



Hybrid AI Models for Intelligent Investment Advisory System in the Saudi Stock Market

Master Thesis

Prepared By

Name: Meshal Manfi Alkhalidi

Academic ID: 251001992

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Artificial Intelligence

College of Informatics

Midocean University

United Arab Emirates

Supervisor: Dr. Khaled Eskaf

Submission Date: April 2026

Acknowledgements

Above all, I owe more than I can put into words to my mother, whose prayers have carried me through every step of this journey. To my uncle, thank you for pushing me to aim higher, both in my studies and in my career; your belief in me shaped a lot of the choices that led here. A special word of appreciation goes to my friend Aqeel, who has always spoken about my work with a pride that kept me going on the days I doubted myself, and to his family, who never stopped cheering me on. Last, but truly closest to my heart, my wife and my children thank you for bearing with the long nights, the missed weekends, and the quiet moments I stole for this thesis. This work belongs to all of you as much as it does to me.

Abstract

In this thesis, I look at how AI can support investment decisions in the Saudi Stock Market (Tadawul). My work centers on the ticker 1202.SR, and I test three predictive baselines against each other: logistic regression, LSTM, and a Transformer. All three use the same technical features I built from OHLCV data. To see if my findings carry beyond a single stock, I also extended the predictive comparison to two other tickers: 2320.SR and 4300.SR.

The results are more nuanced than a single "winner" narrative would suggest. Logistic regression got the highest accuracy at 0.5867, but its recall dropped to 0.0828 and F1 to just 0.1474 — meaning it missed almost every upward move. LSTM sat in the middle (accuracy 0.5604, F1 0.1896). The Transformer, though it did not top the accuracy table (0.5116), was clearly the most balanced model, with recall of 0.4551 and F1 of 0.4444. On the trading side, I aligned both strategies to the same evaluation window to make the comparison fair. My rule-based strategy returned +147.18% while buy-and-hold lost 47.19% — a much lower maximum drawdown as

well. Across the three tickers, the Transformer stayed the strongest balanced choice, while LSTM was inconsistent and logistic kept struggling with recall. What I take away from this work is not that one model "wins", but that model selection in this domain needs to balance predictive behavior with trading relevance rather than chasing accuracy alone. I see my contribution as laying a clean baseline for future hybrid extensions — where graph neural networks, reinforcement learning, and a proper advisory layer would eventually complete the picture.

Keywords: Saudi Stock Market, Tadawul, Deep Learning, Transformer, LSTM, Logistic Regression, Stock Movement Classification, Rule-Based Trading Strategy, Investment Advisory System.

Table of Contents

Contents

Acknowledgements.....	2
Abstract.....	2
Table of Contents.....	4
List of Tables	7
List of Figures	8
List of Abbreviations	8
Chapter 1: Introduction.....	10
1.1 Background.....	10
1.2 Problem Definition.....	10
1.3 Research Objectives	12
1.4 Research Questions	12
1.5 Significance of the Study.....	13
1.6 Thesis Contribution.....	13
1.7 Thesis Organization	14
Chapter 2: Background and Related Work.....	14
2.1 Artificial Intelligence in Financial Markets	14
2.2 Limitations of Classical Financial Models	15

2.3 Deep Learning for Time-Series Forecasting	15
2.4 Transformer-Based Temporal Modeling	16
2.5 Graph Neural Networks in Financial Markets	16
2.6 Reinforcement Learning for Trading Decisions	17
2.7 Human-Centered Advisory Systems.....	18
2.8 Research Gap Summary.....	18
2.9 Comparative Summary	19
Chapter 3: Proposed Approach and Methodology.....	19
3.1 Research Design.....	19
3.1.1 The Proposed Framework	21
3.2 Data Collection and Preprocessing	26
3.3 Predictive Modeling Layer.....	28
3.3.1 Problem Formulation and Class Definition	28
3.4 Fusion Strategy	32
3.5 Reinforcement Learning Environment.....	32
3.6 Advisory Engine	33
3.7 Evaluation Metrics	34
3.7.1 Predictive Metrics	34
3.7.2 Financial Metrics.....	36
3.7.3 On the Non-Applicability of Regression Error Metrics.....	37

3.7.4 Ablation Analysis	37
3.8 Ethical Considerations and Limitations	38
3.9 Proposed System Workflow	39
Chapter 4: Experimental Results and Discussion	39
4.1 Introduction.....	39
4.2 Experimental Setup	40
4.3 Baseline Models	40
4.3.1 Transformer Baseline Results	41
4.3.2 Logistic Regression Baseline Results	43
4.3.3 LSTM Baseline Results	45
4.4 Predictive Performance Analysis	47
4.4.1 Predictive Baseline Comparison	47
4.4.2 Multi-Stock Baseline Comparison	49
4.5 Trading Performance Analysis.....	50
4.5.1 Trading Baseline Comparison.....	51
4.6 Interpretability and Advisory Quality	53
4.6.1 Application-Level Validation Using Sahmk API	53
4.7 Discussion	55
4.8 Threats to Validity	56
4.9 Result Matrix.....	56

Chapter 5: Conclusion and Future Work	58
5.1 Conclusion	58
5.2 Main Contributions	59
5.3 Future Work	60
References.....	61

List of Tables

Table 3.1: Definition of the two movement classes and their advisory interpretation.

Table 3.2: Confusion matrix of the logistic regression baseline on the test partition of ticker 1202.SR.

Table 4.1: Transformer baseline configuration and test performance for ticker 1202.SR.

Table 4.2: Predictive baseline comparison for ticker 1202.SR.

Table 4.3: Multi-stock comparison across Logistic Regression, LSTM, and Transformer baselines for 1202.SR, 2320.SR, and 4300.SR.

Table 4.4: Trading baseline comparison between the Buy-and-Hold and Rule-Based strategies for ticker 1202.SR.

Table 4.5: Result matrix summarizing the main predictive and trading findings of the thesis.

List of Figures

Figure 3.1: Architecture of the proposed hybrid AI investment advisory framework for the Saudi stock market.

Figure 4.1: Transformer training and validation loss curves for ticker 1202.SR, with the best epoch indicated by early stopping.

Figure 4.2: Transformer evaluation metrics on the test set for ticker 1202.SR.

Figure 4.3: Transformer confusion matrix on the test set for ticker 1202.SR.

Figure 4.4: Logistic regression evaluation metrics on the test set for ticker 1202.SR.

Figure 4.5: Logistic regression confusion matrix on the test set for ticker 1202.SR.

Figure 4.6: LSTM training and validation loss curves for ticker 1202.SR, with the best epoch marked by early stopping.

Figure 4.7: LSTM evaluation metrics on the test set for ticker 1202.SR.

Figure 4.8: LSTM confusion matrix on the test set for ticker 1202.SR.

Figure 4.9: Equity curve comparison between the Buy-and-Hold and Rule-Based trading baselines for ticker 1202.SR.

Figure 4.10: Successful retrieval of live watchlist quotes through the developed backend integrated with the Sahmk API.

List of Abbreviations

- **AI:** Artificial Intelligence

- **Acc.:** Accuracy
- **DL:** Deep Learning
- **GNN:** Graph Neural Network
- **GRU:** Gated Recurrent Unit
- **LSTM:** Long Short-Term Memory
- **OHLCV:** Open, High, Low, Close, Volume
- **PPO:** Proximal Policy Optimization
- **RL:** Reinforcement Learning
- **RMSE:** Root Mean Square Error
- **TFT:** Temporal Fusion Transformer

Chapter 1: Introduction

1.1 Background

Financial markets today are noisy, tightly connected, and rarely stable for long. I find price forecasting genuinely hard for these reasons: the series are nonlinear, they carry a lot of noise, and their statistical behavior keeps changing whenever new events hit the market. Because of this, I noticed that recent research has moved away from betting on a single forecasting method. What I see more often now is the design of hybrid AI pipelines that combine preprocessing, temporal learning, optimization, and decision logic together — building a robustness that no single component can achieve alone (Jayanth et al., 2024; Jayanth & Manimaran, 2024; Sun et al., 2024). The Saudi Stock Market (Tadawul) fits into this bigger picture in an interesting way. Its strategic weight keeps growing, and I see clear sector-driven co-movement across banking, materials, energy, and telecommunications sectors — which makes cross-stock generalization more than just a theoretical worry for me. On top of that, there is real demand from local Saudi investors for decision-support tools that can pull multiple signals into one usable recommendation. Put these together, and Tadawul becomes a suitable place to study temporal forecasting and inter-stock dependence side by side, rather than as two separate problems.

Recent AI progress opens the door well beyond the classical forecasting toolkit. Deep learning models can extract temporal structure directly from historical market data. Transformer architectures reach across longer horizons and capture interactions among many variables at the same time. Graph Neural Networks (GNNs) give me a natural way to encode stock-to-stock relationships. And Reinforcement Learning (RL) provides the machinery I need to convert

predictive signals into a sequence of buy, sell, or hold actions, rather than isolated forecasts (Ansari et al., 2022; Ansari, 2024; Friday et al., 2025; Kabbani & Duman, 2022).

1.2 Problem Definition

One gap I keep noticing in this research area is that most AI-based financial studies were built on top of large international exchanges — not the Saudi market itself. Because of that, the assumptions these studies carry (about liquidity depth, sector generalization, and the broader microstructure of the market) do not always survive the move to Tadawul. I see another gap at the framing level too: much of the existing work places itself either in the pure forecasting camp or in the automated trading camp. Very few papers, in my reading, position themselves openly as interpretable advisory systems designed with a real end user in mind (Ansari et al., 2022; Kabbani & Duman, 2022). A second limitation has to do with how market structure gets represented inside the model. It is still common to treat every stock as a stand-alone time series. But in real life, stocks interact through sector membership, correlation patterns, and shifts in the wider market regime. I found recent graph-based work that makes this point clearly: adding a relational layer to the model can improve forecasting quality, precisely because it picks up interactions that independent per-stock models simply cannot see (Ansari, 2024). A third problem is more pragmatic and — I think — often gets less attention than it deserves: usability. Many systems in the literature emit raw numerical predictions or opaque policy signals that most investors have no straightforward way to interpret. In an actual investment setting, what people really need is closer to a short, explainable recommendation with a bit of risk context they can reason about. So for me, the challenge is not only to push predictive accuracy higher, but also to translate whatever the model produces into a human-centered advisory format that supports responsible decision-making rather than hiding it (Ansari et al., 2022).

1.3 Research Objectives

In this thesis, I set out to achieve the following objectives:

- First, I develop deep learning and Transformer-based forecasting models for a selected set of stocks listed on Tadawul.
- Second, I design a graph-based representation of the market and derive relational embeddings from it using Graph Neural Networks.
- Third, I build a Reinforcement Learning trading environment in which the hybrid predictive outputs are incorporated directly into the state representation.
- Fourth, I construct an advisory layer that translates the internal model outputs into concrete buy, sell, and hold recommendations, together with an indication of confidence and basic risk context.
- Finally, I assess the complete framework using both statistical accuracy measures and financial performance metrics.

1.4 Research Questions

- Do hybrid AI architectures deliver measurably better forecasting of Saudi stock movements than single-model baselines used on their own?
- When an RL agent is given hybrid signals as input, does the quality of its sequential trading decisions on Tadawul actually improve?
- What is a workable way to combine the outputs of several AI components into a single, interpretable recommendation that an investor can act on?
- What levels of profitability, and what degree of drawdown control, can the proposed framework realistically achieve when tested on out-of-sample data?

1.5 Significance of the Study

On the academic side, my study adds to the growing body of work on AI in finance by bringing together — inside one coherent framework — sequence modeling, Transformer-based forecasting, relational learning, and sequential decision-making. These are four method families that are usually studied on their own. I want to be clear: the novelty in my work is not in inventing any of these techniques from scratch, since each of them has been validated separately in earlier studies. What I do is combine them intentionally and adapt the combination to the specific dynamics of the Saudi market (Ansari et al., 2022; Ansari, 2024; Friday et al., 2025).

On the practical side, my thesis sketches a concrete path toward an intelligent advisory system aimed at both individual investors and institutions active in Tadawul. Instead of treating AI as a fully autonomous replacement for human judgment, I designed it as an analytical layer. This layer sharpens the investor's situational awareness, makes the trade-offs of any decision more visible, and communicates its output in a form that even a non-technical user can actually work with.

1.6 Thesis Contribution

A The main contributions I make in this thesis are:

- A hybrid AI framework I designed specifically for the Saudi stock market.
- A methodology I developed for coupling forecasting, relational learning, and policy learning within one pipeline.
- An interpretable advisory layer that I built to communicate the recommended action, its confidence, and the associated risk to the user.

- An evaluation setup I put together that explicitly ties machine learning performance metrics to financial outcomes.

1.7 Thesis Organization

I organized this thesis into five chapters. In Chapter 1, I set the stage by explaining the research context, defining the problem I am addressing, stating my objectives, and describing what I expect the work to contribute. Chapter 2 turns to the theoretical background, where I review the strand of literature most relevant to AI-driven financial forecasting and trading. In Chapter 3, I lay out my proposed hybrid framework from end to end, covering data preparation, the predictive models, the construction of the graph, the design of the RL component, and finally the advisory layer that sits on top of everything. Chapter 4 is where I discuss the evaluation design and interpret the kinds of results I need to examine in order to judge my framework fairly. Finally, Chapter 5 closes the thesis with my main conclusions and points to the directions along which I plan to extend the work in the future.

Chapter 2: Background and Related Work

2.1 Artificial Intelligence in Financial Markets

Artificial Intelligence has moved to the front of financial forecasting research largely because stock data resist the assumptions most classical methods rely on: the series are nonlinear, filled with noise, and shift in behavior as market conditions change. Reflecting this, the literature has been steadily leaning toward pipelines that blend statistical preprocessing, deep learning, and optimization together, instead of putting all the weight on any single family of models. Empirical work on hybrid forecasting supports this shift studies that attach a smoothing, decomposition, or

feature-expansion stage to a neural network tend to report performance that stand-alone baselines cannot match, which lines up with the broader move toward hybrid AI in financial applications (Jayanth et al., 2024; Jayanth & Manimaran, 2024; Li et al., 2024; Sun et al., 2024; Swetha & Arya, 2025).

2.2 Limitations of Classical Financial Models

Classical approaches such as ARIMA and GARCH have not lost their relevance: they remain valuable because they are transparent, easy to interpret, and still show up regularly in serious financial research as baselines. Their weakness, though, becomes visible when the data pull away from linear behavior, break into different regimes, or become dominated by noise. Hybrid volatility research that pairs a GARCH-type model with an LSTM makes this trade-off explicit: the statistical layer keeps contributing useful structure, but a neural component is usually needed on top of it to model the more intricate dynamics of financial series (Koo & Kim, 2022). For this reason, the present thesis positions classical methods as reference points and complementary tools rather than as the core advisory engine.

2.3 Deep Learning for Time-Series Forecasting

Deep learning architectures LSTM in particular, along with GRU and its bidirectional variants have become one of the standard choices for stock forecasting, mainly because they are well suited to picking up temporal patterns from rolling windows of price and volume data. What the prior literature adds to this is an important qualification: recurrent networks tend to reach their full potential only when they are paired with stronger feature engineering or with a more deliberate preprocessing stage. Concrete illustrations of this pattern include hybrid recurrent systems augmented with exponential smoothing and Bayesian optimization, decomposition-

based forecasting pipelines, and feature-expansion strategies that widen the input space before training even begins (Jayanth et al., 2024; Jayanth & Manimaran, 2024; Li et al., 2024; Swetha & Arya, 2025). These observations connect directly to the present study, since the first-term proposal already placed the forecasting layer on top of engineered market signals rather than on raw price series alone.

2.4 Transformer-Based Temporal Modeling

Transformer-based architectures have gained traction in this field for a simple reason: self-attention gives the model a way to capture long-range temporal dependencies and to represent complex interactions among variables without leaning entirely on recurrent connections. The reviewed literature adds a further nuance the Temporal Fusion Transformer, in particular, appears to be a strong fit for tasks that combine multi-horizon movement classification with a mix of heterogeneous feature sources. A representative example is the work of Friday et al. (2025), who paired a Vision Transformer with a Temporal Fusion Transformer to process both historical numerical data and candlestick-derived information, and reported solid classification results across a range of indices. In the present thesis, the Transformer component is kept in the pipeline as a dedicated forecasting stream; its role, however, is to reinforce the original proposal rather than to redirect the project into an image-centric study.

2.5 Graph Neural Networks in Financial Markets

Graph Neural Networks become particularly attractive at the point where treating each stock as an isolated time series stops being a reasonable simplification. Ansari (2024) demonstrated that clustering-driven, multi-relation stock graphs can encode stock-to-stock interactions in a way that a set of independent forecasters simply cannot, letting one graph model produce forecasts for

many stocks simultaneously while taking advantage of the shared structure between them. This line of reasoning maps well onto the Saudi market, where sector-based co-movement is a familiar phenomenon especially within the banking, materials, energy, and telecommunications segments of Tadawul. For that reason, the GNN component in this thesis is positioned as a relational layer that sits alongside the temporal forecasting stream and complements it, rather than as a mechanism intended to replace it.

2.6 Reinforcement Learning for Trading Decisions

Reinforcement Learning reframes trading as a sequential decision problem, in which an agent observes the state of the market, chooses an action, and receives a reward that reflects the consequences of that action over time. This framing becomes especially useful once the objective goes beyond simply predicting price direction and starts to include the timing question when to buy, when to sell, and when to hold under the pressure of portfolio constraints and transaction costs. Ansari et al. (2022) put forward a deep reinforcement learning-based decision-support system in which the outputs of a forecasting network are concatenated with historical trend information, giving the agent a richer view of the state. Along similar lines, Kabbani and Duman (2022) modeled trading as a partially observed Markov decision process and showed that deep reinforcement learning can bring prediction and allocation together inside a single decision framework. Together, these studies motivate keeping an RL decision layer inside the present system, while still preserving its overall character as an advisory tool rather than a fully autonomous trader.

2.7 Human-Centered Advisory Systems

While a large share of the published systems in this area is optimized either for higher forecasting accuracy or for the automated generation of profit, comparatively few of them are actually built around the idea of interpretable support for a real end user. The decision-support framing adopted by Ansari et al. (2022) is particularly instructive in this respect: it shows that predictive signals can be put to work in a way that supports the investor's decision without removing human oversight from the loop. The present thesis aligns itself more with that direction than with a fully automated execution model. The intent, accordingly, is to produce advisory outputs buy, sell, or hold recommendations, accompanied by a confidence level and a minimum of risk context in a form that can meaningfully assist users operating in the Saudi stock market.

2.8 Research Gap Summary

Read together, the studies reviewed in this chapter converge on four clear themes. The first is that hybrid deep learning models generally outperform simpler baselines once the financial environment becomes sufficiently noisy (Jayanth et al., 2024; Li et al., 2024; Sun et al., 2024). The second is that Transformer-based architectures add real value in settings where temporal dependency stretches over long horizons or where the input is multivariate in nature (Friday et al., 2025). The third is that graph-based methods sharpen the representation of the market whenever inter-stock relationships carry information that per-stock models cannot recover (Ansari, 2024). The fourth is that reinforcement learning becomes the natural fit once the problem is reframed as sequential decision-making rather than pure prediction (Ansari et al., 2022; Kabbani & Duman, 2022). What the surveyed studies rarely offer, however, is a Saudi-market perspective — most of them are developed on non-Saudi datasets, and they typically

address only one or two of these dimensions at a time. The gap that the present thesis sets out to close is precisely there: the design of a Saudi-market-oriented hybrid advisory framework that stays faithful to the first-term proposal while pulling these four ideas together in a controlled and interpretable way.

2.9 Comparative Summary

Taken as a whole, the picture emerging from the reviewed studies suggests that the most sensible direction for this thesis is not to swap out the original proposal for a completely different architecture, but rather to refine it in a targeted way. Sequence models remain a natural fit for short-term forecasting; Transformer models offer a stronger handle on temporal representation; GNNs bring in the structural view of the market that per-stock forecasters miss; and RL provides the mechanism for turning all of this into a coherent sequence of buy, sell, and hold actions (Ansari et al., 2022; Ansari, 2024; Friday et al., 2025). Combining them, therefore, extends the first-term proposal for Tadawul in a coherent way, rather than pulling the work away from it without justification.

Chapter 3: Proposed Approach and Methodology

3.1 Research Design

This study relies on a quantitative experimental design that builds directly on top of the first-term proposal rather than redrawing the boundaries of the project. The workflow starts with the preparation of historical Tadawul data, moves into feature engineering, and then advances through the development of a forecasting layer, the construction of a simple relational representation of the market, the training of a reinforcement learning decision module, and

finally the generation of the advisory recommendations that the system delivers to the user. This choice of design is in line with prior literature, which points to the practical value of hybrid forecasting pipelines and decision-support architectures; at the same time, the scope is deliberately kept narrow so that the thesis stays anchored to its original objective of serving the Saudi market.

This chapter presents the complete target architecture of the proposed advisory framework, and it is necessary to state at the outset how much of that architecture falls within the empirical scope of the present thesis. The framework is designed as a staged system, and the work reported here implements and evaluates its first stage: data acquisition, preprocessing, feature construction, the predictive modeling layer, and the trading and evaluation components that sit on top of it. The remaining stages — the graph-based relational module introduced in Section 3.3, the fusion strategy of Section 3.4, the reinforcement learning decision module of Section 3.5, and the user-facing advisory engine of Section 3.6 — are specified here at the design level so that the architecture is documented as a coherent whole, but they are not instantiated in this work and no empirical results are claimed for them. Figure 3.1 marks this boundary explicitly, and each affected section is flagged in the text where it occurs.

The decision to present the full architecture rather than only the implemented subset is deliberate. The contribution claimed by this thesis is twofold: a clean and reproducible empirical baseline for movement classification and rule-based trading on the Saudi market, and a complete architectural specification for the hybrid advisory system that the baseline is intended to support. A methodology chapter restricted to the implemented subset would serve the first contribution but not the second, and would leave the later stages undocumented. Chapter 5 returns to these components as the primary directions for future work.

3.1.1 The Proposed Framework

Figure 3.1 presents the architecture of the proposed framework as a single end-to-end pipeline. Reading the diagram from top to bottom, market data enters at the acquisition stage, passes through preprocessing and feature construction, and is then consumed in parallel by the components of the predictive modeling layer. The outputs of that layer, together with the relational representation produced by the graph module, are combined in the fusion stage into one hybrid market signal. That signal informs the trading and decision components, whose outputs the advisory engine finally translates into a recommendation delivered to the user. Evaluation closes the pipeline and is applied to out-of-sample data along the two axes set out in Section 3.7.

The diagram distinguishes deliberately between two categories of component. Blocks drawn with a solid outline denote parts of the framework that were implemented and evaluated empirically, and whose results are reported in Chapter 4: data acquisition, preprocessing, feature engineering, the three predictive baselines, the trading and backtesting stage, and the evaluation stage. Blocks drawn in grey denote parts that are fully specified at the design level in this chapter but were not instantiated within the scope of the present work, namely the graph neural network, the fusion layer, the reinforcement learning agent, and the advisory engine. This distinction is stated openly rather than left implicit, because the contribution claimed by this thesis is the establishment of a clean and reproducible empirical baseline for a Saudi-market advisory system, together with a complete architectural specification for the stages that build on it. The deferred components are revisited in Section 5.3 as the primary directions for future work.

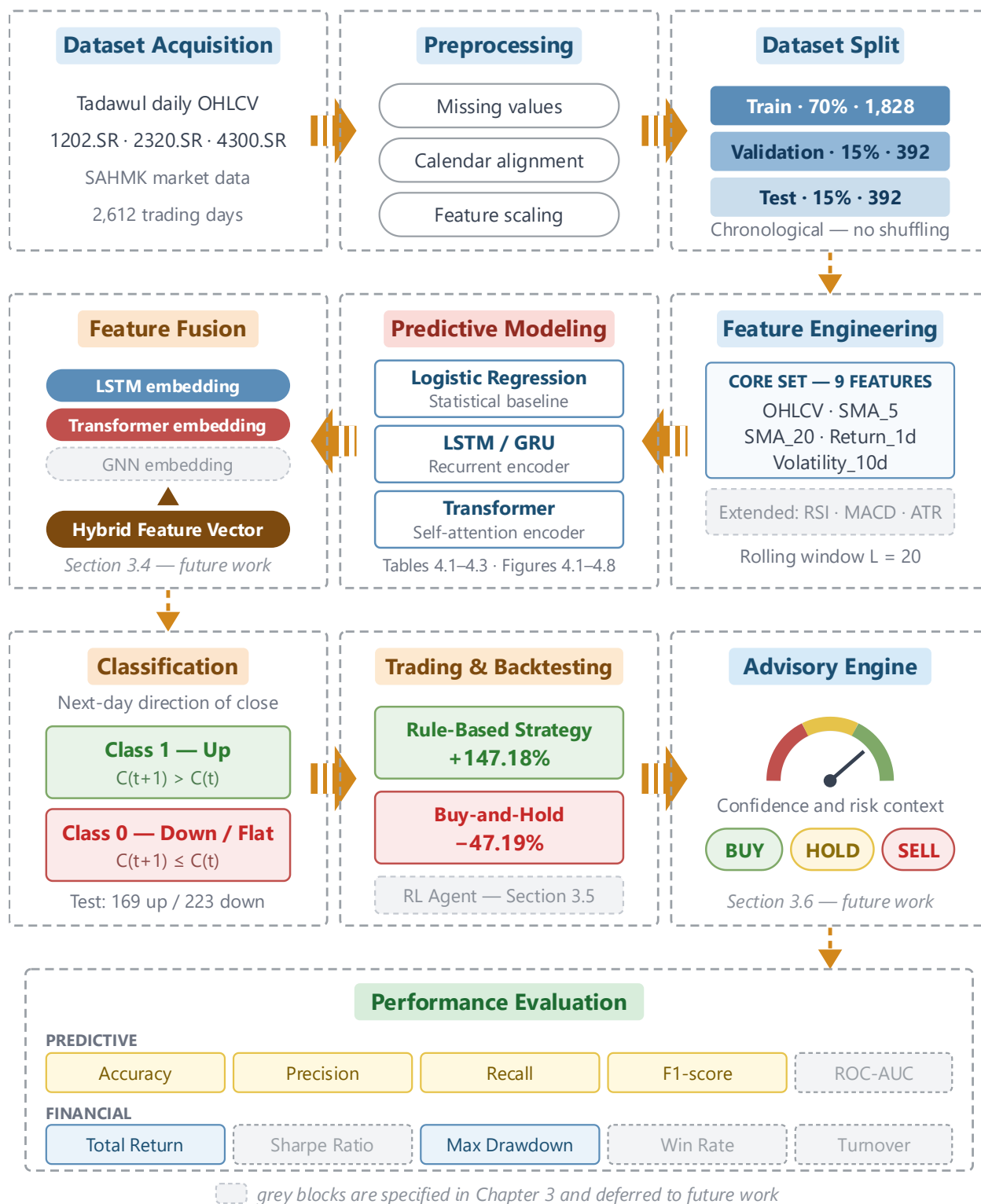


Figure 3.1: Architecture of the proposed hybrid AI investment advisory framework for the Saudi stock market.

3.1.1.1 Dataset Acquisition

The pipeline begins with the retrieval of historical daily OHLCV records for Tadawul-listed equities. Three tickers are selected — 1202.SR, 2320.SR and 4300.SR — deliberately drawn from different sectors, so that any conclusion about model behavior is not an artifact of a single industry. Ticker 1202.SR carries the detailed baseline experiments, and the other two serve as a generalization check. Daily frequency is chosen rather than intraday because it is the granularity at which a reliable and complete history is available for the Saudi market, and because an advisory system aimed at retail investors operates on a horizon of days rather than minutes.

3.1.1.2 Preprocessing

Raw market feeds are not directly usable. This stage handles missing observations, aligns the series to the Tadawul trading calendar so that non-trading days do not enter the sequence as artificial gaps, and scales each feature to a common range. Scaling parameters are estimated on the training partition alone and then applied unchanged to validation and test data. This restriction is not a formality: fitting a scaler on the full series would leak information about the future distribution into the training process and would inflate reported performance above what a live system could achieve.

3.1.1.3 Dataset Split

The sample is divided chronologically into training, validation and test partitions in a 70 / 15 / 15 proportion. The ordering is preserved strictly, and the data are never shuffled. Random partitioning is standard in many classification problems but is invalid here, because it would allow the model to learn from days that follow the ones it is asked to predict. The split is placed

before feature engineering and modeling in the diagram to make explicit that every subsequent stage respects this boundary.

3.1.1.4 Feature Engineering

Raw prices alone give a model very little to work with, so this stage constructs a set of derived indicators: short and medium moving averages, one-day returns, and rolling volatility, alongside the original OHLCV series. Nine features are used in the experiments reported in Chapter 4. A wider set of technical indicators — RSI, MACD, ATR and momentum — is specified in Section 3.2 and shown separately in the figure, because it belongs to the framework design but was not activated in the present experiments; keeping the two sets visually distinct prevents the figure from claiming more than the results support. Each example is finally shaped into a rolling window of the twenty most recent trading days.

3.1.1.5 Predictive Modeling Layer

Three models are trained on identical partitions and identical features, so that any difference between them reflects model capacity rather than an advantage in the data they receive. Logistic regression provides a transparent statistical reference point and consumes the window in flattened form. The recurrent stream, an LSTM or GRU encoder, captures short-range temporal dependence and represents the established baseline for financial sequence learning. The Transformer stream applies self-attention over the same window and is introduced to represent longer-range structure that a purely recurrent design compresses into a single hidden state. Logistic regression is evaluated standalone; the two deep encoders are the ones intended to feed the fusion stage.

3.1.1.6 Feature Fusion

The fusion stage concatenates the representations produced by the recurrent and attention-based encoders — and, in the full design, the relational embedding produced by the graph module — into a single hybrid feature space, which a learned fusion layer then reduces to one market signal. The purpose is to test whether the three components carry complementary rather than redundant information. A deliberately simple mechanism is specified, because the claim being made is about complementarity, not about the merits of any particular fusion architecture. This stage is specified but not implemented in the present work.

3.1.1.7 Classification

The classification head converts the learned representation into a decision on the direction of the next trading day's close, together with the probability attached to it. The task is binary throughout, as formalized in Section 3.3.1. The probability matters as much as the label, because the advisory engine downstream reports a confidence level rather than a bare instruction, and a poorly calibrated probability would make that confidence misleading.

3.1.1.8 Trading and Backtesting

A classification result is not yet an investment outcome, so this stage translates signals into positions and evaluates them on an out-of-sample equity curve. Two reference strategies are implemented: a rule-based technical strategy, representing the kind of approach many investors already apply, and a passive buy-and-hold benchmark, which is the most transparent point of comparison available in any financial evaluation. Their comparison is reported in Table 4.4 and Figure 4.9. The reinforcement learning agent specified in Section 3.5 belongs at this same point in the pipeline, as an alternative decision policy that would be measured against these two references rather than as a separate stage; it is not implemented here.

3.1.1.9 Advisory Engine

The advisory engine is the component that keeps the framework anchored to a practical application. It converts internal model outputs into an actionable recommendation — buy, sell or hold — accompanied by a confidence score and a readable indication of the risk involved. Where the available signals agree, confidence is raised; where they disagree, the engine falls back on a hold decision or issues advice at reduced confidence. This conservative default reflects the framework's positioning as a decision-support tool rather than an autonomous trading system. This stage is specified but not implemented in the present work.

3.1.1.10 Performance Evaluation

Evaluation closes the pipeline and is deliberately split along two axes. Predictive metrics assess classification quality, and financial metrics assess whether acting on those classifications produces sound portfolio behavior. Both are necessary, because a model can classify acceptably yet trade poorly once transaction costs are applied, and a strategy can return well while resting on classification barely better than chance. The full metric definitions are given in Section 3.7. All evaluation is performed on the held-out test partition, which no stage of training or tuning has seen.

3.2 Data Collection and Preprocessing

The primary dataset consists of historical daily OHLCV records for a selected set of equities listed on the Saudi Exchange (Tadawul). The historical daily OHLCV records were retrieved through the SAHMK market data service (SAHMK, 2026), which serves as the primary data source for the Saudi Exchange (Tadawul) equities used in this study and provides the .SR-suffixed tickers of interest. The retrieved series were cross-checked against publicly accessible

Yahoo Finance data (Yahoo Finance, 2026) through the open-source yfinance Python interface (Aroussi, 2024) wherever overlapping coverage was available. Three tickers were selected so that the evaluation would span distinct sectors rather than concentrate on a single industry: 1202.SR (Middle East Paper Company, Materials), 2320.SR (Al Babbtain Power and Telecommunication Company, Capital Goods), and 4300.SR (Dar Al Arkan Real Estate Development Company, Real Estate). Ticker 1202.SR serves as the primary instrument for the detailed baseline experiments, while the remaining two are used to test whether the observed behavior generalizes beyond a single company. Company identity and sector classification follow the official listings published by the Saudi Exchange (Saudi Exchange, 2026). For the live advisory layer, quotes are additionally served by the SAHMK market data API (SAHMK, 2026); this source supports the deployed application and is not used to produce the historical results reported in Chapter 4.

After alignment to the Tadawul trading calendar and removal of incomplete records, the working sample for ticker 1202.SR comprises 2,612 daily observations, partitioned chronologically into 1,828 training, 392 validation and 392 test rows. Applying a rolling window of length 20 to the same sample yields 2,592 supervised sequences for the recurrent and Transformer streams, partitioned as 1,814 training, 389 validation and 389 test sequences. Preprocessing handles missing observations, resolves the data-alignment issues that arise in raw market feeds, applies per-feature scaling fitted on the training partition only, and constructs the rolling windows required for sequential modeling. Scaling parameters are never estimated on validation or test data, so that no distributional information leaks backwards across the split boundaries. This pipeline lines up with the practice adopted in recent forecasting studies, which routinely take

structured market histories as the base input for recurrent, Transformer, and hybrid models (Jayanth et al., 2024; Li et al., 2024; Swetha & Arya, 2025).

Feature engineering is intentionally kept practical and close to what was originally planned in the first-term proposal. The framework brings in a set of technical indicators moving averages, RSI, MACD, ATR, momentum, rolling volatility, and volume-derived measures that together give the models a richer view of the market than raw prices alone would allow. This choice is backed by recent work showing that stronger technical feature sets support model learning and that hybrid systems tend to benefit from a preprocessing or feature-expansion stage placed before the forecasting layer (Moody et al., 2024; Swetha & Arya, 2025). Additional data sources such as market sentiment can be considered in later extensions of the project, but they are not treated as central to the present thesis, in order to avoid drifting too far from the original scope of the work.

3.3 Predictive Modeling Layer

3.3.1 Problem Formulation and Class Definition

The forecasting task is formulated throughout this thesis as **binary classification of next-day directional movement**, not as regression on price level. Making this explicit matters, because the choice determines which evaluation metrics are admissible and how the resulting advisory signal should be interpreted.

Let C_t denote the adjusted closing price of a given stock on trading day t . The target variable associated with day t is defined as

$$y_t = 1 \quad \text{if } (C_{t+1} - C_t) / C_t > \tau \quad \text{and} \quad y_t = 0 \quad \text{otherwise,}$$

where τ is a threshold on the one-day forward return. In the experiments reported in Chapter 4, τ is set to zero, so the two classes carry the following meaning:

Class	Label	Definition	Advisory interpretation
1	Up movement	$C(t+1) > C(t)$ — the close on the following trading day exceeds the close on day t	Positive directional signal; supports a Buy or Hold recommendation
0	Down or flat movement	$C(t+1) \leq C(t)$ — the close on the following trading day is lower than or equal to the close on day t	Negative or neutral signal; supports a Sell or Hold recommendation

Table 3.1: Definition of the two movement classes and their advisory interpretation.

Two consequences of this definition should be noted. First, days on which the price closes unchanged are absorbed into class 0. This is a conservative choice: an unchanged close carries no upward evidence, and assigning it to the positive class would inflate the apparent frequency of profitable opportunities. Second, the target is strictly forward-looking and is computed from information dated $t+1$, while every predictor supplied to the model is dated t or earlier. Combined with the chronological partitioning described in Section 4.2, this ordering removes look-ahead bias from the labeling procedure.

The model receives as input a rolling window of the $L = 20$ most recent trading days, so that each supervised example takes the form of a matrix $\mathbf{X}_t \in \mathbb{R}^{20 \times d}$, where d is the number of engineered features, paired with the scalar label y_t . For the baseline experiments $d = 9$, corresponding to Open, High, Low, Close, Volume, SMA_5, SMA_20, Return_1d and Volatility_10d. The logistic regression baseline consumes the same window in flattened form, ensuring that all three

baselines are exposed to identical information and that the comparison between them isolates model capacity rather than input advantage.

Class distribution

The two classes are not equally represented, and the imbalance is material to how the results in Chapter 4 should be read. Because every model is evaluated on the held-out test partition, it is the composition of that partition that bears on the interpretation of the reported figures. Of its 392 observations, 169 (43.1 per cent) belong to class 1 and 223 (56.9 per cent) to class 0, so a classifier that predicted the majority class on every day would already attain an accuracy of 0.569 without discriminating between the two classes at all.

The imbalance explains the pattern observed among the baselines in Chapter 4. A classifier that predicts the majority class almost everywhere attains an accuracy close to the majority share while achieving very low recall on the minority class, and this is precisely the behavior exhibited by the logistic regression baseline. Its reported test performance — an accuracy of 0.5867, a precision of 0.6667 and a recall of 0.0828 — corresponds to the confusion matrix shown in Table 3.2.

Label	Predicted Up (1)	Predicted Down / Flat (0)
Actual Up (1)	TP = 14	FN = 155
Actual Down / Flat (0)	FP = 7	TN = 216

Table 3.2: Confusion matrix of the logistic regression baseline on the test partition of ticker 1202.SR.

The model issued only 21 positive predictions across 392 test days, of which 14 were correct. Its high precision therefore reflects extreme selectivity rather than skill: it identified fewer than one upward movement in twelve. The Transformer baseline reports a lower accuracy of 0.5116 but a

substantially higher recall of 0.4551, indicating that it discriminates between the classes rather than defaulting to the majority. It follows that accuracy alone is not an adequate criterion for model selection in this setting, and that F1 together with recall on the positive class carries the greater diagnostic weight. This point is developed further in Section 3.7.

The predictive modeling layer is organized around two complementary streams that address different aspects of the same forecasting problem. The first stream relies on a recurrent architecture LSTM or GRU for the simple reason that these models continue to serve as strong baselines for financial sequence learning and are firmly backed by the reviewed literature. The second stream introduces a Transformer-based temporal model, which is added to improve the representation of longer-range dependencies and to capture multivariate interactions more effectively than a purely recurrent design can. The rationale is grounded in the general track record of recurrent-hybrid forecasting models, as well as in the more recent work applying the Temporal Fusion Transformer to stock movement classification (Friday et al., 2025; Jayanth et al., 2024; Li et al., 2024). It should be noted that the model implemented and evaluated in Chapter 4 is a standard Transformer encoder with self-attention over a fixed input window, rather than the full Temporal Fusion Transformer architecture. The latter is cited here as the motivating precedent for introducing attention-based temporal modeling into this problem domain; adopting its complete design, including the variable selection and gating mechanisms, is left to future work. It is worth stressing, though, that the Transformer stream is treated here as a temporal enhancement layered on top of the original proposal, rather than as a signal that the project is being redirected toward a fully different multimodal design.

To bring market structure into the model without inflating the complexity of the implementation, a stock graph is constructed in which the nodes stand for the selected Saudi stocks and the edges

reflect simple but meaningful relationships sector membership, for instance, or a sustained return correlation between two stocks. A Graph Neural Network is then trained on this graph in order to produce relational embeddings that capture the interactions among the nodes. The justification for this choice is the accumulating evidence that graph-based modeling can improve stock forecasting by capturing precisely those interactions that independent time-series models systematically overlook (Ansari, 2024). The graph layer, then, enters the pipeline as a supportive relational module whose role is to enrich the temporal forecast rather than to replace it.

3.4 Fusion Strategy

At the fusion stage, the outputs coming out of the recurrent stream, the Transformer stream, and the graph-based representation are brought together into a single hybrid feature space. For the purposes of this thesis, a straightforward concatenation followed by a learned fusion layer is enough the aim here is to demonstrate that the three components carry complementary information, not to introduce a fusion mechanism whose complexity would exceed the requirements of the problem. This choice keeps the design in line with the central logic of the first-term proposal: multiple AI components should be able to contribute jointly to a stronger advisory signal for the Saudi market

3.5 Reinforcement Learning Environment

A custom trading environment is set up to simulate sequential decision-making on top of historical Tadawul data. At every time step, the agent takes in a view of the recent market features, together with the hybrid predictive signal coming from the earlier layers and a small amount of portfolio state information — chiefly the current cash balance and the position that is presently held. The action space is intentionally limited to buy, sell, or hold, which keeps the

advisory logic transparent and aligned with the intended end-user application. This overall setup is consistent with the wider body of decision-support and trading-automation work, which shows how forecasting outputs can be folded directly into the RL state representation to give the agent a more informative view of the environment (Ansari et al., 2022; Kabbani & Duman, 2022).

The reward function is designed with the intention of encouraging profitable behavior that remains realistic under actual market conditions. It is built around the realized return of the agent, but with explicit penalties for transaction costs, excessive turnover, and drawdowns of an unacceptable magnitude. This care in the design matters, because the reinforcement learning literature on trading has repeatedly shown that a reward function that ignores these frictions tends to produce policies that are either unstable in practice or plainly misleading (Ansari et al., 2022; Kabbani & Duman, 2022). By keeping these penalties inside the reward, the thesis anchors the RL component as a decision-support mechanism, rather than allowing it to drift into a purely theoretical trading exercise.

3.6 Advisory Engine

The advisory engine sits at the user-facing end of the framework, and it is the component that keeps the whole thesis anchored to a practical application. Its role is to convert the internal model outputs into an actionable suggestion buy, sell, or hold accompanied by a confidence score and a simple, readable indication of the risk involved. When the recurrent model, the Transformer stream, the graph context, and the RL policy all point in the same direction, the engine can raise the confidence attached to the recommendation; when the signals disagree, the safer default is to fall back on a hold decision or to issue advice at a lower confidence level. This

design keeps the project firmly aligned with its original ambition of delivering an intelligent investment advisory system for the Saudi stock market.

3.7 Evaluation Metrics

The framework is evaluated along two complementary axes. The predictive axis measures how well the models classify next-day directional movement as defined in Section 3.3.1. The financial axis measures whether acting on those classifications produces sound portfolio behavior. Both are necessary: a model may classify well yet trade poorly once transaction costs are applied, and a strategy may perform acceptably in return terms while resting on classification that is barely better than chance. Splitting the evaluation in this way stays consistent with the practice of the reviewed literature, in which forecasting studies typically report statistical error measures while trading studies report portfolio-level metrics (Ansari et al., 2022; Jayanth et al., 2024; Kabbani & Duman, 2022).

3.7.1 Predictive Metrics

All predictive metrics are computed from the confusion matrix of the binary task, in which TP and TN denote correctly classified up and down movements, and FP and FN denote the corresponding errors.

Accuracy is the proportion of correctly classified trading days:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Precision is the proportion of predicted up movements that were in fact up movements, and therefore measures the reliability of a Buy signal:

$$\text{Precision} = TP / (TP + FP)$$

Recall, or sensitivity, is the proportion of actual up movements that the model succeeded in identifying, and therefore measures how much of the available opportunity is captured:

$$\text{Recall} = TP / (TP + FN)$$

F1 score is the harmonic mean of precision and recall, and is the primary criterion for model selection in this thesis because it penalizes the degenerate majority-class behavior that accuracy rewards:

$$F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall})$$

Given the class imbalance documented in Table 3.1, the **macro-averaged F1** is also reported, computed as the unweighted mean of the per-class F1 scores so that both classes contribute equally:

$$\text{Macro-F1} = (F1_{\text{class } 0} + F1_{\text{class } 1}) / 2$$

ROC-AUC summarizes the trade-off between true positive rate and false positive rate across all decision thresholds, and is threshold-independent, which makes it useful when the operating point of the advisory engine has not yet been fixed.

Finally, because the advisory engine of Section 3.6 emits a confidence score rather than a bare label, the calibration of the predicted probabilities is assessed using the **Brier score**, the mean squared error between the predicted probability p_i and the observed outcome y_i over N test examples:

$$\text{Brier} = (1 / N) \sum_{i=1}^N (p_i - y_i)^2$$

A lower Brier score indicates better-calibrated confidence, which matters directly for a system whose recommendations are accompanied by a stated confidence level.

3.7.2 Financial Metrics

The financial measures adopted here are those routinely reported in the reinforcement learning and decision-support trading literature, where portfolio-level outcomes rather than statistical error are the quantities of interest (Ansari et al., 2022; Kabbani & Duman, 2022). Let V_t denote portfolio value at time t over a horizon of T trading days, and r_t the corresponding daily portfolio return.

Cumulative return measures total growth of capital over the evaluation period:

$$R_T = (V_T - V_0) / V_0$$

Sharpe ratio measures excess return per unit of volatility, annualised over 252 trading days, where r_f is the risk-free rate:

$$\text{Sharpe} = [(r - r_f) / \sigma_r] \cdot \sqrt{252}$$

Maximum drawdown is the largest peak-to-trough decline in portfolio value observed over the period, and captures worst-case exposure rather than average behavior:

$$\text{MDD} = \max_{t \in [0, T]} [(\max_{s \leq t} V_s - V_t) / \max_{s \leq t} V_s]$$

Win rate is the proportion of closed trades that ended in profit:

$$\text{Win rate} = N_{\text{profitable trades}} / N_{\text{total trades}}$$

Turnover measures trading intensity as the mean absolute change in portfolio weight w_t , and is reported because excessive turnover erodes realized return once transaction costs are applied:

$$\text{Turnover} = (1 / T) \sum_{t=1}^T |w_t - w_{t-1}|$$

3.7.3 On the Non-Applicability of Regression Error Metrics

The task addressed in this thesis is defined in Section 3.3.1 as binary classification of next-day directional movement. The models therefore emit a class label together with an associated probability, and never a predicted price level. Regression error measures such as RMSE, MAE and MAPE are defined on the difference between a predicted and an observed continuous quantity, and consequently have no meaningful interpretation when applied to a binary label: the numerical difference between a predicted class and a true class does not correspond to any economic quantity, and an RMSE computed over class indicators is simply a monotone transformation of the error rate, conveying nothing that accuracy does not already convey. These metrics are accordingly excluded from the evaluation protocol.

The distinction is retained here because it is not merely terminological. Should the framework be reformulated as a value-forecasting problem — predicting the closing price or the magnitude of the next-day return rather than its sign — then RMSE, MAE and MAPE would become the appropriate error measures, and the classification metrics defined above would in turn cease to apply. Such a reformulation would also change the design of the advisory engine, since a predicted magnitude permits position sizing whereas a predicted direction does not. This is identified in Section 5.3 as a possible extension, but it lies outside the scope of the present work, and the evaluation reported in Chapter 4 is confined to the classification metrics of Section 3.7.1 and the financial metrics of Section 3.7.2.

3.7.4 Ablation Analysis

Beyond these headline metrics, the study relies on ablation analysis to check whether each component of the pipeline earns its place. The comparisons that are most informative include recurrent-only forecasting, recurrent-plus-Transformer forecasting, recurrent-plus-graph

forecasting, and the full hybrid framework with the reinforcement learning layer on top. Each configuration is evaluated using the identical metric set and the identical chronological partitions, so that any difference is attributable to the component under test rather than to the evaluation procedure. This step matters because the thesis does not claim that additional components improve results by default; the point is to establish whether each added layer contributes a measurable benefit sufficient to justify its inclusion in a Saudi-market advisory system.

3.8 Ethical Considerations and Limitations

The proposed framework is deliberately positioned as a decision-support tool, not as a promise of guaranteed investment profit. Any move toward practical deployment would therefore need to be accompanied by clear communication about the uncertainty embedded in the system, the limitations of the underlying models, and the continuing role of human judgment in the final decision. This point carries particular weight in financial applications, where users have a well-documented tendency to over-trust algorithmic outputs when those outputs are presented without adequate context (Ansari et al., 2022).

From a methodological standpoint, the study also has to guard against several familiar pitfalls: survivorship bias, look-ahead bias, and overfitting to the training window. Alongside these, it has to acknowledge openly that model performance can deteriorate during structural breaks or during unusual market episodes that are underrepresented in the training data. Further limitations worth flagging include the constrained availability of high-quality Saudi market data, the well-known sensitivity of RL training to hyperparameter choices, and the more subtle challenge of translating

a complex internal model behavior into advisory outputs that stay both simple to read and genuinely informative.

3.9 Proposed System Workflow

- Collect the historical Tadawul market data and clean it into a consistent working dataset.
- Derive the technical indicators and the market-structure features from that dataset.
- Train the sequence-based models alongside the Transformer-based forecasting models.
- Build a stock relationship graph and learn the corresponding GNN embeddings from it.
- Fuse the different predictive signals into a single hybrid market signal.
- Feed this hybrid signal into a custom RL trading environment as part of the agent's state.
- Translate the final outputs of the pipeline into buy, sell, or hold recommendations, each carrying a confidence level and a basic risk context.
- Evaluate the framework on out-of-sample data along both the predictive and the financial dimensions.

Chapter 4: Experimental Results and Discussion

4.1 Introduction

This chapter sets out how the proposed framework should be examined empirically and, more importantly, how the evidence produced by those experiments ought to be read. Because the thesis brings forecasting, relational learning, and sequential decision-making under the same roof, no single metric is expected to settle the question of whether the framework is working well. Any meaningful discussion of the results therefore has to bring model quality, trading

behavior, and advisory usefulness into one coherent line of analysis rather than treating them as three unrelated stories.

4.2 Experimental Setup

The experimental design rests on chronological train, validation, and test partitions, chosen specifically to preserve the temporal ordering that is intrinsic to financial data. The models are first calibrated on the historical training window, then tuned on the validation segment, and finally evaluated on a test period that they have never seen during any stage of learning. This ordering is not a formality: any attempt to shuffle the data at random would inject information leakage into the pipeline and would push the reported performance well above what could realistically be expected in a live setting.

4.3 Baseline Models

A meaningful comparison requires a well-chosen set of baselines, and in this study the natural candidates fall into four groups. The first is a plain buy-and-hold strategy, which serves as the most transparent point of reference in any financial evaluation. The second is a rule-based technical trading system, which represents the kind of approach that many investors already apply in practice. The third comprises classical forecasting methods, which anchor the study within the established statistical tradition. The fourth consists of reduced versions of the proposed framework itself, obtained by disabling one component at a time. Taken together, these baselines make it possible to determine whether the full hybrid architecture is contributing something of its own, or whether it is largely reproducing results that simpler methods can already deliver at a lower cost.

4.3.1 Transformer Baseline Results

The Transformer baseline was evaluated on the ticker 1202.SR using a sequence length of 20 and a fixed random seed of 42. The dataset was split into 1,814 training sequences, 389 validation sequences, and 389 test sequences, giving a total of 2,592 sequences to work with. Under this configuration, the model reached its best validation performance at epoch 2, and on the held-out test set it produced an accuracy of 0.5116, a precision of 0.4343, a recall of 0.4551, and an F1 score of 0.4444.

Read together, these results suggest that the baseline Transformer did manage to capture at least some of the temporal structure embedded in the input, but the overall classification quality remained modest. The fact that precision, recall, and F1 all lie within a narrow band of one another points to performance that is balanced yet limited in absolute terms. On that basis, the experiment is best treated as a benchmark reference against which more advanced or better-tuned models can be compared, rather than as a standalone decision-support model in its own right.

The main configuration settings and the evaluation outcomes of this Transformer baseline experiment are summarized in the table below.

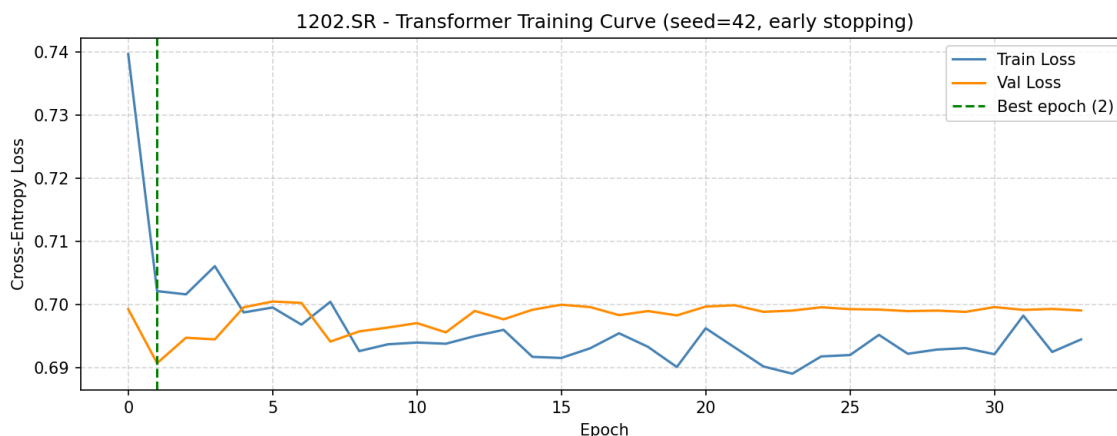


Figure 4.1: Transformer training and validation loss curves for ticker 1202.SR, with the best epoch indicated by early stopping.

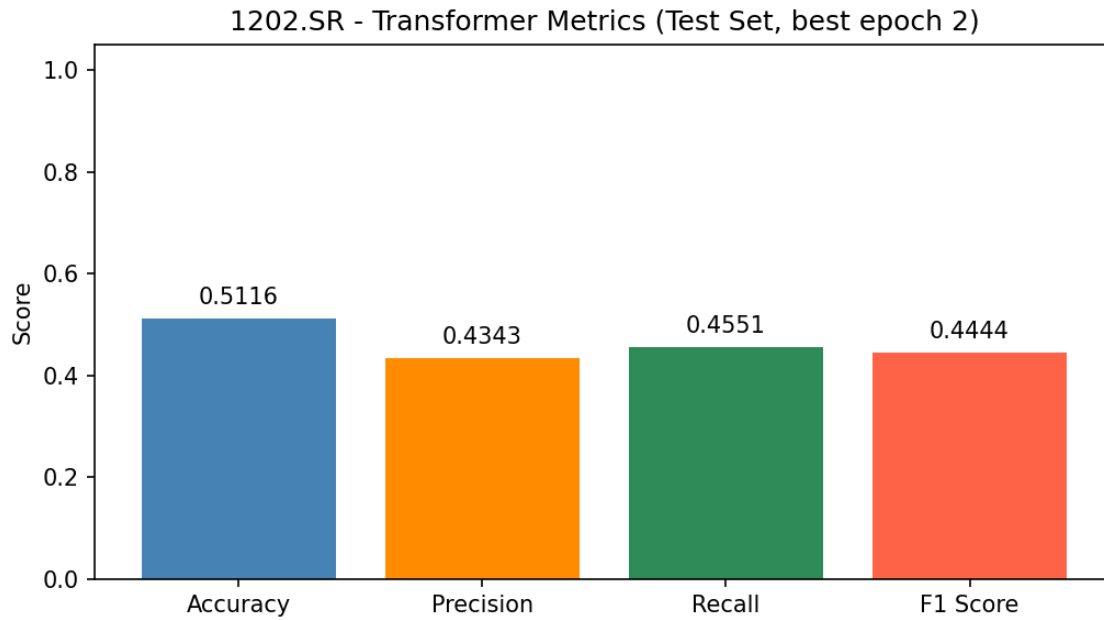


Figure 4.2: Transformer evaluation metrics on the test set for ticker 1202.SR.

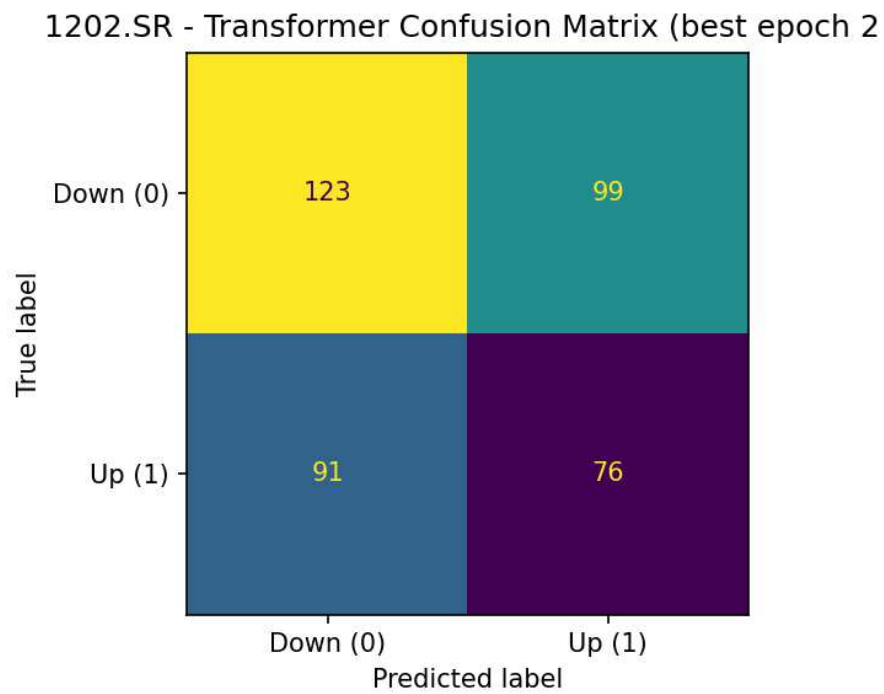


Figure 4.3: Transformer confusion matrix on the test set for ticker 1202.SR.

Table 4.1: Transformer baseline configuration and test performance for ticker 1202.SR.

Parameter / Metric	Value
Ticker	1202.SR
Sequence Length	20
Random Seed	42
Best Epoch	2
Train Sequences	1814
Validation Sequences	389
Test Sequences	389
Total Sequences	2592
Features Used	Open, High, Low, Close, Volume, SMA_5, SMA_20, Return_1d, Volatility_10d
Accuracy	0.5116
Precision	0.4343
Recall	0.4551
F1 Score	0.4444

A terminal execution screenshot is retained in the appendix as supplementary evidence of the experimental runtime and output generation process.

4.3.2 Logistic Regression Baseline Results

The logistic regression baseline was evaluated on the ticker 1202.SR using 1,828 training rows, 392 validation rows, and 392 test rows. The feature set fed into the model brought together

Open, High, Low, Close, Volume, SMA_5, SMA_20, Return_1d, and Volatility_10d. Under this configuration, the model reached an accuracy of 0.5867, a precision of 0.6667, a recall of 0.0828, and an F1 score of 0.1474.

Although the logistic baseline came out on top of the current set of predictive baselines in both accuracy and precision, its recall dropped to a level that tells a different story: the model missed a substantial share of the positive market-movement cases it was supposed to detect. In practical terms, it is useful as a simple statistical reference — a familiar yardstick against which the other models can be measured — but the imbalance between its precision and its recall means that, on its own, it is not adequate as a classifier for a real advisory system.

Figures 4.4 and 4.5 present the logistic regression metrics and the confusion matrix obtained for ticker 1202.SR

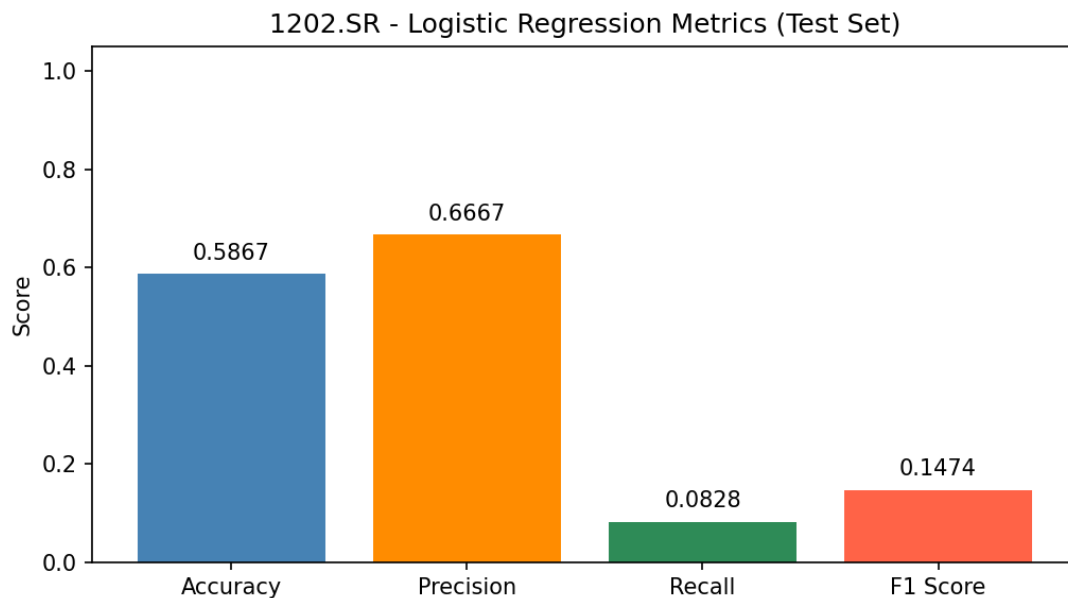


Figure 4.4: Logistic regression evaluation metrics on the test set for ticker 1202.SR.

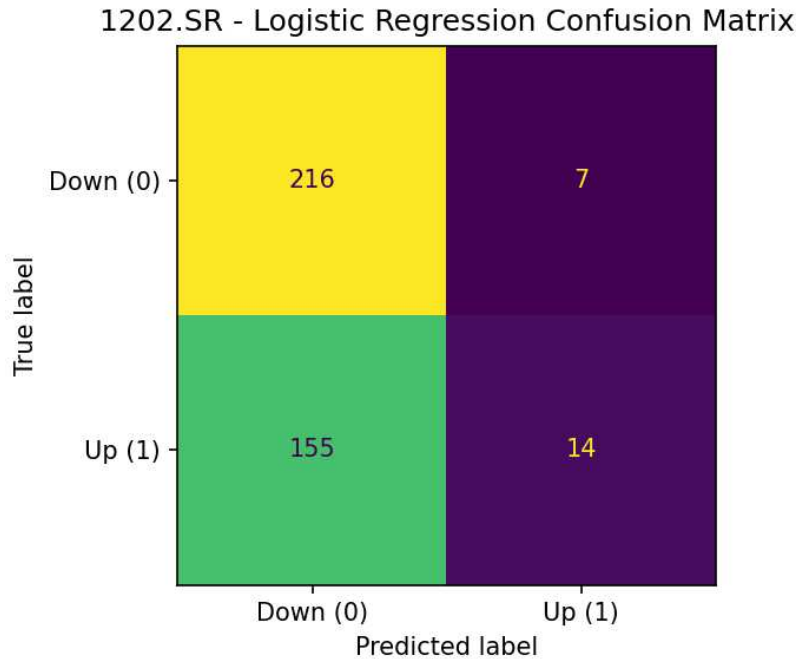


Figure 4.5: Logistic regression confusion matrix on the test set for ticker 1202.SR.

4.3.3 LSTM Baseline Results

The LSTM baseline was evaluated on the ticker 1202.SR using a sequence length of 20 and a random seed of 42. The dataset was divided into 1,814 training sequences, 389 validation sequences, and 389 test sequences, with the best validation performance being recorded at epoch 4. On the held-out test set, the model produced an accuracy of 0.5604, a precision of 0.4545, a recall of 0.1198, and an F1 score of 0.1896.

Taken together, these figures indicate that the LSTM baseline delivered a sequential benchmark that captures the temporal structure of the data slightly more effectively than the logistic model, yet both its recall and its F1 score stayed at a fairly limited level. For that reason, the LSTM baseline is best read as an intermediate sequence-learning benchmark, positioned between the plain logistic reference and the more expressive Transformer stream, rather than as a decision-support model that could stand on its own.

Figures 4.6, 4.7, and 4.8 show, respectively, the LSTM training curve, the evaluation metrics, and the confusion matrix obtained for ticker 1202.SR

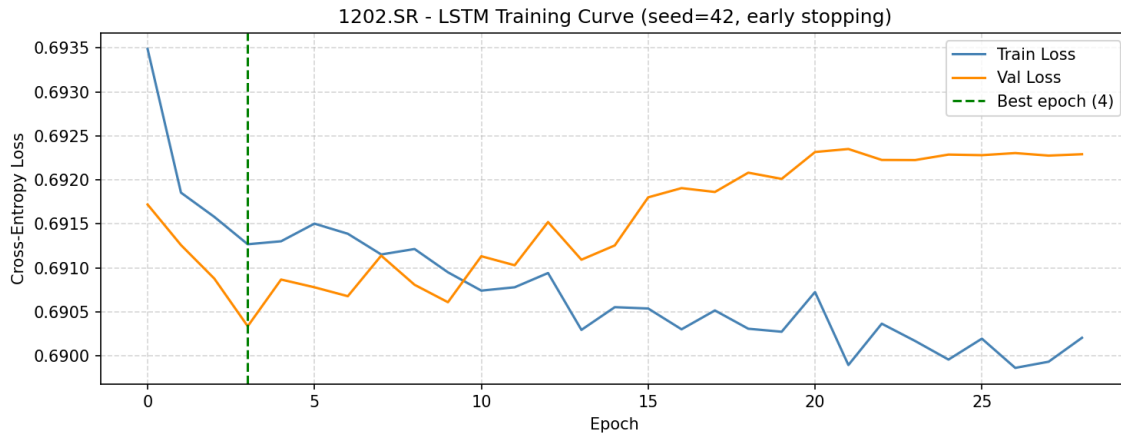


Figure 4.6: LSTM training and validation loss curves for ticker 1202.SR, with the best epoch marked by early stopping.

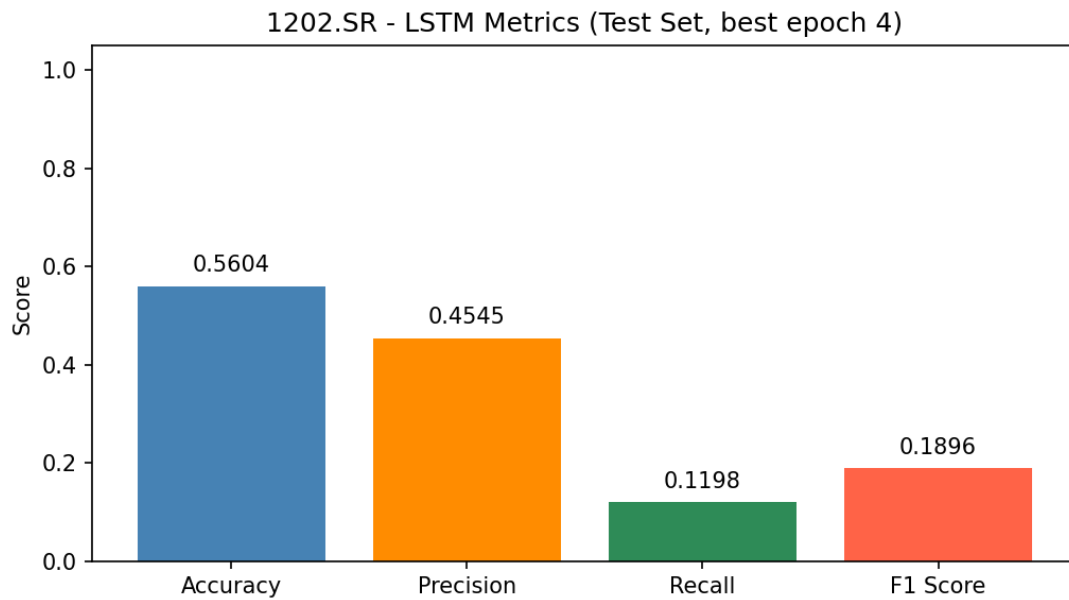


Figure 4.7: LSTM evaluation metrics on the test set for ticker 1202.SR.

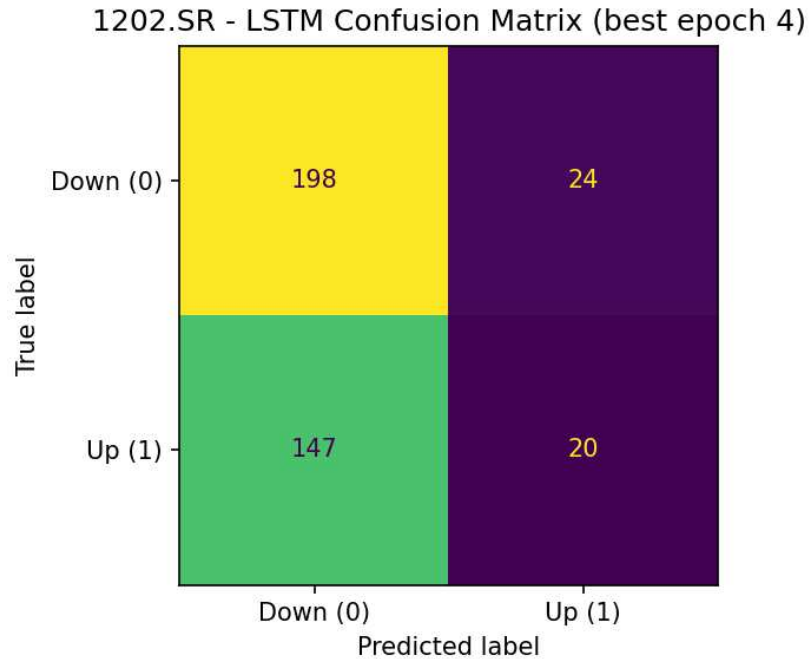


Figure 4.8: LSTM confusion matrix on the test set for ticker 1202.SR.

4.4 Predictive Performance Analysis

The predictive analysis should begin by placing the hybrid model side by side with each of its individual components. Any measurable gain in directional classification, in the calibration of the predicted probabilities, or in the raw forecast error would support the view that the combined architecture is genuinely pulling more information out of the market state than any single component could on its own. That said, the discussion has to remain cautious about what such gains actually mean: a modest improvement in a predictive score only carries real weight if it holds up consistently across different test periods, rather than being concentrated in one narrow market regime that happens to favor the model.

4.4.1 Predictive Baseline Comparison

The predictive performance of the three main baseline models on ticker 1202.SR is summarized below. Looking at the numbers, the logistic baseline came out ahead in accuracy at 0.5867 and in

precision at 0.6667, yet its recall collapsed to just 0.0828 — a signal that the model was largely unable to detect positive market-movement cases when they actually occurred. The LSTM baseline improved on the temporal side of the problem but still ended up with a fairly limited F1 score of 0.1896. The Transformer baseline, by contrast, produced the best trade-off between recall (0.4551) and F1 score (0.4444), which makes it the most balanced predictive baseline of the three under the current experimental setup.

From an advisory perspective, the comparative evidence delivers a clear methodological lesson: accuracy on its own would be a misleading criterion for selecting a model. The Transformer baseline did not deliver the highest headline accuracy, but it did offer the strongest balance between sensitivity and overall classification quality — and it is precisely that balance which matters when the goal is to build a downstream decision-support system rather than to maximize a single scalar. On that basis, the Transformer baseline emerges as the more suitable temporal reference for later integration into a broader hybrid architecture.

The corresponding logistic, LSTM, and Transformer figures visually support the comparative reading offered in this subsection.

Table 4.2: Predictive baseline comparison for ticker 1202.SR.

Model	Train Rows	Validation Rows	Test Rows	Accuracy	Precision	Recall	F1 Score
Logistic Regression	1828	392	392	0.5867	0.6667	0.0828	0.1474
LSTM	1814	389	389	0.5604	0.4545	0.1198	0.1896
Transformer	1814	389	389	0.5116	0.4343	0.4551	0.4444

4.4.2 Multi-Stock Baseline Comparison

To test whether the picture emerging from a single Saudi stock actually generalizes, an additional multi-stock comparison was carried out. This extended experiment ran the three predictive baselines Logistic Regression, LSTM, and Transformer on three Saudi tickers side by side: 1202.SR, 2320.SR, and 4300.SR. The purpose of widening the scope in this way was to check whether the relative strengths of the baseline models remain stable when the market behavior underneath them changes, rather than being an artifact of one particular case study

Table 4.3: Multi-stock comparison across Logistic Regression, LSTM, and Transformer baselines for 1202.SR, 2320.SR, and 4300.SR.

Model	Ticker	Train	Val	Test	Acc.	Prec.	Recall	F1
Logistic	1202.SR	1828	392	392	0.5867	0.6667	0.0828	0.1474
LSTM	1202.SR	1814	389	389	0.5604	0.4545	0.1198	0.1896
Transformer	1202.SR	1814	389	389	0.5116	0.4343	0.4551	0.4444
Logistic	2320.SR	2546	546	546	0.5037	0.5000	0.0923	0.1558
LSTM	2320.SR	2532	543	543	0.5120	0.6111	0.0409	0.0767
Transformer	2320.SR	2532	543	543	0.4991	0.4942	0.4758	0.4848
Logistic	4300.SR	2550	546	547	0.5356	0.3103	0.0370	0.0662
LSTM	4300.SR	2536	543	544	0.5588	0.0000	0.0000	0.0000
Transformer	4300.SR	2536	543	544	0.5699	0.6875	0.0458	0.0859

The results make it clear that model performance varies materially from one Saudi stock to another, which is itself an important finding it confirms that any conclusion drawn from a single ticker would necessarily be incomplete. On 1202.SR and 2320.SR, the Transformer baseline

produced a considerably more balanced profile than Logistic Regression, and this was especially visible in the recall and F1 figures. On 4300.SR, both models struggled with recall, yet even in that harder setting the Transformer still edged out the logistic baseline in accuracy, precision, and F1. Read as a whole, this evidence reinforces two related points: accuracy on its own is not a sufficient criterion for selecting a model in a financial classification task, and richer temporal modeling can deliver stronger advisory relevance on selected Tadawul stocks. Once the LSTM baseline was brought into the same comparison, it remained weaker than the Transformer on all three stocks, and on 4300.SR it collapsed to zero recall and zero F1 a further piece of evidence in favor of the Transformer as the most suitable temporal baseline among the three models evaluated here.

This additional experiment strengthens the empirical footing of the thesis by taking the predictive evaluation beyond a single-stock case study and reframing it as a broader multi-stock analysis. The three-model comparison, run across the three tickers, therefore delivers stronger generalization evidence and a more credible empirical basis on which the later hybrid extensions of the framework can be built.

4.5 Trading Performance Analysis

The more practical question, once the predictive layer has been examined, is whether the stronger signals actually translate into better decision quality once they are placed inside the RL environment. The analysis should therefore ask a specific set of questions: does the agent produce more stable returns than the baseline strategies, does it achieve tighter control over its drawdowns, and does it manage its positions in a more disciplined way rather than trading in and out on every fluctuation. In an advisory setting aimed at real users, consistency and risk control often deserve

as much weight as the raw return figure — and in some cases arguably more (Ansari et al., 2022; Kabbani & Duman, 2022).

4.5.1 Trading Baseline Comparison

The financial outcomes obtained for the Buy-and-Hold baseline and the Rule-Based trading strategy on ticker 1202.SR are summarized below. To ensure a scientifically fair comparison, both strategies are evaluated over the identical window of 2,612 trading days from 2015-05-28 to 2025-11-17; the alignment removes the twenty-day warm-up gap that would otherwise favor either strategy through a different entry price. Over this common window, the Rule-Based strategy outperformed Buy-and-Hold by a wide margin, delivering a total return of 147.18% against a loss of -47.19% for Buy-and-Hold. It also brought the maximum drawdown down from -77.29% to -34.76% and lifted the final equity from SAR 52,811.84 all the way to SAR 247,184.47. The Rule-Based strategy did execute 272 trades in the process, yet the resulting performance still points to a clear conclusion: even a structured, non-learning set of decision rules can already generate materially stronger portfolio performance than passive holding under the market conditions tested here.

These figures constitute an important financial baseline for the thesis. Before reinforcement learning even enters the picture, the comparison is already enough to demonstrate that decision-oriented strategies can meaningfully improve both return and downside control relative to a passive exposure to the same stock. The baseline equity comparison chart is intended to be placed in this subsection in the final version of the thesis, so as to give visual support to this reading.

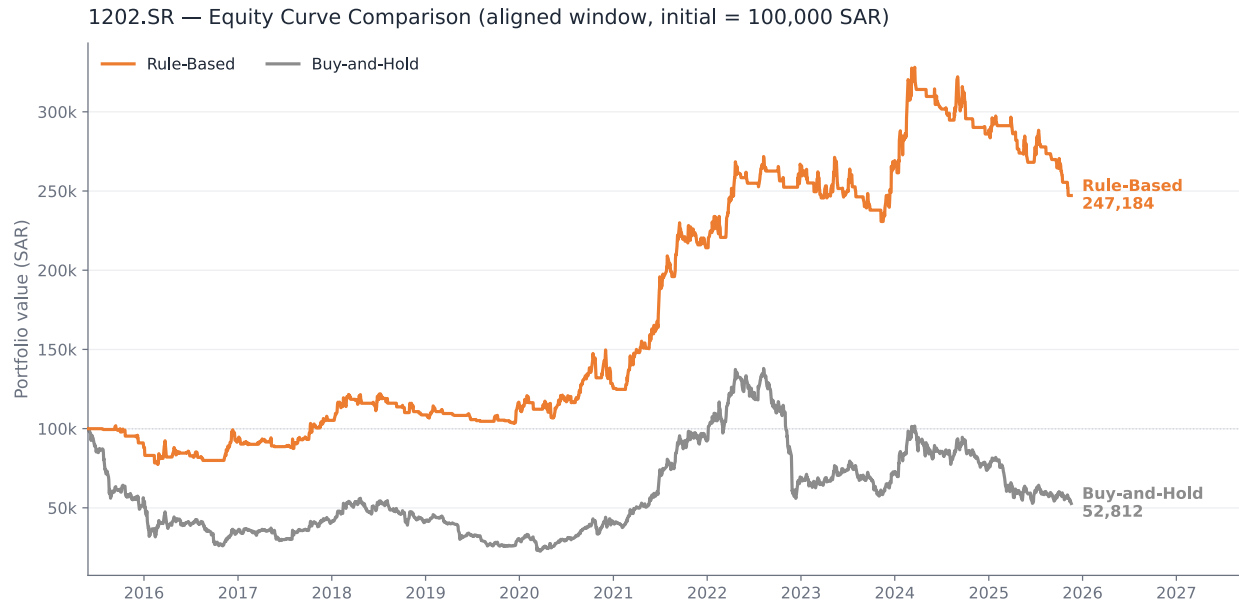


Figure 4.9: Equity curve comparison between the Buy-and-Hold and Rule-Based trading baselines for ticker 1202.SR.

Table 4.4: Trading baseline comparison between the Buy-and-Hold and Rule-Based strategies for ticker 1202.SR.

Metric	Buy-and-Hold	Rule-Based
Ticker	1202.SR	1202.SR
Start Date	2015-05-28	2015-05-28
End Date	2025-11-17	2025-11-17
Number of Rows	2612	2612
Total Return (%)	-47.19	147.18
Number of Trades	1	272
Max Drawdown (%)	-77.29	-34.76
Final Equity (SAR)	52,811.84	247,184.47

4.6 Interpretability and Advisory Quality

Interpretability continues to be one of the core dimensions along which the framework has to be evaluated. The advisory output itself needs to be judged on three fronts at the same time: how clear it is to the user, whether the sequence of recommendations remains internally consistent from one moment to the next, and how well it aligns with the market conditions actually observed at the time. Recommendations that arrive frequently but contradict one another would quickly erode the user's trust in the system, whereas recommendations that communicate their confidence and their attached risk in a stable and readable manner stand a much better chance of being genuinely useful to investors on the ground.

4.6.1 Application-Level Validation Using Sahmk API

In order to confirm that the proposed system holds up in a practical setting, the backend application was linked to the Sahmk market data API, which was chosen to serve as a live external source of Saudi stock market information. The integration was then verified through the application-level API endpoints exposed by the developed backend, and both of the endpoints exercised in the test returned the expected results. The endpoint `/api/v1/market/quote/{symbol}` produced a valid live quote for stock 2222, and the endpoint `/api/v1/market/watchlist/quotes` returned a properly structured list of live quotes for the several Saudi stocks that had been added to the configured watchlist. Stock 2222 (Saudi Aramco) was picked at this point as a representative liquid ticker used solely to validate the API integration itself, and it should be read as separate from the predictive modeling experiments discussed elsewhere.

The HTTP 200 responses received during the test, together with the correctly shaped JSON payloads returned by the service, are enough to confirm that the implemented system is capable of pulling in real market data in real time and then re-exposing that data cleanly through the

project's own API layer. This finding adds weight to the practical side of the thesis, since it shows that the proposed advisory framework does not stop at offline experimentation but is able to run inside a real application environment that is connected to an active market data provider.

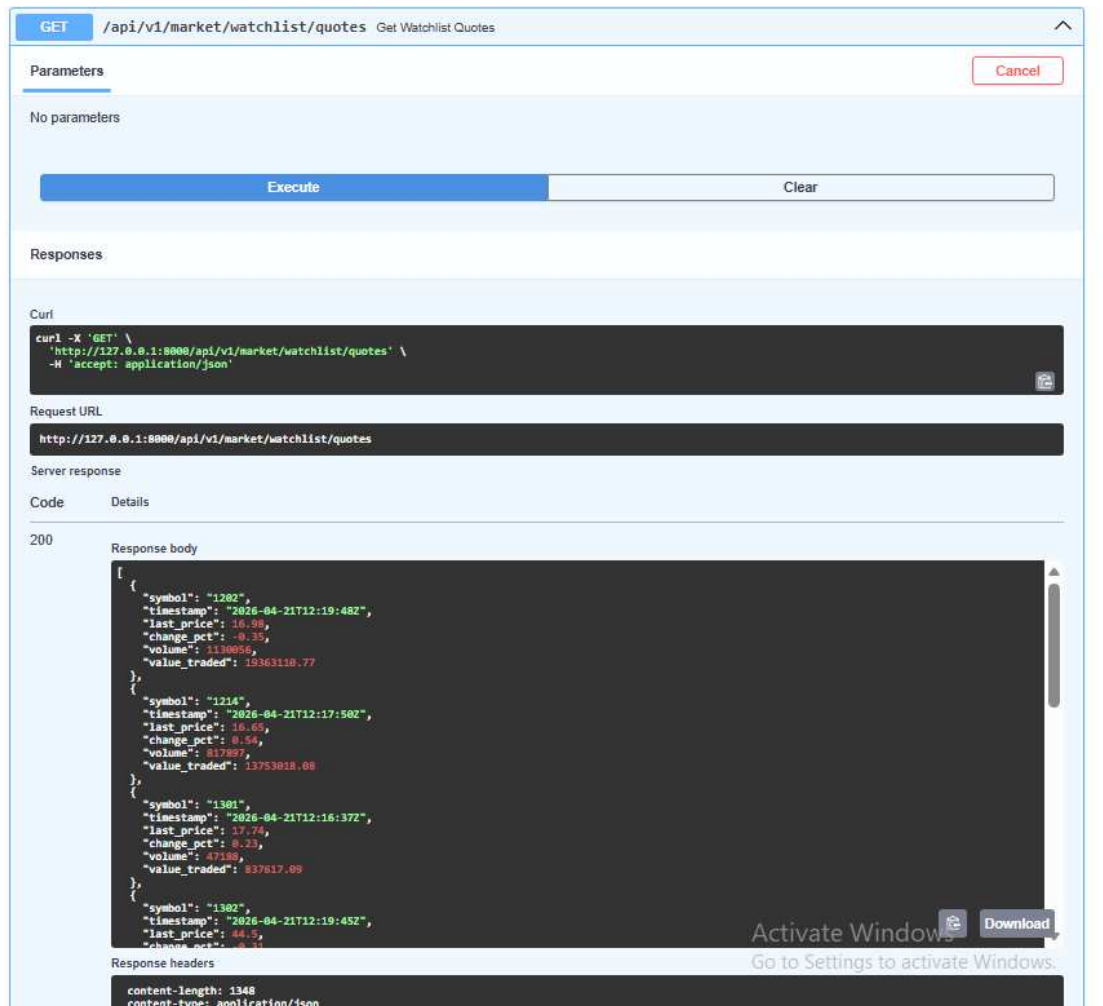


Figure 4.10: Successful retrieval of live watchlist quotes through the developed backend integrated with the Sahmk API. A supplementary screenshot of the successful /api/v1/market/quote/2222 response is retained in the appendix as additional application evidence.

Beyond its immediate validation role, this integration also lays a practical foundation for the future extensions of the system most notably real-time signal generation, dashboard-level visualization of market activity, and watchlist-based monitoring inside the deployed application.

4.7 Discussion

Interpreting the findings honestly requires stepping back and engaging with a few broader questions that the numbers alone cannot answer. The first is whether the graph-based representation of market structure genuinely adds value during sector-led movements or during periods of pronounced co-movement, as opposed to acting as background noise the rest of the time. The second is whether the RL agent is really learning a robust decision policy or is merely overfitting to historical patterns that happen to repeat themselves within the training window. The third is whether the advisory layer manages to simplify the system for the end user without at the same time hiding uncertainty that the user should actually be aware of.

The baseline evidence gathered so far already carries a clear methodological message: leaning on a single metric such as accuracy is enough to produce misleading conclusions in financial classification tasks. The logistic regression model reached the highest accuracy in the comparison, but its recall was so low that a large share of the relevant positive cases went undetected. The Transformer model, in contrast, produced the strongest balance between recall and F1 score, which is what qualifies it as a more suitable predictive benchmark for advisory use. Turning to the financial side, the Rule-Based strategy clearly outperformed Buy-and-Hold in both profitability and drawdown control. This finding holds on an evaluation window aligned to the same 2,612 trading days for both strategies, which removes the small warm-up bias that a naïve comparison would otherwise introduce. Read together, these findings back the central argument of the thesis — that decision-aware and hybrid modeling approaches earn their place not merely because they are theoretically richer, but because they translate into measurably better portfolio performance. The additional multi-stock evidence gathered across 1202.SR, 2320.SR, and 4300.SR further shows that the ranking of models is not perfectly stable when the underlying

stock changes, yet the Transformer still comes out as the strongest overall model once recall and F1 are considered together rather than in isolation.

4.8 Threats to Validity

Several threats to validity have to be acknowledged when reading the results of this study. Assumptions about transaction costs, the specific stock universe that was chosen for the experiments, effects tied to the particular sample period used, and possible shifts in market microstructure over time all belong to that list. The Saudi market, in addition to these general concerns, may go through regime shifts that affect different sectors unevenly — which means that any conclusion drawn from one historical period should not be generalized to future ones without a real degree of caution.

4.9 Result Matrix

The thesis brings the empirical findings together into consolidated result tables that cover both the predictive and the financial sides of the evaluation. One of these tables places the predictive metrics of the different baseline models next to each other, while another summarizes the financially meaningful outcomes obtained for the trading baselines.

A compact result matrix, which summarizes the main empirical findings coming out of both the predictive baselines and the trading baseline comparison, is reported below.

This result matrix reinforces the empirical foundation of the thesis by tying the classification-oriented evidence to the financially meaningful trading outcomes, and by making the connection between the two visible in a single view. It also serves as a structured benchmark against which the subsequent hybrid and reinforcement learning components introduced in later stages of the project can be measured on comparable terms.

Table 4.5: Result matrix summarizing the main predictive and trading findings of the thesis.

Component	Key Results	Interpretation
Logistic Regression Baseline	Accuracy = 0.5867, Precision = 0.6667, Recall = 0.0828, F1 = 0.1474	Highest accuracy and precision, but poor sensitivity to positive cases.
LSTM Baseline	Accuracy = 0.5604, Precision = 0.4545, Recall = 0.1198, F1 = 0.1896	Useful sequential benchmark, but overall classification balance remained limited.
Transformer Baseline	Accuracy = 0.5116, Precision = 0.4343, Recall = 0.4551, F1 = 0.4444	Most balanced predictive baseline and strongest benchmark for further hybrid integration.
Trading Baseline Comparison	Buy-and-Hold return = -47.19%; Rule-Based return = 147.18%; Final equity = SAR 52,811.84 vs SAR 247,184.47	Decision-oriented strategy materially outperformed passive exposure on both return and drawdown control.
Multi-Stock Baseline Comparison	Across 1202.SR, 2320.SR, and 4300.SR, the Transformer delivered the strongest overall balance,	Extends the evidence beyond a single-stock case and reinforces that accuracy

	especially on 1202.SR and 2320.SR, while LSTM remained inconsistent and weak on 4300.SR.	alone is insufficient for model selection.
--	--	--

Chapter 5: Conclusion and Future Work

5.1 Conclusion

In this thesis, I set out to examine — in concrete rather than purely conceptual terms — whether an AI-based investment advisory framework can be built for the Saudi Stock Market. I chose to anchor the discussion in empirical baseline evaluation, and my experiments on stock 1202.SR made two things clear to me. First, predictive baselines do not behave uniformly across metrics; a model can look strong on one measure while quietly failing on another. Second, and closely related, I found that accuracy on its own is a poor guide to model quality in financial classification. Within the set of models I evaluated, the Transformer baseline produced the most balanced classification profile, while the logistic regression baseline reached a higher headline accuracy at the cost of recall too low to be useful in practice. On the financial side, my trading baseline comparison delivered an equally clear message: the rule-based strategy substantially outperformed buy-and-hold in total return, final equity, and maximum drawdown alike. Taken together, these findings give my thesis a real empirical foundation and support my broader argument — any intelligent advisory system built for Tadawul should be judged jointly on its predictive quality and on the financial relevance of its outputs. The multi-stock comparison I later carried out across 1202.SR, 2320.SR, and 4300.SR reinforced this picture: model performance varies materially from one Saudi stock to another, yet the Transformer baseline still

emerged as the strongest overall model once I considered balanced recall and F1 performance together.

5.2 Main Contributions

I can organize the main contributions of my work into four points; with one additional contribution I want to highlight:

First, I put in place a Saudi-market-oriented experimental setup for evaluating stock movement prediction, built on top of structured OHLCV-based features together with a considered set of technical indicators.

Second, I offer a comparative empirical analysis of three predictive baselines — logistic regression, LSTM, and Transformer — and I show that, under my current configuration, the Transformer model provided the most balanced profile in terms of recall and F1 score.

Third, my trading baseline analysis demonstrates that even a simple rule-based strategy can materially outperform a passive buy-and-hold approach on my chosen case study. This ties the predictive analysis directly to financially interpretable outcomes, rather than leaving it on a purely statistical plane.

Fourth, I lay a practical foundation on which my system can be extended into a fuller hybrid advisory architecture. This foundation may later incorporate graph-based market structure, reinforcement learning, and a user-facing recommendation layer.

As a fifth contribution, I extended the predictive evaluation from a single-stock case study to a multi-stock comparison across three Saudi tickers. This produced stronger empirical evidence for the relative robustness of the Transformer baseline.

5.3 Future Work

I plan to carry my current baseline study forward into a more complete hybrid advisory system. My first priority is to bring graph-based market representations into the pipeline, so the sector relationships and cross-stock dependencies present in Tadawul can be modeled explicitly, rather than absorbed implicitly by a per-stock forecaster. My second priority is to layer reinforcement learning on top of the predictive stage, in order to test whether sequential decision policies can genuinely improve trade timing, risk control, and the overall quality of the advisory output beyond what my static rule-based strategies deliver. Closely tied to both of these is the fusion stage itself, which in my current work remains specified rather than implemented. Combining the recurrent, attention-based, and relational representations into a single learned feature space is the step that turns three parallel forecasters into one hybrid model. Evaluating whether that combination outperforms its strongest individual component is, in my view, a necessary test before I can claim the architecture as a whole. Further extensions I want to pursue include enlarging the stock universe under study, drawing on richer intraday and fundamental data, tightening my hyperparameter tuning process, and eventually deploying the framework inside an interactive dashboard that presents its buy, sell, and hold recommendations together with the confidence attached to each and the basic risk context surrounding it.

References

- Ansari, Y. (2024). Multi-Cluster Graph (MCG): A novel clustering-based multi-relation graph neural networks for stock price forecasting. IEEE Access, 12.
<https://doi.org/10.1109/ACCESS.2024.3476159>
- Ansari, Y., Yasmin, S., Naz, S., Zaffar, H., Ali, Z., Moon, J., & Rho, S. (2022). A deep reinforcement learning-based decision support system for automated stock market trading. IEEE Access, 10, 127469–127492.
<https://doi.org/10.1109/ACCESS.2022.3226629>
- Aroussi, R. (2024). yfinance: Download market data from Yahoo! Finance API (Version 0.2) [Computer software]. GitHub. <https://github.com/ranaroussi/yfinance>
- Friday, I. K., Pati, S. P., & Mishra, D. (2025). A multi-modal approach using a hybrid Vision Transformer and Temporal Fusion Transformer model for stock price movement classification. IEEE Access, 13. <https://doi.org/10.1109/ACCESS.2025.3589063>
- Jayanth, T., & Manimaran, A. (2024). Developing a novel hybrid model double exponential smoothing and dual attention encoder-decoder based bi-directional gated recurrent unit enhanced with Bayesian optimization to forecast stock price. IEEE Access, 12.
<https://doi.org/10.1109/ACCESS.2024.3435683>
- Jayanth, T., Manimaran, A., & Siva, G. (2024). Enhancing stock price forecasting with a hybrid SES-DA-BiLSTM-BO model: Superior accuracy in high-frequency financial data analysis. IEEE Access, 12. <https://doi.org/10.1109/ACCESS.2024.3502175>

- Kabbani, T., & Duman, E. (2022). Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2022.3203697>
- Koo, E., & Kim, G. (2022). A hybrid prediction model integrating GARCH models with a distribution manipulation strategy based on LSTM networks for stock market volatility. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2022.3163723>
- Li, Y., Chen, L., Sun, C., Liu, G., Chen, C., & Zhang, Y. (2024). Accurate stock price forecasting based on deep learning and hierarchical frequency decomposition. *IEEE Access*, 12. <https://doi.org/10.1109/ACCESS.2024.3384430>
- Moody, F., Jahangard Rafsanjani, A., Zarifzadeh, S., & Zare Chahooki, M. A. (2024). Fusion of technical indicators and sentiment analysis in a hybrid framework of deep learning models for stock price movement prediction. *IEEE Access*, 12. <https://doi.org/10.1109/ACCESS.2024.3519556>
- SAHMK. (2026). SAHMK market data API documentation. <https://app.sahmk.sa>
- Saudi Exchange. (2026). Listed companies directory. <https://www.saudiexchange.sa>
- Sun, Y., Mutalib, S., Omar, N., & Tian, L. (2024). A novel integrated approach for stock prediction based on modal decomposition technology and machine learning. *IEEE Access*, 12. <https://doi.org/10.1109/ACCESS.2024.3425727>
- Swetha, B., & Arya, K. (2025). Prediction of the stock market using a hybrid model based on feature expansion and LSTM-based algorithms. *IEEE Access*, 13. <https://doi.org/10.1109/ACCESS.2025.3631096>
- Yahoo Finance. (2026). Historical market data for Tadawul-listed equities [Data set]. <https://finance.yahoo.com>

