

Stock Price Forecasting with Technical Indicators: Comparing Windowed and Non-Windowed Deep Learning Models on the Saudi Stock Market

Abdullah Almusned*, Mohammad Mehedi Hassan, Muhammad Al-Qurishi, Bader Alkhamees

Department of Information System, College of Computer and Information Sciences, King Saud University, Saudi Arabia

Abstract—Stock price forecasting increasingly uses recurrent neural networks with sliding-window sequences. However, whether explicit temporal windowing improves next-day prediction when technical indicators already encode recent market behaviour remains unclear, particularly in the Saudi stock market. This study aimed to determine how temporal input design affects LSTM, hybrid LSTM-GRU, and MLP performance. Daily OHLCV data from 2010 to 2023 were collected for 19 companies representing all TASI sectors. Twenty-two technical indicators produced a 27-variable feature space. Each model was evaluated without explicit windowing and with 5-, 10-, and 22-day windows, producing 228 neural model-company experiments. A naïve persistence baseline was evaluated separately for each company, yielding 19 additional evaluations. Performance was assessed using eight regression metrics and Wilcoxon signed-rank and Friedman tests. Non-windowed recurrent models outperformed all windowed variants. Non-windowed LSTM achieved the highest average R^2 of 0.9583, while non-windowed hybrid LSTM-GRU achieved the lowest average RMSE of 1.1585 and led 12 of 19 companies.

Keywords—Stock price prediction; deep learning; long short-term memory; sliding window; technical indicators; Saudi stock market

I. INTRODUCTION

Forecasting stock prices is challenging because financial markets are nonlinear, dynamic, and affected by interacting factors, yet accurate prediction remains important to investors, analysts, and researchers. This has driven growing interest in machine and deep learning for financial forecasting. Neural models such as Long Short-Term Memory (LSTM) [1], Gated Recurrent Unit (GRU) [2], and Multilayer Perceptron (MLP) are widely used to capture complex patterns in historical data. Recent surveys confirm that deep learning methods have become dominant in financial time-series prediction and consistently outperform traditional statistical methods across different market settings [3]. However, as Chong et al. (2017) showed for the Korean stock market, performance depends heavily on data representation [4]. This raises an important question: does sliding-window sequence construction still add value when the input already contains domain-informed technical indicators?

The Saudi Stock Exchange (Tadawul) is a particularly relevant setting for this investigation. As the largest capital market in the Middle East and part of a G20 economy,

it has characteristics that distinguish it from the developed markets examined in most deep learning forecasting studies. These include a $\pm 10\%$ daily price fluctuation limit, high retail investor participation [5], sensitivity to global oil prices, and an evolving regulatory environment shaped by Vision 2030, the Saudi Aramco initial public offering, and the 2019 inclusion in FTSE Russell and S&P emerging market indices [6]. These characteristics may produce dynamics that do not align with models validated on Western or East Asian exchanges, raising questions about whether forecasting design choices transfer effectively to this market.

Prior Saudi-market studies have applied statistical, shallow machine-learning, and deep-learning techniques [7], [8], [9], [10], but three issues remain insufficiently addressed. First, no Saudi-market study has compared LSTM, hybrid LSTM-GRU, and MLP within a unified framework. Prior work tested these architectures separately or in limited combinations: Aseeri (2023) evaluated only GRU [11], while Baamer and Aljamaan (2025) tested LSTM, GRU, and BiRNN without a hybrid recurrent design or non-recurrent baseline [12]. Second, sliding-window construction is commonly used with recurrent models without testing whether it improves next-day forecasting, and studies that examine it typically use one window size. Aseeri (2023) applied a stride-based sliding window without comparing it with a non-windowed alternative or other window lengths [11]; no other Saudi-market study has isolated this design variable. Third, explicit windowing may be redundant when calculated technical indicators that already encode recent trend, momentum, volatility, and volume dynamics. This interaction between feature richness and explicit windowing remains unexamined for Saudi equities, although Alsubaie et al. (2019) demonstrated that indicator selection significantly affects prediction quality in this market [13].

The study makes five contributions. First, it develops an end-to-end Saudi forecasting framework integrating OHLCV data with systematically computed technical indicators. Second, it provides a controlled comparison of non-windowed and windowed forecasting across three window lengths. Third, it compares LSTM, hybrid LSTM-GRU, and MLP for next-day closing-price prediction across 19 companies. Fourth, it demonstrates that sliding-window degradation is consistent across all window sizes, architectures, and companies, indicating a structural interaction with the technical-indicator feature space rather than an artefact of a particular window length. Fifth, it supports the comparisons with Wilcoxon signed-rank and Friedman significance tests, addressing a methodological

*Corresponding author.

limitation in prior Saudi forecasting studies.

Accordingly, the paper is a comparative empirical investigation of input design. It examines how feature engineering, temporal input design, and model choice interact in Saudi stock forecasting. Its central finding is that non-windowed deep learning models significantly outperform all sliding-window variants across three window sizes, all architectures, and all 19 companies, with recurrent-model RMSE reductions exceeding 85% (Wilcoxon $p < 0.00001$). This suggests that sufficiently rich multivariate technical representations reduce the need for explicit temporal windowing in short-horizon forecasting.

The rest of the paper is organised as follows. Section II reviews the relevant literature. Section III describes the dataset and feature construction. Section IV presents the proposed methodology, and Section V sets out the experimental design. Section VI reports the results, Section VII discusses their implications, and the final section concludes.

II. LITERATURE REVIEW

Stock market forecasting has progressed from classical time series models such as ARIMA and GARCH to machine and deep learning. Bollerslev (1986) introduced the generalized ARCH model for volatility modelling, while the Box-Jenkins ARIMA framework established the standard for linear time-series forecasting [14]. In the Saudi context, Alshammari et al. (2020) combined Maximum Overlapping Discrete Wavelet Transform with GARCH models using Tadawul daily closing-price data from 2011 to 2019 and improved volatility forecasting over standalone ARIMA and GARCH models [5].

Machine learning introduced greater modelling flexibility. Campisi et al. (2024) compared 55 stock markets over 65 periods from 2011 to 2022 and found that machine-learning models, particularly random forests and ensembles, consistently outperformed linear baselines in predicting stock-index direction [15]. This demonstrates their ability to capture complex nonlinear dynamics that may not be adequately represented by linear models. Technical indicators also became central to forecasting, rather than a minor preprocessing step, by transforming OHLCV data into measures of momentum, volatility, trend, and volume. Mostafavi and Hooman (2025) evaluated 88 indicators in these categories using XGBoost, Random Forest, Support Vector Regression, and LSTM, showing that appropriate indicator selection improves S&P 500 forecasting [16]. In the Saudi market, Alsubaie et al. (2019) examined 50 indicators across 99 Tadawul-listed stocks using nine classifiers and five feature-selection methods; larger feature sets offered no additional improvement and often degraded performance [13]. Thus, forecasting quality depends on both input representation and model architecture.

Enriched feature representations motivated deep learning models that capture complex temporal dependencies. Long Short-Term Memory networks, introduced by Hochreiter and Schmidhuber (1997), address vanishing gradients through three gates that selectively retain information across sequences [1]. Gated Recurrent Units, proposed by Cho et al. (2014), use fewer parameters while retaining competitive temporal modelling capacity [2]. Their relative performance is context-dependent: Chung et al. (2014) found comparable performance across sequence benchmarks [17], whereas Sivadasan et al.

(2024) reported inconsistent rankings across stock datasets despite both outperforming classical models [18]. Hybrid architectures may combine complementary gating mechanisms; Farhadi et al. (2025) used 42 technical indicators across eight stocks and reported improvements from a hybrid LSTM-GRU model [19]. The distinct gating mechanisms may capture complementary temporal patterns. MLP remains relevant as a non-sequential baseline, particularly when engineered features already encode historical behaviour, and Kara et al. (2011) found it competitive with more complex stock-prediction architectures [20].

Data representation is therefore central to deep-learning performance. Chong et al. (2017) examined deep neural networks on five-minute Korean KOSPI data using principal component analysis, autoencoder, and restricted Boltzmann machine feature extraction, finding that performance depends heavily on input representation [4]. Deep networks extracted predictive information from transformed representations that was inaccessible through raw inputs. This is directly relevant to the present study, which examines the related but distinct question of whether domain-informed technical indicators reduce the need for explicit temporal windowing. If indicators already capture recent market dynamics, overlapping sequences may add redundant rather than new information.

Sliding-window construction remains widely used but rarely tested as a hypothesis. Sequence construction is among the underexamined stages of deep learning-based stock forecasting [21]. Liu and Wang (2025) tested sliding-window effects on LSTM across six major stock indices using daily data from 2010 to 2019, finding that shorter intervals improved predictions and that stationarity affected performance [22]. However, prior research has not directly tested whether windowing adds value when identical inputs already contain technical indicators, and most studies evaluate only one window length. Consequently, to the best of our knowledge, no study has systematically varied window size while comparing windowed and non-windowed configurations using the same data and feature space.

Saudi deep-learning forecasting research remains limited, as Table I shows. The closest studies are Aseeri (2023) [11], Baamer and Aljamaan (2025) [12], and Alreshidi (2025) [23]. Aseeri (2023) used GRU with five technical indicators across 10 TASI companies for multi-step closing-price forecasts but tested one recurrent architecture and a sliding window without non-windowed or alternative-length comparisons [11]. Baamer and Aljamaan (2025) found GRU and LSTM consistently outperformed BiRNN, with Ensemble Averaging strongest overall, but did not include a hybrid architecture, windowed-versus-non-windowed comparisons, or an MLP baseline [12]. Alreshidi (2025) compared seven models across eight Saudi stocks and found XGBoost produced the lowest errors with substantially less computation than LSTM [23], but used lag-based features without technical indicators and tested neither GRU nor hybrid architectures.

Other Saudi studies provide complementary context. Alotaibi et al. (2018) applied neural networks to four Tadawul sectors [7]; Jarrah and Salim (2019) combined wavelet transforms with recurrent neural networks for closing-price prediction [8]; Alamro et al. (2019) integrated sentiment analysis with RNN to predict the TASI index [9]; and Alsulmi et al.

TABLE I. POSITIONING OF THE PRESENT STUDY RELATIVE TO KEY PRIOR WORKS ON SAUDI STOCK MARKET FORECASTING

Study	Year	Models	No. of Companies	Sliding Window Tested	Target Type	No. of Evaluation Metrics
Alotaibi et al. [7]	2018	Neural Network, MLP, LSTM	4 sectors	No	Closing price regression	4 (MSE, RMSE, R, NRMSE)
Alsubaie et al. [13]	2019	NB, FTNB, ANN, SVM, Meta-Cost, CSFTNB (9 classifiers)	99 + TASI	No	Direction classification (Up/Down/Flat)	3 (Accuracy, Cost, Sharpe)
Jarrah & Salim [8]	2019	DWT + RNN, ARIMA	Selected	No	Closing price regression (next-day)	3 (MSE, RMSE, MAE)
Alamro et al. [9]	2019	RNN + Sentiment	TASI index	No	Closing price regression (next-day)	Not clearly reported
Alshammari et al. [5]	2020	MODWT-GARCH, ARIMA	1 (TASI index)	No	Volatility forecasting	4 (RMSE, MAE, MAPE, MASE)
Alsulmi et al. [10]	2022	ML classifiers + risk management	10 (portfolio)	No	Direction classification	Accuracy, ROI
Aseeri [11]	2023	GRU, VARMAX	10	No	Closing price regression (multi-step)	4 (MAE, MSE, RMSE, R^2)
Baamer & Aljamaan [12]	2025	LSTM, GRU, BiRNN, Ensemble Averaging	Tadawul dataset	No	Closing price regression	Multiple (MAE, MSE, RMSE, R^2)
Alreshidi [23]	2025	LSTM, DT, RF, XGBoost, SVR, ARIMA, KNN (7 models)	8	No	Closing price regression	4 (MAE, MSE, RMSE, R^2)
Present Study	2025	LSTM, Hybrid LSTM-GRU, MLP	19 (19 sectors)	Yes (windowed vs. non-sliding)	Closing price regression (next-day)	8 (MAE, MSE, MBE, RMSE, MSLE, RMSLE, NRMSE, R^2)

(2022) developed a machine-learning trading framework using 31 features [10]. However, these studies differ considerably in preparation quality, comparison design, and evaluation transparency. Alshammari et al. (2020) examined volatility rather than individual-stock deep-learning prediction [5]; Alsubaie et al. (2019) analysed feature selection for direction classification without recurrent architectures [13]; and Aseeri (2023) demonstrated GRU effectiveness without comparing architectures or controlling window use [11]. The present study addresses these limitations together.

Three observations motivate the study. First, technical indicators enrich raw market data, although their optimal number and selection are market-dependent, as Alsubaie et al. (2019) showed [13]. Second, LSTM, GRU, hybrid architectures, and MLP have demonstrated value, but their relative performance remains context-dependent and has not been compared consistently for Saudi equities [12]. Third, sliding-window construction is commonly assumed to improve recurrent forecasting, but as far as we know no Saudi study has tested that assumption and no study in any market has examined it across multiple window sizes when technical indicators already enrich the inputs [22]. The present study therefore compares three architectures under four settings comprising three window lengths and a non-windowed baseline.

III. DATASET AND FEATURE CONSTRUCTION

This study uses historical daily data for selected companies listed on the Saudi stock market. To represent the diversity of the Tadawul All Share Index (TASI), one company was randomly selected from each sector, subject to the company having been listed for more than ten years. This criterion ensures a sufficiently long historical series for feature construction and longitudinal forecasting while maintaining sector coverage. Table II reports the company names, tickers, and TASI sectors obtained from official Saudi Exchange records.

Historical data were retrieved from Yahoo Finance [24] using Saudi-market ticker formats. For each company, daily

TABLE II. SELECTED COMPANIES INCLUDED IN THE STUDY

Company	Ticker	TASI Sector
Dar Al Arkan	4300	Real Estate Management & Development
Almarai	2280	Food & Beverages
STC	7010	Telecommunication Services
Al-Baha Investment	4130	Financial Services
Tihama	4070	Media & Entertainment
SPIMACO	2070	Pharma, Biotech & Life Science
AlRajhi Bank	1120	Banks
AlSagr Insurance	8180	Insurance
Bahri	4030	Energy
Al-Othaim Markets	4001	Consumer Staples Distribution & Retail
CATRION	6004	Commercial & Professional Services
Herfy	6002	Consumer Services
Saudi Cable	2110	Capital Goods
Naseej	1213	Consumer Durables & Apparel
Saudi Electricity	5110	Utilities
SABIC	2010	Materials
SAPTCO	4040	Transportation
Jarir	4190	Consumer Discretionary Distribution & Retail
Al-Mouwasat Hospital	4002	Health Care Equipment & Services

open, high, low, close, and volume (OHLCV) observations were collected from 1 January 2010 to 31 December 2023. These variables form the raw market layer and support target and feature construction. OHLCV data capture daily price behaviour and trading activity and provide the inputs for technical indicators [25].

The target is the next-day closing price, formulated as a one-step-ahead regression in which features at time t predict the closing price at time $t+1$. Each company's closing-price

series was shifted one trading day forward so that every observation aligned with the following day's close.

Technical feature construction transforms OHLCV observations into measures of trend, momentum, volatility, oscillator movement, and volume pressure [26]. The raw variables were combined with indicators selected with reference to Alsubaie et al. (2019), who studied 50 indicators across 99 Tadawul-listed stocks [13]. Twenty-two indicators were retained, spanning all four functional categories. This allows the neural models to identify the most informative dimensions during training.

The 22 indicators are Accumulation/Distribution, Average Directional Index, Average Directional Movement Rating, Aroon Up, Aroon Down, Aroon Oscillator, Average True Range, Bollinger Upper Band, Money Flow Index, Midpoint, Normalised Average True Range, On-Balance Volume, rolling standard deviation, T3, Chaikin Oscillator, Fast Stochastic %K and %D, Stochastic RSI components, and Williams %R. Together with OHLCV, they yield 27 variables per observation. Collectively, the indicators represent trend direction and strength, momentum and oscillator shifts, price variability, and accumulation, distribution, and money-flow dynamics. This domain-informed layer is a substantive contribution of the unified Saudi forecasting framework. Table III groups the inputs by financial function.

After feature construction, only complete and usable observations were retained. Technical-indicator lookback periods naturally generate missing values at the beginning of each company's series; these rows and any other rows containing missing values were removed to maintain consistency across the final feature matrix for longitudinal modeling across all selected companies. Every retained observation therefore contained valid values for all 27 variables (22 technical indicators and 5 OHLCV).

Four versions of the dataset were prepared. The first retained individual multivariate observations without explicit sliding windows, allowing direct learning from the structured feature representation. The other three transformed each company's feature matrix into overlapping, temporally ordered samples using the preceding 5, 10, or 22 trading days to predict the next day's close. These periods correspond approximately to one trading week, two trading weeks, and one trading month and span commonly used lookbacks: Aseeri (2023) used 5 days for Saudi stocks [11], Alsubaie et al. (2019) used 20 days for feature calculation [13], and Liu and Wang (2025) found shorter intervals tended to improve LSTM performance [22]. Preparing all four forms enables direct comparison of non-windowed and sequence-based designs across multiple temporal scales under the same data source, feature space, and prediction target.

Each prepared dataset was divided chronologically: observations from 2010 to 2020 formed the training set, and observations from 2021 to 2023 formed the test set. This preserves temporal integrity and evaluates the models on future unseen observations rather than randomly sampled records [27]. Feature normalisation was applied before model training; the specific scaling procedure and its rationale are detailed in the methodology section.

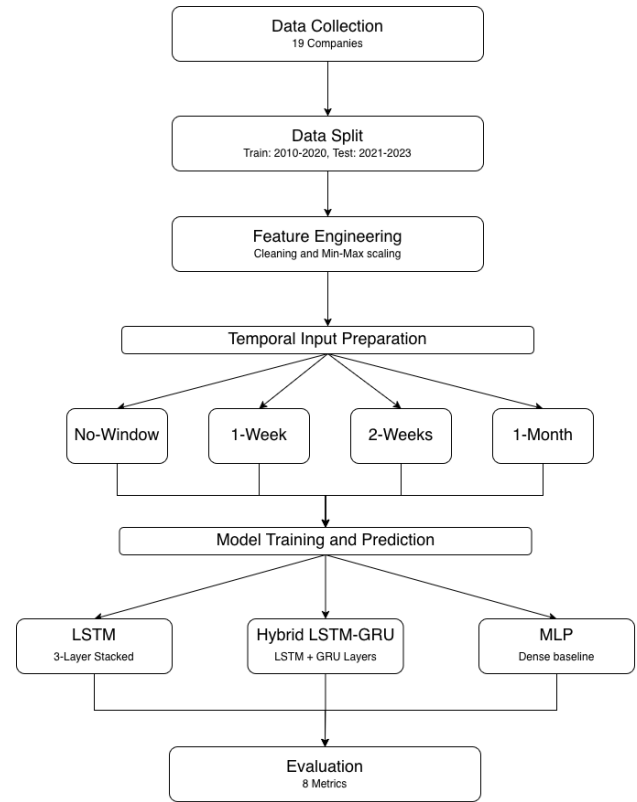


Fig. 1. Overview of the proposed forecasting framework.

IV. PROPOSED METHODOLOGY

This section describes the methodology used to forecast next-day closing prices for the selected Saudi stocks. It presents the overall forecasting framework, contrasts the sequence-based and non-windowed input designs, and specifies the LSTM, hybrid LSTM-GRU, and MLP models together with the naïve persistence baseline.

A. Overview of the Forecasting Framework

This study proposes a comparative deep learning framework for next-day Saudi stock closing-price forecasting using multivariate OHLCV data and systematically calculated technical indicators. It evaluates how alternative neural architectures perform under different temporal input designs within a unified Saudi forecasting setting. It compares LSTM, hybrid LSTM-GRU, and MLP under identical features and targets in four settings: without sliding windows and with 5-, 10-, and 22-day windows. Holding the underlying feature space and prediction target constant enables performance differences to be attributed to architecture and temporal input design. The resulting 12 configurations identify the strongest model and test whether explicit sequence construction improves performance, and whether its effect depends on window length, when inputs already contain domain-informed technical features. Fig. 1 shows the complete pipeline from raw data collection and feature engineering through scaling, four-form input preparation, model training, and evaluation.

TABLE III. SUMMARY OF INPUT FEATURES USED IN THE STUDY

Feature Group	Variables	Description
Raw market variables	Open, High, Low, Close, Volume	Daily OHLCV variables collected for each selected stock and used as the base market representation.
Volume-based indicators	Accumulation/Distribution (AD), On-Balance Volume (OBV), Chaikin Oscillator (ADOSC), Money Flow Index (MFI)	Indicators derived from price and trading volume to capture accumulation, distribution, and money flow dynamics.
Trend indicators	Average Directional Index (ADX), Average Directional Movement Rating (ADXR), Aroon Up, Aroon Down, Aroon Oscillator, T3	Indicators used to capture trend direction, persistence, and trend strength.
Volatility indicators	Average True Range (ATR), Normalized Average True Range (NATR), Bollinger Upper Band, Rolling Standard Deviation	Indicators used to represent price variability and market volatility.
Momentum and oscillator indicators	Fast Stochastic %K, Fast Stochastic %D, Stochastic RSI %K, Stochastic RSI %D, Williams %R, Midpoint	Indicators used to capture relative price movement, momentum shifts, and overbought/oversold conditions.

B. Data Scaling and Input Preparation

Before training, Min-Max scaling was applied because the raw market variables and technical indicators have substantially different numerical ranges that may adversely affect neural-network optimization [28]. Separate scalers were fitted to the feature matrix and target. This separation is required because the dedicated target scaler enables predictions to be returned to the original closing-price scale, while the feature scaler normalises the input dimensions independently. Both scalers were fitted only on training data and then applied unchanged to testing data to prevent information leakage.

C. Sequence-Based Forecasting with Sliding Window

Sequence-based forecasting constructs each input from a fixed number of preceding trading days and predicts the following day's close. Three window lengths of 5, 10, and 22 days were tested, representing approximately one trading week, two trading weeks, and one trading month. These lengths span common lookbacks: Aseeri (2023) used 5 days for Saudi stocks [11], Alsubaie et al. (2019) used 20 days for feature calculation [13], and Liu and Wang (2025) found shorter intervals tended to improve LSTM performance [22]. For each length, each company's multivariate feature matrix was transformed into overlapping fixed-length sequences that preserve observation order and expose recurrent models to explicitly structured short-term historical patterns. Testing multiple lengths distinguishes a length-specific effect from a general effect of windowing on feature-rich inputs and follows common recurrent forecasting practice [3].

D. Non-Windowed Forecasting Without Sliding Window

Non-windowed forecasting uses each multivariate observation directly to predict the next day's close, without an explicit multi-day window. MLP naturally processes each observation as a single fixed-length feature vector. Because LSTM and hybrid LSTM-GRU require three-dimensional inputs of the form samples $\times$ timesteps $\times$ features, each observation is reshaped with a single timestep. The recurrent gates therefore remain active but process only one timestep, reducing their temporal modelling capacity to a minimal level. This design choice is intentional: comparing single-step input with full 5-, 10-, and 22-day windows under the same architectures

and features isolates the contribution of explicit multi-step sequence construction rather than assuming that it is beneficial.

E. LSTM Model

The Long Short-Term Memory (LSTM) network is the primary recurrent benchmark and one of the most widely used architectures in financial forecasting. Its memory cells and input, forget, and output gates regulate information retention, updating, and removal, addressing vanishing gradients and supporting temporal learning [1]. These mechanisms can retain or discard useful signals as market conditions change, making LSTM suitable for historical financial patterns and short-horizon forecasting [29]. LSTM is evaluated under all four settings; its layers, hyperparameters, and training procedure are specified in the experimental design section.

The LSTM model consisted of three stacked LSTM layers containing 128, 64, and 32 units, respectively. The first two recurrent layers returned the complete sequence to the subsequent LSTM layer, whereas the final LSTM layer returned only the final hidden representation. Each LSTM layer used a tanh activation function and was followed by a dropout layer with a rate of 0.2. A single linear output unit produced the next-day closing-price prediction. For the non-windowed configuration, each observation was represented as a sequence of one timestep, whereas the windowed configurations used sequences of 5, 10, or 22 trading days.

F. Hybrid LSTM-GRU Model

The hybrid architecture stacks an LSTM layer followed by a GRU layer, passing the LSTM output sequence to the GRU. The first layer captures longer-range dependencies through its three-gate mechanism, while the second refines the representation through a simpler two-gate structure. This combines LSTM's temporal learning with GRU's more compact behaviour, allowing complementary longer- and shorter-term patterns to be represented within one recurrent pipeline [19]. The hybrid model tests whether combining the two gating mechanisms improves Saudi stock forecasting over LSTM alone.

The hybrid model combined an LSTM layer with 100 units and a GRU layer with 50 units. The LSTM layer returned the full sequence to the GRU layer, which produced

the final recurrent representation. Both recurrent layers used tanh activation with sigmoid recurrent activation. A ReLU activation was subsequently applied before a single linear output unit generated the next-day closing-price prediction. As with the LSTM model, the non-windowed configuration used a sequence length of one, while the windowed configurations used 5, 10, or 22 timesteps.

G. MLP Model

The Multilayer Perceptron (MLP) is the dense neural baseline. Unlike recurrent models, it has no temporal memory and processes fixed-length inputs through successive fully connected layers with nonlinear activation functions, learning hierarchical nonlinear mappings between features and the target. MLP is evaluated in all four settings; each windowed input is flattened into one feature vector before entering the network. Including it permits direct assessment of whether the technical indicators already encode sufficient historical behaviour without temporal memory. If MLP performs competitively, this would reduce the apparent value of explicit sequential modelling. This fully crossed design tests whether the calculated technical-indicator representation alone supports effective forecasting and whether explicit recurrent sequence processing adds value relative to a non-recurrent architecture.

The MLP provided a non-recurrent comparison using the same feature space. It consisted of three fully connected hidden layers containing 256, 128, and 64 units with ReLU activation. Dropout with a rate of 0.3 was applied after the first and second hidden layers. The output layer consisted of a single linear unit for next-day closing-price regression. For windowed configurations, the temporally ordered inputs were flattened before being passed to the network, whereas the non-windowed configuration directly used the feature vector for each observation.

H. Naïve Persistence Baseline

A naïve persistence baseline was included as a non-trained benchmark. For each company, the next trading day's closing price was predicted as the current closing price, $\hat{y}_{t+1} = y_t$. The baseline required no parameter estimation or optimization and tested whether each neural configuration added predictive value beyond simple price persistence. It was evaluated once for each of the 19 companies using the same 2021–2023 testing period and eight regression metrics as the neural models.

V. EXPERIMENTAL DESIGN

This section sets out the experimental design: the forecasting scenarios, compared models, training configuration, evaluation metrics, and statistical tests. Data preparation and evaluation are held constant so that performance differences reflect architecture and temporal input design alone.

A. Experimental Method

A controlled comparison was designed under identical data conditions. Every model used the same next-day closing-price target, 27-variable feature space, and chronological partition, so performance differences reflect architecture and temporal input design rather than inconsistent preparation or evaluation.

The design produced 228 neural model–company experiments (12 model configurations $\times$ 19 companies) and 19 naïve-baseline evaluations, yielding 247 evaluations in total.

B. Forecasting Scenarios

Four forecasting scenarios were examined: direct use of each multivariate observation without explicit windowing and sequence-based input constructed from the preceding 5, 10, or 22 trading days. These windows represent approximately one trading week, two trading weeks, and one trading month. Evaluating every model in all four scenarios permits direct comparison across temporal scales and distinguishes a window-length effect from a general effect of sequence construction.

C. Compared Models

The comparison included LSTM, hybrid LSTM-GRU, and MLP, each evaluated in four scenarios across 19 companies, yielding 12 neural model–scenario configurations. A naïve persistence baseline was evaluated separately for each company because it required neither an architecture nor a window setting. The recurrent models tested explicit sequence learning, while MLP provided a non-sequential baseline with flattened windowed inputs. The naïve benchmark established whether the trained configurations improved on simple price persistence.

D. Training Configuration

All experiments used a chronological split: 2010–2020 observations for training and 2021–2023 observations for testing. Min-Max scaling used separate feature and target scalers fitted only on training data to prevent information leakage. Models were trained for at most 1,000 epochs with early stopping patience of 50 epochs. The feature set and next-day target remained identical throughout to ensure fair comparison.

E. Evaluation Metrics

Performance was evaluated using eight regression metrics. Table IV defines each metric, where y_i is the actual closing price, $\hat{y}_i$ is the predicted closing price, $\bar{y}$ is the mean of actual values, and n is the number of observations. Together, MAE and RMSE measure average deviation, MSE emphasises large errors, MBE captures directional bias, MSLE and RMSLE measure relative error, NRMSE supports cross-company comparison, and R^2 measures explained variance. This breadth avoids the limitations of single-metric reporting [30].

These metrics were selected to evaluate forecasting error from different perspectives, including average deviation (MAE, RMSE), sensitivity to large errors (MSE), directional bias (MBE), relative error magnitude (MSLE, RMSLE), normalised cross-company comparability (NRMSE), and coefficient of determination (R^2). Using eight metrics addresses a common limitation in prior Saudi studies, where single-metric reporting can obscure important aspects such as systematic bias or scale sensitivity [30].

TABLE IV. MATHEMATICAL DEFINITIONS OF THE EVALUATION METRICS

Metric	Formula	Interpretation
MAE	$MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $	Average absolute deviation; robust to outliers
MSE	$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$	Average squared error; penalises large errors
MBE	$MBE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)$	Directional bias; positive = overestimation
RMSE	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$	Scale-consistent error in original units
MSLE	$MSLE = \frac{1}{n} \sum_{i=1}^n (\log(1 + y_i) - \log(1 + \hat{y}_i))^2$	Penalises relative error; suited to skewed targets
RMSLE	$RMSLE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(1 + y_i) - \log(1 + \hat{y}_i))^2}$	Scale-independent relative error
NRMSE	$NRMSE = \frac{RMSE}{y_{\max} - y_{\min}}$	Normalised error for cross-stock comparison
R^2	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	Proportion of variance explained; 1.0 = perfect

F. Statistical Significance Testing

The Wilcoxon signed-rank test was used for pairwise comparisons of configuration RMSE across the 19 companies. This non-parametric test does not assume normally distributed performance differences. The Friedman test assessed overall differences across all 12 configurations. Both tests used $\alpha = 0.05$, following the Wilcoxon and Friedman approach used by Alsubaie et al. (2019) in Saudi stock prediction [13].

G. Comparison Strategy

Results were compared at three levels: aggregate performance across companies, company-level consistency, and the effect of window length within each architecture. Results were also examined across architectures and forecasting scenarios, identifying both the strongest overall model and whether any sliding-window length provided a measurable advantage.

VI. RESULTS

Results from 228 model–company experiments and 19 naïve-baseline evaluations on the 2021–2023 test period are presented by aggregate performance, forecasting setting, window size, company, and statistical significance using eight regression metrics.

A. Aggregate Performance

Table V summarises the mean performance across 19 companies for the 12 neural configurations and the separate naïve persistence baseline.

The two non-windowed recurrent models led every metric. Hybrid LSTM-GRU achieved the lowest average RMSE (1.1585) and R^2 of 0.9560, while LSTM achieved RMSE of 1.1826 and the highest R^2 of 0.9583. Sliding-window recurrent RMSE ranged from 1.80 to 2.59 across architectures and window sizes; MAE, MSE, MSLE, RMSLE, and NRMSE confirmed the same ordering.

MLP recorded negative average R^2 in all four settings, from -0.25 (non-sliding) to -1.60 (22-day), indicating performance below the mean-value baseline. Non-windowed MLP was its strongest configuration (RMSE 5.67), but it performed worse

than all recurrent configurations, including those using sliding windows. Windowed MLP produced the highest forecasting errors among the neural configurations, with RMSE ranging from 6.54 to 8.02, and did not generalise effectively across the stocks.

The naïve baseline produced the worst aggregate performance on all six non-bias error measures. It recorded the highest average among the 13 evaluated configurations. Thus, every neural configuration reduced aggregate forecasting error relative to naïve persistence.

The MBE values show differences in directional bias across model types. The non-windowed recurrent models produced near-zero average MBE values (0.0893 for hybrid LSTM-GRU and 0.1593 for LSTM), which suggests balanced prediction behaviour with no systematic over- or under-estimation. The sliding-window recurrent models also showed relatively small bias, with MBE values ranging from -0.15 to 0.64 . In contrast, the MLP models exhibited consistent negative bias across all settings (MBE ranging from -1.05 to -2.86), which indicates systematic underestimation of closing prices.

B. Performance by Forecasting Setting

Table VI presents the average performance grouped by forecasting setting. Because MLP's poor performance across all settings would distort the group averages, the recurrent model averages are reported separately from MLP. This separation is necessary to show the true magnitude of the difference between sliding and non-windowed configurations for the recurrent architectures.

The comparison of recurrent models across forecasting settings showed that the non-windowed configuration produced an average RMSE of 1.17 compared to 1.85, 2.00, and 2.45 for the 5-day, 10-day, and 22-day sliding-window configurations, respectively. This represents RMSE reductions of approximately 37%, 42%, and 52%, respectively. The average R^2 was 0.957 for non-windowed compared to 0.903, 0.886, and 0.816 for the 5-day, 10-day, and 22-day sliding windows, respectively. This indicates that removing explicit sliding-window construction substantially improved forecasting accuracy for the recurrent architectures under the implemented experimental design.

TABLE V. AVERAGE FORECASTING PERFORMANCE ACROSS ALL COMPANIES

Model	Size	MAE	MSE	MBE	RMSE	MSLE	RMSLE	NRMSE	R ²
Hybrid LSTM-GRU	0	0.8680	2.3987	0.0893	1.1585	0.0021	0.0328	0.0413	0.9560
LSTM	0	0.8916	2.3145	0.1593	1.1826	0.0015	0.0317	0.0437	0.9583
MLP	0	5.1770	59.0101	-1.0461	5.6657	0.0334	0.1527	0.2251	-0.2479
Hybrid LSTM-GRU	5	1.4623	6.9260	0.2274	1.8937	0.0045	0.0488	0.0642	0.8928
LSTM	5	1.3274	5.5423	0.1414	1.8040	0.0034	0.0467	0.0633	0.9127
MLP	5	5.8915	72.6786	-2.8593	6.5448	0.0367	0.1683	0.2628	-0.8893
Hybrid LSTM-GRU	10	1.5087	7.1847	0.0199	1.9618	0.0042	0.0513	0.0695	0.8867
LSTM	10	1.5483	7.4500	0.3031	2.0456	0.0041	0.0517	0.0714	0.8859
MLP	10	6.0545	75.2413	-2.0551	6.7895	0.0429	0.1804	0.2726	-0.7629
Hybrid LSTM-GRU	22	2.0838	15.5436	0.6358	2.5881	0.0095	0.0647	0.0820	0.7991
LSTM	22	1.7624	9.5022	-0.1508	2.3090	0.0135	0.0709	0.0809	0.8337
MLP	22	7.2452	113.4205	-2.2435	8.0175	0.0585	0.2122	0.3236	-1.5957

TABLE VI. AVERAGE FORECASTING PERFORMANCE BY FORECASTING SETTING (RECURRENT MODELS ONLY)

Setting	Average RMSE	Average R ²
Non-windowed (recurrent only)	1.17	0.957
5-day sliding (recurrent only)	1.85	0.903
10-day sliding (recurrent only)	2.00	0.886
22-day sliding (recurrent only)	2.45	0.816

TABLE VII. AVERAGE RMSE BY ARCHITECTURE AND WINDOW SIZE

Architecture	Non-windowed	5-day	10-day	22-day
LSTM	1.18	1.80	2.05	2.31
Hybrid LSTM-GRU	1.16	1.89	1.96	2.59
MLP	5.67	6.54	6.79	8.02

C. Effect of Window Size

Table VII presents the average RMSE by architecture and window size. This comparison is important because it shows whether the performance degradation caused by sliding-window construction depends on the specific window length or reflects a more general pattern.

Within windowed recurrent models, RMSE increased as windows lengthened: 5-day outperformed 10-day, which outperformed 22-day. The 5-to-22-day differences were 0.51 for LSTM and 0.70 for hybrid LSTM-GRU, whereas the gap between non-windowed and the best windowed configuration was 0.62 for LSTM and 0.73 for hybrid LSTM-GRU. Window presence therefore had a substantial effect, while window length also influenced forecasting performance. MLP showed the same length pattern, with higher errors for longer windows, and remained weakest overall.

D. Company-Level Analysis

Table VIII reports the number of companies for which each model achieved the lowest RMSE.

Across all 19 companies, the lowest RMSE was achieved exclusively by the two non-windowed recurrent models. The hybrid LSTM-GRU without sliding window achieved the best

TABLE VIII. NUMBER OF COMPANIES FOR WHICH EACH MODEL ACHIEVED THE LOWEST RMSE

Model	Company-Level Wins
Hybrid LSTM-GRU without sliding window	12
LSTM without sliding window	7
All sliding-window models (6 configurations)	0
All MLP models (4 configurations)	0

result for 12 companies, while the LSTM without sliding window achieved the best result for the remaining 7. None of the six sliding-window configurations and none of the four MLP configurations achieved the lowest RMSE for any individual company. The R² values for the best model per company ranged from 0.9247 (Naseej) to 0.9940 (Herfy), which means that the non-windowed recurrent models explained more than 92% of the closing price variance for every company in the sample.

To show the company-level results in more detail, Table IX presents the full metric results for SABIC, one of the largest companies in the dataset and a representative example from the Materials sector.

For SABIC, the same overall pattern from the aggregate results holds at the individual company level. The hybrid LSTM-GRU without sliding window achieved the lowest RMSE and the highest R², followed closely by the LSTM without sliding window. All sliding-window models produced RMSE values substantially higher than the non-windowed recurrent models, regardless of window size. The MLP configurations showed the weakest performance and the strongest directional bias. Fig. 2 presents the actual-versus-predicted closing price series for SABIC under selected model configurations during the testing period.

E. Statistical Significance

To confirm that the observed performance differences are statistically meaningful, the Wilcoxon signed-rank test was applied for pairwise comparisons and the Friedman test across all 12 configurations. Table X summarises the key pairwise results.

The Friedman test across all 12 model configurations confirmed a highly significant overall difference ($\chi^2 = 116.69$,

TABLE IX. FORECASTING RESULTS FOR SABIC ACROSS ALL 12 MODEL CONFIGURATIONS

Model	Size	MAE	MSE	MBE	RMSE	MSLE	RMSLE	NRMSE	R ²
LSTM	0	0.9270	1.4535	0.1711	1.2056	0.0002	0.0141	0.0260	0.9890
Hybrid LSTM-GRU	0	0.8365	1.2869	0.0462	1.1344	0.0002	0.0133	0.0244	0.9902
MLP	0	3.3939	26.0343	-3.3228	5.1024	0.0028	0.0529	0.1099	0.8027
LSTM	5	1.4317	3.7748	0.0585	1.9429	0.0005	0.0226	0.0419	0.9712
Hybrid LSTM-GRU	5	1.3694	3.4307	0.0880	1.8522	0.0005	0.0213	0.0399	0.9739
MLP	5	6.9069	64.2174	-6.9057	8.0136	0.0084	0.0917	0.1726	0.5107
LSTM	10	1.5536	4.2040	0.0632	2.0504	0.0006	0.0240	0.0442	0.9676
Hybrid LSTM-GRU	10	1.6761	4.7454	0.0646	2.1784	0.0006	0.0254	0.0469	0.9634
MLP	10	10.7621	135.2841	-10.7621	11.6312	0.0201	0.1419	0.2506	-0.0434
LSTM	22	1.8395	5.8304	0.0393	2.4146	0.0008	0.0288	0.0520	0.9525
Hybrid LSTM-GRU	22	1.5964	4.3802	-0.3596	2.0929	0.0006	0.0247	0.0451	0.9643
MLP	22	8.9422	100.8312	-8.9422	10.0415	0.0140	0.1182	0.2163	0.1780

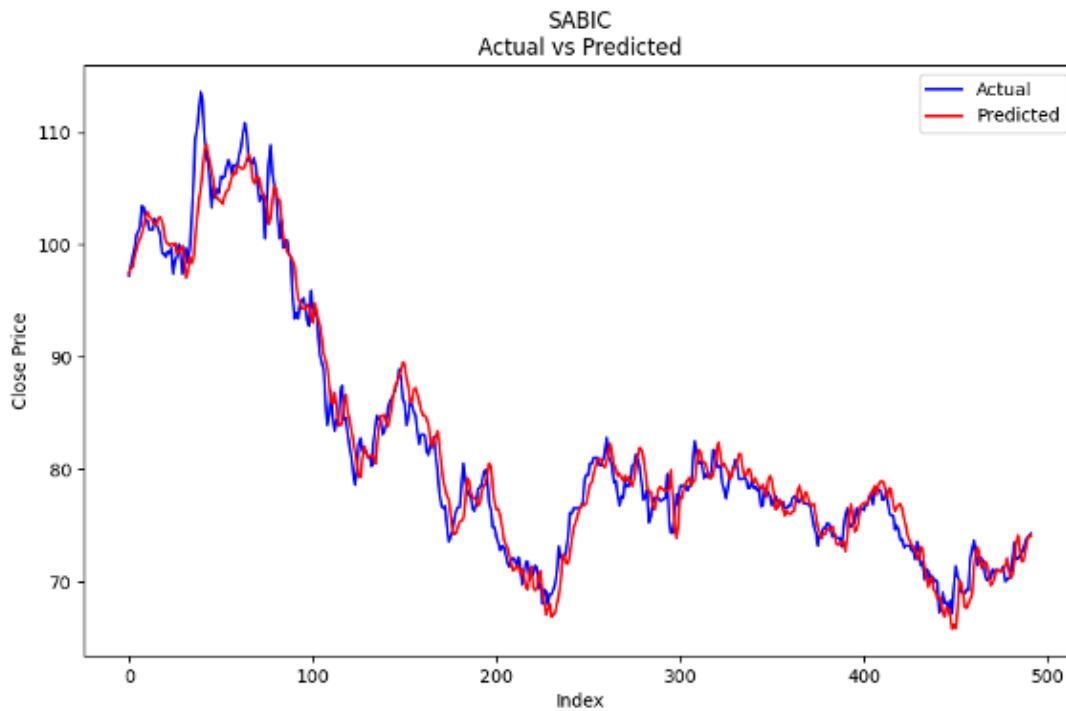


Fig. 2. Actual and predicted closing prices for SABIC using the LSTM model with a 5-day window.

$p < 10^{-10}$). All reported comparisons between non-windowed and 10-day sliding-window configurations were statistically significant ($p \leq 0.013$), providing strong statistical evidence of systematic performance differences between the two configurations. The comparison between the non-windowed LSTM and the non-windowed hybrid LSTM-GRU was not significant ($p = 0.145$), which indicates that no statistically significant performance difference was detected between these two architectures when the sliding window was removed. The window-size comparisons showed that the differences between 5-day and 10-day windows and between 10-day and 22-day windows were statistically significant for LSTM ($p = 0.009$ and $p = 0.035$, respectively), while neither comparison was significant for hybrid LSTM-GRU ($p = 0.13$ and $p = 0.46$, respectively). This suggests that within the sliding-window

setting, window size had a statistically detectable effect for LSTM but not for hybrid LSTM-GRU.

VII. DISCUSSION

The results clearly address the paper's comparative objective. Non-windowed recurrent models consistently produced the lowest errors, while MLP models produced the highest. Non-windowed LSTM achieved the best average R^2 of 0.9583, and non-windowed hybrid LSTM-GRU achieved the lowest average RMSE of 1.1585 and the best company-level result for 12 companies, compared with 7 for LSTM. No sliding-window or MLP configuration led any company. The difference between the non-windowed recurrent models was not statistically significant (Wilcoxon $p = 0.145$), indicating that no significant performance difference was detected between these two archi-

TABLE X. WILCOXON SIGNED-RANK TEST RESULTS FOR KEY MODEL PAIRS

Comparison	<i>p</i> -value	Significant ($\alpha = 0.05$)
LSTM: non-windowed vs 10-day sliding	4×10^{-5}	Yes
Hybrid LSTM-GRU: non-windowed vs 10-day sliding	$p < 10^{-5}$	Yes
MLP: non-windowed vs 10-day sliding	0.013	Yes
Non-sliding: LSTM vs hybrid LSTM-GRU	0.145	No
Non-windowed LSTM vs non-windowed MLP	$p < 10^{-5}$	Yes
Non-windowed hybrid LSTM-GRU vs non-windowed MLP	$p < 10^{-5}$	Yes
LSTM: 5-day vs 10-day sliding	0.009	Yes
LSTM: 10-day vs 22-day sliding	0.035	Yes
Hybrid LSTM-GRU: 5-day vs 10-day sliding	0.13	No
Hybrid LSTM-GRU: 10-day vs 22-day sliding	0.46	No

textures, while the non-windowed configurations outperformed their sliding-window counterparts.

Comparison with the naïve persistence baseline further showed that the neural configurations achieved lower forecasting errors than simply carrying the latest closing price forward. The baseline had the highest aggregate errors on MAE, MSE, RMSE, MSLE, RMSLE, and NRMSE, whereas every trained configuration reduced these measures. This benchmark strengthens the interpretation of the neural results by showing that their error reductions were not obtained merely from price-level persistence.

The principal finding is that non-windowed recurrent models outperformed sliding-window recurrent models across the reported error metrics and three window lengths. Average recurrent-model RMSE decreased from 1.85 (5-day), 2.00 (10-day), and 2.45 (22-day) to 1.17 without windowing, representing reductions of approximately 37%, 42%, and 52%, respectively. The comparisons between non-windowed and 10-day sliding-window configurations were statistically significant for both recurrent architectures (Wilcoxon $p = 0.00004$ for LSTM and $p < 0.00001$ for hybrid LSTM-GRU). This challenges treating sliding windows as standard in recurrent stock forecasting [22]. It extends Liu and Wang (2025), who found shorter windows improved LSTM performance across six major indices [22], by showing that removing the window performed better still. It also extends Chong et al. (2017), who linked performance to representation [4], by identifying temporal input structure as an additional determinant.

The testing of multiple window sizes strengthens this conclusion in two ways. First, it shows that the degradation associated with sliding-window construction is not an artefact of a poorly chosen window length. Whether the window spans one week, two weeks, or approximately one month, the non-windowed configuration remained superior by a substantial margin. Second, it reveals that within the sliding-window setting, window size affected the recurrent architectures differently. The 5-day window produced the lowest errors among the sliding-window configurations, while the 22-day window produced the highest. The Wilcoxon test showed that the differences between 5-day and 10-day windows and between 10-day and 22-day windows were statistically significant for LSTM ($p = 0.009$ and $p = 0.035$, respectively), but not

for hybrid LSTM-GRU ($p = 0.13$ and $p = 0.46$, respectively). This indicates that window length had a statistically detectable effect on LSTM performance, while no significant differences were detected between the tested adjacent window sizes for hybrid LSTM-GRU.

A reasonable explanation for these results is that the technical-indicator layer already embeds a substantial amount of recent historical information. Indicators such as ADX, ATR, stochastic measures, OBV, and MFI are themselves functions of previous market movements. Therefore, the current multivariate observation already provides a condensed technical summary of short-term price and volume behaviour. Under such conditions, reorganising the same information into overlapping sliding-window sequences may present the model with redundant temporal information. The same recent market dynamics are summarised both within each observation's indicator values and across the multi-day sequence. This redundancy may introduce noise rather than complementary signal. Additionally, the larger input dimensionality of the windowed setting may increase model complexity and the number of learnable parameters, depending on the architecture, which could contribute to overfitting on training-period patterns that do not generalise well to the testing period.

The comparison between recurrent and non-recurrent models confirms that recurrent architectures achieved substantially better forecasting performance than MLP across all tested configurations. The MLP recorded negative average R^2 alues across all four settings, meaning it performed worse than a mean-value baseline regardless of whether a sliding window was used. By contrast, both non-windowed recurrent models achieved R^2 above 0.92 for every individual company. This suggests that the recurrent architectures captured relationships in the multivariate technical feature space more effectively than the feedforward baseline. The 27-feature space employed in this study falls below the 30-indicator threshold beyond which Alsubaie et al. (2019) observed performance degradation on Tadawul data [13], which supports the design choice of the present feature set.

From a methodological perspective, the results reinforce the importance of feature construction as a substantive part of stock forecasting design. The strong performance of the non-windowed recurrent models suggests that the systematic calculation of technical indicators contributed meaningfully to the predictive framework and that a sufficiently rich technical representation may reduce the practical benefit of explicit short-term sequence construction. This finding contributes to the Saudi stock forecasting literature by showing that performance differences cannot be attributed solely to the use of deep learning in general. They must be interpreted in light of how the input representation is built and how the forecasting task is formulated.

VIII. CONCLUSION

This study developed a deep learning framework for next-day Saudi stock closing-price forecasting using daily OHLCV data and systematically calculated technical indicators. LSTM, hybrid LSTM-GRU, and MLP were evaluated without sliding windows and with 5-, 10-, and 22-day windows, producing 228 experiments across 12 configurations and 19 companies spanning 19 TASI sectors.

Explicit sliding windows substantially increased forecasting errors when technical indicators enriched the inputs, regardless of window length. Non-windowed recurrent models achieved the best overall performance: LSTM achieved the highest average R^2 of 0.9583, while hybrid LSTM-GRU achieved the lowest average RMSE of 1.1585 and led 12 of 19 companies. The non-windowed and 10-day sliding-window configurations showed statistically significant differences for both recurrent architectures (Wilcoxon $p = 0.00004$ for LSTM and $p < 0.00001$ for hybrid LSTM-GRU), and the Friedman test confirmed overall differences across 12 configurations ($p < 0.0000000001$). Consistent degradation across all three window lengths suggests a possible structural effect: indicators already encode recent temporal context, so explicit multi-step sequences may introduce redundancy and complexity rather than improving prediction.

The study makes five contributions. First, it integrates OHLCV data with 22 systematically calculated indicators and demonstrates strong prediction without explicit windowing. Second, it provides, to the best of our knowledge, the first controlled Saudi comparison of non-windowed inputs with three window sizes. Third, it compares LSTM, hybrid LSTM-GRU, and MLP in one setting and shows that aggregate and company-level rankings can differ. Fourth, it demonstrates that window-related degradation persists across all tested lengths and architectures, with non-windowed recurrent models achieving the lowest RMSE for all 19 companies. Fifth, Wilcoxon signed-rank and Friedman tests provide statistical evidence of performance differences, addressing a common methodological limitation [13].

At the practical level, the results suggest that researchers and analysts working on short-horizon Saudi stock forecasting may achieve stronger results by investing in well-constructed multivariate technical representations rather than explicit sliding-window sequence designs. The non-windowed recurrent models not only achieved lower forecasting errors but also used simpler input structures, potentially reducing implementation complexity and computational cost.

The findings should be interpreted within the scope of the study. The analysis was limited to next-day closing-price forecasting using OHLCV data and technical indicators. The framework did not include macroeconomic, textual, or sentiment-based variables. Although three window sizes were tested, other lengths such as 1-day or 30-day windows may show different patterns.

REFERENCES

- [1] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [2] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Sep. 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [3] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Appl. Soft Comput.*, vol. 90, p. 106181, May 2020, doi: 10.1016/j.asoc.2020.106181.
- [4] E. Chong, C. Han, and F. C. Park, "Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, Data Representations, and Case Studies," 2017, doi: 10.1016/j.eswa.2017.04.030.
- [5] T. S. Alshammari, M. T. Ismail, S. Al-Wadi, M. H. Saleh, and J. J. Jaber, "Modeling and Forecasting Saudi Stock Market Volatility Using Wavelet Methods," *The Journal of Asian Finance, Economics and Business*, vol. 7, no. 11, pp. 83–93, Nov. 2020, doi: 10.13106/jafeb.2020.vol7.no11.083.
- [6] Tadawul, "Saudi Arabia Inclusion to FTSE Russell and S&P Dow Jones Into Emerging Market Indices." Accessed: Apr. 06, 2026. [Online]. Available: <https://www.prnewswire.com/news-releases/saudi-arabia-inclusion-to-ftse-russell-and-sp-dow-jones-into-emerging-market-indices-300813730.html>
- [7] T. Alotaibi *et al.*, "Saudi Arabia Stock Market Prediction Using Neural Network," *International Journal on Computer Science and Engineering*, vol. 9, no. 2, pp. 62–70, 2018, doi: 10.21817/ijcse/2018/v10i2/181002024.
- [8] M. Jarrah and N. Salim, "A recurrent neural network and a discrete wavelet transform to predict the Saudi stock price trends," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, pp. 155–162, 2019, doi: 10.14569/ijacsa.2019.0100418.
- [9] R. Alamro, A. McCarren, and A. Al-Rasheed, "Predicting Saudi Stock Market Index by Incorporating GDELT Using Multivariate Time Series Modelling," *Communications in Computer and Information Science*, vol. 1097 CCIS, pp. 317–328, 2019, doi: 10.1007/978-3-030-36365-9_26.
- [10] M. Alsulmi and N. Al-Shahrani, "Machine Learning-Based Decision-Making for Stock Trading: Case Study for Automated Trading in Saudi Stock Exchange," 2022, doi: 10.1155/2022/6542862.
- [11] A. O. Aseeri, "Effective short-term forecasts of Saudi stock price trends using technical indicators and large-scale multivariate time series," *PeerJ Comput. Sci.*, vol. 9, 2023, doi: 10.7717/peerj-cs.1205.
- [12] R. Baamer and H. Aljamaan, "Enhancing Stock Price Forecasting with Deep Learning: Insights from the Saudi Stock Market," *Comput. Econ.*, Jul. 2025, doi: 10.1007/s10614-025-11033-9.
- [13] Y. Alsubaie, K. El Hindi, and H. Als Salman, "Cost-Sensitive Prediction of Stock Price Direction: Selection of Technical Indicators," *IEEE Access*, vol. 7, pp. 146876–146892, 2019, doi: 10.1109/ACCESS.2019.2945907.
- [14] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econom.*, vol. 31, no. 3, pp. 307–327, Apr. 1986, doi: 10.1016/0304-4076(86)90063-1.
- [15] G. Campisi, S. Muzzioli, and B. De Baets, "A comparison of machine learning methods for predicting the direction of the US stock market on the basis of volatility indices," *Int. J. Forecast.*, vol. 40, no. 3, pp. 869–880, Jul. 2024, doi: 10.1016/j.ijforecast.2023.07.002.
- [16] S. M. Mostafavi and A. R. Hooman, "Key technical indicators for stock market prediction," *Machine Learning with Applications*, vol. 20, p. 100631, Jun. 2025, doi: 10.1016/j.mlwa.2025.100631.
- [17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014. [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [18] E. T. Sivadasan, N. Mohana Sundaram, and R. Santhosh, "RETRACTED ARTICLE: Stock market forecasting using deep learning with long short-term memory and gated recurrent unit," *Soft Comput.*, vol. 28, no. 4, pp. 3267–3282, Feb. 2024, doi: 10.1007/s00500-023-09606-7.
- [19] A. Farhadi, A. Zamanifar, A. Alipour, A. Taheri, and M. Asadolahi, "A Hybrid LSTM-GRU Model for Stock Price Prediction," *IEEE Access*, vol. 13, pp. 117594–117618, 2025, doi: 10.1109/ACCESS.2025.3586558.
- [20] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "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.*, vol. 38, no. 5, pp. 5311–5319, May 2011, doi: 10.1016/j.eswa.2010.10.027.
- [21] Z. D. Akşehir and E. Kılıç, "Analyzing the critical steps in deep learning-based stock forecasting: a literature review," *PeerJ Comput. Sci.*, vol. 10, 2024, doi: 10.7717/peerj-cs.2312.
- [22] X. Liu and W. Wang, "Improving Sliding Window Effect of LSTM in Stock Prediction Based on Econometrics Theory," *Comput. Econ.*, vol. 65, no. 4, pp. 2057–2080, Apr. 2025, doi: 10.1007/s10614-024-10627-z.

- [23] E. Alreshidi, "Forecasting of Saudi Stock Prices Using Statistical and Machine Learning Models: A Multi-Model Comparative Approach," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 11, 2025, doi: 10.14569/IJACSA.2025.0161114.
- [24] Yahoo, "Yahoo Finance." Accessed: Feb. 22, 2023. [Online]. Available: <https://finance.yahoo.com/>
- [25] H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," Dec. 01, 2023, *Springer Science and Business Media Deutschland GmbH*, doi: 10.1186/s40854-022-00441-7.
- [26] A. S. Saud and S. Shakya, "Technical indicator empowered intelligent strategies to predict stock trading signals," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 4, Dec. 2024, doi: 10.1016/j.joitmc.2024.100398.
- [27] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021.
- [28] A. Paccagnini, M. Bhasin, R. Mahadeva, A. K. Goyal, and V. Sarda, "Academic Editors: Massimo Guidolin Bio-Inspired Hyperparameter Optimization of LSTM Networks for Long-Horizon Stock Price Forecasting," 2026, doi: 10.3390/forecast8050084.
- [29] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PLoS One*, vol. 15, no. 1, Jan. 2020, doi: 10.1371/journal.pone.0227222.
- [30] A. Jadon, A. Patil, and S. Jadon, "A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting," Nov. 2022.