100.00	378.60	82.69	114.11	42.96	15.10

A correlation analysis was conducted to understand feature interdependencies. The correlation heatmap revealed strong positive correlations between the Close price and both the TR_INT ($r = 0.98$) and ATR ($r = 0.91$), and a positive correlation with the USDX ($r = 0.59$). Negative correlations were observed between the Close price and both the Unemployment Rate ($r = -0.75$) and the TR_CCI ($r = -0.24$). The MACD exhibited a moderate positive correlation with RSI ($r = 0.52$). These patterns are broadly consistent with economic theory: rising interest rates in the US context co-occurred with the post pandemic monetary tightening cycle, while the strong ATR-Close correlation reflects that absolute price range naturally scales with index level. The Open price, being nearly perfectly correlated with Close ($r \approx 1$), was excluded from the feature set following the initial exploratory analysis to avoid redundancy. The final feature set thus comprises nine variables: Close, MACD, RSI, ATR, VIX, USDX, TR_INT, TR_UNEMP, and TR_CCI. Figure 1 below shows the correlation analysis performed.

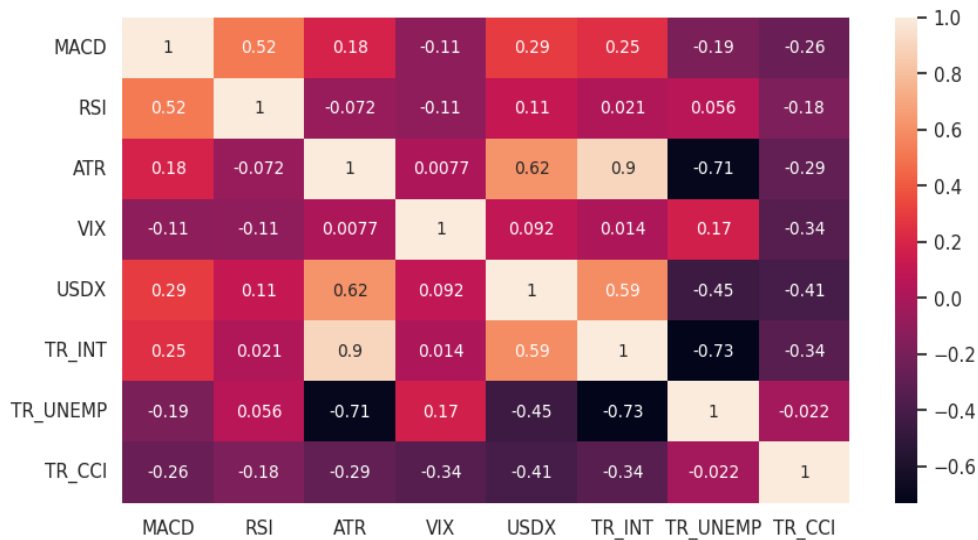


Figure 1. Correlation Matrix of final feature set comprising nine variables: Close, MACD, RSI, ATR, VIX, USDX, TR_INT, TR_UNEMP, and TR_CCI

These exploratory findings supported the use of a nonlinear sequential forecasting framework while also indicating the need for appropriate preprocessing to control scale differences, missing observations and high frequency noise.

3.3 Data Preprocessing

This section presents the data, preprocessing procedures, LSTM modelling strategy, hyperparameter optimisation and statistical evaluation framework used in the study. This

section describes the preprocessing and modelling procedures used to preserve temporal ordering, minimise information leakage, and provide a consistent basis for comparing single-layer and multi-layer LSTM architectures. Following data collection, exploratory analysis and missing-value treatment, the observations were divided chronologically into training and test sets. Wavelet denoising and Min-Max normalisation were then applied before sequence construction and model training. Figure 2 summarises the methodological workflow, which is described in detail in the following subsections.

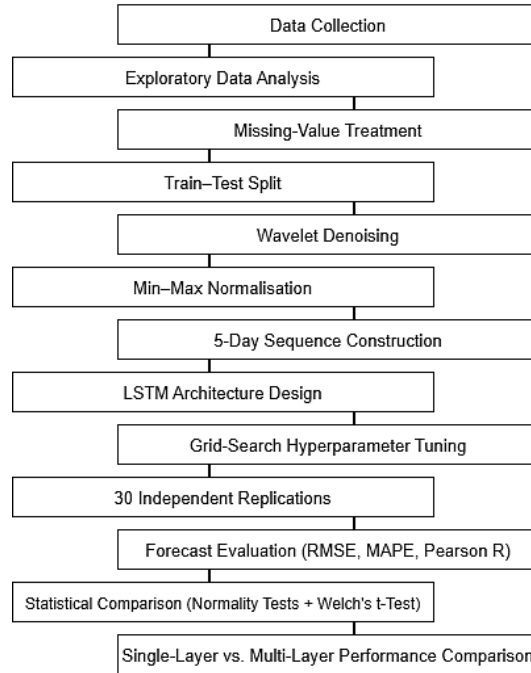


Figure 2. Methodological workflow of the study

3.3.1 Missing Value Treatment

Missing values were present in the VIX (59 observations) and USDX (7 observations) series, attributable to non overlapping trading calendars between the Turkish and US markets. These were imputed using forward fill (last observation carried forward), with any remaining leading missing values handled by backward fill. This approach preserves the temporal structure of the data and is consistent with standard practice for financial time series imputation (Moritz & Bartz-Beielstein, 2017).

3.3.2 Data Splitting

The dataset was divided chronologically into training and test sets, with 80% of the observations assigned to training and the remaining 20% reserved for out-of-sample testing. This temporal split was applied before subsequent signal processing and scaling procedures to ensure that information from the test period did not influence model development. During hyperparameter tuning, 20% of the training data was further withheld as a validation set, resulting in an effective train/validation/test structure of approximately 64%/16%/20% of the full sample.

3.3.3 Wavelet Denoising

Following the chronological split, wavelet preprocessing was implemented without using information from future observations relative to each forecast origin. The denoising procedure was therefore applied in a leakage-safe manner before scaling and sequence construction. The Haar wavelet was combined with the VisuShrink thresholding method in soft mode, as implemented in the scikit-image library (Donoho & Johnstone, 1994; van der Walt et al., 2014).

Applying the transformation separately to each segment prevented information from the test period from entering the training-stage preprocessing. The denoised Close series was then used as the target variable, while the remaining predictors retained their original values.

3.3.4 Normalisation

After wavelet denoising, Min-Max normalisation was applied to the model inputs. The scaling parameters were estimated exclusively from the training data and subsequently applied unchanged to the validation and test sets. This procedure ensured that the scale of the test observations did not influence model training or hyperparameter selection and therefore reduced the risk of information leakage. Predictions were inverse-transformed to the original price scale using the training-set scaling parameters before the final evaluation.

3.3.5 Sequence Construction

After normalisation, the scaled observations were converted into supervised learning sequences for LSTM input. A five-trading-day lookback window was used, with each input sample containing the lagged values of all nine features over the preceding five trading days and the target corresponding to the Close price at the next time step. This window size was chosen to capture short term market dependencies while maintaining a tractable sequence length consistent with related literature on equity index forecasting with LSTMs (Fischer & Krauss, 2018).

3.4 LSTM Model Architecture

LSTM was selected because the research question concerns whether increasing recurrent architectural depth improves one step ahead forecasting performance in a temporally dependent financial series. Unlike conventional feed forward models, LSTM networks retain information through gated memory mechanisms and can therefore represent temporal dependencies in sequential observations (Hochreiter & Schmidhuber, 1997; Graves, 2012). This makes LSTM suitable for the present multivariate forecasting problem, where recent price dynamics, technical indicators and macro financial variables are jointly used to predict the next closing value. At each time step t , an LSTM unit updates its hidden state $\mathbf{h}_t$ and cell state $\mathbf{c}_t$ through gating mechanisms — an input gate $\mathbf{i}_t$, a forget gate $\mathbf{f}_t$, an output gate $\mathbf{o}_t$, and a cell candidate $\tilde{\mathbf{c}}_t$ — governed by the following equations:

$$\begin{aligned} \mathbf{f}_t &= \sigma(W_f \cdot [h_{\{t-1\}}, x_t] + b_f), & \mathbf{i}_t &= \sigma(W_i \cdot [h_{\{t-1\}}, x_t] + b_i) \\ \tilde{\mathbf{c}}_t &= \tanh(W_c \cdot [h_{\{t-1\}}, x_t] + b_c), & \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{\{t-1\}} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \\ \mathbf{o}_t &= \sigma(W_o \cdot [h_{\{t-1\}}, x_t] + b_o), & \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \end{aligned}$$

where σ denotes the sigmoid activation function, $\tanh$ is the hyperbolic tangent function, $\odot$ denotes element wise multiplication, and W and b are the weight matrices and bias vectors learned during training. All LSTM models were implemented using TensorFlow 2.x and Keras.

The main advantage of LSTM in this setting is its ability to model nonlinear temporal relationships while retaining information across sequential observations. However, LSTM models are also computationally intensive and sensitive to architectural and optimisation choices, and increasing the number of layers and parameters may increase overfitting risk. These considerations motivate the explicit comparison of single layer and multi layer architectures in this study.

Two architecture families were investigated: single layer (SL) LSTMs and multi layer (ML) LSTMs. The SL family included networks with 10, 30, 50, 100, 150 and 200 neurons, whereas the ML family included two layer configurations of (10, 5), (20, 10), (50, 20), (100, 50) and (150, 100), together with a three layer (100, 50, 20) configuration. This design allowed model

capacity to be increased systematically while keeping the input variables, lookback window and evaluation framework constant. In both families, all LSTM layers feed into a single Dense output neuron with a linear activation function, appropriate for regression. For multi layer configurations, the `return_sequences=True` argument is set on all intermediate LSTM layers to propagate the full sequence output to the subsequent LSTM layer, while the final LSTM layer outputs only the last hidden state. No dropout regularisation was applied in the final configurations. Instead, early stopping was used to limit unnecessary training once improvements plateaued.

The build model function accepts an arbitrary list of layer sizes, enabling both single layer configurations (e.g., [50]) and deep stacked configurations (e.g., [100, 50]). All models were compiled with Mean Squared Error (MSE) as the loss function, consistent with the regression objective of minimising prediction error on the scaled Close price.

3.5 Hyperparameter Tuning

Hyperparameter tuning was conducted to reduce the risk that differences between model architectures were driven by arbitrary optimisation settings rather than architectural structure. The optimisation procedure was therefore applied separately to each model configuration using the same predefined search space and validation criterion. The search space covered three optimisation algorithms Adam (Kingma & Ba, 2015), Adagrad (Duchi et al., 2011), and Nadam (Dozat, 2016), three learning rates (0.1, 0.01, and 0.001), and three batch sizes (4, 8, and 16), yielding 27 combinations per architecture. The grid search approach was preferred because it provides a transparent and reproducible comparison of the predefined optimisation settings, while the validation based selection criterion prevents the test set from influencing hyperparameter selection. For each combination, 10 stochastic replicates were trained for up to 50 epochs with an early stopping callback monitoring validation loss (patience = 5). The average RMSE on the validation set across all replicates was used as the selection criterion, and the combination producing the lowest average validation RMSE was retained for full scale training. Table 3 summarises the hyperparameter search space, and Tables 4 and 5 present the best identified configurations for the single layer and multi layer model families, respectively.

Table 3. Hyperparameter Search Space and General Configuration

Hyperparameter	Values Tested	Best Configuration(s)
Optimizers	Adam, Adagrad, Nadam	Adagrad (most models); Adam (10N SL)
Learning Rates	0.1, 0.01, 0.001	0.01 (dominant); 0.001 (200N, 100-50-20N)
Batch Sizes	4, 8, 16	Model-specific (see Tables 4 & 5)
Epochs (tuning)	50 (with early stopping, patience=5)	—
Epochs (full training)	100 (with early stopping, patience=5)	—
Time Step (lookback)	5 trading days	Fixed at 5
Train/Test Split	80% / 20%	Fixed
Validation Split (tuning)	20% of training set	Fixed
Replicates	10 (tuning); 30 (full training)	—

Table 3 summarises the common hyperparameter search space and the fixed training settings used throughout the optimisation procedure. The same candidate optimisers, learning rates and batch sizes were evaluated across architectures, while the final configuration was selected separately for each model according to validation RMSE. This design ensured that the

subsequent comparison of model architectures was based on independently optimised configurations rather than a common set of potentially suboptimal hyperparameters.

Table 4. Best Hyperparameters Identified for Single Layer LSTM Models

Neurons	Optimizer	Learning Rate	Batch Size	Elapsed Time (s)
10	Adam	0.01	16	11.33
30	Adagrad	0.01	8	86.59
50	Adagrad	0.01	4	144.62
100	Adagrad	0.01	16	45.21
150	Adagrad	0.01	16	45.60
200	Adagrad	0.001	4	170.65

Table 4 reports the best performing hyperparameter configuration for each single-layer architecture. Although Adagrad was selected for most single layer models, the 10-neuron configuration performed best with Adam. The learning rate was generally 0.01, whereas the largest 200-neuron model required a lower rate of 0.001. Batch size selection also varied across architectures, indicating that optimisation settings interacted with model capacity rather than remaining constant across configurations.

A separate tuning process was conducted for the multi layer architectures because their optimisation landscape differs from that of single layer networks as model depth and parameterisation increase. The resulting configurations are reported in Table 5.

Table 5. Best Hyperparameters Identified for Multi Layer LSTM Models

Architecture	Optimizer	Learning Rate	Batch Size	Elapsed Time (s)
(10, 5)	Adagrad	0.1	4	99.95
(20, 10)	Adagrad	0.01	16	59.99
(50, 20)	Adagrad	0.01	16	59.49
(100, 50)	Adagrad	0.01	16	59.04
(150, 100)	Adagrad	0.01	16	59.81
(100, 50, 20)	Adagrad	0.001	8	135.34

The tuning results indicate that Adagrad with a learning rate of 0.01 was selected for most configurations across the two model families. The main exceptions were the 10-neuron single layer model, for which Adam was selected, and the largest single and multi layer configurations, for which a learning rate of 0.001 was preferred. This pattern suggests that optimisation settings interacted with model capacity, providing a further reason for tuning hyperparameters separately for each architecture rather than imposing a common configuration.

3.6 Full Scale Model Training and Evaluation

After hyperparameter selection, each model configuration moved into full scale training and evaluation across 30 independent stochastic replicates. LSTM training depends on random weight initialisation and stochastic gradient descent, so single runs vary in result. Running 30 replicates lets you map the sampling distribution of performance metrics, identify the best model with confidence, and run formal statistical tests. Final models trained on the same 64% training subset used during hyperparameter tuning, with the same 16% validation subset held out for early stopping. Training ran for up to 100 epochs, with early stopping based on

validation loss and a patience of 5 epochs. The test set played no role in training, hyperparameter selection, or early stopping at any stage.

Each of the 30 replicates used a new random weight initialisation and a new random seed for stochastic gradient descent. The selected hyperparameters stayed fixed across all 30 replicates. This design separates variation caused by stochastic training from variation caused by architecture or hyperparameter choice, and gives a stronger basis for comparing model families than a single training run offers.

Performance was measured on the held out test set with three complementary metrics. Root Mean Squared Error (RMSE) penalises large errors and reports in the same units as the target variable, BIST 100 index points. Mean Absolute Percentage Error (MAPE) gives a scale independent measure of relative forecast accuracy. Pearson Correlation Coefficient (R) measures the linear concordance between true and predicted values. Together these three metrics capture forecast error size, relative accuracy, and linear association between actual and predicted values. For each model configuration, the results report the mean, standard deviation, minimum, and maximum of each metric across the 30 replicates.

3.7 Statistical Comparison of Model Families

To rigorously assess whether the observed performance difference between the lowest-error single-layer configuration and the representative multi-layer benchmark is statistically significant, a two stage statistical analysis was conducted. First, the normality of the RMSE distributions (across 30 replicates) for each model was assessed using D'Agostino and Pearson's omnibus normality test (D'Agostino et al., 1990). Second, given the potential for unequal variances between the two distributions, Welch's two sample t-test was preferred because the preliminary results indicated that the two model families could differ in the variability of their RMSE distributions, making the equal variance assumption of the conventional Student's t-test inappropriate. (Welch, 1947). The null hypothesis is that the mean RMSE of the best single layer model equals the mean RMSE of the best multi layer model, the alternative hypothesis is that there is a statistically significant difference. A significance level of $\alpha = 0.05$ was adopted. This procedure allowed the comparison to focus on differences in mean forecasting error while remaining robust to unequal variances between the model distributions.

4. Results and Analysis

This section presents the empirical findings obtained from the single layer and multi layer LSTM models and evaluates whether increasing architectural complexity improves the forecasting performance of the BIST 100 index. The analysis first reports the predictive accuracy of each model family, then compares their performance across RMSE, MAPE and Pearson correlation metrics, and finally assesses the statistical significance of the observed differences through formal testing.

4.1 Single Layer LSTM Model Performance

Six single layer LSTM configurations, with 10, 30, 50, 100, 150, and 200 neurons, were trained across 30 independent replicates using the best hyperparameters identified during tuning. Table 6 summarises the test set performance, while Figures 3 and 4 show the distribution and average performance of the models across the three evaluation metrics.

Table 6. Test Set Performance of Single Layer LSTM Models (30 replicates). Asterisk (*) marks the best mean RMSE; dagger (†) marks the most stable (lowest Std) viable model.

Neurons	Mean RMSE	Std RMSE	Min RMSE	Max RMSE	Mean MAPE (%)	Mean R	Std R
10*	231.32	44.88	165.46	362.11	1.87	0.9797	0.0030

Neurons	Mean RMSE	Std RMSE	Min RMSE	Max RMSE	Mean MAPE (%)	Mean R	Std R
30	305.53	35.81	239.47	373.30	2.50	0.9614	0.0080
50†	240.09	16.97	205.88	269.71	1.92	0.9713	0.0021
100	275.82	25.68	227.13	326.74	2.24	0.9599	0.0055
150	257.60	18.71	227.43	299.98	2.07	0.9622	0.0036
200	454.99	35.23	391.59	515.62	3.75	0.9099	0.0125

Note. R = Pearson correlation coefficient between true and predicted test set values. Asterisk (*) = lowest mean RMSE; Dagger (†) = lowest standard deviation among comparable models.

The results reveal a clear non monotonic relationship between model size and forecasting performance. The smallest single layer architecture delivered the strongest average predictive performance, while increasing the number of neurons did not produce systematic gains. The 50-neuron configuration was particularly notable for its training stability, whereas performance deteriorated markedly at the highest model size. Overall, increasing model capacity beyond a relatively compact architecture did not improve generalisation in the present forecasting task.

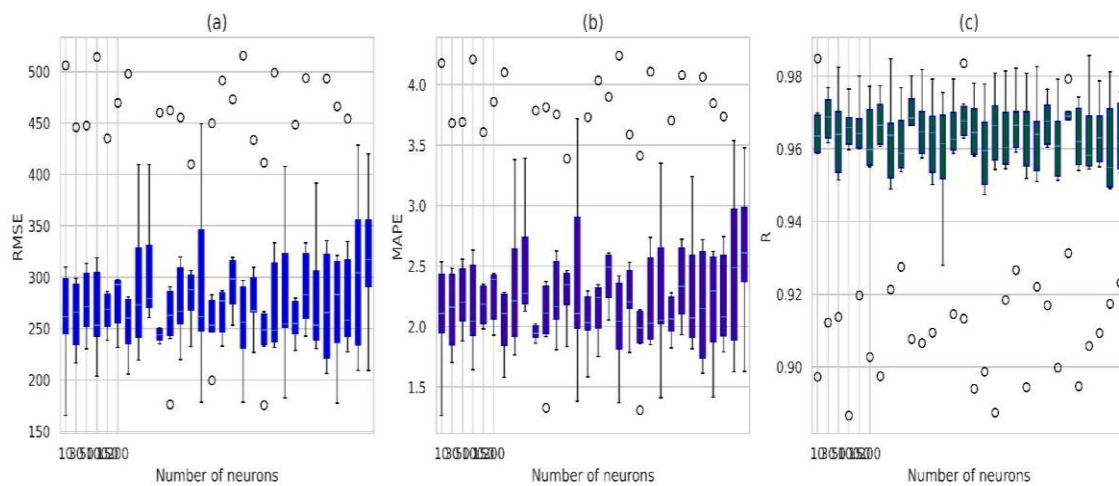


Figure 3. Distribution of (a) RMSE, (b) MAPE, and (c) Pearson R across 30 replicates for the six single layer LSTM configurations (10, 30, 50, 100, 150, 200 neurons). Outlier replicates are shown as open circles.

Figure 3 provides complementary evidence on the variability of model performance across replications. The 50-neuron configuration stands out for its comparatively narrow error distribution, indicating greater stability than the other single layer configurations. In contrast, the 10-neuron model combines the strongest average performance with greater sensitivity across replications. This distinction is important because forecasting accuracy and training stability do not necessarily identify the same architecture.

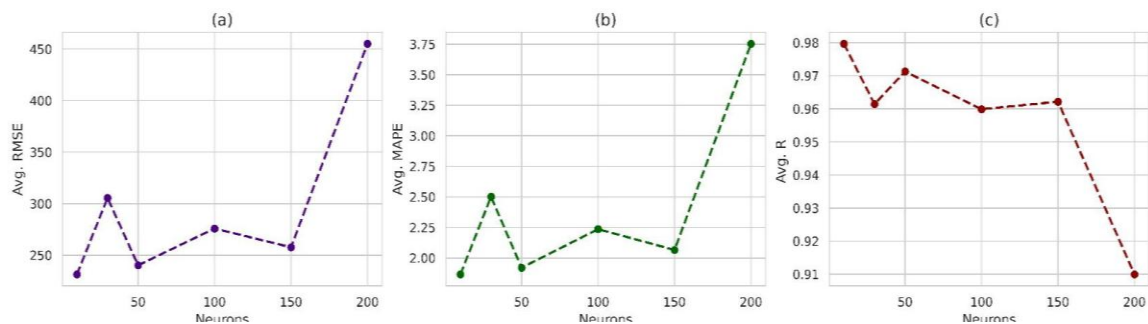


Figure 4. Average (a) RMSE, (b) MAPE, and (c) Pearson R as functions of neuron count for the six single-layer LSTM models. Clear non-monotonic trends are visible, with the 10- and 50-neuron models performing best and the 200-neuron model degrading markedly.

Figure 4 highlights the non-linear relationship between network width and forecasting performance. The intermediate architectures remain broadly competitive, but increasing the number of neurons beyond the compact configurations does not yield further gains. The sharp deterioration at the largest network size is particularly notable, as it is accompanied by weaker performance across all three evaluation metrics. The pattern is consistent with a capacity generalisation trade-off, suggesting that additional parameters may become detrimental when they exceed the informational structure of the forecasting problem.

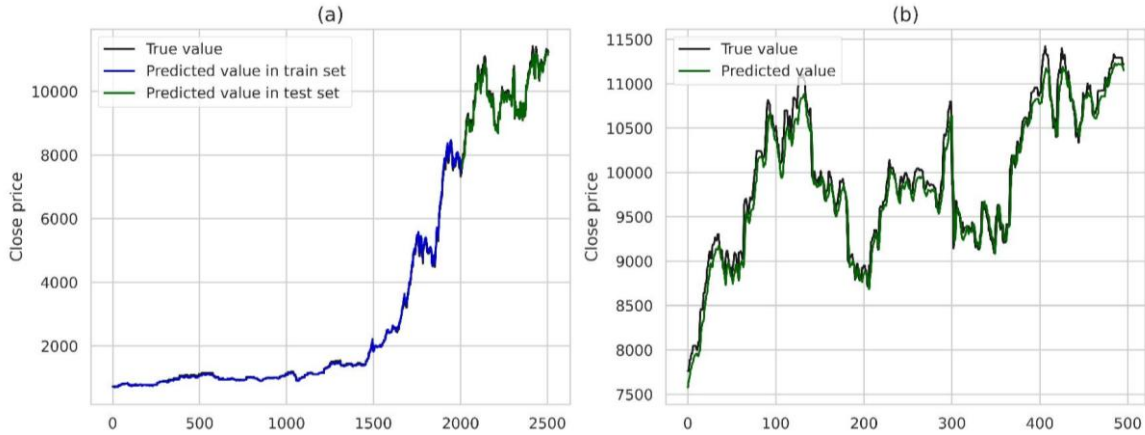


Figure 5. Best single layer LSTM model predictions (10 neurons, best replicate, RMSE = 165.46, MAPE = 1.27%). (a) Full time series showing training period predictions (blue) and test period predictions (green) against true values (black). (b) Test period only, showing the model's ability to track short term oscillations.

Figure 5 illustrates the predictive behaviour of the best single layer configuration. The model closely follows the overall trajectory of the BIST 100 and captures the main short-term movements during the test period. Some deviations remain around sharper fluctuations, but the overall alignment indicates that the compact architecture was able to reproduce both the long-term trend and a substantial part of the short term variation.

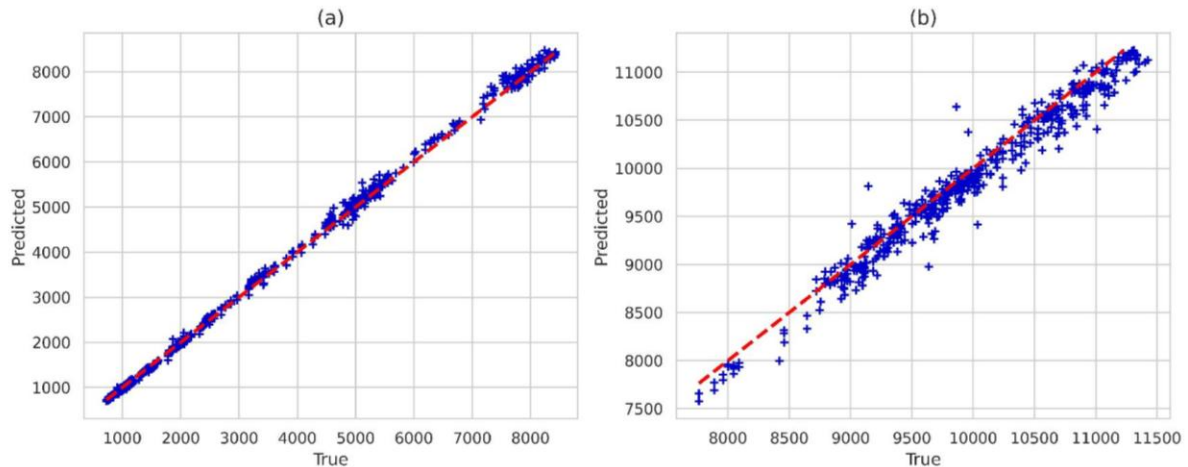


Figure 6. True versus predicted BIST 100 close prices for the best single layer LSTM model (10N, best replicate). (a) Training set predictions show near perfect alignment with the 45° identity line (red dashed). (b) Test set predictions exhibit strong but slightly more dispersed alignment, consistent with the increased volatility of the 2023–2025 test window.

Figure 6 further illustrates the agreement between actual and predicted index values. The close alignment is particularly evident in the training period, while the test period observations show greater dispersion, consistent with the more volatile market conditions in the later sample. Nevertheless, the overall relationship remains strong, supporting the predictive performance indicated by the aggregate evaluation metrics.

4.2 Multi Layer LSTM Model Performance

Six multi layer LSTM architectures were evaluated across 30 independent replicates: (10,5), (20,10), (50,20), (100,50), (150,100), and (100,50,20). Table 7 summarises their test set performance, while Figure 7 illustrates the distribution of RMSE, MAPE, and Pearson R across replications.

Table 7. Test Set Performance of Multi Layer LSTM Models (30 replicates). Asterisk (*) marks the best mean RMSE configuration.

Architecture	Mean RMSE	Std RMSE	Min RMSE	Max RMSE	Mean MAPE (%)	Mean R	Std R
(10, 5)	339.99	40.55	266.28	438.72	2.81	0.9567	0.0062
(20, 10)	489.17	102.84	319.17	699.07	4.08	0.9111	0.0284
(50, 20)	392.82	47.37	299.51	494.14	3.25	0.9312	0.0137
(100, 50)	358.91	35.00	290.47	438.67	2.95	0.9362	0.0081
(150, 100)*	351.97	27.04	300.64	412.23	2.89	0.9360	0.0073
(100, 50, 20)	655.82	41.75	591.15	754.78	5.52	0.8620	0.0175

Note: Asterisk (*) marks the representative multi layer benchmark selected based on the balance between accuracy and replication stability. The (10,5) architecture produced the lowest raw mean RMSE within the multi-layer family.

The multi layer models generally showed weaker predictive performance than the single-layer configurations. Within this family, the (10,5) architecture produced the lowest raw mean error, but its performance was less stable across replications. The (150,100) configuration provided a more balanced accuracy stability profile and was therefore retained as the representative multi-layer benchmark for the subsequent model family comparison. This distinction is important: (150,100) is not the lowest error multi layer architecture, but rather the selected benchmark based on its more favourable balance between accuracy and replication stability.

The remaining architectures show that greater depth or width did not generate a systematic performance improvement. The (20,10) configuration exhibited particularly high instability, while the three layer (100,50,20) architecture produced the weakest overall performance. Taken together, these results indicate that increasing architectural complexity beyond the compact two layer configurations was not advantageous for this forecasting problem.

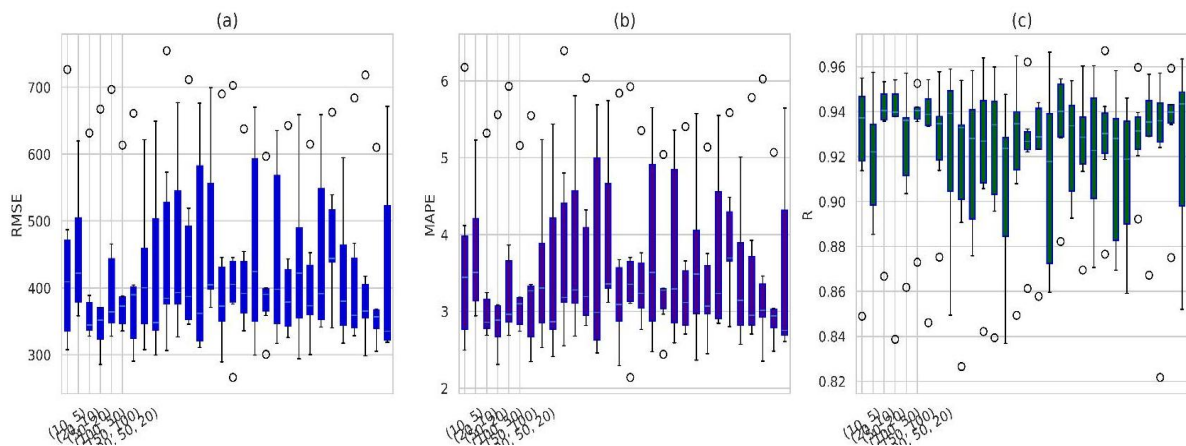


Figure 7. Distribution of (a) RMSE, (b) MAPE, and (c) Pearson R across 30 replicates for the six multi-layer LSTM configurations [(10,5), (20,10), (50,20), (100,50), (150,100), (100,50,20)]. Note the markedly higher RMSE range compared with the single-layer models in Figure 3, and the dramatically poor performance of the three-layer (100,50,20) architecture.

Figure 7 reinforces the distributional differences observed in Table 7. The multi layer configurations generally exhibit wider error distributions than the stronger single layer models, while the three layer architecture shows the clearest deterioration. The visual evidence therefore supports the conclusion that additional architectural depth did not translate into more stable or accurate forecasts.



Figure 8. Test set predictions for all six multi layer LSTM model configurations (best replicate of each). Blue = true BIST 100 close price; Green = predicted values. Panels (a)–(f) correspond to architectures (10,5), (20,10), (50,20), (100,50), (150,100), and (100,50,20), respectively. Note the systematic lag and magnitude underestimation relative to Figure 5(b).

Figure 8 presents the test set predictions for the six multi layer configurations. Across the models, the broad direction of the BIST 100 is generally captured, but short term peaks and troughs are more frequently underestimated or mistimed than in the best single layer configuration. This pattern is consistent with the higher forecast errors and indicates that the additional architectural complexity did not improve short term pattern capture.

4.3 Comparative Analysis: Single Layer vs. Multi Layer Models

A direct comparison between the best single layer configuration (10N) and the representative multi layer benchmark (150,100) is presented in Table 8.

The comparison shows a clear and consistent forecasting advantage for the single layer architecture across the principal accuracy measures, while the multi layer benchmark exhibits greater replication stability.

Table 8. Comparative Performance Summary: Best Single-Layer vs. Representative Multi-Layer LSTM Model

Metric	Best SL-LSTM (10N)	Representative ML-LSTM (150,100)	Difference
Mean RMSE	231.32	351.97	−120.65 (SL better)

Metric	Best SL-LSTM (10N)	Representative ML-LSTM (150,100)	Difference
Std RMSE	44.88	27.04	+17.84 (ML more stable)
Min RMSE (best run)	165.46	300.64	−135.18 (SL better)
Mean MAPE (%)	1.87%	2.89%	−1.02% (SL better)
Mean R	0.9797	0.9360	+0.0437 (SL better)
Welch's t-statistic	—	—	−12.20
p-value	—	—	< 0.001
Normality (p-value)	0.386 (normal)	0.929 (normal)	Both passed

The comparison reveals a clear accuracy stability trade-off between the two architectures. The single-layer model provides stronger forecasting accuracy across the evaluation criteria, whereas the representative multi layer benchmark produces more consistent results across replications. Thus, greater architectural depth does not appear to offer an accuracy advantage in this setting, even though it may reduce variability between training runs.

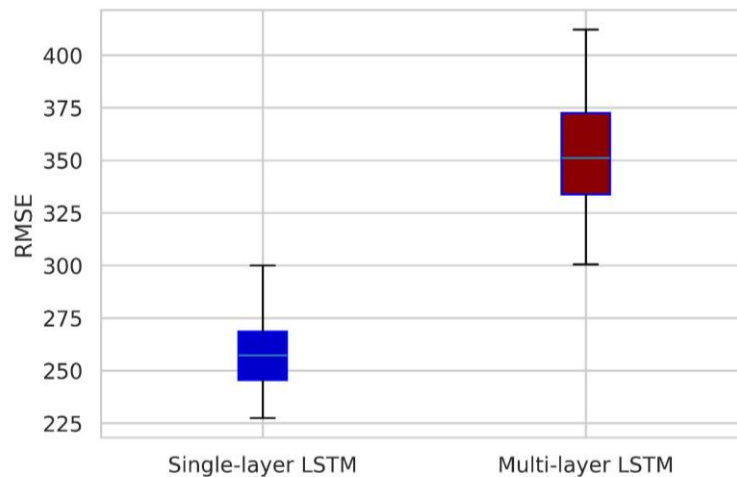


Figure 9. Comparative boxplot of RMSE distributions across 30 replicates for the best single layer LSTM (10 neurons, blue) and the representative multi-layer LSTM (150, 100 neurons, dark red). The near complete separation of the two distributions visually confirms the statistically significant performance advantage of the single layer architecture.

Figure 9 provides visual support for this result, showing substantial separation between the RMSE distributions of the two model families. The limited overlap indicates that the single-layer advantage is systematic across replications rather than driven by a small number of unusually successful runs.

The lower variability of the representative multi-layer benchmark highlights an important practical trade-off. Although the 10N configuration provides the strongest average accuracy, the 50N single-layer model may offer a particularly attractive compromise between accuracy and training stability, as it combines near-best predictive performance with the lowest variability among the viable configurations.

4.4 Statistical Analysis

To formally test whether the performance difference between the best single-layer (10N) and best multi-layer (150, 100) models is statistically significant, a two-stage statistical analysis was performed.

4.4.1 Normality Testing

D'Agostino and Pearson's omnibus test provided no evidence of substantial departures from normality for either RMSE distribution. The Q–Q plots in Figure 10 are consistent with this result, showing close alignment between the observed and theoretical quantiles. These findings support the use of Welch's t-test for the subsequent comparison.

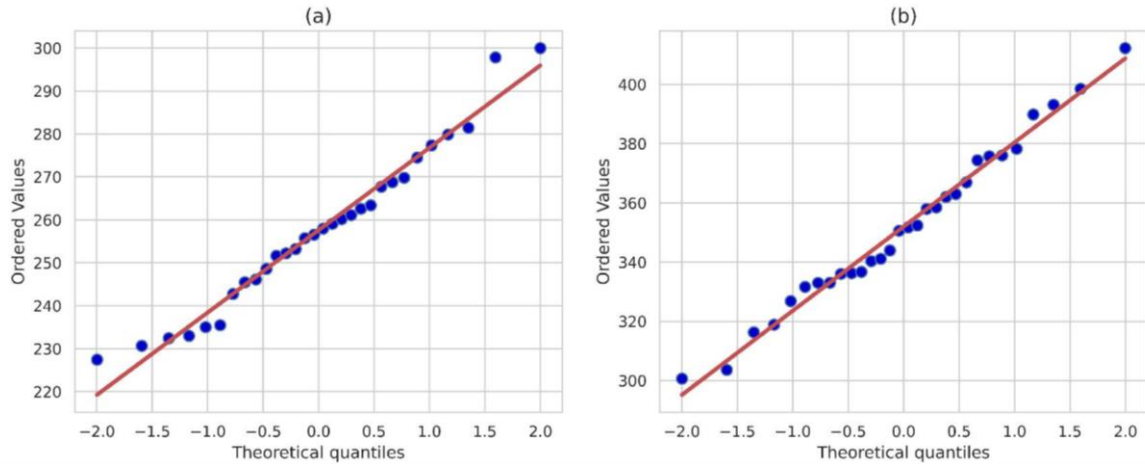


Figure 10. Q-Q plots of RMSE distributions across 30 replicates for (a) the best single-layer LSTM (10N) and (b) the representative multi-layer LSTM (150,100). Close alignment of observed quantiles (blue dots) with the theoretical normal line (red) confirms approximate normality for both distributions, validating the use of Welch's t-test.

4.4.2 Welch's Two-Sample t-Test

Welch's two-sample t-test confirmed a statistically significant difference in mean RMSE between the two model configurations ($t = -12.20$, $p < 0.001$). The result provides strong evidence that the observed accuracy advantage of the single-layer configuration is systematic rather than attributable to random variation across training runs.

A supplementary comparison using the 50-neuron single-layer configuration, selected for its greater training stability, led to the same conclusion ($p < 0.001$). This indicates that the observed advantage of the single-layer family is not dependent on selecting the lowest-error configuration alone.

4.5 Discussion

The consistent outperformance of single-layer LSTM models suggests that greater architectural depth was not advantageous for the present forecasting task. One plausible explanation is that the five-day lookback window captures relatively local temporal dependencies. After wavelet denoising, the BIST 100 series may therefore not contain sufficient sequential complexity to benefit from the hierarchical feature extraction associated with deeper recurrent architectures.

A second explanation concerns model capacity. Multi-layer architectures introduce substantially more parameters, while the effective information available for learning remains constrained by the short sequence window and the dependence structure of financial

observations. The weaker performance of the deeper configurations is therefore consistent with a mismatch between model capacity and the informational structure of the forecasting problem.

The findings consequently support a parsimonious approach to LSTM design in this setting. The objective is not to minimise model complexity per se, but to match architectural capacity to the information available in the data. The strong performance of the compact single-layer configurations provides empirical support for this principle.

This interpretation is broadly consistent with financial forecasting studies showing that deeper or more sophisticated architectures do not universally outperform simpler alternatives (Sezer et al., 2020; Jiang, 2021). The present results extend this perspective by isolating architectural depth within the LSTM family while keeping the forecasting task and information environment constant.

These findings should nevertheless be interpreted within the scope of the experimental design. The analysis uses a fixed five-day lookback window and a specific set of predictors and preprocessing choices. Longer sequences, alternative information sets, attention mechanisms, or hybrid architectures may produce different results. These extensions provide promising directions for future research.

Conclusion

This study examined whether increasing architectural complexity improves the forecasting accuracy of Long Short-Term Memory (LSTM) models for the BIST 100 index. Using 2,507 daily observations from January 2016 to December 2025, the study incorporated technical indicators together with global and macroeconomic variables and applied wavelet denoising, sequence construction, hyperparameter optimisation, and repeated model training to establish a consistent forecasting framework. Single layer and multi layer LSTM architectures were evaluated across multiple independent replications using complementary measures of predictive accuracy and stability.

The overall evidence indicates that greater architectural complexity did not translate into better generalisation in this setting. Compact single layer architectures provided the strongest forecasting performance, while additional layers and substantially larger architectures failed to deliver a consistent accuracy advantage. At the same time, the results revealed a relevant accuracy stability trade-off, suggesting that model selection should consider not only average predictive accuracy but also the consistency of performance across repeated training runs. More broadly, the findings suggest that architectural parsimony can be advantageous when model complexity exceeds the informational structure of the data.

Several limitations should be acknowledged. The analysis was conducted for a single emerging market index and relied on a fixed five day lookback window, a specific set of technical and macroeconomic predictors, and a predefined preprocessing and modelling framework. The findings therefore should not be interpreted as evidence that simpler LSTM architectures will universally outperform deeper alternatives. In addition, the study focused on forecasting accuracy and training stability rather than the economic profitability of trading strategies based on the forecasts. These design choices define the scope within which the reported conclusions should be interpreted.

Future research could examine whether the observed relationship between architectural complexity and forecasting performance persists under alternative sequence lengths, predictor sets, market conditions, and emerging or developed market indices. Further studies could also compare the present LSTM framework with attention based, transformer, hybrid, or other deep-learning architectures and assess whether complexity becomes beneficial when longer temporal dependencies or richer information sets are available. Extending the analysis from statistical

forecasting accuracy to economic and portfolio level outcomes would provide an additional perspective on the practical value of alternative model architectures.

Ultimately, the findings suggest that in volatile emerging market forecasting, more complex LSTM architectures are not necessarily better; rather, effective model design depends on matching architectural capacity to the information contained in the data.

References

- Achelis, S. B., & Achelis, S. (2001). *Technical Analysis from A to Z*. pp. 199-200. New York: McGraw Hill.
- Akyildirim, E., Nguyen, D. K., Sensoy, A., & Šikić, M. (2023). Forecasting High-Frequency Excess Stock Returns Via Data Analytics and Machine Learning. *European Financial Management*, 29(1), pp.22–75.
- Akturk, Z., Kılıc, E., Algorabi, O., Ulu, M., Turkan, Y. S., & Namlı, E. (2025). Real-Time Forecast of BIST100 Index Under Market Volatility and Uncertainty. *Computational Economics*, 67, pp. 3249–3274.
- Bao, W., Yue, J., & Rao, Y. (2017). A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders And Long-Short Term Memory. *PLOS ONE*, 12(7), e0180944.
- Bekiros, S., Gupta, R., & Kyei, C. (2016). On Economic Uncertainty, Stock Market Predictability and Nonlinear Spillover Effects. *North American Journal of Economics and Finance*, 36, pp.184–191.
- Bennaton, E. M., Baroudi, M. N., Khan, W. A., Kouatly, R., Karunanithi, M., & Hassan, A. (2025). Enhancing Stock Forecasting Accuracy Through LSTM-Based Recurrent Models. *2nd International Conference on Integration of Computational Intelligent System (ICICIS) Pune, India. Sep 09-12, 2025*.
- Bergmeir, C., & Benítez, J. M. (2012). On the Use of Cross-Validation for Time Series Predictor Evaluation. *Information Sciences*, pp. 192–213.
- Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3), pp. 307–327.
- Box, G. E. P., & Jenkins, G. M. (1970). *Time series analysis: Forecasting and control*. Holden-Day.
- Bulus, E. (2025). Comparison of Deep Learning Models in Stock Price. In *2025 10th International Conference on Computer Science and Engineering (UBMK)* pp. 52-57. IEEE.
- Chen, Z., Dai, Z., Xing, H., Chen, J., Huo, M., & Lu, K. (2025). Multi-Model Approach for Stock Price Prediction and Trading Recommendations. In *Proceedings of the 2025 9th International Conference on Electronic Information Technology and Computer Engineering (EITCE 2025)*. ACM.
- Chowdhury, M. S., Nabi, N., Rana, M. N. U., Shaima, M., Esa, H., Mitra, A., Mozumder, M. A. S., Liza, I. A., Sweet, M. M. R., & Naznin, R. (2024). Deep Learning Models for Stock Market Forecasting: A Comprehensive Comparative Analysis. *Journal of Business and Management Studies*, 6(2), pp. 95–99.
- D'Agostino, R. B., Belanger, A., & D'Agostino Jr, R. B. (1990). A Suggestion for Using Powerful and Informative Tests of Normality. *The American Statistician*, 44(4), pp. 316-321.
- Demir, C. (2019). Macroeconomic Determinants of Stock Market Fluctuations: The case of BIST-100. *Economies*, 7(1), pp. 8.

- Diebold, F. X., & Mariano, R. S. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3), pp. 253–263.
- Donoho, D. L., & Johnstone, I. M. (1994). Ideal Spatial Adaptation By Wavelet Shrinkage. *Biometrika*, 81(3), pp. 425–455.
- Dozat, T. (2016). Incorporating Nesterov momentum into Adam. *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 1–4.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12, 2121–2159.
- Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of The Variance of United Kingdom Inflation. *Econometrica*, 50(4), 987–1007.
- Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), pp. 383–417.
- Fischer, T., & Krauss, C. (2018). Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. *European Journal of Operational Research*, 270(2), pp. 654–669.
- Fischer, T., Sterling, M., & Lessmann, S. (2024). Fx-Spot Predictions with State-Of-The-Art Transformer and Time Embeddings. *Expert Systems with Applications*, 249, pp. 123538.
- Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer.
- Gur, Y. E. (2024). Evaluation Of BIST100 Index Prediction Performance of Deep and Machine Learning Algorithms. *Ekonomik ve Sosyal Arastirmalar Dergisi*, 20(2), pp. 394–409.
- Hartanto, R. D., Shidik, G. F., Alzami, F., Fanani, A. Z., Marjuni, A., & Syukur, A. (2026). Attention-Augmented GRU for Stock Forecasting: A Trade-Off Between Directional Accuracy and Price Prediction Error. *Journal of Computing Theories and Applications*, 3(4), pp. 457-472.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
- Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning: The Mit Press, 2016, 800 pp, isbn: 0262035618. *Genetic Programming and Evolvable Machines*, 19(1), pp. 305-307.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), pp. 1735–1780.
- Jiang, W. (2021). Applications of Deep Learning in Stock Market Prediction: Recent Progress. *Expert Systems with Applications*, 184, 115537.
- Kilimci, Z. H., & Duvar, R. (2020). An Efficient Word Embedding And Deep Learning Based Model To Forecast The Direction of Stock Exchange Market Using Twitter And Financial News Sites: A Case of Istanbul Stock Exchange (BIST 100). *IEEE Access*, 8, pp. 188186–188198.
- Kim, H. Y., & Won, C. H. (2018). Forecasting The Volatility of Stock Price index: A Hybrid Model Integrating LSTM with Multiple GARCH-Type Models. *Expert Systems with Applications*, 103, pp. 25–37.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representation (ICLR)*, 1–4.

- Kirelli, Y. (2024). Comparative Analysis of LSTM And ARIMA Models in Stock Price Prediction: A Technology Company Example. *Black Sea Journal of Engineering and Science*, 7(5), pp. 866–873.
- Kobiela, D., Krefta, D., Król, W., & Weichbroth, P. (2022). ARIMA vs LSTM on NASDAQ Stock Exchange Data. *Procedia Computer Science*, 207, pp. 3830–3839.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PLOS ONE*, 13(3), pp. e0194889.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 Time Series and 61 Forecasting Methods. *International Journal of Forecasting*, 36(1), pp. 54–74.
- Mazzouz, A., & Elmoufidi, A. (2025). Comparative Analysis of LSTM and CNN Architectures for Prediction Enhancement: Case of Moroccan Stock Market Return. *International Journal of Computing and Digital Systems*, 18(1), pp. 1–15.
- Moritz, S., & Bartz-Beielstein, T. (2017). Imputets: Time Series Missing Value Imputation in R. *The R Journal*, 9(1), pp. 207–218.
- Nelson, D. M. Q., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock Market's Price Movement Prediction with LSTM Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* pp. 1419–1426. IEEE.
- Oksak, Y., Buyukkor, Y., & Saritas, T. (2025). Wavelet-Enhanced Multimodel Framework for Stock Market Forecasting: A Comprehensive Analysis Across Market Regimes. *Borsa Istanbul Review*, 100771.
- Osman, E. G. A., & Otaibi, F. A. (2025). Integrating Deep Learning and Econometrics For Stock Price Prediction: A Comprehensive Comparison of LSTM, Transformers, and Traditional Time Series Models. *Machine Learning with Applications*, 22, pp. 100730.
- Rehman, M. S. U., & Gul, F. (2026). Intelligent Forecasting in Emerging Markets: A Comparison of AI, Linear, and Hybrid Forecasting Models at Pakistan Stock Exchange. *Future Business Journal*, 12, pp. 110.
- Renaud, O., Starck, J.-L., & Murtagh, F. (2005). Wavelet-Based Combined Signal Filtering and Prediction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(6), pp. 1241–1251.
- Sarikoc, M., & Celik, M. (2024). PCA-ICA-LSTM: A Hybrid Deep Learning Model Based On Dimension Reduction Methods To Predict S&P 500 index price. *Computational Economics*, 65, pp. 2249–2315.
- Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019. *Applied Soft Computing*, 90, pp. 106181.
- Sivri, M. S., & Ustundag, A. (2024). An Adaptive And Enhanced Framework for Daily Stock Market Prediction Using Feature Selection and Ensemble Learning Algorithms. *Journal of Business Analytics*, 7(1), pp. 42–62.
- Ustuner, F., & Cankal, E. (2025). The Effectiveness of Artificial Intelligence in Financial Analysis: Evidence from Istanbul Stock Exchange 100 (ISE 100). *Third Sector Social Economic Review*, 60(4), pp. 3792–3825.
- Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart E. & Yu, T. (2014). Scikit-Image: Image Processing in Python. *PeerJ*, 2, pp. e453.

- Wang, T. W., Lu, Y. S., Ku, C. S., Mohammed, O., Raj, B., & Por, L. Y. (2026). A multimodal Deep Fusion Method for Stock Movement Prediction Using Heterogeneous Data Source. *IEEE Access*, 14, pp. 9882–9896.
- Welch, B. L. (1947). The Generalization of 'Student's' Problem when Several Different Population Variances are Involved. *Biometrika*, 34(1/2), pp. 28–35.
- Whaley, R. E. (2000). The Investor Fear Gauge. *Journal of Portfolio Management*, 26(3), pp. 12–17.
- Xiao, J., Bi, S., & Deng, T. (2024). Comparative Analysis of LSTM, GRU, and Transformer Models for Stock Price Prediction. *arXiv preprint*, <https://doi.org/10.48550/arXiv.2411.05790>
- Yan, X. (2025). A Comparative Study of LSTM, ARIMA-LSTM, and CNN-LSTM Models for S&P 500 Closing Price Prediction. *2025 IEEE 3rd International Conference on Electrical, Automation and Computer Engineering (ICEACE)*.
- Yigit, Ö. E., Alp, S., & Öz, E. (2020). Prediction of BIST Price Indices: A Comparative Study Between Traditional and Deep Learning Methods. *Sigma Journal of Engineering and Natural Sciences*, 38(4), pp. 1693–1704.
- Zein, M. H., Yudistira, N., & Adikara, P. P. (2024). Indonesian Stock Price Prediction Using Neural Basis Expansion Analysis for Interpretable Time Series Method. *Journal of Information and Communication Technology*, 23(3), pp. 361–392.
- Zhang, G. P. (2003). Time Series Forecasting Using A Hybrid ARIMA and Neural Network Model. *Neurocomputing*, 50, pp. 159–175.
- Ziółkowski, K., & Wieceński, P. (2025). Enhancing Financial Time Series Forecasting: A Comparative Study of Discrete Wavelet Transform and LSTM Models For Selected Global Indices. *Quality & Quantity*, 60, pp. 1995–2015.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent: This article does not contain any studies with human participants, so informed consent is not applicable here.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [FinalResults.zip](#)
- [TRUNEMP.csv](#)
- [VariablesandDataSources1.docx](#)
- [TRCCL.csv](#)
- [BIST10020152025.xlsx](#)
- [FinalTRDataAllCleaned.csv](#)
- [VIXHistory.csv](#)
- [LSTMSDMCodeImplementation.ipynb](#)