Journal of Economics, Finance and Accounting Studies (JEFAS)
ISSN: 2709-0809
DOI: 10.32996/jefas
Journal Homepage: www.al-kindipublisher.com/index.php/jefas

JEFAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Bridging Prediction and Profit: Deep Learning models with Trading Evaluation for FTSE 100

S. Ehsan Hosseiny N.[1] and Daqing Chen[2]
[12]London South Bank University, London, UK
Corresponding Author: S. Ehsan Hosseiny N, **E-mail**: S4335313@LSBU.ac.uk

| ABSTRACT
This study examines stock price forecasting for FTSE 100 companies using deep learning and XAIl. The research addresses the disconnect between predictive accuracy and interpret ability in financial models by integrating data-driven forecasting with transparent feature attribution. Four neural architectures: LSTM2, Bi-LSTM3, GRU4 and CNN5 are compared to classical benchmarks: SMA6 and EMA7. Models are trained on OHLCV8 data augmented with technical indicators. Evaluation uses a threshold-based trading strategy. The findings indicate that a lower prediction error does not necessarily result in higher profitability. Although LSTM achieved the lowest prediction error, GRU and Bi-LSTM produced more stable cumulative returns (16%), compared to the EMA benchmark (2%). SHAP9 analysis demonstrates that recent price movements and momentum indicators, particularly SMA, drive model decisions.

| KEYWORDS
FTSE 100, LSTM, GRU, Explainable AI, SHAP, Trading strategy, Stock market prediction, BILSTM, financial data, machine learning

| ARTICLE INFORMATION
ACCEPTED: 01 January 2026 **PUBLISHED:** 15 January 2026 **DOI:** 10.32996/jefas.2026.8.1.4

1 – Introduction

The central challenge in stock price forecasting lies not only in increasing predictive accuracy but also in turning such gains into profitable, actionable investment decisions. Traditional methods, such as statistical time-series forecasting and financial ratio analysis, often rely on linear assumptions which limits their applicability. Machine and deep learning approaches have improved pattern recognition in market data. However, a critical disconnect remains: research often celebrates lower error rates without proving impact on trading performance or model explainability. This paper tackles this gap by integrating feature engineering, XAI, and simulated trading evaluation, we demonstrate how linking predictive outcomes to financial impact and transparent model behavior advances both theoretical and practical finance.

[1] Explainable artificial intelligence [2]LSTM: Long Short-Term Memory
[3]Bidirectional LSTM
[4]Gated Recurrent Unit
[5]Convolutional Neural Network
[6]Simple Moving Average
[7]Exponential Moving Average
[8]0pen; high; Low; Close; Volume
[9]SHapley Additive exPlanations

Feature engineering adds new features, such as EMA, SMA, VIX, and Bollinger Bands, to help models spot market trends and volatility [Mos25]. Combining these features aims to give algorithms better market insight. The input data are used to train deep learning models (LSTM, Bi-LSTM, GRU, CNN) and for comparison against classical statistical measures (EMA and SMA) to support interpretation. Model performance is evaluated by MSE[10] and a simulation approach that applies a threshold to assess profitability. XAI strategies, such as SHAP, are utilized to interpret the influence of specific features and input dimensions on predictions. The remainder of this paper is organized as follows. 2 reviews related work on stock prediction and explainable finance. 3 presents the methodology and model design. 4 describes the experimental setup and the results. Section 5 discusses the findings, implications, wraps up the paper, and outlines future research directions.

2. Related Work

Early attempts at stock market prediction relied on classical statistical techniques such as ARIMA and GARCH [BJ76] [BJR15] [Bol86]. Although successful for linear series, they could not properly characterize the non-linear behavior and volatility of financial markets [Fam70] [Tsa10]. To address this challenge, Deep learning and ML[11] techniques have been developed in stock data models as sequence data to take ad-vantage of temporal relationships to learn momentum, volatility, and market trends [FK18] [NPdO17]. Recurrent Neural Network architectures based on LSTM and GRU layers have also gained acceptance for financial projection because they can learn from temporal relationships in sequence data [NPdO17] [FK18] [ZW21]. Hybrid models have also demonstrated successful applications. For example, [MK25] reported that CNN-LSTM learning improved the predictability of returns relative to the benchmarks. [RJ25] de-signed an LSTM and real-time sentiment analysis coupled with SHAP to generate interpretable buy or sell recommendations. Similarly, [KK19] and [Lan25] emphasized the effectiveness of CNN-LSTM learning to identify patterns by focusing on local features and dependencies to improve performance. Moreover, A Few researchers have tried to go beyond error measures for validating their models. For example, [LKY+24] ranked portfolios to generate +31% annual excess returns using LSTM on CSI 300 benchmarks. Additionally, all ML models rely on the input features provided to them and giving more high quality data could help with its accuracy and wise versa, as widely established by the garbage-in-garbage-out principle [ZW21]. Based on this rationale, [Mos25] examined 88 technical features, including EMA and HMA[12], RSI, and BB[13], among others, using machine learning classifiers such as XGBoost, Random Forest, SVR, and LSTM Regressor. Their result demonstrated that on average, EMA and HMA are among the most prominent features for describing market trends and regularizing irregular price movements, thereby accentuating smoother futures for visionary advancements through engineered features rather than simple fundamentals. Nevertheless, while hybrid architectures have succeeded in learning temporal features, they may lack interpretability, which is addressed by XAI frameworks applied to hybrid representations at a later stage. According to the systematic review conducted by [vK24], the fusion of deep learning and XAI techniques improves the trustworthiness of financial prediction tasks. [MAKB24] applied momentum and volatility measures for classification using deep learning techniques. Their approach outperformed the SVM and Random Forest classifiers and, using SHAP and LIME explanations, confirmed that short-term momentum features influence prediction outcomes the most. [SL25] fused sentiment analysis and feature analysis with LSTM classification to achieve a 51.8% improvement in annual performance, as indicated by a higher Shap ratio for financial significance than RMSE values. De-spite advances in deep learning-based stock prediction, most of the works continue to emphasize statistical validity rather than financial viability [LKY+24]. In conclusion, most financial forecasting works include only price input features and are primarily evaluated using error metrics. Although technical analysis has improved trend volatility and removed inefficiencies, studies rarely assess ROI[14] in depth, focusing mainly on U.S. or Asian markets and largely ignoring UK business concerns.

[10] Mean Squared Error
[11] Machine learning
[12] Hull Moving Average
[13] Bollinger Bands

3. Methodology

This study adopts the CRISP-DM [WH00] framework to ensure a structured, reproducible, and business-oriented process for the development and evaluation of stock forecasting models. The workflow begins with data acquisition for FTSE 100 companies, and daily OHLCV data are enriched through feature engineering to generate technical indicators that capture trend and volatility. The data is then split into sliding windows, which serve as inputs to deep learning models (LSTM, GRU, CNN) for next-day forecasting. Daily OHLCV data are collected from Yahoo Finance for the period 2015 to 2025. To capture the market structure beyond the raw price, a set of technical indicators is engineered. These include SMA, EMA, BB, and rolling standard deviation for volatility. Engineered indicators improve trend detection, but introduce correlation risks. All continuous inputs are scaled using Min-Max normalization, while volume data is log-transformed to reduce skewness. Forecasting was framed as a supervised time-series task using sliding windows of trading days to predict the next-day closing price. the Window (30 to 120 days) was empirically selected based on the stability of validation-loss to balance short-term sensitivity with contextual depth. Four deep learning architectures were implemented: LSTM, Bi-LSTM, GRU, and 1D-CNN, comparing shallow (single-layer) and deep (stacked) variants to evaluate temporal learning capacity. Each module in the proposed framework offers specific advantages and trade-offs. These Models are trained using an 80-20 chrono-logical split with the Adam optimizer and a training loss function based on MSE. While deep learning models can effectively capture temporal dependencies, they re-quired careful regularization to prevent overfitting. The evaluation of the models was conducted in these complementary stages: First, statistical performance was tested using two input configurations: one with raw OHLCV data and another incorporating engineered technical indicators, across multiple window lengths (30-120 days). Each configuration and input is compared using MSE, and the model achieving the lowest MSE is selected for subsequent testing. To contextualize deep learning performance within traditional forecasting methods, SMA and EMA models were also tested as baseline benchmarks. These indicators are widely used by traders to smooth short-term volatility, highlight trend reversals, and generate buy-sell signals, making them suitable reference points for both metric and trading evaluations. Second, the best-performing structures were evaluated in a trading simulation to assess practical effectiveness. The simulation generated buy or sell signals when predicted price deviations exceeded a decision threshold, measuring cumulative portfolio gain or loss over 200 simulated trading days. Third is return on investment test: Each experiment began with a notional capital of $1000 and applied a threshold-based trading strategy driven by the model's predicted price relative to the real market price. When the model estimated that a stock was undervalued (the predicted price exceeded the actual price by more than a defined threshold), buy action is opted. When the stock was judged overvalued (the predicted price fell sufficiently below the actual price), stocks will be sold, and capital was fully moved to cash. If the difference lay within the neutral band, the position was left unchanged. The strategy is therefore intended to exit the market before downward moves and re-enter ahead of upward moves. Finally, the interpretability of the model was examined using SHAP, applied in two analyzes: (1) assessing the contribution of each time step within the input window, and (2) ranking the most influential technical indicators driving the models' forecasts. This integrated process (Figure 1) ensures that the framework captures not only predictive accuracy but also the relationship between loss-based metrics and real-world financial performance through trading simulation. By linking interpretability and return evaluation, the framework aligns with the study's objective of developing a transparent, practical, and actionable forecasting system for FTSE 100 stocks

[14]Return on Investment

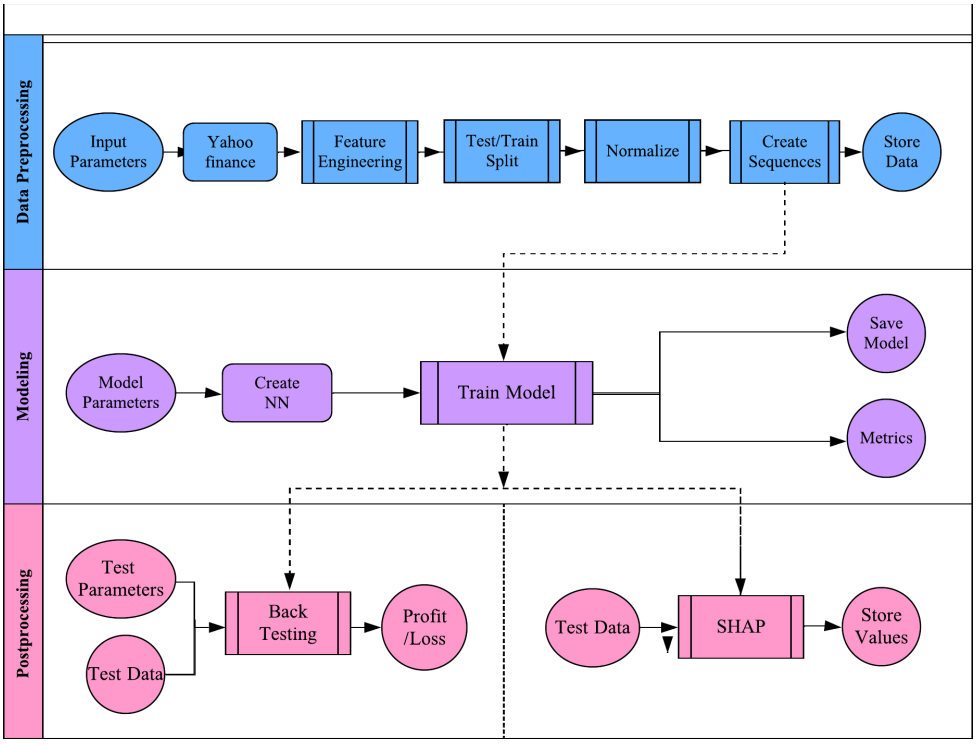


Figure 1: Methodology 0verview

4. Experiments and results

4.1 Experimental Settings

All experiments are conducted in Python using TensorFlow and Scikit-learn and were executed on a laptop with an Intel Core i7 processor, 16GB RAM, and NVIDIA GPU acceleration, ensuring efficient model training.

4.2 Models' metrics

The performance of six forecasting models (SMA, EMA, CNN, LSTM, GRU, and Bi-LSTM) is summarized in Table 1. Each model is trained with window sizes ranging from 30 to 120 days. The first training uses only the OLHCV value. The next training includes Feature Engineering (FE) to find the best conditions for predictive accuracy.

Table 1: Model Performance Comparison

Model	Best Window	Loss	Loss with FE	Training Time
SMA (baseline)	5	0.019	–	1 sec
EMA (baseline)	5	0.016	–	1 sec
CNN (shallow)	30	$4.2\text{-}5.2\times10^{-4}$	$11\text{-}22\times10^{-4}$	39 sec
GRU (shallow)	120	15.1×10^{-4}	16.0×10^{-4}	362 sec
LSTM (shallow)	30	3.85×10^{-4}	3.82×10^{-4}	66 sec
Bi-LSTM (shallow)	30	$2.4\text{-}4.2\times10^{-4}$	$1.2\text{-}1.6\times10^{-4}$	630 sec

Among the baseline models, EMA outperformed SMA because it reacts more quickly to recent price movements. All deep learning models had lower validation loss than the baselines, with LSTM performing best, followed by Bi-LSTM and CNN. GRU was slightly weaker. Deeper architectures reduced RMSE by 8-12%, though improvements plateaued after 2 layers. Analyzing the window size revealed that 30 to 60-day look-back periods were the most effective, confirming that recent data has the highest predictive value. Feature engineering mainly improved results for deeper models, especially Bi-LSTM, while shallow LSTM saw little benefit. This suggests that more complex architectures better utilize engineered features. Figure 2 depicts the predicted values for LSTM and actual FTSE ticker prices during

validation. The plot shows that LSTM follows the actual values more closely than SMA. It is also evident from this plot that, even for short-term fluctuations in both upward and downward markets, LSTM follows actual values more closely than SMA.

Figure 2: Actual vs Predicted deep learning and baseline price

Since actual values closely follow predicted values from LSTM, one can safely conclude that LSTM captures temporal dependencies to some extent but may be overfitting in some cases, as discussed in further detail in section 4.3 below.

4.3 Trading Test

In this section, all models tested in section 4.2 are evaluated using a unified back-testing framework to assess both predictive accuracy and trading profitability.

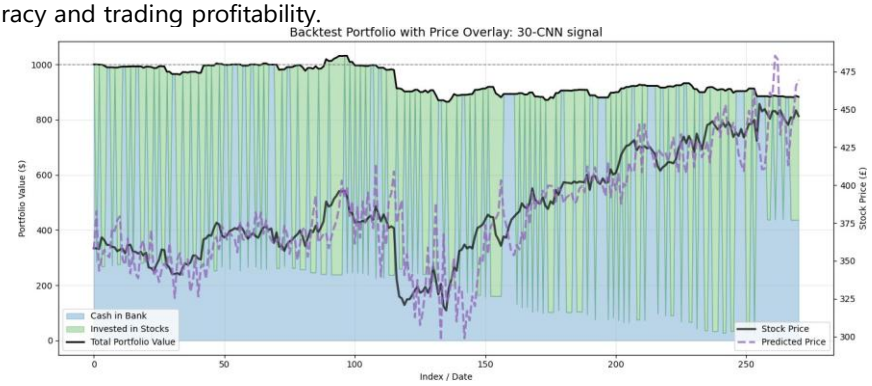


Figure 3: Testbed with prices for CNN 30-day

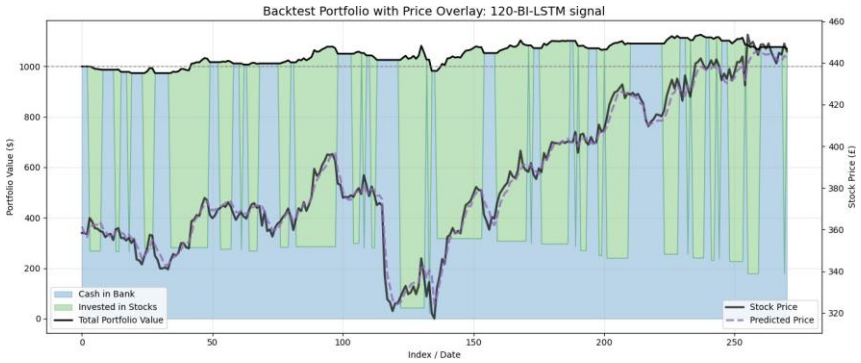
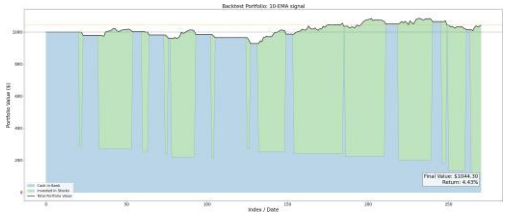


Figure 4: Testbed with prices for BI-LSTM 120-day

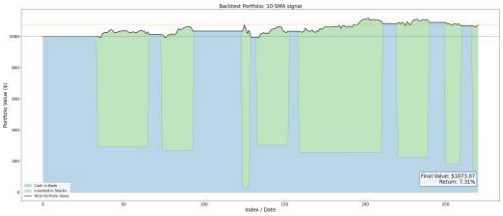
For each, training and validation are performed as before, using additional features and two window sizes. In the following figures, the blue-shaded area represents uninvested cash, the green area indicates invested capital, and the black line shows the total value of the portfolio over time. Some figures also show the predicted (purple) and actual (black) stock values, with their scales on the right. For starters, two of them are studied: 3 (CNN-30) shows that the model struggles to capture sustained market trends. It predicts sudden directional shifts every few days, leading to rapid buy-sell actions and unstable portfolio movement. This suggests that CNN reacts to short-term fluctuations rather than genuine trend changes. By contrast, figure 4 (Bi-LSTM-120) shows smooth and intentional trades. The strategy reacts only to major trades and takes longer to liquidate positions. It is apparent that during the market downturn from Day 210 to 230, Bi-LSTM adjusted correctly, moving to cash before recovering afterward.

Figure 5 depict the performance of portfolios for the remaining setting parameters in the test data. For comparison, the basic SMA and EMA strategies resulted in few trades and missed several profitable opportunities to cash out, notably at the very end of the testing, leading to premature exit

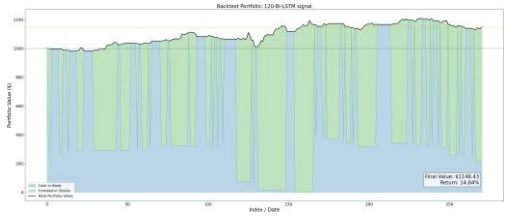
CNN and GRU tended to trade too frequently while experiencing erratic re-turns-likely acting too quickly while reacting to market fluctuations. Unlike others, LSTM and Bi-LSTM models-particularly Bi-LSTM-120-traded positions steadily while growing portfolios smoothly-suggesting good market awareness and effective capital management.
Table 2 shows the return ranges associated with the corresponding testing results. Model stability: Model stability ranged from very high to very low: Bi-LSTM and GRU showed higher stability than others, but CNN-30 and LSTM-30 showed higher volatility. EMA-30, having high accuracy for deep learning tasks, showed equally good returns (8 to 13%), thereby proving.



(a) EMA-lO

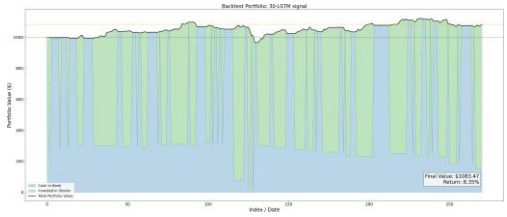


(b) SMA-lO

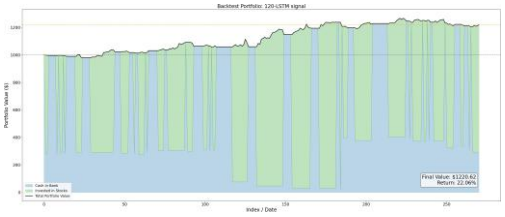


(c) BI-LSTM-l2O

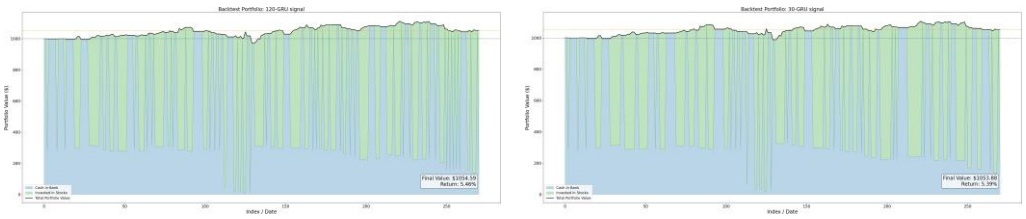
(d) BI-LSTM-3O



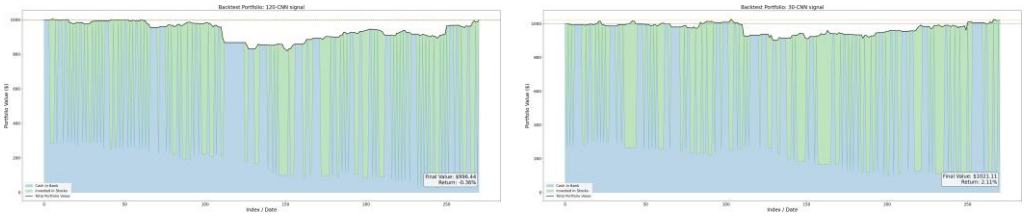
(e) LSTM-3O



(f) LSTM-l2O



(g) GRU-I2O (h) GRU-3O



(i) CNN-I2O (j) CNN-3O

Figure 5: Back-Test Result Examples

Table 2: Model Cumulative Return Ranges

Model	Return	Model	Return Range
Bi-LSTM-30	1% to 8%	Bi-LSTM-120	7% to 11%
GRU-30	2% to 16%	GRU-120	3% to 9%
LSTM-30	-3% to 13%	LSTM-120	3% to 12%
CNN-30	-16% to 17%	CNN-120	-15% to 12%
EMA-10	-2% to 4%	EMA-30	8% to 13%
SMA-10	-4% to 9%	SMA-30	9% to 11%

These observations clearly indicate that reliance on metrics alone is not an effective measure of market success. Models for stock prediction combined with market evaluation through actual market performance can capture market behavior associated with financial decision-making processes more effectively.

4.4 Explainability

Explainability analysis using SHAP showed consistent behavior across models. Values applied for days decline quickly over time, confirming that predictions relied mainly on recent price movements. The last three trading days always ranked in the top five, and the values returned to nominal after the tenth (Table 3).

Table 3: SHAP Values by Day

Days	SHAP Value	Days	SHAP Value
1	46×10^{-4}	3	25×10^{-4}
2	44×10^{-4}	26	35×10^{-4}

At first glance, this emphasis on short-term features may appear inconsistent with the back-testing results, which showed that longer lookback windows (120 days) produced higher and more stable profitability. This can be

reconciled by noting that although the decision signal arises from the most recent days, the longer historical window provides structural context, reduces overfitting, and smooths reactions to market noise. In other words, short-term patterns determine trade direction, while long-term context improves confidence and robustness, explaining why the longer-window models outperform despite SHAP showing localized influence. Feature-level SHAP scores further showed that raw price components (Close, High, Low, and Open) were the most influential inputs, while engineered indicators such as SMA-10, EMA-10, and Bollinger Bands contributed at lower magnitudes (Table 4).

Table 4: Feature SHAP Values

Feature	SHAP Value	Feature	SHAP Value
Close Price	14-57 × 10^{-4}	EMA-10	11-15 × 10^{-5}
High Price	10-38 × 10^{-4}	BB Bands	20-23 × 10^{-5}
Low Price	10-38 × 10^{-4}	Log Return	0.7 × 10^{-5}
SMA-10	17-18 × 10^{-5}	Volume	0.2-2 × 10^{-5}

To evaluate feature importance in a trading context, days with unusually high feature influence (>1.5X IQR above the median) were flagged and linked to trading outcomes. This "importance spike" analysis tested whether extreme feature dominance was associated with higher predictive confidence or better next-day returns (Table 5).

Table 5: Model Important Features and Performance

Model	Most Important Features	High Influence Days	Profit Rate (%)
Bi-LSTM	Close −−**t** Open −−**t** High	148 / 110 / 102	32 / 32 / 32
LSTM	Close −−**t** High −−**t** Open	250 / 95 / 70	32 / 30 / 29
GRU	Close −−**t** High −−**t** Open	183 / 222 / 60	32 / 29 / 29

No single feature consistently showed higher profit ratios during spiking events. While Close Price, High Price, and Open Price were the most significant input features across all models, high importance for any particular price feature did not necessarily translate into better trade performance. Technical indicators were secondary to price information and served as additional rather than major decision-making fac-tors. Altogether, the analysis of explanations indicates the financial reasonableness of the behavior of the respective models: they mostly focus on contemporary price dynamics, while features such as SMA and volatility help create a complete picture. This is consistent with other studies, such as [GU25], which also point to trend and volatility measures as crucial for predicting stock returns. Nevertheless, these models tend to use long time windows to smooth training and avoid making decisions based on transient observations.

One of the primary contributions of this study is that it goes beyond the normal analysis of SHAP values. It not only aims to identify which features have high influence but also to check whether days with high feature importance values also result in high performance. The result shows that high influence for one feature, whether it is Close or High or Open prices, does not necessarily lead to high returns, confirming once again that profitability comes from combinations of different market scenarios and not from peak feature values alone. To the best of our knowledge, no one has explored this direct link between explanations and actual performance in XAI Finance before.

5. Conclusion and Future Work

This work demonstrated that deep learning techniques can be combined with feature engineering and back testing to generate effective, interpretable short-term trading signals for FTSE-100 stocks. While LSTM and Bi-LSTM achieved the lowest prediction errors, back-testing showed that high accuracy is not necessarily linked to high profits, as a simple EMA strategy also performed well. The most successful approaches were those that provided reliable directional information rather than minimizing loss, and for which GRU-30 and Bi-LSTM-120 provided the most reli-able cumulative returns and drawdown performance. Alternatively, CNN and short-window LSTM approaches demonstrated erratic performance and signs of overfit-ting. The outcomes point to the capabilities and limitations of deep learning for trading. This is because traders are presented with outcomes differentiated by deep learning architecture, modest returns, and performance being affected by markets experiencing high volatility or market shifts. XAI analysis helped validate financial institutions: all models were mainly driven by price behavior across consecutive

periods. A new task evaluating the impact of SHAP importance spikes on trade outcomes' performance showed no benefit but underscored the importance of explanation capabilities for verifying model performance rather than raw metric scores. There are several ways future work can build on this study. First, it may be possible to integrate the testbed developed for this study and use reinforcement learning to automatically determine which strategy and model combination is best suited to each market condition. Second, to further improve robustness for market structure changes, researchers should implement shift-aware learning and stress tests. Third, future studies should explore new architectures for sequence modeling and introduce new modalities, such as market-related news, to enhance overall market sensitivity. Lastly, researchers should conduct further realistic back-testing while considering transaction costs and turnover limits to facilitate this, and present interpretable explanations to end users involved.

Acknowledgments

The author would like to thank Dr. Daqing Chen for his valuable advice and guidance throughout this research. The author also expresses sincere gratitude to his family, especially his mother, for their financial and emotional support.

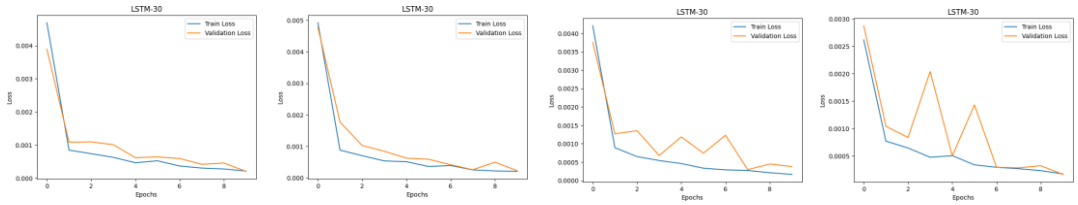
Funding

This research did not receive external funding.

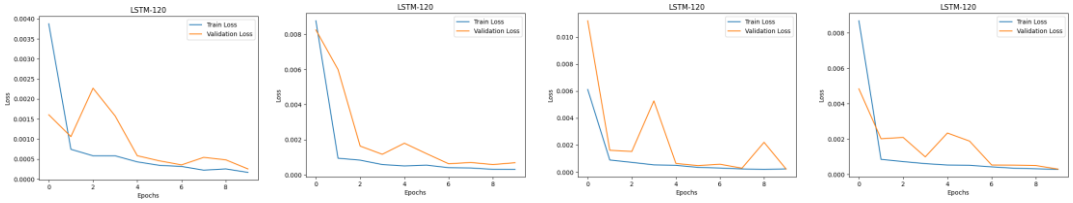
References

[1]. [BJ76] G. E. P. Box and G. M. Jenkins. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco, 1976.
[2]. [BJR15] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. Time Series Analysis: Forecasting and Control. Wiley, 2015.
[3]. [Bol86] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3):307–327, 1986.
[4]. [Fam70] E. F. Fama. Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2):383–417, 1970.
[5]. [FK18] T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2):654–669, 2018.
[6]. [GU25] B. Goswami and A. Uddin. Significance of predictors: revisiting stock return predictions using explainable ai. Annals of Operations Research, 2025.
[7]. [KK19] H. Kim and J. Kim. Cnn-lstm hybrid models for short-term stock price forecasting. Applied Soft Computing, 2019.
[8]. [Lan25] Y. Lan. A hybrid cnn-lstm model for stock price prediction with spatial and temporal dependencies. Applied and Computational Engineering, 155(1):236–242, 2025.
[9]. [LKY+24] Q. Li, N. Kamaruddin, S. S. Yuhaniz, et al. Forecasting stock price changes using a long short-term memory neural network with symbolic genetic programming. Scientific Reports, 14:422, 2024.
[10]. [MAKB24] I. A. Muhammad, I. Ahmed, N. Khwaja, and M. Bendechache. An explainable deep learning approach for stock market trend prediction. Heliyon, 10(21):e40095, 2024.
[11]. [MK25] Rahul Maheshwari and Vivek Kapoor. A deep approach for forecasting the nse opening index employing cnn-lstm hybrid framework. Procedia Computer Science, 258:170–182, 2025. International Conference on Machine Learning and Data Engineering.
[12]. [Mos25] S. M. Mostafavi. Key technical indicators for stock market prediction. International Journal of Data Science and Forecasting, 7(2):45–62, 2025. 10
[13]. [NPdO17] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira. Stock market's price movement prediction with lstm neural networks. In International Joint Conference on Neural Networks (IJCNN), pages 1419–1426. IEEE, 2017.
[14]. [RJ25] L. Ruan and H. Jiang. Stock price prediction using finbert-enhanced sentiment with shap explainability and differential privacy. Mathematics, 13(17):2747, 2025.
[15]. [SL25] G. Sun and Y. Li. Intraday and post-market investor sentiment for stock price prediction: A deep learning framework with explainability and quantitative trading strategy. Systems, 13(5):390, 2025.
[16]. [Tsa10] R. S. Tsay. Analysis of Financial Time Series. Wiley, New Jersey, 3 edition, 2010.
[17]. [vK24] J. Černevičien˙e and A. Kabašinskas. Explainable artificial intelligence (xai) in finance: a systematic literature review. Artificial Intelligence Review, 57(8):216, 2024.
[18]. [WH00] R. Wirth and J. Hipp. Crisp-dm: Towards a standard process model for data mining. In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, pages 29–39, 2000.
[19]. [ZW21] Y. Zhang and S. Wang. Data quality issues in machine learning and deep learning: A survey. Knowledge-Based Systems, 222:106964, 2021.

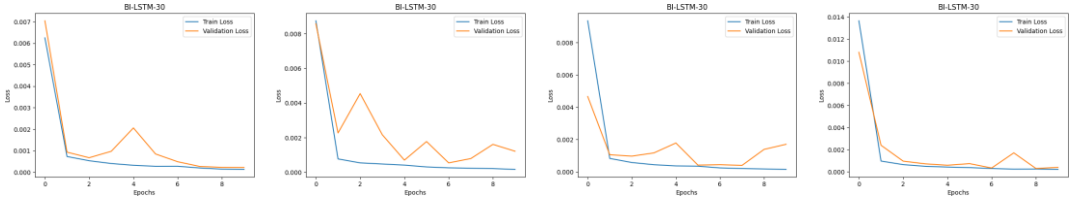
Appendix A. Train and validation curves



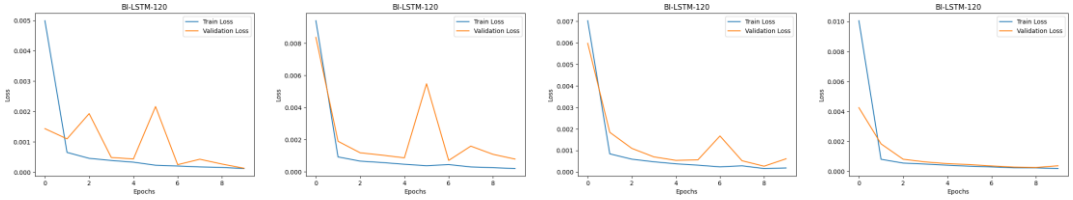
(a) LSTM 30 (b) (c) (d)



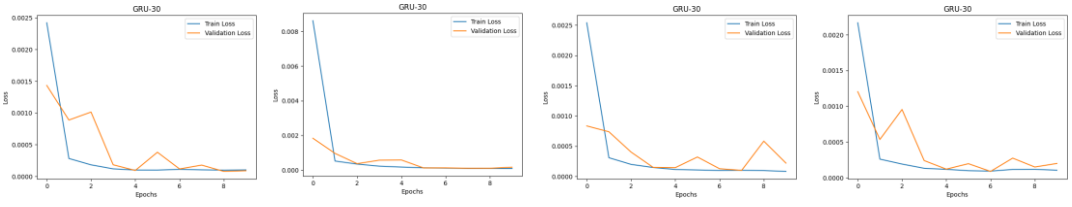
(a) LSTM 120 (b) (c) (d)



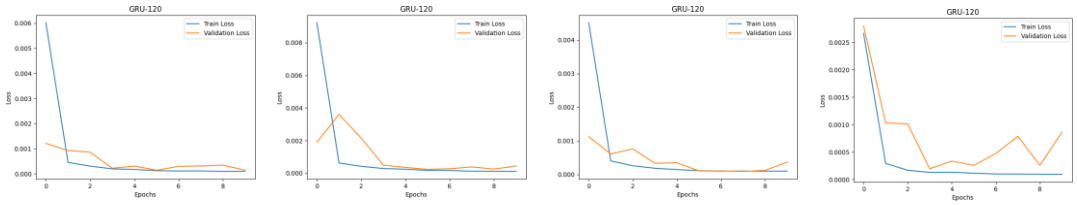
(a) BI-LSTM 30 (b) (c) (d)



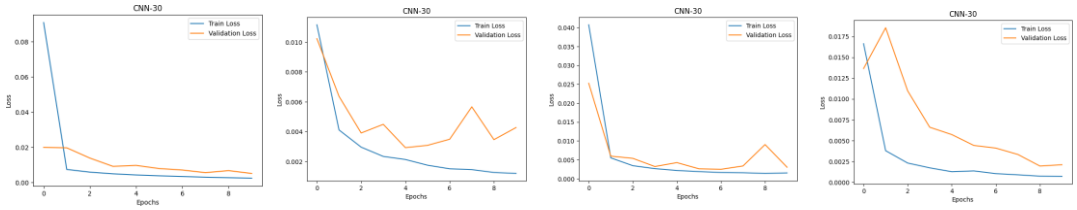
(a) BI-LSTM 120 (b) (c) (d)



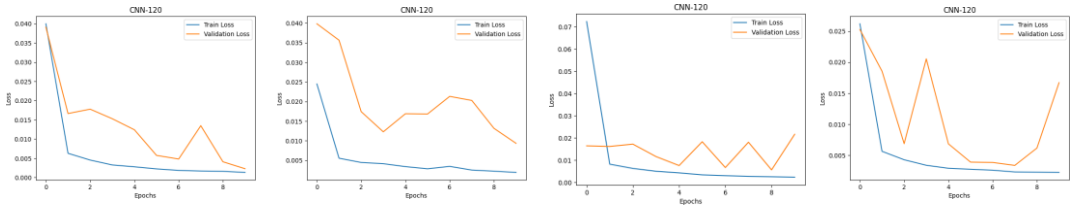
(a) GRU 30 (b) (c) (d)



(a) GRU 120 (b) (c) (d)



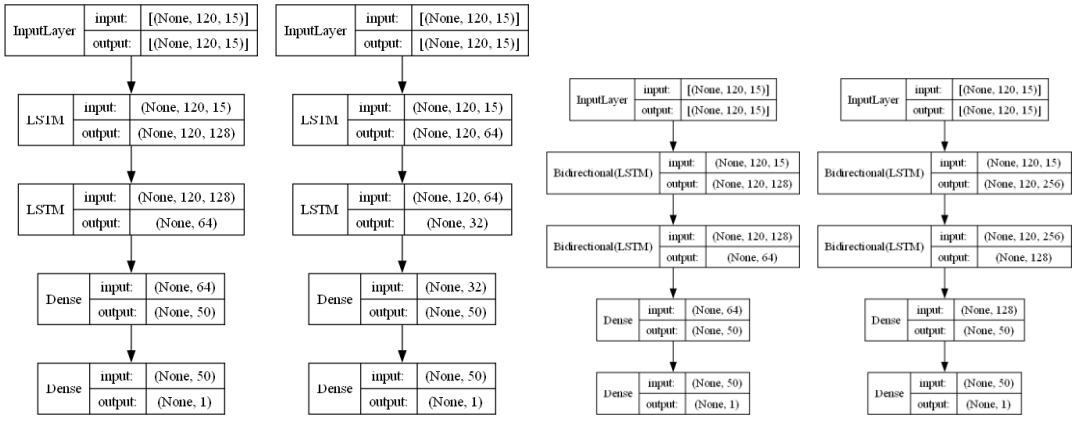
(a) CNN 30 (b) (c) (d)



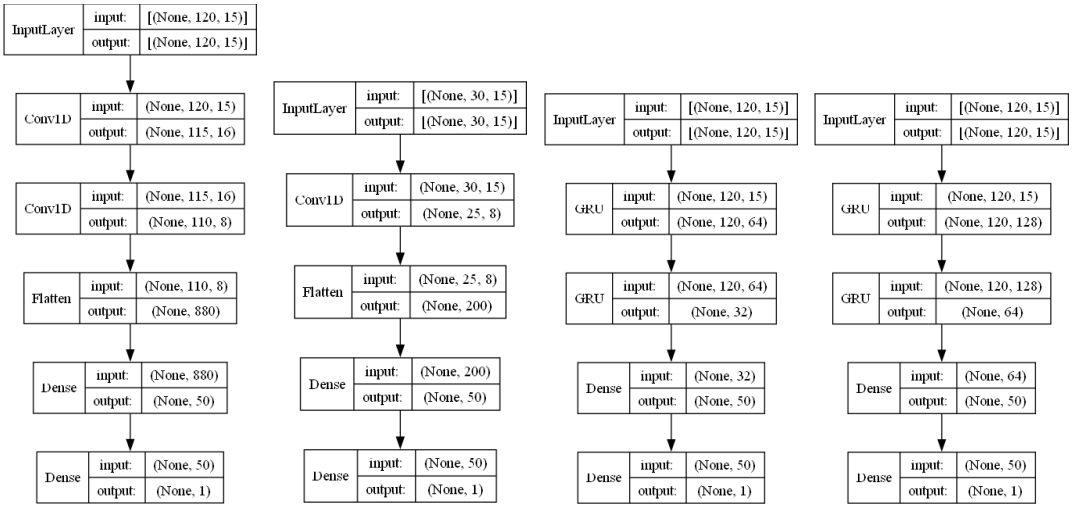
(a) CNN 120 (b) (c) (d)

Figure A.13: Train and validation curve Examples

Appendix B. Models' layers details



(a) LSTM Deep (b) LSTM Shallow (c) BI-LSTM Shallow (d) BI-LSTM Deep



(a) CNN Deep (b) CNN Shallow (c) GRU Shallow (d) GRU Deep

Figure B.15: Model Figures