mathematics

MDPI

Article

Comparative Evaluation of Deep Learning Architectures for Next-Day Stock Price Forecasting Using Technical Indicators

Theofanis Aravanis [1,*] and Andreas Kanavos [2,*]

[1] Department of Digital Systems, University of the Peloponnese, 23100 Sparta, Greece
[2] Department of Informatics, Ionian University, 49100 Corfu, Greece
* Correspondence: taravanis@uop.gr (T.A.); akanavos@ionio.gr (A.K.)

Abstract

Accurate next-day stock price forecasting remains challenging because daily price changes have a low signal-to-noise ratio and can be strongly affected by short-lived news shocks, order-flow imbalances, and abrupt changes in volatility or market sentiment. This study presents a controlled empirical comparison of deep learning architectures for next-day stock price forecasting using technical indicators. Using a decade-long daily dataset covering four large-cap NASDAQ equities (AAPL, META, SBUX, and TSLA), multivariate input sequences are constructed by combining historical prices with five widely used technical indicators: exponential moving average (EMA), relative strength index (RSI), moving average convergence divergence (MACD), on-balance volume (OBV), and average true range (ATR). Four deep sequence architectures—long short-term memory (LSTM), bidirectional LSTM (BiLSTM), gated recurrent unit (GRU), and convolutional LSTM (ConvLSTM)—are evaluated across multiple lookback windows (5, 15, and 30 trading days) and chronological train/validation/test splits (60–20–20, 70–15–15, and 80–10–10). Hyperparameters are optimized through random search, and forecasting performance is assessed on held-out test sets using normalized-scale root mean squared error (RMSE) and out-of-sample R^2. Within the examined fixed chronological partitions, ConvLSTM records the lowest observed RMSE for all four equities, attaining values between 0.0256 and 0.0394 and out-of-sample R^2 values above 0.90. Because the evaluation does not include walk-forward validation or formal statistical significance testing, these results should be interpreted as descriptive evidence within the present experimental setting rather than as proof of general architectural superiority. To assess practical utility, forecasts are translated into a transparent long-only trading rule that enters the market when the predicted next-day closing price exceeds the current closing price. Out-of-sample backtesting shows that the frictionless forecast-driven strategy achieves higher terminal cumulative returns than Buy-and-Hold for AAPL, SBUX, and TSLA, while Buy-and-Hold remains superior for META. Approximate five-day-frequency risk-adjusted estimates generally reinforce these relative patterns: the ConvLSTM strategy improves the Sharpe, Sortino, and Calmar ratios for AAPL, SBUX, and TSLA, although TSLA remains exposed to substantial drawdown risk. Transaction-cost sensitivity analysis further indicates that the terminal-return gains weaken under trading frictions and are particularly sensitive for AAPL. The findings demonstrate the value of evaluating forecasting architectures through both statistical and financial criteria, while emphasizing that lower point-forecast error does not necessarily translate into superior economic or risk-adjusted performance.

Keywords: stock price forecasting; technical indicators; deep learning; financial time series; convolutional LSTM; trading strategy evaluation

Academic Editors: Raymond Lee and Zengjing Chen

Received: 24 June 2026
Revised: 21 July 2026
Accepted: 29 July 2026
Published: 2 August 2026

MSC: 62M45; 68T07; 62M10; 91B84; 91G15

1. Introduction

Financial markets aggregate vast amounts of information into prices, yet short-horizon price movements remain difficult to predict reliably [1]. Equity time series are difficult to model at the next-day horizon because the signal-to-noise ratio of daily price changes is low, short-term returns are strongly affected by transient order-flow imbalances and news shocks, and the data-generating process can change rapidly after earnings announcements, macroeconomic releases, monetary-policy updates, or sector-specific events. These effects are especially relevant for one-step-ahead forecasting because the model must infer a small expected price movement from recent observations while being exposed to abrupt changes in volatility, trend direction, and market sentiment.

Financial returns also exhibit well-documented stylized properties, including volatility clustering, conditional heteroscedasticity, and deviations from simple random-walk behavior [2–5]. These characteristics further complicate the design of forecasting models that generalize out-of-sample. Despite these challenges, next-day forecasting continues to attract sustained interest because even small predictive edges can be valuable when translated into systematic decision rules. This argument is consistent with quantitative-finance research showing that modest return predictability can have economic value when evaluated through portfolio allocation, market timing, or mean–variance utility rather than statistical accuracy alone [6–8]. Furthermore, empirical findings suggesting mean-reverting behavior over certain horizons continue to motivate the search for exploitable predictive structure in financial markets [9].

Traditional approaches to short-horizon forecasting have ranged from linear time-series models and econometric specifications to modern machine-learning frameworks capable of capturing complex nonlinear relationships among financial variables. The importance of modeling nonlinear dependencies in economic and financial time series has long been recognized, as linear formulations are often unable to represent the complex dynamics observed in real-world markets [10,11]. In practice, a large body of trading and quantitative analysis relies on *technical indicators*—transformations of price, volume, and volatility intended to summarize trend, momentum, and market pressure [12]. Indicators such as the exponential moving average (EMA), relative strength index (RSI), moving average convergence divergence (MACD), on-balance volume (OBV), and average true range (ATR) are widely used, as they compactly encode interpretable market dynamics and can be computed consistently across assets and time periods [12,13]. Empirical finance has also examined the behavior of simple technical trading rules in historical data [14]. Nevertheless, the predictive value of indicators is often sensitive to market regimes and may not be captured well by models that impose strong linearity assumptions or require strict stationarity.

Deep learning models for sequential data offer a complementary perspective [15]. Recurrent neural networks (RNNs), including long short-term memory (LSTM) [16] and gated recurrent unit (GRU) [17] architectures, can model temporal dependencies and nonlinear interactions without requiring explicit parametric forms. Bidirectional variants can exploit contextual relationships within an observed input window [18]; when applied to forecasting, such models must be evaluated under strict time-ordered splits to avoid look-ahead bias. In addition, convolutional–recurrent hybrids can extract local patterns from multivariate sequences before integrating them over time [19,20], potentially improving sensitivity to short-term structures such as transient momentum bursts or volatility clustering. These

properties make deep architectures attractive candidates for next-day prediction when the input includes multiple indicator streams alongside raw prices. Accordingly, the literature increasingly adopts deep architectures for financial forecasting and trading-oriented prediction, as summarized in recent empirical studies and surveys [21–23].

A persistent gap in much of the forecasting literature is the connection between *predictive accuracy* and *practical utility*. Models that reduce error metrics do not necessarily produce profitable decisions once forecasts are converted into trades. Conversely, modest improvements in directional accuracy can sometimes be more useful than small gains in point-forecast root mean squared error (RMSE). Therefore, evaluating a forecasting model solely by statistical metrics can be misleading if the intended application is decision-making [24].

This article addresses these issues through a focused empirical investigation of next-day stock price forecasting for four large-cap NASDAQ-listed equities: Apple (AAPL), Meta Platforms (META), Starbucks (SBUX), and Tesla (TSLA). Using a decade-long daily dataset, we construct multivariate input sequences combining historical prices with five widely used technical indicators—EMA, RSI, MACD, OBV, and ATR—and evaluate four deep sequence architectures: LSTM [16], bidirectional LSTM (BiLSTM) [18], GRU [17], and convolutional LSTM (ConvLSTM) [19]. The models are assessed across multiple lookback windows (5, 15, and 30 trading days) and chronological train/validation/test partitions (60–20–20, 70–15–15, and 80–10–10), with hyperparameters selected through random search using KerasTuner (version 1.4.8) [25].

The present study does not propose a new neural architecture, learning algorithm, or optimization method. Its scientific contribution is empirical and evaluation-oriented, and the study is positioned as a controlled benchmark of established deep sequence architectures within a clearly defined next-day financial forecasting setting. Rather than claiming model-level methodological novelty, the analysis examines whether architectural differences remain observable when the dataset, technical-indicator representation, forecasting target, preprocessing pipeline, hyperparameter-tuning procedure, and chronological evaluation protocol are held constant.

Within this scope, the study makes three contributions. First, it provides a controlled comparison of LSTM, BiLSTM, GRU, and ConvLSTM using the same technical-indicator-enhanced multivariate representation across four equities with heterogeneous market behavior. Second, it evaluates the sensitivity of the architectures to multiple lookback windows and chronological train/validation/test partitions, thereby examining whether their relative performance remains consistent across alternative experimental settings. Third, it connects point-forecast accuracy with terminal-return, transaction-cost, and approximate risk-adjusted evaluations. This combined analysis demonstrates that the architecture producing the lowest forecasting error does not necessarily provide the strongest economic or risk-adjusted performance. The contribution, therefore, lies in the controlled evaluation design and in the empirical distinction between statistical forecasting quality and practical financial usefulness, rather than in the introduction of a new forecasting model. Accordingly, the reported architecture rankings should be interpreted as hypothesis-generating comparative evidence within the defined experimental setting, rather than as a definitive benchmark of general model superiority.

Within the examined configurations, the observed ConvLSTM results suggest that convolutional–recurrent processing may be beneficial for next-day forecasting; however, this descriptive pattern does not establish broader or statistically significant architectural superiority. The paper also emphasizes the limits of inference from backtests. In particular, any conclusions regarding deployable profitability must be tempered by realistic trading frictions (transaction costs, bid–ask spreads, slippage, and market impact) and by the need

for robustness checks across time periods and market regimes. The objective is not to claim a universally profitable trading system, but to evaluate whether technical-indicator-based deep learning forecasts contain actionable structure under a clear and reproducible experimental protocol.

The remainder of the article is organized as follows: Section 2 reviews related work on technical-indicator forecasting and deep learning for financial time series. Section 3 describes the dataset, feature construction, and preprocessing pipeline. Section 4 presents the forecasting models and the hyperparameter tuning procedure. Section 5 details the experimental design, evaluation metrics, and the forecast-driven trading rule. Section 6 presents the empirical results and discusses their implications. Section 7 concludes with limitations and directions for future work. Finally, Appendix A provides the mathematical formulations of the technical indicators used in the study.

2. Related Work

The literature on stock-market forecasting has evolved from traditional technical trading rules and statistical models to machine-learning and deep-learning approaches capable of extracting complex temporal patterns from financial data. Alongside advances in forecasting algorithms, increasing attention has been devoted to the role of technical indicators, model evaluation protocols, and the relationship between predictive accuracy and trading performance. This section reviews the most relevant strands of the literature and positions the present study within the broader context of financial forecasting research.

Research on stock forecasting using historical market data spans technical trading rules, classical machine-learning approaches based on engineered features, and more recent deep-learning architectures capable of learning temporal representations directly from financial time series. Comprehensive reviews have documented the evolution of forecasting methodologies from statistical and soft-computing approaches toward increasingly sophisticated machine-learning frameworks [26]. Recent research has also demonstrated the growing importance of machine-learning techniques in empirical asset pricing and financial prediction tasks, further motivating the adoption of nonlinear modeling approaches in financial applications [11]. Early studies examined whether transformations of past prices and volume contain exploitable information, reporting mixed evidence regarding the profitability of technical trading rules [14,27]. Subsequent work highlighted the importance of rigorous evaluation procedures, showing that apparent gains may be inflated by data snooping and that results often depend on market conditions, testing protocols, and transaction costs [28–30].

Before the widespread adoption of deep learning, many studies used technical indicators as engineered inputs for shallow neural networks, support vector machines, and hybrid forecasting systems. Previous work demonstrated that indicator-based feature engineering could improve stock prediction performance across different markets and forecasting settings [31–35]. Collectively, these studies established the practical usefulness of technical indicators, while also revealing the limitations of manually specified predictors and models that struggle to capture long-range nonlinear temporal dependencies.

The transition to deep learning shifted the literature toward representation learning from sequential financial data. Early applications of deep learning demonstrated that neural architectures could model complex temporal dependencies more effectively than many traditional forecasting approaches, including LSTM-based forecasting frameworks and hybrid deep-learning models combining feature extraction and sequential prediction components [36–40]. A large-scale empirical study [21] showed that LSTM-based models can outperform several benchmark approaches in predicting daily stock movements, while later surveys confirmed the growing dominance of recurrent and hybrid deep-learning ar-

chitectures in financial forecasting research [22,23]. More recent studies have also leveraged high-frequency and limit-order-book representations, further demonstrating the flexibility of deep-learning architectures for extracting predictive information from complex financial data structures [41].

A parallel line of work augments recurrence with convolutional feature extraction to capture local temporal motifs before longer-range sequence integration. Convolution-based approaches have been applied both to transformed technical-indicator representations and to multivariate financial inputs, demonstrating the usefulness of local pattern extraction in forecasting and trading-oriented tasks [42,43]. Particularly relevant to the present study, ConvLSTM-based approaches have shown promise in learning hierarchical temporal representations from structured financial sequences [44]. Although originally developed outside the financial domain, ConvLSTM combines convolutional feature extraction with gated temporal modeling, making it a natural candidate for forecasting tasks based on multivariate technical-indicator windows.

Recent time-series forecasting research has expanded beyond recurrent and convolutional–recurrent architectures to include attention-based, Transformer-based, and multilayer-perceptron-based models. For example, the Temporal Fusion Transformer combines recurrent processing, variable-selection mechanisms, and interpretable self-attention for multivariate forecasting [45], while Informer introduces an efficient sparse-attention mechanism for long-sequence prediction [46]. More recently, TSMixer has demonstrated that temporal and cross-variable interactions can also be modeled through multilayer-perceptron mixing operations [47]. These architectures represent important contemporary forecasting alternatives. However, the present study is intentionally restricted to a controlled comparison of recurrent and convolutional–recurrent architectures and therefore does not constitute a comprehensive benchmark against the current state of the art.

Several recent studies lie especially close to the setting considered herein, as they combine daily market variables with technical indicators and then assess predictive or trading performance. Prior work has demonstrated that both forecasting accuracy and trading performance depend strongly on the selected indicator set, model architecture, and forecast-to-trade mapping [48–51]. Forecast performance can also depend on the length of the historical input window used to construct model features, an issue that has received comparatively less attention despite its potential influence on model behavior and generalization [52]. These findings highlight the importance of evaluating forecasting models under consistent datasets, feature sets, and validation procedures. Another important research direction has explored classification-oriented formulations of financial prediction problems using deep neural networks, demonstrating that machine-learning approaches can be effective even when the forecasting objective is transformed into a directional decision task [53].

Another notable distinction in the literature concerns the prediction target itself. Many deep-learning studies formulate the problem as a classification task, focusing on directional movement prediction rather than direct estimation of future price levels [21,38,49]. While directional formulations are attractive for trading-rule design, they do not directly evaluate the accuracy of next-day price forecasts. In contrast, regression-based approaches aim to estimate future closing prices explicitly, providing a different perspective on forecasting performance and practical utility [48].

A further important issue concerns model evaluation. Forecasting accuracy and economic decision value are related but not equivalent concepts [54]. In financial applications, improvements in statistical error metrics do not necessarily translate into superior trading outcomes, particularly when trading rules, directional errors, transaction costs, and market frictions are taken into account [21,49]. Moreover, methodological studies have highlighted

the importance of using evaluation procedures that respect temporal dependencies and avoid information leakage [24,55]. These considerations motivate the strictly chronological train/validation/test partitions adopted in the present study and the explicit separation between forecasting accuracy and trading-performance evaluation.

Although numerous comparative studies of financial forecasting methods already exist, their findings are often difficult to compare directly because of substantial differences in datasets, input features, prediction targets, lookback horizons, data-partitioning procedures, evaluation metrics, and trading assumptions. Therefore, the purpose of the present comparison is not to address an absence of comparative research, but to provide a controlled evaluation in which the dataset, technical-indicator representation, forecasting target, hyperparameter-tuning procedure, and chronological evaluation protocol are held constant across architectures. This design helps distinguish architecture-related performance differences from those arising from heterogeneous experimental settings. In addition, the joint assessment of forecasting accuracy and trading performance makes it possible to examine whether improvements in statistical prediction translate into greater economic utility.

Deep learning is not the only viable approach to financial time-series forecasting. Classical statistical models, such as ARIMA [56], and non-neural machine-learning methods, including Random Forests, gradient-boosting models, XGBoost, and support vector machines, constitute important forecasting alternatives and comparative baselines. Recent adaptive forest-based approaches, such as improved multi-granularity cascade forest frameworks, combine multi-granularity temporal processing, tree-based feature derivation, and embedded feature selection to improve forecasting performance in complex dynamic environments [57]. These complementary capabilities may be particularly beneficial for high-dimensional, heterogeneous, and noisy inputs. Since the present study does not evaluate these models using the same financial dataset and experimental protocol, no theoretical or empirical superiority of ConvLSTM over adaptive tree-based approaches is claimed. The present study intentionally adopts a narrower scope by comparing deep sequence architectures under a common experimental protocol. These architectures were selected because they can directly process ordered multivariate lookback windows and learn nonlinear temporal dependencies and interactions among technical indicators without requiring the sequence to be represented solely through manually specified lag features. This choice does not imply that deep learning is universally necessary or superior to statistical and non-neural methods; rather, the objective is to investigate how different recurrent and convolutional–recurrent mechanisms perform when the dataset, input representation, forecasting target, and evaluation procedure are held constant.

Collectively, the literature suggests that technical indicators remain a practical means of enriching daily market data, while recurrent and convolutional–recurrent architectures represent some of the most promising approaches for modeling nonlinear temporal dependencies in financial time series. At the same time, existing studies differ substantially in terms of datasets, feature sets, forecasting targets, evaluation methodologies, and trading assumptions, making direct comparisons difficult. Motivated by these observations, the present study provides a systematic comparison of LSTM, BiLSTM, GRU, and ConvLSTM architectures using a common set of widely used technical indicators (EMA, RSI, MACD, OBV, and ATR) across multiple lookback horizons and evaluation settings. In addition to conventional forecasting metrics, the study investigates whether improvements in statistical prediction accuracy translate into superior out-of-sample trading performance.

3. Dataset and Feature Construction

For the conducted analysis, we use a publicly available daily equity dataset covering ten large, actively traded NASDAQ-listed companies—Apple, Starbucks, Microsoft, Cisco Systems, Qualcomm, Meta Platforms, Amazon, Tesla, Advanced Micro Devices, and Netflix [58]. Each record corresponds to one company on one trading day and includes `Company` (ticker), `Date`, `Close/Last`, `Volume`, `Open`, `High`, and `Low`. The sample spans approximately a decade of daily observations (18 July 2013 to 17 July 2023) and was collected via web scraping from publicly available NASDAQ webpages.

For the experiments in this study, we focus on four large-cap equities: Apple (AAPL), Meta Platforms (META), Starbucks (SBUX), and Tesla (TSLA). These stocks were selected to provide a compact but heterogeneous evaluation set drawn from the available dataset. The selection was guided by three criteria: (i) sectoral diversity, since the selected companies represent different business areas, including consumer technology, social media and digital advertising, consumer services, and electric vehicles; (ii) heterogeneity in price dynamics, since the selected equities exhibit different levels of trend persistence, volatility, growth behavior, and drawdown patterns; and (iii) data continuity and liquidity, since all four stocks have long daily trading histories and are actively traded large-cap NASDAQ equities. The purpose of this selection is not to represent the entire equity market, but to test the forecasting models on assets with different market behaviors under a consistent experimental protocol.

Figure 1 presents the daily closing-price evolution of the four selected equities over the full sample period. The series show substantial differences in price level, trend structure, and volatility. META and TSLA exhibit pronounced growth phases followed by sharp corrections, while AAPL displays a smoother long-term upward trajectory. SBUX shows comparatively lower volatility and a more moderate price range. These differences support the use of multiple equities rather than a single stock, as they allow the forecasting models to be evaluated under heterogeneous market dynamics.

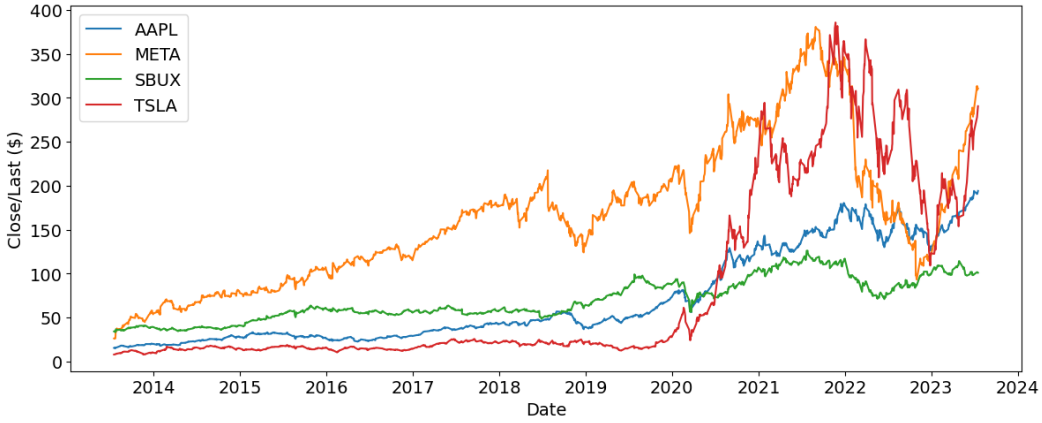

Figure 1. Daily closing price evolution of AAPL, META, SBUX, and TSLA (2013–2023).

The raw data are first filtered by ticker and sorted chronologically by `Date`. Dates are parsed into a standard datetime format, and non-numeric characters, such as currency symbols, are removed from the price fields before casting them to floating-point values. Observations with missing values are discarded, including the initial warm-up period in which certain technical indicators are undefined due to their lookback windows. Because the analysis is conducted on a per-asset basis, each ticker's time series is processed independently to prevent cross-asset leakage.

To construct the model inputs, we combine the closing price with a compact set of widely used technical indicators intended to summarize trend, momentum, volatility, and

volume dynamics [12,13]. Specifically, we construct the exponential moving average (EMA), relative strength index (RSI), moving average convergence divergence (MACD), on-balance volume (OBV), and average true range (ATR). All indicators are computed causally from the available OHLCV fields, and the first observations for which an indicator is undefined are removed. The exact formulas and parameterizations used are provided in Appendix A.

Table 1 summarizes the input variables used for forecasting. The selected features combine the original closing price with indicators capturing complementary aspects of market behavior: trend, momentum, volume pressure, and volatility.

Table 1. Input features used for forecasting.

Feature	Description
Close	Daily closing price
EMA	Exponential Moving Average
RSI	Relative Strength Index
MACD	Moving Average Convergence Divergence
OBV	On-Balance Volume
ATR	Average True Range

The forecasting task is next-day closing price prediction. Let P_t denote the closing price (*Close/Last*) on day t, and let $\mathbf{x}_t \in \mathbb{R}^d$ denote the feature vector at day t, consisting of the scaled closing price and the indicator values. In this study, each daily feature vector contains six variables: the closing price together with EMA, RSI, MACD, OBV, and ATR. Given a lookback length L, each supervised sample is formed as the input sequence $\mathbf{X}_t = [\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t]$ with target $y_t = P_{t+1}$. We evaluate $L \in \{5, 15, 30\}$ to study how different historical contexts affect next-day forecasting performance. These values were selected to represent short-, medium-, and longer-term input horizons commonly used in daily financial forecasting studies while also enabling an assessment of model sensitivity to the amount of historical information available [52].

All features are scaled using Min–Max normalization fitted on the training segment of each ticker's time series, and the same scaler is applied to the validation and test segments to avoid information leakage. Model predictions are inverse-transformed back to the original price scale for plotting and for the trading-rule analysis. Forecasting metrics, including RMSE and out-of-sample R^2, are computed on the Min–Max-normalized target scale.

To preserve temporal causality, each ticker's series is split chronologically into training, validation, and test segments. We consider three split configurations (60–20–20, 70–15–15, and 80–10–10) to assess robustness to the amount of training data and reduce sensitivity to any single partition. All modeling choices, including hyperparameter selection, are based solely on the training and validation segments, and final results are reported exclusively on the held-out test segment.

4. Deep Learning Models and Training Procedure

We formulate next-day closing-price forecasting as a one-step-ahead supervised learning problem in which a model receives a multivariate sequence of historical observations and predicts the closing price of the subsequent trading day. For a lookback length L, each model maps a multivariate window $\mathbf{X}_t \in \mathbb{R}^{L \times d}$ to a scalar prediction $\widehat{P}_{t+1}$. This section presents the deep learning architectures considered in the study together with the training, regularization, and hyperparameter-optimization procedures applied consistently across all assets and experimental configurations.

4.1. Model Architectures

To investigate the effect of architectural design on forecasting performance, we compare four representative sequence-model families that differ in the way temporal dependencies are encoded and aggregated: LSTM [16], GRU [17], bidirectional LSTM (BiLSTM) [18], and ConvLSTM [19]. LSTM and GRU are recurrent baselines designed to capture nonlinear temporal dependencies via gated hidden-state dynamics, with GRU offering a parameter-efficient alternative. BiLSTM computes forward and backward representations within the observed input window and concatenates them; in our setup, bidirectionality is confined to the historical window and does not introduce access to future test observations. ConvLSTM combines convolutional operators with LSTM-style gating to learn short-range motifs in multivariate sequences and aggregate them over time.

The LSTM, GRU, and BiLSTM models share the same high-level backbone, i.e., two stacked recurrent layers followed by a linear regression head. The first recurrent layer returns a sequence of hidden states, while the second layer returns a single summary vector. Batch Normalization [59] and Dropout [60] are applied after each recurrent layer to stabilize training and reduce overfitting. The final Dense layer has one unit and linear activation to produce $\widehat{P}_{t+1}$. For BiLSTM, each recurrent layer is wrapped with a bidirectional operator and the forward/backward outputs are concatenated.

ConvLSTM replaces the affine transformations inside the LSTM gates with convolutional operations [19]. Let N denote the number of samples, L the lookback length, and $d = 6$ the number of input features. The original input to all architectures is therefore represented as $\mathbf{X} \in \mathbb{R}^{N \times L \times d}$. Because the standard `ConvLSTM2D` layer expects a five-dimensional tensor under the channels-last convention, the input is reshaped as

$$\mathbf{Z} \in \mathbb{R}^{N \times L \times 1 \times d \times 1}, \qquad Z_{n,\ell,1,j,1} = X_{n,\ell,j}. \tag{1}$$

The five dimensions correspond respectively to the batch size, recurrent time steps, spatial height, feature-axis width, and input channels. For example, a lookback window of $L = 15$ is represented as a tensor of shape $N \times 15 \times 1 \times 6 \times 1$. The ConvLSTM recurrent state evolves across the L historical observations, while the adopted $(1, 3)$ convolutional kernel operates across groups of three neighboring positions on the feature axis at each recurrent step. Consequently, the convolutional component learns localized combinations among the supplied price and technical-indicator features, while the recurrent gates aggregate information across the ordered lookback window. The final ConvLSTM feature map is passed through Batch Normalization and Dropout and is subsequently flattened before being mapped to the scalar next-day price forecast by a linear Dense output layer.

Figure 2 summarizes the common forecasting pipeline adopted in this study. All models receive the same multivariate input window and share the same normalization, regularization, and prediction stages. The only component that varies across configurations is the sequence encoder, which is instantiated as an LSTM, GRU, BiLSTM, or ConvLSTM architecture.

4.2. Training Objective, Regularization, and Optimization

All models are trained to minimize mean squared error (MSE) on the training set, while mean absolute error (MAE) is monitored during training for additional diagnostics. To ensure a fair comparison, identical optimization and regularization principles are applied across all architectures whenever applicable. To mitigate overfitting, recurrent and ConvLSTM layers employ Dropout and an L_1–L_2 kernel regularizer. Optimization uses one of {`Adam`, `SGD`, `RMSprop`, `AdaGrad`} with a tuned learning rate.

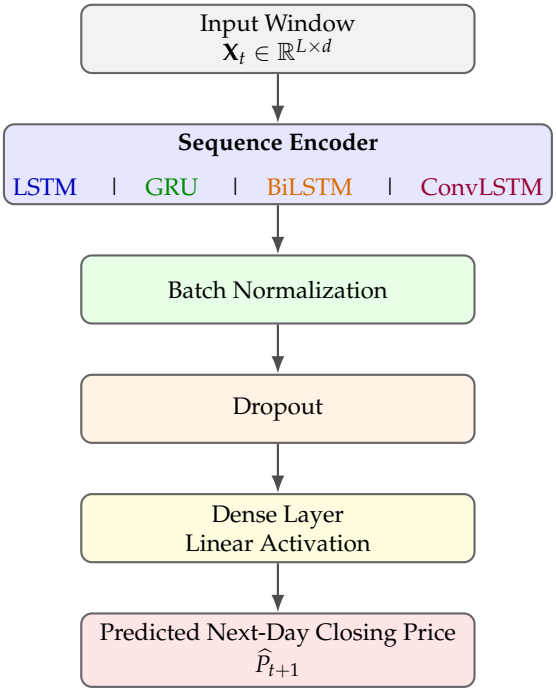

Figure 2. Common forecasting pipeline used across the evaluated deep learning architectures.

4.3. Hyperparameter Tuning and Early Stopping

Hyperparameters are selected with KerasTuner `RandomSearch` [25] by minimizing validation loss (`val_loss`). For each configuration, defined by asset, lookback window L, split ratio, and architecture, the tuner evaluates up to five trials, each trained for 100 epochs with batch size 32. The number of random-search trials was set to five as a fixed and computationally feasible tuning budget applied uniformly across all experimental configurations. This choice was motivated by the size of the complete experimental design: four stocks, four architectures, three lookback windows, and three chronological split ratios, resulting in 144 forecasting configurations. Since tuning is performed separately for each configuration, five trials already require up to 720 hyperparameter-search training runs, followed by final retraining of the selected models. A full grid search over the reported hyperparameter space would lead to a combinatorial number of candidate models and was therefore impractical for the present comparative study. Random search was preferred because it provides a more efficient exploration strategy under limited computational budgets, particularly when only a subset of hyperparameters has a dominant influence on validation performance [61]. The best trial is then rebuilt and retrained using Early Stopping on validation loss, with patience set to 10 and best weights restored. In this final training stage, batch size is selected from $\{16, 32, 64\}$ and the maximum number of epochs from $\{50, 100, 200\}$.

Although random search provides a simple and reproducible strategy for exploring the hyperparameter space under a fixed computational budget, more advanced optimization methods could also be considered. Bayesian optimization, Hyperband, evolutionary search, and multi-fidelity optimization strategies may improve search efficiency by adaptively allocating trials toward more promising regions of the hyperparameter space [62,63]. Such approaches can be particularly useful when the objective function is expensive to evaluate, as in deep learning models for financial time series. In the present study, random search was selected to maintain a consistent and transparent tuning protocol across all assets, architectures, lookback windows, and split configurations. A direct comparison between random search and more advanced hyperparameter-optimization strategies is beyond the scope of the present work, but represents an important direction for future research.

Table 2 summarizes the hyperparameter search space and training settings. The selected ranges were chosen to provide sufficient architectural flexibility while maintaining a computationally feasible tuning procedure across all assets, lookback windows, and train/validation/test configurations.

Table 2. Hyperparameter search space and training settings.

Item	Values
Lookback window L	$\{5, 15, 30\}$
Recurrent units (Layer 1)	10–100 (step 10)
Recurrent units (Layer 2)	5–50 (step 5)
ConvLSTM filters	8–64 (step 8)
ConvLSTM kernel size (height$\times$feature axis)	$(1, 3)$
Activation function	{`tanh`, `relu`}
Dropout rate	$\{0.1, 0.2, 0.3, 0.4, 0.5\}$
L_1 coefficient	$\{0, 10^{-6}, 10^{-5}, 10^{-4}\}$
L_2 coefficient	$\{0, 10^{-6}, 10^{-5}, 10^{-4}\}$
Optimizer	{`Adam`, `SGD`, `RMSprop`, `AdaGrad`}
Learning rate	$\{5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}\}$
Random search trials	5
Epochs per trial/batch size	100/32
Final training epochs/batch size	$\{50, 100, 200\}/\{16, 32, 64\}$

To ensure fair comparison, all architectures are trained using the same preprocessing pipeline, chronological data partitions, tuning methodology, and evaluation procedure. Consequently, differences in forecasting performance can be attributed primarily to architectural characteristics rather than inconsistencies in experimental settings.

5. Experimental Design and Evaluation Metrics

This section describes the experimental protocol used to compare the forecasting architectures and evaluate their performance. The analysis considers both statistical forecasting accuracy and practical trading utility, enabling the relationship between predictive performance and decision-making outcomes to be examined under a consistent evaluation framework. All experiments are conducted separately for each stock to avoid cross-asset leakage and to account for the distinct statistical characteristics of individual equity time series.

5.1. Experimental Setup

For each stock and each model family (LSTM, BiLSTM, GRU, ConvLSTM), we evaluate settings formed by the Cartesian product of (i) lookback window length $L \in \{5, 15, 30\}$ and (ii) chronological split ratio in {60–20–20, 70–15–15, 80–10–10} (train/validation/test). This yields $3 \times 3 = 9$ configurations per model per stock and 36 configurations per stock when considering all four architectures. Considering the four examined equities, the complete experimental campaign comprises 144 forecasting configurations, providing a broad basis for comparative evaluation across architectures, lookback horizons, and data-partition settings.

After training, next-day forecasts are generated on the held-out test inputs $\mathbf{X}_{\text{test}}$. Forecasts are produced on the Min–Max-normalized scale used during training. Predictions are inverse-transformed back to the original price scale only for plotting and for the trading-rule analysis, using the scaler fitted on the training data.

The use of multiple chronological train/validation/test partitions provides a controlled way to assess sensitivity to the amount of training and test data while preserving temporal ordering. However, this protocol does not replace walk-forward or rolling-

window validation, where models are repeatedly re-estimated over time. Therefore, the reported results should be interpreted as evidence under fixed out-of-sample chronological partitions rather than as a complete assessment of performance under continual model updating. Walk-forward and rolling-window evaluation are left as important extensions for future work.

5.2. Forecasting Evaluation

Forecasting performance is evaluated using widely adopted regression metrics that capture average prediction error, error magnitude, and explained variance.

Let $\{P_i\}_{i=1}^n$ denote the true next-day closing prices in the test set on the Min–Max-normalized scale and $\{\widehat{P}_i\}_{i=1}^n$ the corresponding predictions on the same normalized scale. We report mean squared error (MSE),

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(P_i - \widehat{P}_i)^2, \tag{2}$$

root mean squared error (RMSE),

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \tag{3}$$

and mean absolute error (MAE),

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|P_i - \widehat{P}_i|. \tag{4}$$

MSE and RMSE emphasize larger prediction errors, while MAE provides a direct measure of average absolute deviation. Together, these metrics provide complementary views of point-forecast accuracy.

We also report the out-of-sample (test-set) coefficient of determination,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(P_i - \widehat{P}_i)^2}{\sum_{i=1}^{n}(P_i - \overline{P})^2}, \tag{5}$$

where $\overline{P}$ is the mean of $\{P_i\}_{i=1}^n$. All forecasting metrics are computed on the Min–Max-normalized target scale, with the scaler fitted exclusively on the training segment and subsequently applied to the validation and test segments to avoid information leakage.

5.3. Trading Evaluation

While forecasting metrics quantify statistical accuracy, they do not directly indicate whether forecasts can support economically useful trading decisions. To provide a practical perspective on model usefulness, forecasts are additionally evaluated through a simple rule-based trading framework.

The Buy-and-Hold strategy serves as a passive benchmark representing continuous market exposure throughout the test period. Its daily return is computed from consecutive actual closing prices as

$$R_t = \frac{P_{t+1}}{P_t} - 1. \tag{6}$$

For the forecast-driven strategy, a binary trading signal is generated each day by comparing the predicted next closing price to the current actual closing price:

$$S_t = \begin{cases} 1, & \text{if } \widehat{P}_{t+1} > P_t, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The signal S_t is formed at the end of day t using information available up to P_t, and the strategy earns the close-to-close return from day t to day $t+1$ whenever $S_t = 1$. The resulting model-based strategy return is therefore defined as

$$R_t^{\text{model}} = S_t \times R_t. \tag{8}$$

To examine the sensitivity of the forecast-driven strategy to trading frictions, we additionally evaluate cumulative returns under different transaction-cost assumptions. Let c denote the one-way proportional transaction cost, combining commissions and slippage. The cost is applied whenever the strategy changes position, that is, when it enters or exits the market. With the initial previous position set to zero, daily turnover is computed as

$$\tau_t = |S_t - S_{t-1}|. \tag{9}$$

The transaction-cost-adjusted return of the forecast-driven strategy is therefore defined as

$$R_{t,c}^{\text{model}} = S_t R_t - c\tau_t, \tag{10}$$

where $c \in \{0.001, 0.002, 0.003, 0.004, 0.005\}$, corresponding to transaction costs from 0.1% to 0.5%.

For Buy-and-Hold, the terminal cumulative-return multiple over the test horizon is computed as

$$CR^{\text{B\&H}} = \prod_t (1 + R_t). \tag{11}$$

For the frictionless forecast-driven strategy, cumulative return is computed as

$$CR^{\text{model}} = \prod_t (1 + R_t^{\text{model}}), \tag{12}$$

while for the transaction-cost-adjusted strategy it is computed as

$$CR_c^{\text{model}} = \prod_t \left(1 + R_{t,c}^{\text{model}}\right). \tag{13}$$

To complement the terminal cumulative return with risk-adjusted performance measures, we additionally estimate the annualized return, the Sharpe ratio, the Sortino ratio, the maximum drawdown (MDD), and the Calmar ratio. These measures are calculated from cumulative-wealth observations sampled at five-trading-day intervals. Let W_k denote cumulative wealth at sampled observation k, with $W_0 = 1$, and let the corresponding five-day return be

$$R_k^{(5)} = \frac{W_k}{W_{k-1}} - 1. \tag{14}$$

Assuming 252 trading days per year and a zero risk-free rate, annualized return is estimated as

$$R_{\text{ann}} = W_K^{\frac{252}{5K}} - 1, \tag{15}$$

where K denotes the number of five-day return observations. Let $\overline{R}^{(5)}$ and s_5 denote the mean and sample standard deviation of the five-day returns. The annualized Sharpe ratio is calculated as

$$\text{Sharpe} = \sqrt{\frac{252}{5}}\frac{\overline{R}^{(5)}}{s_5}. \tag{16}$$

The annualized Sortino ratio is calculated using downside deviation:

$$\text{Sortino} = \sqrt{\frac{252}{5}}\frac{\overline{R}^{(5)}}{\sqrt{\frac{1}{K}\sum_{k=1}^{K}\min\left(R_k^{(5)},0\right)^2}}. \tag{17}$$

Let $M_k = \max_{0\le j\le k} W_j$ denote the running wealth maximum. Maximum drawdown and the Calmar ratio are defined as

$$\text{MDD} = \max_k\left(\frac{M_k - W_k}{M_k}\right), \qquad \text{Calmar} = \frac{R_{\text{ann}}}{\text{MDD}}. \tag{18}$$

Because supervised sequences of length L shift the effective start of the test examples, the current and next actual prices used for trading evaluation are aligned with the first available test prediction after accounting for the lookback window and split boundaries.

6. Results and Discussion

This section presents the empirical results obtained from the comparative evaluation of the four deep learning architectures across the examined equities. The analysis considers both forecasting accuracy and trading performance, enabling the relationship between predictive quality and practical decision-making outcomes to be investigated. We first compare the best-performing configurations of each architecture using quantitative forecasting and trading metrics. Subsequently, stock-specific visualizations are examined to assess prediction behavior over time and identify differences in tracking performance across models. Finally, the results are synthesized through a cross-asset comparison that highlights common patterns and asset-specific characteristics.

6.1. Forecasting Performance Comparison

Table 3 presents the best out-of-sample results achieved by each architecture for each stock across all evaluated experimental configurations. For every model, the table reports the configuration yielding the lowest test RMSE together with the corresponding out-of-sample R^2, cumulative Buy-and-Hold return, cumulative return of the forecast-driven trading strategy, selected lookback window, data split, activation function, and optimizer.

Table 4 presents a cross-asset statistical summary based on the best out-of-sample RMSE obtained by each architecture for each equity. ConvLSTM achieves the lowest mean RMSE (0.0300) and median RMSE (0.0275), together with a mean rank of 1.00 and four first-place results. GRU obtains the second-best mean rank of 2.75, followed by LSTM and BiLSTM with mean ranks of 3.00 and 3.25, respectively. The mean RMSE of ConvLSTM is approximately 27.4% lower than that of GRU, 28.8% lower than that of LSTM, and 33.3% lower than that of BiLSTM. These aggregated results demonstrate the consistency of the observed ConvLSTM performance across the four examined equities and complement the stock-specific comparisons reported in Table 3.

The results presented in Tables 3 and 4 reveal several consistent patterns. First, ConvLSTM records the lowest observed RMSE for all four equities. This descriptive pattern suggests that combining convolutional feature extraction with recurrent temporal modeling may be beneficial within the examined configurations. The improvement is particularly

pronounced for AAPL and META, where ConvLSTM reduces RMSE from 0.0362 to 0.0260 and from 0.0388 to 0.0256, respectively, relative to the strongest recurrent competitor. Similar gains are observed for SBUX and TSLA, where ConvLSTM achieves RMSE values of 0.0290 and 0.0394, outperforming all alternative architectures. ConvLSTM also attains the highest or near-highest out-of-sample R^2 values across the examined equities, reaching 0.9770 for META and remaining above 0.90 for all four stocks.

Table 3. Best out-of-sample performance per stock and architecture.

Stock	Model	L	Split	Activation	RMSE	R^2	B&H Return	Strategy Return	Optimizer
AAPL	LSTM	15	80–10–10	`relu`	0.0362	0.8828	1.26×	1.02×	`RMSprop`
AAPL	BiLSTM	5	60–20–20	`tanh`	0.0385	0.8260	1.30×	1.29×	`RMSprop`
AAPL	GRU	5	70–15–15	`relu`	0.0414	0.8827	1.27×	1.15×	`RMSprop`
AAPL	ConvLSTM	15	80–10–10	`tanh`	0.0260	0.9220	1.26×	1.27×	`Adam`
META	LSTM	15	80–10–10	`tanh`	0.0447	0.9643	1.76×	1.64×	`RMSprop`
META	BiLSTM	15	80–10–10	`relu`	0.0416	0.9506	1.76×	1.31×	`RMSprop`
META	GRU	5	80–10–10	`tanh`	0.0388	0.9693	1.85×	1.41×	`RMSprop`
META	ConvLSTM	5	80–10–10	`tanh`	0.0256	0.9770	1.85×	1.53×	`RMSprop`
SBUX	LSTM	30	80–10–10	`tanh`	0.0415	0.7961	1.22×	1.20×	`RMSprop`
SBUX	BiLSTM	15	80–10–10	`tanh`	0.0410	0.8961	1.23×	1.32×	`RMSprop`
SBUX	GRU	30	80–10–10	`relu`	0.0372	0.8840	1.22×	0.91×	`RMSprop`
SBUX	ConvLSTM	5	80–10–10	`tanh`	0.0290	0.9043	1.27×	1.50×	`RMSprop`
TSLA	LSTM	5	80–10–10	`tanh`	0.0462	0.8783	1.17×	0.78×	`RMSprop`
TSLA	BiLSTM	5	70–15–15	`tanh`	0.0589	0.8658	0.80×	0.52×	`RMSprop`
TSLA	GRU	5	80–10–10	`tanh`	0.0480	0.9274	1.17×	0.51×	`RMSprop`
TSLA	ConvLSTM	30	80–10–10	`tanh`	0.0394	0.9118	1.04×	1.19×	`Adam`

Table 4. Cross-asset statistical summary of the best observed architecture-specific RMSE values.

Architecture	Mean RMSE	Median RMSE	Mean Rank	First-Place Counts
LSTM	0.0422	0.0431	3.00	0
BiLSTM	0.0450	0.0413	3.25	0
GRU	0.0414	0.0401	2.75	0
ConvLSTM	0.0300	0.0275	1.00	4

Second, although LSTM, BiLSTM, and GRU generally provide competitive results, their performance appears more sensitive to the selected lookback window and data partition. BiLSTM does not consistently outperform the corresponding unidirectional architectures, suggesting that bidirectional processing within a fixed historical window offers limited benefits for this forecasting task. Moreover, no recurrent architecture emerges as a consistently superior alternative across all assets, highlighting the dependence of forecasting performance on stock-specific characteristics.

Third, forecasting accuracy and trading performance do not exhibit a direct one-to-one relationship. For AAPL, SBUX, and TSLA, the best-performing ConvLSTM configurations also achieve higher cumulative returns than Buy-and-Hold, reaching 1.27× versus 1.26× for AAPL, 1.50× versus 1.27× for SBUX, and 1.19× versus 1.04× for TSLA. In contrast, META produces the strongest forecasting metrics in the study (RMSE = 0.0256, R^2 = 0.9770), yet the corresponding trading strategy achieves a cumulative return of only 1.53×, compared with 1.85× for Buy-and-Hold. This result illustrates that minimizing forecasting error does not necessarily maximize investment performance when predictions are translated into trading decisions.

The selected lookback windows also vary across assets. The best-performing configurations favor short or intermediate histories for AAPL and META ($L = 15$ and $L = 5$,

respectively), while TSLA benefits from a substantially longer context ($L = 30$). SBUX likewise achieves its strongest ConvLSTM performance using a short lookback window ($L = 5$). These differences suggest that the amount of useful historical information is asset-dependent and influenced by the underlying dynamics of each equity.

Overall, ConvLSTM records the lowest observed forecasting error across the four examined equities and produces the strongest trading outcomes for SBUX and TSLA, while remaining competitive for AAPL. This pattern suggests that combining convolutional feature extraction with recurrent temporal modeling may be beneficial within the present experimental setting, but it does not establish superiority across broader assets or market conditions. At the same time, the META results demonstrate that superior predictive accuracy alone is insufficient to guarantee superior trading performance, emphasizing the importance of evaluating forecasting models through both statistical and economic criteria.

It should be noted that the performance comparison reported in Table 3 is descriptive and is based on the best out-of-sample configuration identified for each architecture and stock. Although ConvLSTM achieves the lowest RMSE for all four examined equities, no formal statistical significance test is conducted to determine whether these differences are statistically significant across all configurations. Therefore, the reported advantage of ConvLSTM should be interpreted as an empirical tendency observed within the present experimental setting rather than as statistically proven dominance over the alternative architectures.

6.2. AAPL Results

For AAPL, Figure 3 presents the actual closing-price trajectory together with the corresponding predictions generated by the four evaluated architectures. The plots correspond to the best-performing configuration of each model and provide a qualitative assessment of tracking accuracy, responsiveness to turning points, and the ability to reproduce the overall market trend during the test period.

The four architectures exhibit markedly different forecasting behavior. LSTM and GRU successfully capture the dominant trend of the series, including the mid-period decline and subsequent recovery, although both models produce smoother trajectories than the observed prices and tend to underestimate the strongest upward movements toward the end of the test period. This smoothing effect is particularly visible for GRU, which remains consistently below the actual trajectory during the final rally.

BiLSTM produces the weakest visual agreement with the observed series. Although it captures some local fluctuations, the predicted trajectory remains confined to a relatively narrow range and fails to reproduce the magnitude of the sustained upward trend. As a result, the model exhibits a persistent level bias and substantial underestimation throughout the second half of the test period.

ConvLSTM provides the closest correspondence to the actual price series. The model accurately follows both the mid-period trough and the subsequent recovery while maintaining a close alignment with the observed values during the final upward trend. Compared with the recurrent architectures, ConvLSTM better preserves local temporal variations and exhibits less lag around turning points, resulting in a trajectory that remains consistently close to the ground-truth series.

The qualitative evidence presented in Figure 3 is consistent with the quantitative findings reported in Table 3. ConvLSTM delivers the most faithful reconstruction of the test-period dynamics, while LSTM and GRU capture the broader trend but smooth short-term fluctuations. In contrast, BiLSTM exhibits clear calibration issues and weaker trend tracking, which is reflected in its inferior forecasting performance.

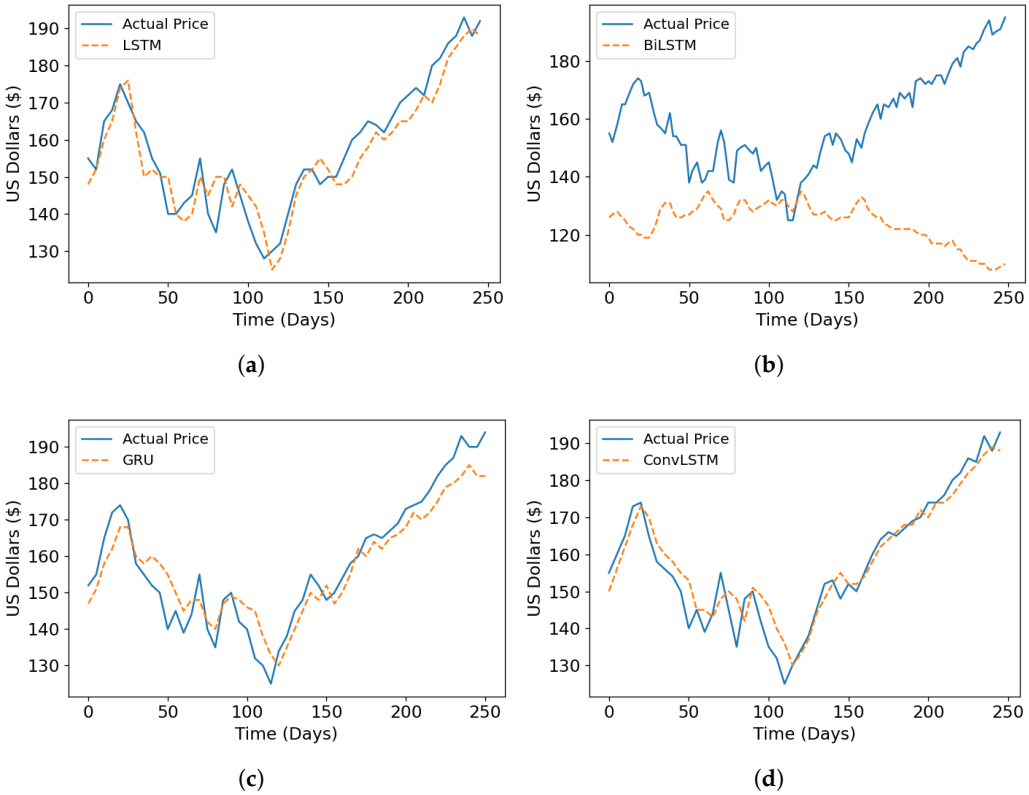

Figure 3. Actual and predicted AAPL closing prices for the evaluated architectures. (**a**) AAPL–LSTM. (**b**) AAPL–BiLSTM. (**c**) AAPL–GRU. (**d**) AAPL–ConvLSTM.

6.3. META Results

For META, Figure 4 presents the actual closing-price trajectory together with the corresponding predictions generated by the four evaluated architectures. The plots correspond to the best-performing configuration of each model and provide a qualitative assessment of tracking accuracy, responsiveness to turning points, and the ability to reproduce the overall market trend during the test period.

The four architectures successfully capture the major phases of the META price trajectory, including the initial decline, the pronounced mid-period trough, and the sustained upward trend that follows. Compared with the AAPL results, all models exhibit stronger agreement with the observed series, indicating that the dominant market dynamics of META are captured effectively by both recurrent and convolutional–recurrent architectures.

LSTM and GRU display very similar behavior throughout the test period. Both models reproduce the location of the trough and follow the subsequent recovery closely, although they smooth abrupt transitions and tend to underestimate the strongest upward movements during the final phase of the rally. This effect is slightly more pronounced for GRU, whose predictions remain consistently below the actual prices near the end of the test horizon.

BiLSTM demonstrates strong overall tracking performance and follows the observed trajectory closely across most of the evaluation period. The model captures both the decline and the subsequent recovery with relatively small deviations, although minor local overestimations are visible around parts of the rebound and during sections of the upward trend. Nevertheless, BiLSTM exhibits substantially better calibration than observed in the corresponding AAPL experiment.

ConvLSTM again provides the closest correspondence to the actual price series. The model accurately reproduces both the depth and timing of the trough while maintaining close alignment with the observed values throughout the recovery and final upward trend.

Although some smoothing remains visible around the sharp rebound, ConvLSTM preserves local temporal structure more effectively than the recurrent baselines and exhibits minimal lag during major trend transitions.

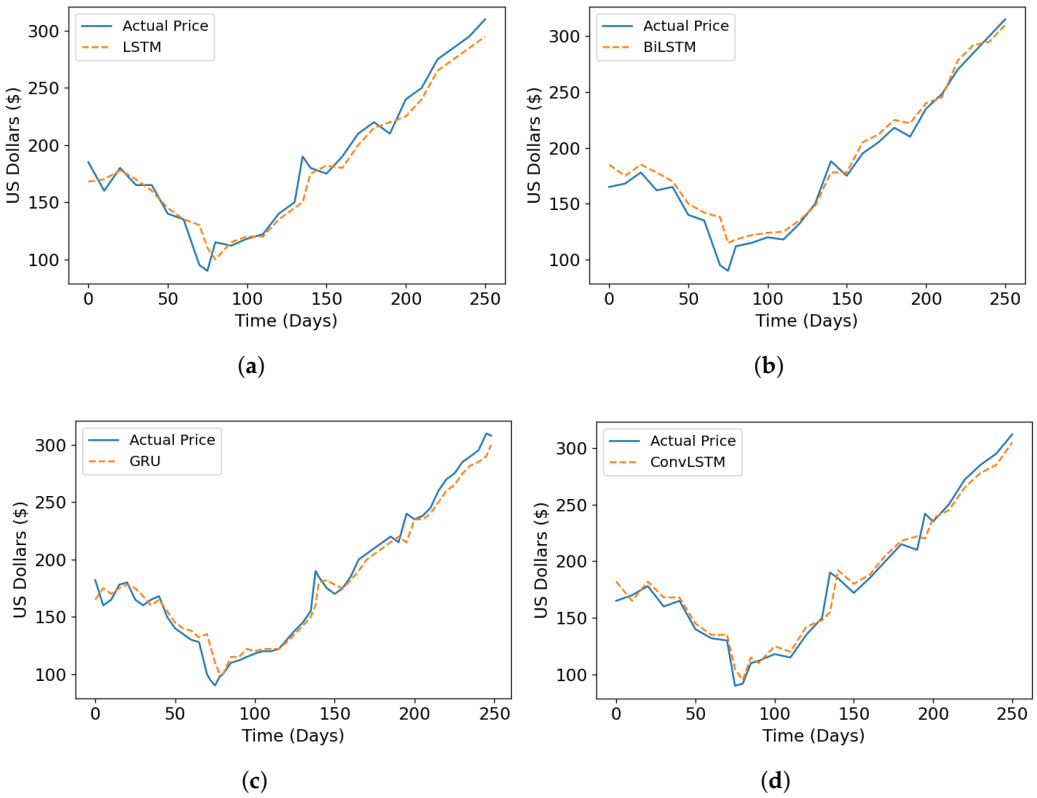

Figure 4. Actual and predicted META closing prices for the evaluated architectures. (**a**) META–LSTM. (**b**) META–BiLSTM. (**c**) META–GRU. (**d**) META–ConvLSTM.

The qualitative evidence presented in Figure 4 is consistent with the quantitative findings reported in Table 3. ConvLSTM delivers the most accurate reconstruction of the META test trajectory, while BiLSTM remains highly competitive and closely follows the observed series. LSTM and GRU successfully capture the dominant market trend but exhibit stronger smoothing effects during periods of rapid price appreciation. Overall, the META results are consistent with the lower observed forecasting error of ConvLSTM within the present test period.

6.4. SBUX Results

For SBUX, Figure 5 presents the actual closing-price trajectory together with the corresponding predictions generated by the four evaluated architectures. The plots correspond to the best-performing configuration of each model and provide a qualitative assessment of tracking accuracy, responsiveness to turning points, and the ability to reproduce the overall market trend during the test period.

The SBUX price series exhibits a different behavior from AAPL and META, with more moderate long-term movements and a pronounced peak during the latter part of the test period followed by a noticeable correction. All four architectures successfully capture the overall evolution of the series, including the gradual rise, the attainment of the local maximum, and the subsequent decline toward the end of the evaluation horizon.

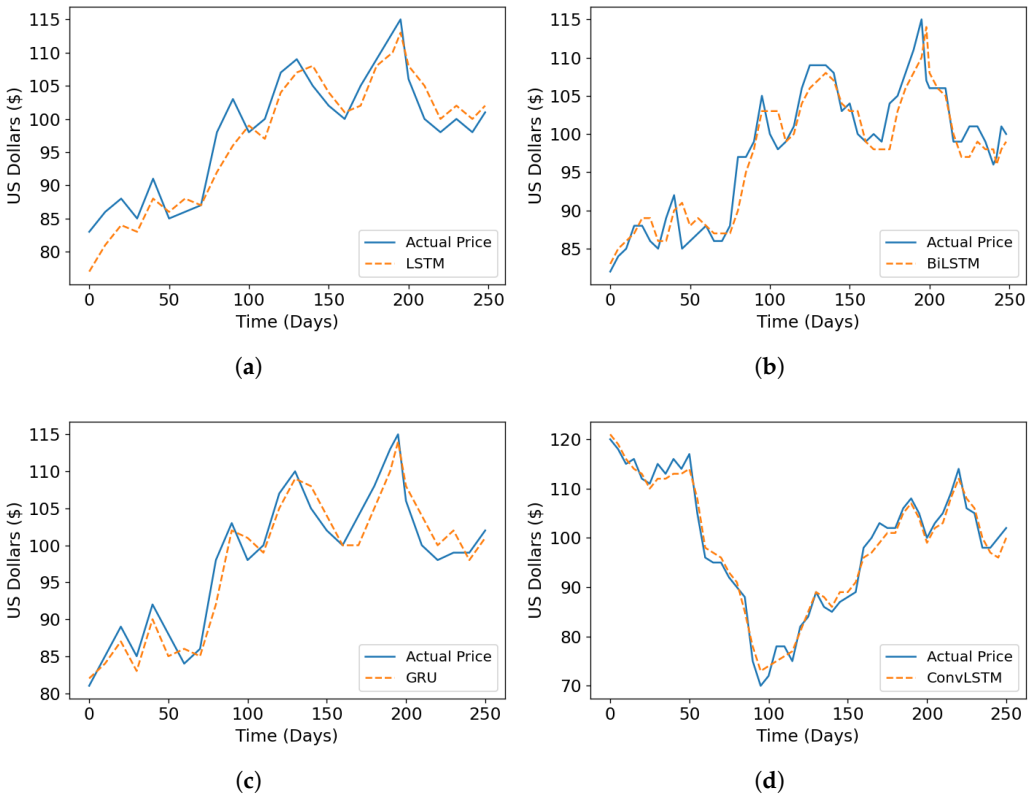

Figure 5. Actual and predicted SBUX closing prices for the evaluated architectures. (**a**) SBUX–LSTM. (**b**) SBUX–BiLSTM. (**c**) SBUX–GRU. (**d**) SBUX–ConvLSTM.

LSTM follows the dominant trend reasonably well but exhibits the strongest smoothing effect among the evaluated models. The predicted trajectory captures the general upward movement and subsequent correction, although it tends to underestimate the magnitude of the peak and compress the amplitude of short-term fluctuations. As a result, the model produces a smoother representation of the series and responds conservatively around local extrema.

BiLSTM demonstrates strong agreement with the observed trajectory throughout most of the test period. The model successfully reproduces both the rise toward the local maximum and the subsequent decline while maintaining close alignment with the actual price levels. Although some smoothing remains visible around abrupt changes, BiLSTM tracks the overall structure of the series accurately and exhibits only minor deviations from the ground-truth trajectory.

GRU also captures the principal movements of the series, including the intermediate rises and declines preceding the major peak. Similar to LSTM, the model tends to smooth local fluctuations and slightly underestimate extreme values. Nevertheless, it follows the overall trend closely and reproduces the timing of the dominant turning points with relatively small lag.

ConvLSTM again provides the closest visual correspondence to the actual series. The model accurately tracks both the growth phase and the subsequent correction while preserving local temporal variations more effectively than the purely recurrent architectures. In particular, ConvLSTM captures the timing and magnitude of the major turning points with high fidelity and remains closely aligned with the observed trajectory throughout most of the evaluation period.

The qualitative evidence presented in Figure 5 is consistent with the quantitative findings reported in Table 3. All four architectures demonstrate satisfactory forecasting

capability for SBUX, but ConvLSTM provides the most accurate reconstruction of the observed price dynamics. BiLSTM also performs strongly, while LSTM and GRU exhibit more pronounced smoothing effects around local peaks and troughs. Overall, the SBUX results indicate that convolutional–recurrent modeling performed well for this asset and test period, without implying general superiority beyond the examined setting.

6.5. TSLA Results

For TSLA, Figure 6 presents the actual closing-price trajectory together with the corresponding predictions generated by the four evaluated architectures. The plots correspond to the best-performing configuration of each model and provide a qualitative assessment of tracking accuracy, responsiveness to turning points, and the ability to reproduce the overall market trend during the test period.

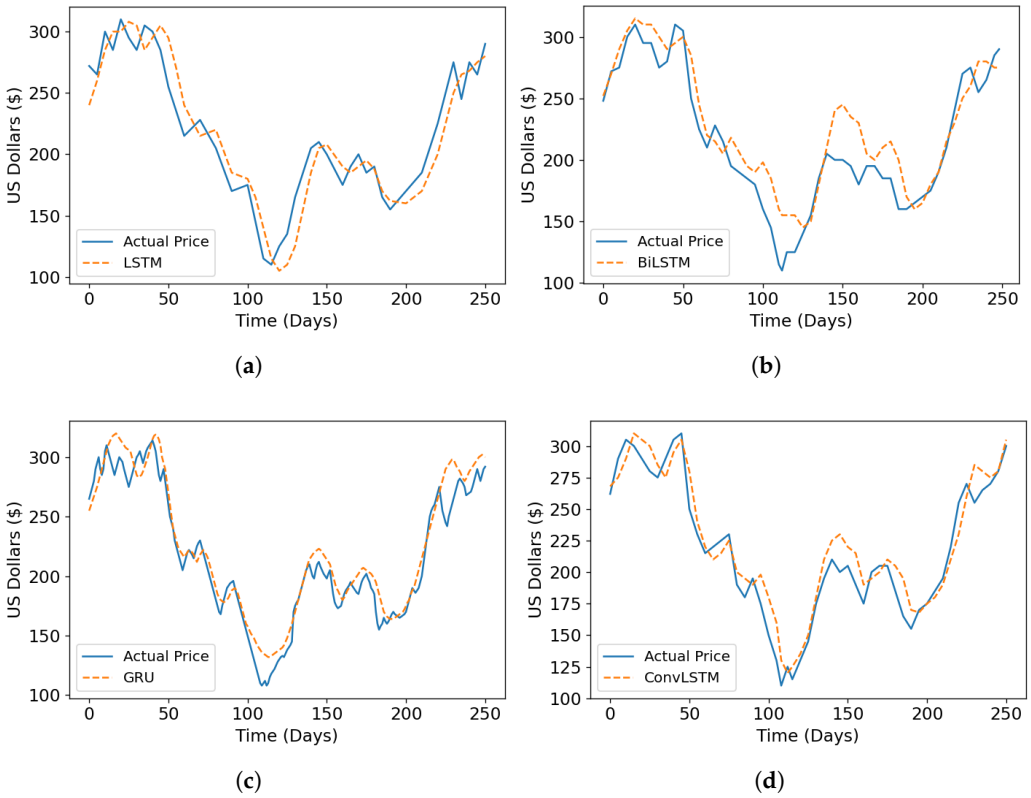

Figure 6. Actual and predicted TSLA closing prices for the evaluated architectures. (**a**) TSLA–LSTM. (**b**) TSLA–BiLSTM. (**c**) TSLA–GRU. (**d**) TSLA–ConvLSTM.

TSLA displays the highest volatility among the examined equities, with sharp declines, rapid rebounds, and substantial price variation throughout the evaluation period. These characteristics make TSLA the most challenging forecasting task in the study and provide a useful benchmark for assessing how well each architecture adapts to rapid changes in market conditions.

LSTM successfully captures the overall trajectory of the series, including the extended decline toward the middle of the test period and the strong recovery that follows. The model reproduces the major turning points reasonably well, although the predicted trajectory is smoother than the observed series and exhibits a small lag immediately after the deepest trough. During the final upward movement, the model follows the trend closely but slightly underestimates the highest observed values.

BiLSTM produces the weakest visual agreement among the four architectures. Although the model correctly identifies the major phases of the series, it substantially overes-

timates prices during the post-trough recovery and remains above the observed trajectory for extended periods. This positive bias is particularly evident around the first major rebound, where the predicted values diverge noticeably from the actual series. The resulting trajectory captures the overall direction of movement but struggles to maintain accurate calibration under rapidly changing market conditions.

GRU reproduces the principal turning points and follows the broad decline–recovery pattern successfully. However, the model also exhibits a tendency toward overestimation, particularly during the middle portion of the evaluation period. While the timing of the recovery is captured correctly, the predicted values often remain above the actual series and fail to reproduce the full depth of the observed trough. Despite this bias, the model maintains reasonable alignment with the overall market trend and accurately reflects the final upward movement.

ConvLSTM again provides the closest visual correspondence to the observed trajectory. The model accurately captures both the timing and depth of the major trough and follows the subsequent recovery with relatively small deviations. Compared with the recurrent architectures, ConvLSTM responds more effectively to abrupt changes in direction and maintains stronger alignment during periods of elevated volatility. Although some overestimation is visible during portions of the recovery phase, the model consistently remains close to the actual series and reproduces the major market movements with high fidelity.

The qualitative evidence presented in Figure 6 is consistent with the quantitative findings reported in Table 3. ConvLSTM records the lowest forecasting error and provides the closest visual correspondence to the TSLA trajectory among the evaluated models. However, these results do not establish superior robustness across volatile market regimes. LSTM and GRU successfully capture the dominant market dynamics but exhibit stronger smoothing effects and localized bias, while BiLSTM shows the largest deviations from the observed series. Overall, the TSLA findings indicate that ConvLSTM tracked the observed price dynamics more closely than the alternative models during the examined test period.

6.6. Cross-Asset Comparison and Trading Performance

Table 5 summarizes the strongest forecasting configuration identified for each equity across all evaluated architectures, lookback windows, and chronological train/validation/test partitions. For every stock, the lowest observed forecasting error is achieved by a ConvLSTM model. This descriptive pattern suggests that combining convolutional feature extraction with recurrent temporal modeling may be beneficial within the examined experimental setting; however, it does not establish statistically significant or general superiority over the alternative architectures.

Table 5. Best out-of-sample ConvLSTM configurations across equities.

Item	AAPL	META	SBUX	TSLA
Model	ConvLSTM	ConvLSTM	ConvLSTM	ConvLSTM
Split ratio (train/val/test)	80–10–10	80–10–10	80–10–10	80–10–10
Lookback window L	15	5	5	30
Activation function	`tanh`	`tanh`	`tanh`	`tanh`
Optimizer	`Adam`	`RMSprop`	`RMSprop`	`Adam`
Test RMSE	0.0260	0.0256	0.0290	0.0394
Test R^2	0.9220	0.9770	0.9043	0.9118
Buy-and-Hold cumulative return	1.26×	1.85×	1.27×	1.04×
Model strategy cumulative return	1.27×	1.53×	1.50×	1.19×

Several observations emerge from Table 5. First, ConvLSTM achieves the lowest observed forecasting error for all four equities, showing a consistent descriptive pattern

within the examined configurations. The strongest predictive performance is observed for META, which attains the lowest RMSE (0.0256) and highest out-of-sample R^2 (0.9770). AAPL also achieves strong predictive accuracy, while SBUX and especially TSLA exhibit higher RMSE values, reflecting the greater difficulty of accurately tracking assets with more variable price dynamics.

Second, although ConvLSTM is selected in all cases, the optimal lookback window varies substantially across assets. META and SBUX achieve their best performance using short-term histories ($L = 5$), AAPL favors an intermediate window ($L = 15$), while TSLA benefits from a considerably longer context ($L = 30$). This variability suggests that no single temporal horizon is universally optimal and that forecasting performance depends on the specific dynamics of each equity.

Finally, beyond forecasting accuracy, it is important to examine whether predictive improvements translate into economically meaningful outcomes. Since the ultimate objective of many financial forecasting systems is decision support rather than error minimization alone, trading-based evaluation provides an additional perspective on model performance. Consistent with established forecasting practice, all reported results are obtained from strictly out-of-sample test data that were not used during model training or hyperparameter selection [56].

To examine more systematically the relationship between prediction accuracy and economic utility, we compare the forecasting metrics of the best ConvLSTM configuration with the corresponding trading outcomes. In this study, economic utility is approximated by the terminal cumulative return of the forecast-driven strategy relative to the Buy-and-Hold benchmark. Let $CR_{\mathrm{B\&H}}$ denote the terminal cumulative return of the Buy-and-Hold benchmark, and let CR_{model} denote the terminal cumulative return of the forecast-driven trading strategy. We define the cumulative-return difference as

$$\Delta CR = CR_{\mathrm{model}} - CR_{\mathrm{B\&H}},$$

and the relative utility ratio as

$$UR = \frac{CR_{\mathrm{model}}}{CR_{\mathrm{B\&H}}}.$$

Values of $\Delta CR > 0$ or $UR > 1$ indicate that the forecast-driven strategy achieves a higher terminal cumulative return than Buy-and-Hold over the test period. These measures do not, by themselves, establish superior overall financial performance because they do not account for volatility or drawdown risk.

Table 6 shows that prediction accuracy and terminal cumulative return are not monotonically related. META achieves the strongest statistical forecasting performance, with the lowest RMSE and highest out-of-sample R^2, yet its forecast-driven strategy underperforms Buy-and-Hold. Conversely, SBUX and TSLA produce positive terminal-return differences despite having higher RMSE values than META. These results suggest that low point-forecast error is not sufficient for superior trading performance. Economic utility also depends on the directional correctness of the trading signal, the timing of market exposure, and the interaction between forecast errors and realized return dynamics.

Table 6. Prediction accuracy and economic utility of the best ConvLSTM configurations.

Stock	RMSE	R^2	$CR_{\mathrm{B\&H}}$	CR_{model}	ΔCR	UR	Terminal-Return Outcome
AAPL	0.0260	0.9220	1.26	1.27	+0.01	1.01	Model marginally higher
META	0.0256	0.9770	1.85	1.53	−0.32	0.83	Buy-and-Hold higher
SBUX	0.0290	0.9043	1.27	1.50	+0.23	1.18	Model higher
TSLA	0.0394	0.9118	1.04	1.19	+0.15	1.14	Model higher

Table 7 reports a sensitivity analysis of the ConvLSTM-based trading strategy under increasing transaction-cost assumptions, where transaction costs combine commissions and slippage. The results show that cumulative returns decline monotonically as transaction costs increase. The degradation is especially important for AAPL, where the frictionless strategy only slightly outperforms Buy-and-Hold; after introducing even modest costs, the advantage disappears. META remains below Buy-and-Hold under all transaction-cost assumptions, confirming that its strong forecasting accuracy does not translate into superior trading performance. SBUX remains the most robust case, preserving a positive advantage over Buy-and-Hold even under higher transaction-cost assumptions. TSLA also remains competitive under moderate costs, although its advantage disappears at the highest examined cost level of 0.5%. Overall, the sensitivity analysis indicates that the economic value of the forecast-driven strategy is materially affected by market frictions and depends on both turnover and the size of the original performance margin.

Table 7. Transaction-cost sensitivity of the ConvLSTM-based trading strategy.

Stock	0.0%	0.1%	0.2%	0.3%	0.4%	0.5%
AAPL	1.27	1.24	1.22	1.19	1.17	1.15
META	1.53	1.48	1.43	1.39	1.34	1.30
SBUX	1.50	1.46	1.42	1.38	1.34	1.31
TSLA	1.19	1.15	1.12	1.08	1.05	1.02

To complement the terminal cumulative-return comparison, Table 8 reports approximate risk-adjusted performance measures for Buy-and-Hold and the ConvLSTM-based strategy. The reported measures include annualized return, Sharpe ratio, Sortino ratio, maximum drawdown (MDD), and Calmar ratio.

Table 8. Approximate risk-adjusted performance based on five-day sampled cumulative-return trajectories.

Stock	Strategy	Ann. Return	Sharpe	Sortino	MDD	Calmar
AAPL	Buy-and-Hold	26.15%	1.03	1.46	28.26%	0.93
AAPL	ConvLSTM	27.14%	1.13	1.68	17.97%	1.51
META	Buy-and-Hold	82.68%	1.42	2.14	49.07%	1.68
META	ConvLSTM	53.22%	1.18	1.72	42.72%	1.25
SBUX	Buy-and-Hold	26.64%	1.03	1.59	14.29%	1.87
SBUX	ConvLSTM	49.29%	2.10	4.22	5.41%	9.12
TSLA	Buy-and-Hold	4.03%	0.39	0.57	64.91%	0.06
TSLA	ConvLSTM	18.16%	0.57	0.90	57.89%	0.31

The risk-adjusted measures in Table 8 are estimated at a five-trading-day frequency from the sampled cumulative-wealth trajectories. Annualization assumes 252 trading days per year and a zero risk-free rate. Because cumulative wealth is observed at five-day intervals and recorded with finite numerical precision, drawdowns occurring between consecutive observations are not captured. The reported MDD values should therefore be interpreted as lower-bound estimates, while the Sharpe, Sortino, and Calmar ratios represent approximate five-day-frequency risk-adjusted indicators rather than exact daily-frequency measures.

The risk-adjusted results reveal substantial differences across equities. For AAPL, the ConvLSTM strategy achieves only a small annualized-return advantage over Buy-and-Hold, but produces higher Sharpe, Sortino, and Calmar ratios together with a considerably lower estimated MDD. For META, Buy-and-Hold retains stronger return and risk-adjusted

performance, despite the ConvLSTM strategy exhibiting a moderately smaller drawdown. SBUX represents the strongest case for the forecast-driven strategy, which achieves higher annualized return and risk-adjusted ratios together with a substantially lower estimated MDD; however, the magnitude of these ratios should be interpreted cautiously given the five-day sampling frequency. For TSLA, ConvLSTM improves the annualized return, Sharpe ratio, Sortino ratio, and Calmar ratio relative to Buy-and-Hold, but both strategies remain exposed to exceptionally large drawdowns. Overall, these findings confirm that terminal cumulative return alone provides an incomplete assessment of financial performance and that the economic usefulness of the forecasting strategy remains strongly asset dependent.

Figure 7 presents the cumulative returns of the Buy-and-Hold benchmark and the forecast-driven trading strategy generated using the best-performing ConvLSTM configuration for each stock.

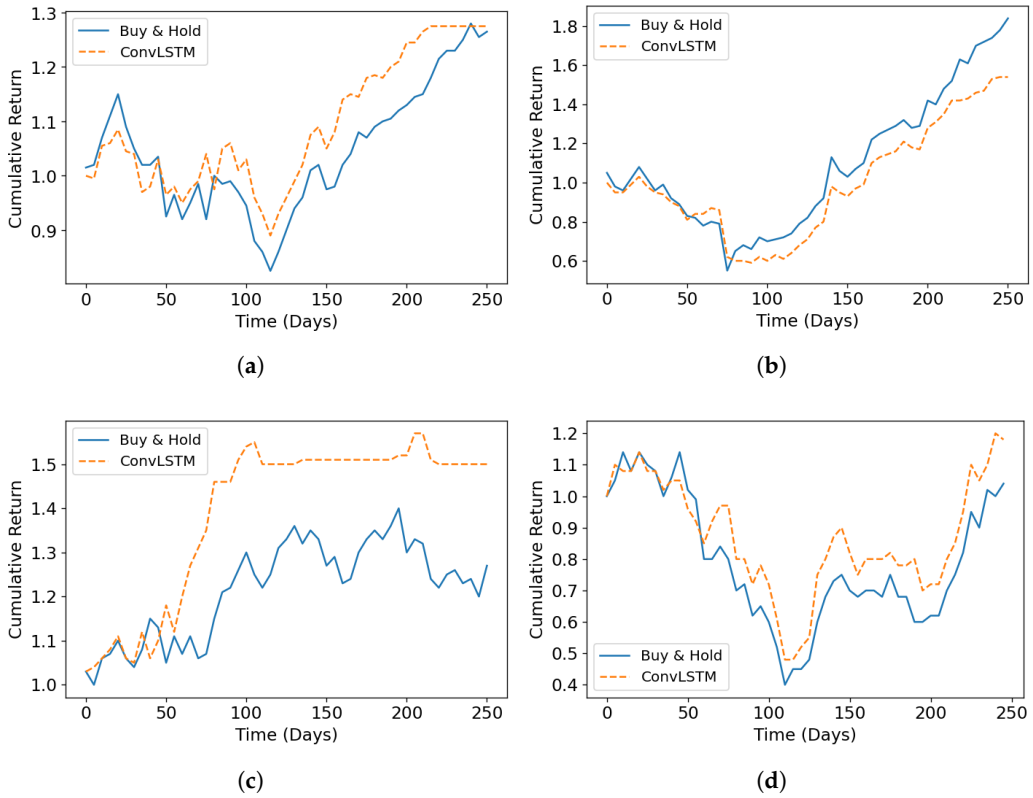

Figure 7. Cumulative returns of Buy-and-Hold and ConvLSTM-based trading strategies. (**a**) AAPL. (**b**) META. (**c**) SBUX. (**d**) TSLA.

For AAPL, both strategies follow a similar overall trajectory and experience a noticeable drawdown during the middle of the evaluation period. The forecast-driven strategy declines less severely and begins recovering slightly earlier, resulting in a small but measurable advantage by the end of the test horizon. Overall, AAPL represents a case where the trading rule provides only a modest improvement over Buy-and-Hold, primarily through reduced drawdowns and earlier participation in the recovery phase.

META represents the only case in which Buy-and-Hold clearly outperforms the forecast-driven strategy. Although both approaches recover strongly following the mid-horizon decline, the benchmark compounds more rapidly during the second half of the evaluation period and finishes substantially higher. This result demonstrates that strong forecasting accuracy alone is insufficient to guarantee superior trading performance.

For SBUX, the forecast-driven strategy achieves the largest relative improvement over Buy-and-Hold among the examined equities, finishing at approximately 1.50× compared with 1.27× for the benchmark. The performance gap emerges gradually throughout the evaluation period and remains stable toward the end of the horizon, suggesting that the forecasting model successfully identifies a substantial portion of the favorable market movements while avoiding part of the adverse periods.

For TSLA, Buy-and-Hold experiences a severe decline during the first half of the test period, while the forecast-driven strategy remains above the benchmark during much of the subsequent recovery and ultimately achieves a higher terminal return (1.19× versus 1.04×). The approximate risk-adjusted measures also favor the ConvLSTM strategy relative to Buy-and-Hold. Nevertheless, its estimated MDD remains exceptionally high at 57.89%, demonstrating that the strategy continues to carry substantial downside risk despite its relative improvement over the benchmark. The TSLA findings should therefore be interpreted as evidence of comparative drawdown mitigation rather than robust performance under highly volatile market conditions.

6.7. Discussion

Within the fixed chronological partitions examined in this study, ConvLSTM records the lowest observed forecasting error across all four equities and ranks among the strongest models in terms of out-of-sample R^2. This pattern suggests that combining convolutional feature extraction with recurrent temporal modeling may provide a useful representation of technical-indicator sequences. However, because the evaluation is limited to four equities, does not employ walk-forward or rolling-window validation, and does not include formal statistical significance testing, the results should not be interpreted as evidence of universal or statistically proven superiority over LSTM, GRU, or BiLSTM.

One possible explanation for the stronger performance of ConvLSTM is its ability to combine local feature extraction with recurrent temporal modeling. Each input window contains multiple correlated variables representing price, trend, momentum, volume, and volatility. With the adopted $(1, 3)$ kernel, the convolutional operations learn localized combinations among the price and technical-indicator features represented along the feature axis, while the recurrent gating mechanism aggregates information across the ordered observations in the lookback window. This combination may enable ConvLSTM to represent interactions among the supplied indicators while preserving relevant temporal dependencies more effectively than the purely recurrent architectures examined in this study. Although this architectural interpretation provides a plausible explanation for the observed results, the present experiments do not establish the underlying mechanism causally.

It is important to clarify that ConvLSTM does not learn technical indicators directly from raw OHLCV data in the present framework. The input features are manually engineered indicators, and the model learns representations from these predefined feature sequences. Therefore, the advantage of ConvLSTM should be interpreted as its ability to learn localized combinations among the provided price and indicator features and to aggregate their temporal dependencies, rather than as evidence that it discovers entirely new market variables or technical indicators. In this sense, ConvLSTM may act partly as a nonlinear feature combiner and temporal smoother over the engineered input representation. However, unlike simple smoothing operations, the convolutional–recurrent structure applies trainable filters and gated temporal dynamics, allowing the model to adaptively combine short-term indicator configurations with sequential dependencies over the lookback window.

The present framework uses a fixed set of six input variables and does not include an adaptive feature-selection stage prior to sequence modeling. Given the compact feature

set used herein, the main objective was to evaluate how different sequence architectures exploit the same technical-indicator representation. However, adaptive feature selection could be beneficial, especially if the input space is expanded to include additional technical indicators, macroeconomic variables, sentiment features, or options-implied measures. Tree-based feature-selection methods, such as Random Forest or XGBoost importance, could reduce redundant or noisy predictors before sequence modeling and improve interpretability. Such feature selection must be performed strictly within each training segment to avoid information leakage into validation or test periods. Integrating adaptive feature selection with recurrent or convolutional–recurrent forecasting models is therefore a promising extension of the current framework.

The present study also does not include indicator-level ablation experiments or post-hoc feature-attribution analysis. Consequently, the reported results do not isolate the marginal contribution of EMA, RSI, MACD, OBV, and ATR, or determine whether some indicators are redundant when used jointly. A more detailed interpretability analysis could retrain each architecture after removing one indicator at a time and compare the resulting change in forecasting and trading performance. This leave-one-feature-out analysis could be complemented by permutation importance or SHAP-based explanations, with feature contributions aggregated across the lookback window. To preserve the chronological integrity of the evaluation, all feature selection, permutation, and explanation procedures should be performed without using information from future validation or test observations.

An important observation is that forecasting accuracy, terminal return, and risk-adjusted financial performance are related but distinct evaluation dimensions. Although ConvLSTM achieves the lowest forecasting error across all four equities, the corresponding trading strategy does not uniformly outperform Buy-and-Hold. For AAPL and SBUX, the approximate risk-adjusted measures favor the ConvLSTM strategy, while META retains stronger terminal and risk-adjusted performance under Buy-and-Hold. For TSLA, ConvLSTM improves the relative return and risk-adjusted ratios, but its estimated MDD remains exceptionally high, indicating substantial residual downside risk. These findings demonstrate that small point-forecast errors do not necessarily produce robust trading decisions and that economic performance remains strongly dependent on asset-specific return dynamics, market exposure, transaction costs, and drawdown behavior.

A further observation concerns the relative behavior of the recurrent architectures. Although LSTM, BiLSTM, and GRU frequently achieved competitive forecasting accuracy, none emerged as a consistently superior alternative across all equities. Their ranking varied across assets and experimental configurations, suggesting that the predictive information contained in technical-indicator sequences interacts with asset-specific market dynamics. This variability reinforces the importance of comparative evaluation across multiple assets and highlights the difficulty of identifying a universally optimal recurrent architecture for financial forecasting tasks.

The experiments also demonstrate that the optimal amount of historical information is asset dependent. While META and SBUX achieved their best results using short lookback windows, TSLA benefited from substantially longer historical context. This suggests that different equities exhibit different temporal dependencies and that model configuration should be adapted to the characteristics of the underlying asset rather than relying on a universal forecasting horizon.

The sample period also covers heterogeneous market conditions, including the pre-pandemic period, the COVID-19 shock and recovery, the inflationary and rising-rate environment, and subsequent developments related to technology and artificial intelligence. The present evaluation reports aggregate out-of-sample performance over chronological test segments and does not conduct a formal regime-specific performance decomposition.

Therefore, the results should not be interpreted as evidence that model performance is invariant across market regimes. From a practical perspective, this suggests that forecasting models should be monitored and revalidated under changing market conditions, preferably using rolling or walk-forward evaluation, periodic retraining, and regime-specific robustness checks. Future work should explicitly examine whether the relative advantage of ConvLSTM persists across distinct market regimes and during periods of elevated volatility or structural change.

Several limitations should also be acknowledged. The analysis focuses on four large-cap NASDAQ equities drawn from a single publicly available dataset and uses a predefined set of technical indicators, which may not fully capture the diversity of behaviors observed across broader equity markets, alternative datasets, asset classes, or time periods. Therefore, future work should replicate the analysis across additional datasets and market settings to further assess the robustness and generalizability of the reported findings. In addition, the trading evaluation is intentionally simplified and serves primarily as a mechanism for comparing forecasting utility across models. Consequently, the reported results should be interpreted as evidence of relative model performance within the examined experimental setting rather than as estimates of deployable trading profitability. Accordingly, stronger conclusions regarding architectural rankings would require matched walk-forward forecasts, repeated model estimation, formal statistical testing, and replication across a broader cross-section of assets.

7. Conclusions and Future Work

This study investigated next-day closing-price forecasting for four large-cap NASDAQ equities (AAPL, META, SBUX, and TSLA) using deep learning models trained on multivariate sequences enriched with widely used technical indicators. Within the fixed chronological partitions examined in this study, ConvLSTM records the lowest observed normalized-scale RMSE for all four equities, attaining values of 0.0260 for AAPL, 0.0256 for META, 0.0290 for SBUX, and 0.0394 for TSLA, with corresponding out-of-sample R^2 values of 0.9220, 0.9770, 0.9043, and 0.9118. A plausible architectural explanation is that ConvLSTM combines localized feature extraction with recurrent temporal modeling, allowing it to learn combinations among the supplied price and indicator features while aggregating information across the lookback window. Nevertheless, these results constitute descriptive evidence from four equities under fixed chronological evaluation settings. They do not establish statistically significant or generally applicable dominance of ConvLSTM because the study does not include matched significance testing, walk-forward validation, or broader replication across datasets and market environments.

Beyond forecasting accuracy, the study evaluated the practical usefulness of the predictions through a forecast-driven long-only trading strategy. The strategy achieved higher frictionless terminal cumulative returns than Buy-and-Hold for AAPL, SBUX, and TSLA, while underperforming for META. Approximate five-day-frequency risk-adjusted measures generally favor the ConvLSTM strategy for AAPL, SBUX, and TSLA; however, the TSLA strategy retains an exceptionally high estimated maximum drawdown, while Buy-and-Hold remains superior for META. Transaction-cost sensitivity analysis further shows that these terminal-return advantages weaken as trading frictions increase. These findings confirm that strong predictive accuracy and higher terminal return do not necessarily imply robust risk-adjusted financial performance.

The interpretation of these results should be considered within the scope of the study, which focuses on four large-cap NASDAQ equities and a predefined set of technical indicators. Although transaction-cost sensitivity is examined, the trading evaluation remains a simplified backtesting exercise and does not fully account for bid–ask spreads, execution

timing, market impact, portfolio-level risk management, or other practical implementation constraints.

Future work should extend this framework in several concrete directions. First, the analysis should be replicated on the full set of available equities and on additional equity datasets, including non-NASDAQ stocks, market indices, and different time periods, to test whether the ConvLSTM advantage remains stable across assets and market regimes [64]. This extension should also incorporate walk-forward or rolling-window validation protocols and formal statistical significance testing, such as Diebold–Mariano tests, Wilcoxon signed-rank tests, or Friedman tests across matched configurations, to determine whether the observed differences among architectures are statistically significant. A regime-specific evaluation should also examine model performance separately during bullish, bearish, high-volatility, and low-volatility periods, using methods such as volatility-based segmentation, change-point detection, or hidden Markov models.

Second, the trading evaluation should be extended with more realistic implementation assumptions, including exact daily-frequency risk-adjusted evaluation based on unsampled return paths, time-varying transaction costs, bid–ask spreads, slippage, turnover, directional accuracy, value at risk, and conditional value at risk, to provide a more realistic and risk-aware assessment of economic usefulness. Future research should also consider probabilistic forecasting methods that generate prediction intervals, quantiles, or full predictive distributions rather than only point forecasts. Such forecasts would quantify uncertainty and could support trading decisions based on both expected price movements and forecast confidence.

Third, future experiments should examine whether adding exogenous predictors, such as news sentiment, earnings announcements, macroeconomic variables, and options-implied volatility, improves next-day forecasting performance beyond technical indicators alone [65]. Relevant macroeconomic inputs may include interest rates, inflation, market volatility indices, and monetary-policy announcements, while sentiment measures may be extracted from financial news, earnings-call transcripts, or social-media data. These variables should be aligned according to their actual publication times to avoid look-ahead bias.

Fourth, future work should broaden the architectural comparison by evaluating Transformer-based and other contemporary forecasting models, including Temporal Fusion Transformer, Informer, temporal convolutional networks, and MLP-mixing architectures such as TSMixer, under the same chronological evaluation protocol. Finally, indicator-level ablation experiments, permutation importance, SHAP-based feature-attribution analyses, and adaptive feature-selection mechanisms should be conducted to quantify the contribution of each technical indicator, identify potentially redundant inputs, and improve the interpretability of the forecasting framework. Future work should also compare the evaluated deep sequence architectures with classical statistical and non-neural forecasting approaches, including ARIMA, support vector machines, Random Forests, XGBoost, LightGBM, and multi-granularity cascade forest variants [66].

Overall, the findings suggest that indicator-enriched deep sequence models, and ConvLSTM in particular, may capture useful short-horizon structure within the examined equity forecasting setting. At the same time, the results emphasize that forecasting models should be assessed not only through statistical accuracy but also through their ability to support effective financial decision-making.

Author Contributions: Conceptualization, A.K.; methodology, T.A. and A.K.; formal analysis, T.A. and A.K.; investigation, T.A. and A.K.; data curation, T.A. and A.K.; writing—original draft, T.A.; writing—review and editing, T.A. and A.K.; supervision, A.K.; project administration, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available in Kaggle at https://www.kaggle.com/datasets/khushipitroda/stock-market-historical-data-of-top-10-companies (accessed on 23 June 2026).

Acknowledgments: The authors are sincerely thankful to the anonymous reviewers for their valuable and constructive comments, which greatly enhanced the quality of the article.

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

Appendix A. Technical Indicator Formulations

This appendix presents the mathematical formulations of the technical indicators used to construct the feature set described in Section 3. All indicators are computed *causally* from past and present observations only, using daily OHLCV (Open, High, Low, Close, Volume) data: Open O_t, High H_t, Low L_t, Close C_t, and Volume V_t. Indicator values that are undefined during the initial lookback period are discarded, so minor differences in initialization conventions do not affect the supervised learning samples used for training and evaluation. The definitions below follow standard technical-analysis formulations [12,13]:

Exponential Moving Average (EMA):

For a window length n, the exponential moving average (EMA) of the closing price is

$$\mathrm{EMA}_t^{(n)} = \alpha\, C_t + (1-\alpha)\, \mathrm{EMA}_{t-1}^{(n)}, \qquad \alpha = \frac{2}{n+1}. \tag{A1}$$

We use EMA with $n = 14$.

Relative Strength Index (RSI):

Let $\Delta C_t = C_t - C_{t-1}$. Define gains and losses as

$$G_t = \max(\Delta C_t, 0), \qquad D_t = \max(-\Delta C_t, 0). \tag{A2}$$

Using Wilder's smoothing with window n,

$$\overline{G}_t = \frac{(n-1)\,\overline{G}_{t-1} + G_t}{n}, \tag{A3}$$

$$\overline{D}_t = \frac{(n-1)\,\overline{D}_{t-1} + D_t}{n}. \tag{A4}$$

The relative strength is

$$\mathrm{RS}_t = \frac{\overline{G}_t}{\overline{D}_t}, \tag{A5}$$

and RSI is

$$\mathrm{RSI}_t^{(n)} = 100 - \frac{100}{1+\mathrm{RS}_t}. \tag{A6}$$

We use RSI with $n = 14$.

Moving Average Convergence Divergence (MACD):

MACD is defined as the difference between a fast and a slow exponential moving average of the closing price:

$$\mathrm{MACD}_t = \mathrm{EMA}_t^{(n_f)} - \mathrm{EMA}_t^{(n_s)}, \tag{A7}$$

where n_f and n_s denote the fast and slow windows, respectively.

Although only MACD is used as an input feature in the forecasting models, the associated signal line and histogram are included here for completeness:

$$\text{Signal}_t = \text{EMA}_t^{(n_{\text{sig}})}(\text{MACD}), \tag{A8}$$

$$\text{Hist}_t = \text{MACD}_t - \text{Signal}_t. \tag{A9}$$

We use the conventional $(n_f, n_s, n_{\text{sig}}) = (12, 26, 9)$.

On-Balance Volume (OBV):

OBV is a cumulative volume measure that adds or subtracts daily volume depending on the sign of the close-to-close change:

$$\text{OBV}_t = \text{OBV}_{t-1} + \begin{cases} V_t, & \text{if } C_t > C_{t-1}, \\ 0, & \text{if } C_t = C_{t-1}, \\ -V_t, & \text{if } C_t < C_{t-1}. \end{cases} \tag{A10}$$

The initial value OBV_0 can be set to 0 without loss of generality.

Average True Range (ATR):

First define the true range (TR) as

$$\text{TR}_t = \max\Big(H_t - L_t,\ |H_t - C_{t-1}|,\ |L_t - C_{t-1}|\Big). \tag{A11}$$

ATR applies Wilder's smoothing to TR [13]:

$$\text{ATR}_t^{(n)} = \frac{(n-1)\,\text{ATR}_{t-1}^{(n)} + \text{TR}_t}{n}. \tag{A12}$$

We use ATR with $n = 14$.

References

1. Fama, E.F. Efficient Capital Markets: A Review of Theory and Empirical Work. *J. Financ.* **1970**, *25*, 383–417. [CrossRef]
2. Engle, R.F. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica* **1982**, *50*, 987–1007. [CrossRef]
3. Bollerslev, T. Generalized Autoregressive Conditional Heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [CrossRef]
4. Lo, A.W.; MacKinlay, A.C. Stock Market Prices do not Follow Random Walks: Evidence from a Simple Specification Test. *Rev. Financ. Stud.* **1988**, *1*, 41–66. [CrossRef]
5. Cont, R. Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues. *Quant. Financ.* **2001**, *1*, 223–236. [CrossRef]
6. Kandel, S.; Stambaugh, R.F. On the Predictability of Stock Returns: An Asset-Allocation Perspective. *J. Financ.* **1996**, *51*, 385–424. [CrossRef]
7. Pesaran, M.H.; Timmermann, A. Predictability of Stock Returns: Robustness and Economic Significance. *J. Financ.* **1995**, *50*, 1201–1228. [CrossRef]
8. Campbell, J.Y.; Thompson, S.B. Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average? *Rev. Financ. Stud.* **2008**, *21*, 1509–1531.
9. Poterba, J.M.; Summers, L.H. Mean Reversion in Stock Prices: Evidence and Implications. *J. Financ. Econ.* **1988**, *22*, 27–59. [CrossRef]
10. Granger, C.W.J. Strategies for Modelling Nonlinear Time-Series Relationships. *Econ. Rec.* **1993**, *69*, 233–238. [CrossRef]
11. Gu, S.; Kelly, B.; Xiu, D. Empirical Asset Pricing via Machine Learning. *Rev. Financ. Stud.* **2020**, *33*, 2223–2273. [CrossRef]
12. Murphy, J.J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*; New York Institute of Finance: New York, NY, USA, 1999.
13. Wilder, J.W. *New Concepts in Technical Trading Systems*; Trend Research: Greensboro, NC, USA, 1978.

14. Brock, W.; Lakonishok, J.; LeBaron, B. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *J. Financ.* **1992**, *47*, 1731–1764. [CrossRef]
15. Aravanis, T.; Kanavos, A. Comparative Evaluation of Deep Learning Architectures for Electricity Demand Forecasting. *Mathematics* **2026**, *14*, 827. [CrossRef]
16. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
17. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1724–1734.
18. Schuster, M.; Paliwal, K.K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
19. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA; pp. 802–810.
20. Savvopoulos, A.; Kanavos, A.; Mylonas, P.; Sioutas, S. LSTM Accelerator for Convolutional Object Identification. *Algorithms* **2018**, *11*, 157. [CrossRef]
21. Fischer, T.; Krauss, C. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
22. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [CrossRef]
23. Bao, W.; Cao, Y.; Yang, Y.; Che, H.; Huang, J.; Wen, S. Data-Driven Stock Forecasting Models Based on Neural Networks: A Review. *Inf. Fusion* **2025**, *113*, 102616. [CrossRef]
24. Hewamalage, H.; Ackermann, K.; Bergmeir, C. Forecast Evaluation for Data Scientists: Common Pitfalls and Best Practices. *Data Min. Knowl. Discov.* **2023**, *37*, 788–832. [PubMed]
25. O'Malley, T.; Bursztein, E.; Long, J.; Chollet, F.; Jin, H.; Invernizzi, L. KerasTuner. 2019. Available online: https://github.com/keras-team/keras-tuner (accessed on 23 June 2026).
26. Atsalakis, G.S.; Valavanis, K.P. Surveying Stock Market Forecasting Techniques—Part II: Soft Computing Methods. *Expert Syst. Appl.* **2009**, *36*, 5932–5941. [CrossRef]
27. Lo, A.W.; Mamaysky, H.; Wang, J. Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *J. Financ.* **2000**, *55*, 1705–1765. [CrossRef]
28. Sullivan, R.; Timmermann, A.; White, H. Data-Snooping, Technical Trading Rule Performance, and the Bootstrap. *J. Financ.* **1999**, *54*, 1647–1691. [CrossRef]
29. Park, C.H.; Irwin, S.H. What Do We Know About the Profitability of Technical Analysis? *J. Econ. Surv.* **2007**, *21*, 786–826. [CrossRef]
30. Nazário, R.T.F.; e Silva, J.L.; Sobreiro, V.A.; Kimura, H. A Literature Review of Technical Analysis on Stock Markets. *Q. Rev. Econ. Financ.* **2017**, *66*, 115–126. [CrossRef]
31. Kara, Y.; Boyacioglu, M.A.; Baykan, Ö.K. Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* **2011**, *38*, 5311–5319. [CrossRef]
32. Güresen, E.; Kayakutlu, G.; Daim, T.U. Using Artificial Neural Network Models in Stock Market Index Prediction. *Expert Syst. Appl.* **2011**, *38*, 10389–10397. [CrossRef]
33. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. *Expert Syst. Appl.* **2015**, *42*, 259–268. [CrossRef]
34. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting Stock Market Index Using Fusion of Machine Learning Techniques. *Expert Syst. Appl.* **2015**, *42*, 2162–2172. [CrossRef]
35. Ballings, M.; den Poel, D.V.; Hespeels, N.; Gryp, R. Evaluating Multiple Classifiers for Stock Price Direction Prediction. *Expert Syst. Appl.* **2015**, *42*, 7046–7056. [CrossRef]
36. Chen, K.; Zhou, Y.; Dai, F. A LSTM-Based Method for Stock Returns Prediction: A Case Study of China Stock Market. In *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015*; IEEE: New York, NY, USA, 2015; pp. 2823–2824.
37. Bao, W.; Yue, J.; Rao, Y. A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long-Short Term Memory. *PLoS ONE* **2017**, *12*, e0180944. [CrossRef] [PubMed]
38. Nelson, D.M.Q.; Pereira, A.C.M.; de Oliveira, R.A. Stock Market's Price Movement Prediction with LSTM Neural Networks. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017*; IEEE: New York, NY, USA, 2017; pp. 1419–1426.
39. Selvin, S.; Vinayakumar, R.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K.P. Stock Price Prediction Using LSTM, RNN and CNN-Sliding Window Model. In *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017*; IEEE: New York, NY, USA, 2017; pp. 1643–1647.

40. Saravanos, C.; Kanavos, A. Forecasting Stock Market Volatility using Social Media Sentiment Analysis. *Neural Comput. Appl.* **2025**, *37*, 10771–10794. [CrossRef]
41. Tsantekidis, A.; Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; Iosifidis, A. Using Deep Learning for Price Prediction by Exploiting Stationary Limit Order Book Features. *Appl. Soft Comput.* **2020**, *93*, 106401. [CrossRef]
42. Sezer, O.B.; Ozbayoglu, A.M. Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach. *Appl. Soft Comput.* **2018**, *70*, 525–538. [CrossRef]
43. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-Based Stock Market Prediction Using a Diverse Set of Variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [CrossRef]
44. Lee, S.W.; Kim, H.Y. Stock Market Forecasting with Super-High Dimensional Time-Series Data Using ConvLSTM, Trend Sampling, and Specialized Data Augmentation. *Expert Syst. Appl.* **2020**, *161*, 113704. [CrossRef]
45. Lim, B.; Arik, S.O.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
46. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 11106–11115. [CrossRef]
47. Chen, S.A.; Li, C.L.; Yoder, N.; Arik, S.O.; Pfister, T. TSMixer: An All-MLP Architecture for Time Series Forecasting. *Trans. Mach. Learn. Res.* **2023**. Available online: https://openreview.net/forum?id=wbpxTuXgm0 (accessed on 23 June 2026).
48. Bhandari, H.N.; Rimal, B.; Pokhrel, N.R.; Rimal, R.; Dahal, K.R.; Khatri, R.K.C. Predicting Stock Market Index Using LSTM. *Mach. Learn. Appl.* **2022**, *9*, 100320. [CrossRef]
49. Lee, M.C.; Chang, J.W.; Yeh, S.C.; Chia, T.L.; Liao, J.S.; Chen, X.M. Applying Attention-Based BiLSTM and Technical Indicators in the Design and Performance Analysis of Stock Trading Strategies. *Neural Comput. Appl.* **2022**, *34*, 13267–13279. [CrossRef] [PubMed]
50. Sezer, O.B.; Ozbayoglu, M.; Dogdu, E. An Artificial Neural Network-Based Stock Trading System Using Technical Analysis and Big Data Framework. In *Proceedings of the 2017 ACM SouthEast Conference, Kennesaw, GA, USA, 13–15 April 2017*; ACM: New York, NY, USA, 2017; pp. 223–226.
51. Sezer, O.B.; Ozbayoglu, M.; Dogdu, E. A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters. *Procedia Comput. Sci.* **2017**, *114*, 473–480. [CrossRef]
52. Shynkevich, Y.; McGinnity, T.; Coleman, S.A.; Belatreche, A.; Li, Y. Forecasting Price Movements Using Technical Indicators: Investigating the Impact of Varying Input Window Length. *Neurocomputing* **2017**, *264*, 71–88. [CrossRef]
53. Dixon, M.; Klabjan, D.; Bang, J.H. Classification-Based Financial Markets Prediction Using Deep Neural Networks. *Algorithmic Financ.* **2017**, *6*, 67–77. [CrossRef]
54. Granger, C.W.J.; Pesaran, M.H. Economic and Statistical Measures of Forecast Accuracy. *J. Forecast.* **2000**, *19*, 537–560. [CrossRef]
55. Cerqueira, V.; Torgo, L.; Mozetič, I. Evaluating Time Series Forecasting Models: An Empirical Study on Performance Estimation Methods. *Mach. Learn.* **2020**, *109*, 1997–2028. [CrossRef]
56. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, 2021.
57. Guo, J.; Wang, Y.; Chen, Z.S.; Chiclana, F.; Pedrycz, W. Enhancing Multivariate Time Series Forecasting for Container Reloading Operations: An Improved Adaptive Multi-granularity Cascade Forest Model Integrating. *Transp. Res. Part E Logist. Transp. Rev.* **2026**, *211*, 104847. [CrossRef]
58. Pitroda, K. Stock Market: Historical Data of Top 10 Companies. Available online: https://www.kaggle.com/datasets/khushipitroda/stock-market-historical-data-of-top-10-companies (accessed on 23 June 2026).
59. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
60. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
61. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
62. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2012; Volume 25.
63. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* **2018**, *18*, 1–52.
64. Saravanos, C.; Kanavos, A. Forecasting Stock Market Alternations Using Social Media Sentiment Analysis and Deep Neural Networks. In *Proceedings of the 14th International Conference on Information, Intelligence, Systems & Applications (IISA), Volos, Greece, 10–12 July 2023*; IEEE: New York, NY, USA, 2023; pp. 1–8.

65. Kanavos, A.; Vonitsanos, G.; Mohasseb, A.; Mylonas, P. An Entropy-Based Evaluation for Sentiment Analysis of Stock Market Prices Using Twitter Data. In *Proceedings of the 15th International Workshop on Semantic and Social Media Adaptation and Personalization, SMA, Zakynthos, Greece, 29–30 October 2020*; IEEE: New York, NY, USA, 2020; pp. 1–7.
66. Vonitsanos, G.; Kanavos, A.; Mylonas, P. A Context-Enriched Hybrid ARIMAX-Deep Learning Framework for Robust Cryptocurrency Price Forecasting. In Proceedings of the 21st International Conference on Web Information Systems and Technologies (WEBIST), Marbella, Spain, 21–23 October 2025; pp. 257–266.