Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

HybridFinOracle: A Gated-Fusion Deep Learning Framework for Directional Stock Return Prediction on the Tehran Stock Exchange

Marzieh Bagherinia Amiri [1], Heshaam Faili [2]
[1] *School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran.*
marziehbagherinia@gmail.com , *Orcid: https://orcid.org/0009-0003-7822-1268*
[2] *School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran*

ARTICLE INFO

Received: 15 Mar 2025

Revised: 04 May 2025

Accepted: 12 May 2025

ABSTRACT

This study introduces **HybridFinOracle, a novel hybrid deep learning framework designed to enhance the directional prediction of stock returns on the Tehran Stock Exchange (TSE).** Leveraging a comprehensive dataset spanning January 1, 2014, to December 31, 2024, our approach fuses structured financial indicators with qualitative textual information through two dedicated processing streams. **The Integrated Time-Series Stream** ingests a 30-day sequence of normalized OHLCV data and a set of key, effective technical indicators, alongside 30-day trend and residual vectors derived via classical time-series decomposition. **Simultaneously, the NLP Stream** processes relevant news texts from the preceding 7 calendar days—collected from 10 leading and widely trusted Persian-language news platforms—filtered by a zero-shot Llama-3 classifier for general and stock-specific impact, and encodes them using a pre-trained ParsBERT-based model, with document embeddings aggregated via global max pooling. A gated fusion mechanism dynamically weights these modalities before final dense layers, while Monte Carlo Dropout provides uncertainty estimates. Hyperparameters are optimized with Bayesian methods (Optuna) to maximize AUC-ROC and F1-score. **Empirical evaluation on an unseen test set (10,000 stock-day observations)** yields **76.2%** directional accuracy and an AUC of **0.835**, showing approximately **12% improvement in accuracy over the best baseline model (LSTM)**, and significantly outperforming logistic regression, SVM, and random-walk baselines. These results demonstrate the framework's capability to capture complex temporal patterns, market regimes, and sentiment signals, offering a scalable solution for more accurate and robust financial forecasting.

Keywords: Financial Forecasting, Deep Learning, Natural Language Processing (NLP), Hybrid Model, Time-Series Analysis, Feature Fusion, Stock Market Prediction.

INTRODUCTION

Financial forecasting is a high-stakes challenge that has increasingly benefited from advances in artificial intelligence (AI) and deep learning. In recent years, researchers have applied a variety of machine learning and deep neural network models to capture complex patterns in financial time-series data. In fact, hybrid deep learning approaches (e.g. combining ARIMA and LSTM networks) have been shown to **outperform traditional methods** by modeling both linear and non-linear patterns in the data. Deep learning techniques now often represent the state-of-the-art in financial forecasting tasks, offering significant improvements in predictive accuracy over earlier statistical approaches. These AI-driven methods hold great promise for investors and analysts, as even marginal gains in forecast precision can translate into substantial economic value.

However, relying solely on historical price patterns (the realm of **technical analysis**) is inherently limiting – unexpected events and shifts in market sentiment can rapidly destabilize trends. To capture such effects, a growing body of work augments quantitative data with information from news, social media, and other textual sources (the domain of **fundamental analysis**). By applying natural language processing (NLP) to financial news and reports, researchers aim to gauge the market's qualitative sentiment and incorporate it into predictive models. Indeed, financial news has been proven to be a crucial driver of stock price fluctuations, and numerous studies have found that blending textual sentiment indicators with numerical features yields significantly improved forecasting performance. For example, past research on the Tehran Stock Exchange (TSE) showed that incorporating sentiment from Persian news (often overlooked in prior work) can noticeably enhance the prediction of market indices. These developments underscore a clear trend: integrating textual sentiment signals with traditional time-series data is becoming essential for next-generation AI-driven financial forecasting.

Despite this progress, several challenges persist in current deep learning models for finance. One issue is that early approaches to text analysis in this domain frequently used shallow representations like bag-of-words, which ignore the nuanced sentiment and context of news articles. This is problematic because investors react to news based on its perceived sentiment (positive or negative), which in turn drives their trading decisions. Ignoring the **tone** of news can thus lead to missing critical predictive signals. Additionally, financial time-series are notoriously noisy and non-stationary, which means complex models risk overfitting to historical patterns that may not repeat. Market regimes change over time, and models that lack mechanisms for adaptation or regularization may perform inconsistently across different market conditions. Another limitation of many deep learning predictors is the absence of uncertainty quantification – they typically output point estimates with no indication of confidence. In volatile markets, this can be problematic, as decision-makers need to understand the reliability of a prediction. Furthermore, effectively fusing heterogeneous data (numerical and textual) is non-trivial. A naive fusion (such as simply concatenating features from price and text models) fails to account for the context-dependent influence of information – for instance, a major news headline might dramatically outweigh recent technical trends, whereas during quiet periods, historical price patterns might dominate. Designing a model that can dynamically balance these information sources remains an open challenge.

To address these gaps, we propose **HybridFinOracle**, a novel deep learning architecture for financial forecasting that combines the strengths of time-series models and NLP-driven sentiment analysis. The HybridFinOracle model is built as a two-stream neural network: one stream is dedicated to quantitative market data (e.g. historical prices and technical indicators), and the other stream processes textual news data to extract sentiment and semantic features. This design is conceptually inspired by recent multi-stream approaches that separately handle different data modalities, but we extend it with an innovative fusion and inference strategy. At the core of our architecture is a **gated fusion** mechanism that learns to integrate the two streams adaptively. Rather than merging the data sources in a fixed or ad-hoc manner, the gated fusion layer acts as a dynamic filter – it weighs and combines the time-series signals and the news-driven signals based on their relevance at each moment. In practice, this means the model can **accentuate** the information source that is more predictive under current conditions (for example, giving more weight to the news stream when a significant event occurs, or relying more on the technical stream when markets are calm). This gating approach effectively addresses the fusion challenge by letting the model itself decide how much each modality should influence the forecast.

Another distinguishing feature of HybridFinOracle is its incorporation of **Bayesian learning** methods into the forecasting framework. In contrast to standard deterministic neural networks, our model includes a Bayesian component that enables probabilistic reasoning about predictions. Concretely, we train the model in a manner that yields not just a single predicted value, but a distribution over possible outcomes – reflecting the model's confidence or uncertainty in its prediction. This can be implemented, for example, via Bayesian neural network layers or Monte Carlo dropout techniques that treat certain model weights as random variables. By doing so, HybridFinOracle provides **confidence intervals** or uncertainty estimates alongside point forecasts, directly addressing the need for risk-aware predictions in finance. The Bayesian aspect also acts as a form of regularization, helping to prevent overfitting by effectively averaging predictions over many plausible model configurations. In summary, our architecture marries the pattern-capturing power of deep learning with the principled uncertainty estimation of Bayesian inference – a combination that is particularly well-suited for the uncertain nature of financial markets.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

We evaluate the HybridFinOracle framework through an extensive case study focused on the **Tehran Stock Exchange (TSE)**, using **Persian financial news** as the source of textual data. By constructing a novel dataset of Persian news articles aligned with TSE stock data, we demonstrate the model's ability to handle a less-resourced language and to capture culturally specific sentiment cues. The news data are processed with appropriate NLP techniques (including tokenization and sentiment scoring tailored for Persian) to ensure that the text stream of our model receives meaningful inputs. HybridFinOracle is then trained to predict stock price movements or trends in the TSE, given both the time-series history and the contemporaneous news information. This real-world evaluation showcases the practical applicability of our approach and provides insights into the influence of news on an actual financial market.

We conduct comprehensive experiments to benchmark HybridFinOracle against a range of baseline and state-of-the-art models. Our results show that **HybridFinOracle consistently delivers superior performance** across multiple predictive metrics. In particular, it achieves higher forecasting accuracy than the baseline models, confirming that neither historical data nor news alone is sufficient – both are necessary for the best results. The gated fusion strategy proves effective: HybridFinOracle outperforms fusion methods that lack an adaptive gating mechanism, indicating that letting the model learn **when** to trust news over technical data (and vice versa) yields tangible gains. We also observe that our hybrid model produces more stable and reliable predictions, attributable to its Bayesian component which provides well-calibrated outputs. Overall, HybridFinOracle achieved **state-of-the-art results** on the TSE dataset, outperforming existing deep learning benchmarks in this context. These findings validate our architecture and underscore the value of integrating deep learning, NLP-derived sentiment, and Bayesian inference for financial market forecasting.

While our study is grounded in stock market prediction, the framework and insights from HybridFinOracle have broad implications. The two-stream, gated fusion architecture serves as a general template that can be applied to other predictive modeling tasks beyond equities – essentially any scenario where one needs to combine time-series signals with unstructured textual (or even other modality) information. Potential applications range from **macroeconomic indicator forecasting** (e.g. merging economic time-series data with news about policy changes) to **commodity and energy markets** (integrating price histories with news about supply disruptions or weather events) and even domains outside finance (such as blending sensor measurements with textual incident reports for predictive maintenance, or epidemiological time-series with health news for disease outbreak prediction). By demonstrating how to effectively fuse and weight different information sources, our work bridges methodologies in machine learning and finance, contributing a versatile approach that others can adapt and build upon. In the big picture, HybridFinOracle points toward more holistic AI-driven forecasting models – ones that break down data silos and exploit both numeric data and textual narratives to make more informed, context-aware predictions.

In summary, the key contributions and innovations of this work include:

HybridFinOracle Architecture: A novel hybrid deep learning architecture that integrates a time-series prediction model and an NLP-based sentiment analysis model in a two-stream design, enabling simultaneous processing of quantitative market data and textual news data for forecasting.

Gated Fusion Mechanism: A learnable gated fusion module that dynamically combines the outputs of the two streams. This mechanism allows the model to adaptively weight the importance of historical trends versus news-driven signals, effectively capturing context-dependent market drivers.

Bayesian Integration: The incorporation of Bayesian learning techniques into the framework, providing probabilistic predictions with uncertainty estimates. This feature improves the model's robustness and risk-awareness, which are crucial for financial decision-making under volatile conditions.

Persian News & TSE Case Study: A comprehensive evaluation using Persian financial news and Tehran Stock Exchange data, demonstrating the model's effectiveness in an emerging market and a non-English language context. This case study fills a gap in the literature by showing how local-language news sentiment can be leveraged for stock prediction.

Superior Performance: Empirical results showing that HybridFinOracle outperforms multiple baseline models and recent deep-learning approaches. The hybrid model achieved higher accuracy and more stable predictions than models using only price or only text inputs, highlighting the value of its two-stream fused design.

Generalizability: A discussion of the framework's generalizability. The HybridFinOracle approach can be extended to other forecasting and predictive analytics tasks involving multi-modal data, suggesting a broad potential impact and a new direction for integrating deep learning with heterogeneous financial data sources.

LITERATURE REVIEW

The advent of deep learning (DL) has substantially reshaped the landscape of financial market prediction, offering powerful tools to navigate the inherent complexities of financial data, characterized by high dimensionality, non-linearity, and pronounced volatility (Rajendran et al., 2024; Zheng et al., 2024). Traditional statistical methods, while foundational, often struggle to adequately capture these multifaceted dynamics. In contrast, DL architectures, particularly Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), have demonstrated remarkable efficacy in forecasting diverse financial indicators, including stock prices and exchange rates. These models leverage sophisticated architectures to discern intricate patterns within financial time series, often yielding more accurate and reliable forecasts than their conventional counterparts (Dokumacı, 2024; Idowu, 2024).

Financial markets generate vast quantities of time series data, which are inherently dynamic and non-linear. LSTM networks, specifically designed to capture long-term temporal dependencies, are particularly well-suited for such data (Leng, 2024; Zhang et al., 2024a). For instance, multiple studies highlight the superior performance of LSTMs in stock price prediction when benchmarked against traditional models like Autoregressive Integrated Moving Average (ARIMA) and Moving Average Convergence Divergence (MACD) (Rajendran et al., 2024; Sonkavde et al., 2023). While LSTMs excel at temporal sequences, CNNs contribute by effectively identifying spatial hierarchies or local patterns, which can be analogously applied to feature maps derived from financial data or even visual representations like candlestick charts.

Recent innovations in DL have introduced even more potent models for time series analysis. Transformer models, initially engineered for natural language processing (NLP), have been successfully adapted for financial forecasting due to their self-attention mechanisms, which allow them to weigh the importance of different parts of an input sequence and capture global dependencies more effectively than recurrent architectures (Patel et al., 2023; Li & Bastos, 2020). Recognizing that different architectures possess unique strengths, hybrid models that synergize components like CNNs and LSTMs (Farimani et al., 2022; Takale, 2024) or Transformers and LSTMs have emerged. These hybrids aim to concurrently process spatial and temporal features or different facets of sequential data, often leading to enhanced robustness and predictive power for complex financial time series.

A pivotal development has been the integration of NLP techniques with DL for financial prediction. As noted by Heaton et al. (2016) and corroborated by more recent studies (Ohliati & Yuniarty, 2024), sentiment extracted from news articles, social media, and other textual sources can provide invaluable leading indicators of investor behavior and market movements. DL models, especially Transformer-based architectures, excel at analyzing large volumes of textual data, extracting sentiment, and correlating these sentiments with market fluctuations (Li, 2024; Krishna et al., 2023). The sophistication of text representation algorithms (e.g., word embeddings, contextual embeddings from models like BERT) has further augmented NLP's utility, enabling the extraction of salient features from unstructured text that can complement traditional financial indicators, thereby enhancing overall prediction accuracy (Zhao & Huang, 2024; Ti, 2024). Indeed, studies demonstrate that combining sentiment analysis with technical indicators can bolster the resilience and performance of financial forecasting models (Mienye et al., 2024; Zhang et al., 2024b), suggesting the power of multi-modal data fusion.

The field has witnessed the application of numerous DL architectures, each presenting distinct advantages and limitations in the context of financial market prediction. A comparative overview is presented below:

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1. Comparison of Deep Learning Model Types

Model Type	Description	Strengths	Potential Limitations
LSTM Networks	Designed to capture temporal dependencies in sequential data.	Effective for time series forecasting; handles long-term dependencies well.	Can be computationally intensive; may struggle with extremely long sequences.
CNNs	Useful for processing spatial hierarchies in data.	Captures local patterns; effective for image-like data or feature map extraction.	Less inherently suited for sequential order without modifications or hybrid use.
Transformer Models	Leverage self-attention mechanisms for sequence modeling.	Excellent for capturing global dependencies; suitable for parallel processing.	Requires significant data; can be complex to tune and interpret.
Hybrid Models	Combine different architectures, such as CNN-LSTM or Transformer-LSTM.	Captures both spatial/local and temporal/global dependencies; robust for complex data.	Increased model complexity; potential for overfitting if not carefully designed.

These models have found application across a spectrum of financial prediction tasks, including forecasting stock prices, exchange rates, and commodity prices, with the choice of model often contingent upon the specific characteristics of the data and the prediction task at hand (Teixeira & Barbosa, 2024; Sahani, 2024).

Beyond direct market prediction, DL has become instrumental in optimizing trading strategies and enhancing market understanding. By analyzing historical market data, DL algorithms can identify patterns and trends that inform adaptive trading strategies, with reinforcement learning (RL) being explored for developing systems that react dynamically to market conditions (De Avila & Salgado, 2023). The fusion of technical analysis with DL, where models incorporate indicators like RSI and moving averages, has also proven effective in generating more reliable trading signals (Sharma & Gupta, 2022). Furthermore, generative adversarial networks (GANs) are being investigated for their potential to simulate realistic market conditions and generate synthetic data for augmenting training sets, which is particularly useful given the often limited and noisy nature of financial data (Karthik et al., 2023).

Despite these significant strides, several challenges persist in the application of DL to financial market prediction. The "black box" nature of many deep learning models poses a substantial hurdle to interpretability, which is critical for high-stakes financial decision-making where understanding the rationale behind a prediction is paramount (Kumar et al., 2021; Jethani et al., 2023). Overfitting remains a persistent issue, especially when models are trained on volatile and noisy financial data; while regularization techniques like dropout and early stopping are employed, developing more inherently robust models is an ongoing research area (Raut et al., 2024; Olorunnimbe & Viktor, 2023). Data quality is another critical determinant of success; financial data are often plagued by missing values and noise, necessitating sophisticated preprocessing techniques like normalization and outlier detection to ensure high-quality training inputs (Hassan et al., 2022; Li et al., 2022).

The utility of DL extends broadly across financial management, encompassing algorithmic trading, portfolio management, credit risk analysis, and fraud detection. For instance, DL models have been effectively deployed to assess creditworthiness by analyzing complex patterns in credit data (Jain et al., 2022; Sable et al., 2022). In portfolio management, these models can optimize asset allocation by analyzing historical returns and volatility to identify configurations that maximize returns while mitigating risk (Yekrangi & Abdolvand, 2021; Zhang et al., 2021). Similarly, in fraud detection, DL excels at identifying anomalous transactions indicative of fraudulent activity (Le et al., 2020; Guo & Tuckfield, 2020).

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Looking ahead, the prospects for DL in financial market prediction are promising, with several avenues for future exploration. Integrating diverse data sources, such as satellite imagery or sensor data, holds potential for enhancing prediction accuracy. A crucial area of development is Explainable AI (XAI), as interpretable DL models are essential for fostering trust and adoption in an industry where accountability is key. Furthermore, the nascent field of quantum machine learning offers intriguing possibilities for solving complex optimization problems in finance more efficiently than classical algorithms, potentially leading to breakthroughs in financial modeling and forecasting.

This evolving landscape, with its demonstrated successes in leveraging DL for various financial tasks and the concurrent push towards integrating diverse data modalities like textual information, highlights both the potential and the existing complexities. The trend towards hybrid models and the incorporation of NLP underscores a recognition that no single data type or model architecture holds all the answers. However, the effective and optimized combination of these sophisticated approaches remains a frontier.

2.1. Research Gap

Despite the burgeoning body of literature on the application of deep learning and natural language processing (NLP) in financial market prediction, a significant research gap persists, particularly concerning the *systematic optimization* and *holistic integration* of these methodologies within unified frameworks. While many studies explore the deployment of DL architectures like LSTMs or Transformers for identifying temporal patterns (as reviewed above), and others separately investigate sentiment analysis, the optimal strategies for fusing these information streams and tuning the resultant complex models are less understood. Specifically:

- **Optimization of Hybrid Architectures:** While hybrid models are acknowledged for their potential, there is a lack of in-depth research into rigorous optimization techniques tailored for architectures that combine, for instance, advanced sequential models for price data with Transformer-based NLP components for textual data. Algorithmic optimization is crucial not only for maximizing predictive accuracy but also for preventing overfitting and enhancing model robustness in the face of market volatility.
- **Comprehensive Frameworks for Multi-Modal Data:** Current literature often treats quantitative market indicators and qualitative textual data as inputs to separate models or combines them in relatively simplistic ways. There is a dearth of comprehensive, end-to-end hybrid frameworks that are explicitly designed to concurrently process and dynamically weigh the contributions of both numerical patterns and sentiment-driven fluctuations through sophisticated fusion mechanisms.
- **Integration Beyond Simple Concatenation:** Many existing studies that do combine DL and NLP often approach them as distinct elements whose outputs are merely concatenated or averaged. Insufficient attention has been paid to developing more intricate decision-making models where NLP insights actively inform or gate the processing of time-series data, or vice-versa, within a single, cohesively trained system.

This disparity is particularly noticeable in research addressing the erratic nature of emerging or volatile financial markets, where a nuanced understanding of both quantitative trends and qualitative, sentiment-driven shifts is paramount. Therefore, there is a pressing need for empirical studies that develop and evaluate sophisticated, optimized hybrid frameworks. Closing this gap by focusing on advanced optimization methods embedded within such integrated models could significantly enhance predictive accuracy and operational resilience in financial forecasting. Furthermore, it could pave the way for scalable models applicable across diverse international economic sectors, moving beyond ad-hoc combinations to methodologically sound, optimized solutions.

METHODOLOGY

3.1. Data Acquisition and Preprocessing

This study leverages a multifaceted dataset integrating quantitative financial metrics and qualitative textual information to comprehensively model and predict dynamics within the Tehran Stock Exchange (TSE). The data

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

collection period spans from January 1, 2014, to December 31, 2024, ensuring a significant historical scope covering diverse market cycles and economic conditions.

3.1.1. Structured Financial Data Sources

3.1.1.1. Primary Source and Scope:

Quantitative financial data were collected for the 20 most valuable companies in the market based on market capitalization, which have had historical data available since January 1, 2014. The data were sourced from the Tehran Securities Exchange Technology Management Co. (tsetmc.com), the official and most comprehensive data provider for Iranian capital markets.

3.1.1.2. Collected Variables:

The following daily historical data points were collected for each listed entity:

- **Market Trading Data**: Open, High, Low, Close (OHLC) prices, Trading Volume (number of shares), Trading Value (monetary value of shares traded).
- **Company Fundamental Metrics:** Earnings Per Share (EPS), Net Asset Value (NAV) per Share.
- **Valuation Ratios:** Price-to-Earnings (P/E) Ratio, Price-to-Sales (P/S) Ratio, Industry Group P/E Ratio (average P/E of the stock's designated sector).
- **Company Size and Sector:** Market Capitalization, Stock Group (Industry Sector classification as per TSE).
- **Investor Participation Metrics:** Individual Investor Buy Volume, Institutional/Corporate Investor Buy Volume, Individual Investor Sell Volume, Institutional/Corporate Investor Sell Value (monetary).

3.1.1.3. Initial Processing of Structured Data:

Following data collection, structured financial data underwent a rigorous preprocessing pipeline to ensure data quality, consistency, and suitability for machine learning model input.

- **Identification and Exclusion of Non-Trading Days:** Dates corresponding to recognized TSE holidays and weekends, during which no trading occurs, were identified. All data entries for these non-trading days were removed from the dataset for each stock. This ensures that the time series exclusively reflects active trading periods.
- **Handling Missing Data (for remaining active trading days):** Company Fundamental Metrics (EPS, NAV) and Valuation Ratios (P/E, P/S, Industry Group P/E): For these metrics, which are typically reported periodically and remain static between reporting dates, missing values on active trading days were addressed using a forward-fill imputation strategy. This involves propagating the last known valid observation forward to fill the gaps, reflecting the persistence of such information until a new update is available.

Market Trading Data (OHLC, Trading Volume, Trading Value) and Investor Participation Metrics (on active trading days):

Sporadic Missing Data: For isolated missing data points (e.g., 1-2 consecutive active trading days) for a specific stock, linear interpolation was applied to OHLC prices. For trading volume and value, if interpolation was not sensible, forward-filling the value from the most recent active trading day for that stock was used. Alternatively, a moving average calculated from recent active trading days for that specific stock could be considered.

Prolonged Missing Data on Active Trading Sequences: Stocks exhibiting extensive, unrecoverable missing data for critical fields (e.g., close price or volume) over a significant continuous sequence of active trading days were flagged. If the period of missingness was substantial enough to compromise the integrity of the time series for that stock, the stock might be excluded from the analysis, either for the affected period or entirely if the overall data quality was deemed insufficient.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Adjustments for Corporate Actions:** Historical OHLC prices and trading volumes (on active trading days) were systematically adjusted for corporate actions such as stock splits, reverse splits, stock dividends, and rights issues. This adjustment ensures that the historical price and volume data are comparable over time, reflecting true value changes rather than artificial shifts due to corporate restructuring. Standard adjustment formulas were applied based on the ex-dates of these actions.

3.1.1.4 Labeling

For the primary task of predicting market or stock movement from structured financial data, the label at each time step t (corresponding to an active trading day) was defined as the **sign of the future return**.

- **Prediction Target:** The model aims to predict whether a stock's price will increase, decrease, or remain relatively unchanged over a defined future period h.
- **Return Calculation:** The future return $R(t+h)$ was calculated as:

$$R(t+h) = \frac{P(t+h) - P(t)}{P(t)}$$

where $P(t)$ is the closing price at the current active trading day t, and $P(t+h)$ is the closing price of h active trading days into the future. In this research, the h is equal to 1.

- **Label Generation:** The continuous return $R(t+h)$ was then converted into a categorical label representing the direction of movement:
  - UP (+1): If $R(t+h) > \theta$
  - DOWN (-1): If $R(t+h) < \theta$

The θ value is equal to 0.5% of the initial price, which corresponds to the commission and tax of the transaction.

3.1.2. Unstructured Textual Data Sources

To incorporate the influence of public discourse, news sentiment, and macroeconomic narratives, a diverse corpus of textual data was aggregated daily. All textual documents were collected with their original publication timestamps to enable subsequent temporal alignment with financial data and event-based analysis.

3.1.2.1. Domestic News Sources (Persian Language):

- **Telegram Channels:** Content was collected from the official Telegram channels of approximately ten leading Iranian news organizations, selected for their reputation and comprehensive coverage of economic, financial, and political news. Examples include *Donya-e-Eqtesad, Tejarat News, IRNA,* and *Bourse News*. Data collection utilized publicly available Telegram APIs.
- **Twitter (X) Platform:** Tweets were gathered from a curated list of over 100 Twitter accounts representing prominent Iranian economists, financial analysts, and economic journalists. Data was accessed via the official Twitter API, filtering for relevant Farsi and English tweets.

3.1.2.2. Initial Processing of Unstructured Data:

The primary goal of preprocessing textual data was to clean and standardize the content for ingestion by advanced NLP models while preserving original semantic richness.

- **Timestamp Synchronization and Duplicate Removal:** Timestamps were maintained, and identical articles/posts were deduplicated.
- **Basic Cleaning:** HTML/XML tags were removed, and whitespace was normalized.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- **Language-Specific Normalization:** Persian text involved character normalization (e.g., unifying "ی" and "ک" forms). The English text was converted to lowercase, and common contractions were expanded.
- **Handling of URLs, Hashtags, and Mentions:** URLs and mentions were replaced with generic tokens (`[URL]`, `[MENTION]`). Hashtags were retained for their semantic value.
- **Special Character and Emoji Handling:** Most standard punctuation was retained. Emojis were retained or converted to textual descriptions (e.g., `[SMILEY_FACE]`).
- **Tokenization Deferral:** Aggressive tokenization, stemming, lemmatization, and extensive stop-word removal were deliberately **avoided**. The actual tokenization is a part of the subsequent NLP feature extraction phase using the specific model's tokenizer.
- **Final Data Structure:** Preprocessed texts were stored with metadata (timestamp, source, language, unique ID) in a structured format for efficient access.

3.1.2.3. Labeling

Each raw text document was automatically classified, one by one, using a pre-trained Llama-3 70B Chat model API in a zero-shot setting. We designed a single prompt that guided the model to assign each document to exactly one of three impact categories, then parsed its response to extract both the label and any named stocks or sectors.

- **Prompt Structure:** For each document (with its original timestamp), the prompt asked the model to judge its likely effect on the Tehran Stock Exchange, choosing one of:
  - **General_Impact**: Broad news expected to move most TSE stocks (e.g., national interest-rate changes, major political events, international sanctions).
  - **Stock_Specific_Impact**: News targeting a particular company or sector (e.g., “Company X posts record profits,” “automotive subsidies announced”). When this label is chosen, the model also names the affected stock(s) or sector(s).
  - **Negligible_Impact**: Items unlikely to influence market prices directly (e.g., sports results, cultural events, unrelated international news).
- **Output Processing**
  - We parse the model’s textual reply to extract the single label **(General_Impact, Stock_Specific_Impact, or Negligible_Impact)**.
  - If **Stock_Specific_Impact** is selected, any mentioned stocks or sectors are logged alongside the label.
  - Both label and entities are stored with the document’s timestamp for downstream use.

3.2. Feature Engineering

Following data acquisition, preprocessing, and initial labeling, a comprehensive set of features was engineered to capture various aspects of market dynamics. This includes technical price patterns, the influence of textual information, and broader market regime characteristics. All features were constructed considering the chronological nature of the data to prevent lookahead bias. Scaling and normalization were applied after splitting the data into training, validation, and test sets, using parameters derived only from the training set or appropriate rolling windows.

3.2.1. Technical Indicators from Structured Data

A selection of widely recognized technical indicators was calculated from the preprocessed OHLCV (Open, High, Low, Close, Volume) data for each stock on active trading days.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

3.2.1.1 Selected Indicators and Parameters:

- **Trend Indicators:**
  - **Simple Moving Average (SMA):** SMA(20), SMA(50) (calculated on closing prices).
  - **Exponential Moving Average (EMA):** EMA(12), EMA(26) (calculated on closing prices).
  - **Moving Average Convergence Divergence (MACD):** Using EMA(12) and EMA(26) for the MACD line, and EMA(9) of the MACD line for the Signal Line. The MACD Histogram (MACD - Signal Line) was also included.
- **Momentum Indicators:**
  - **Relative Strength Index (RSI):** RSI(14) (calculated on closing prices).
  - **Stochastic Oscillator (%K, %D):** Using a 14-day period for %K, and a 3-day SMA of %K for %D. (%K parameters: 14, %D parameters: 3).
  - **Rate of Change (ROC):** ROC(10) (percentage change over 10 trading days).
- **Volatility Indicators:**
  - **Bollinger Bands (BBands):** Using an SMA(20) for the middle band and 2 standard deviations for the upper and lower bands. Features derived include the Upper Band, Lower Band, and Bandwidth ((Upper - Lower) / Middle).
  - **Average True Range (ATR):** ATR(14).
- **Volume-Based Indicators:**
  - **On-Balance Volume (OBV).**
  - **Money Flow Index (MFI):** MFI(14).

The raw OHLCV values for each day were also retained as base features alongside these indicators.

3.2.1.2. Windowed Normalization of Technical Features:

The raw OHLCV values and all calculated technical indicators for each stock formed a set of time-series features. To prepare these for input into sequence models and ensure consistent scaling, a **rolling window Z-score normalization** was applied.

For each feature X at each time step t for each stock, it was normalized using the mean (μ_w) and standard deviation (σ_w) calculated over a **lookback window of N=30 preceding active trading days** for that specific stock and that specific feature:

$$X_{normalized}(t) = \frac{X(t) - \mu_x(t)}{\sigma_w(t)}$$

where $\mu_x(t)$ and $\sigma_w(t)$ are the mean and standard deviation of feature X over the window $(t - 30, t]$. This adaptive normalization uses only past data within the 30-day window. For the initial periods where a full 30-day window was not available, statistics from an expanding window were used.

3.2.2. Collection of Relevant Textual Data (LLM-Filtered)

Based on the LLM-assigned impact labels, relevant textual documents were identified and associated with each stock and trading day. This step filters the vast corpus of text to a more targeted set for subsequent NLP processing.

- **Data Collection Criteria:** For each active trading day t and for each stock s:

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- **General Market News:** All textual documents published within a lookback period of $w_{text} = 7$ calendar days and labeled by the LLM as General_Impact were collected and associated with day t.
- **Stock-Specific News:** All textual documents published within the same $w_{text} = 7$ calendar day lookback period and labeled as Stock_Specific_Impact, where the LLM-identified entity (stock or stock group name) matched stock s (or the group to which stock s belong), were collected and associated with stock s for day t.

- **Output of this Stage:**

The output of this stage is, for each stock s and day t, two sets of raw textual documents: one set of general market impact news and one set of stock-specific impact news. The actual transformation of these collected texts into numerical features will be detailed in the NLP module description within the Hybrid Deep Learning Model Architecture section. This stage focuses on the curated *collection* of relevant texts based on LLM labeling.

3.2.3. Market Regime Feature Vector via Time-Series Decomposition

To incorporate information about the broader market trend and cyclical patterns, which can define the prevailing market regime, **Classical Time-Series Decomposition** was applied to a market proxy. This provides a vector of components representing different underlying dynamics. A trend $T(w)$, a seasonal component $S(w)$, and a residual/random component $R(w)$. An additive decomposition is assumed:

$$Y(w) = T(w) + S(w) + R(w)$$

For each active trading day t, classical time-series decomposition was applied to the market index closing prices within a **lookback window of N=30 preceding active trading days** (i.e., data from $(t - 30, t]$). The **trend component (**$T(w)$**)** and the **residual component (**$R(t)$) extracted from the 30-day window of market index prices were used as feature vectors. This results in two vectors, each of length 30, representing the recent trend and unexplained variations in the market index.

These feature vectors (Trend_Vector and Residual_Vector) aim to provide the model with a richer representation of the recent market behavior and prevailing conditions beyond a single volatility number. This information, reflecting the market's state over.

3.3. Hybrid Deep Learning Model Architecture

To predict the next-day return sign for individual stocks with associated uncertainty, a sophisticated hybrid deep learning architecture was designed. This model integrates an advanced Time-Series Processing Stream for numerical financial and market regime data, and an NLP Processing Stream for textual news data, culminating in a gated fusion mechanism and a Bayesian-interpretable output.

3.3.1. Overview of the Hybrid Approach

The model comprises two primary processing streams whose outputs are subsequently fused using a learned gating mechanism:

- **Integrated Time-Series Processing Stream:** Handles stock-specific technical indicators and market-wide trend/residual components using a deep LSTM network.
- **NLP Processing Stream:** Processes the collected relevant textual news data using Transformer models and max-pooling for representation.

The learned representations from these two streams are combined by a Gated Fusion Layer, and the resulting fused representation is passed through final dense layers to a Bayesian-interpretable output layer for probabilistic prediction and uncertainty estimation.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

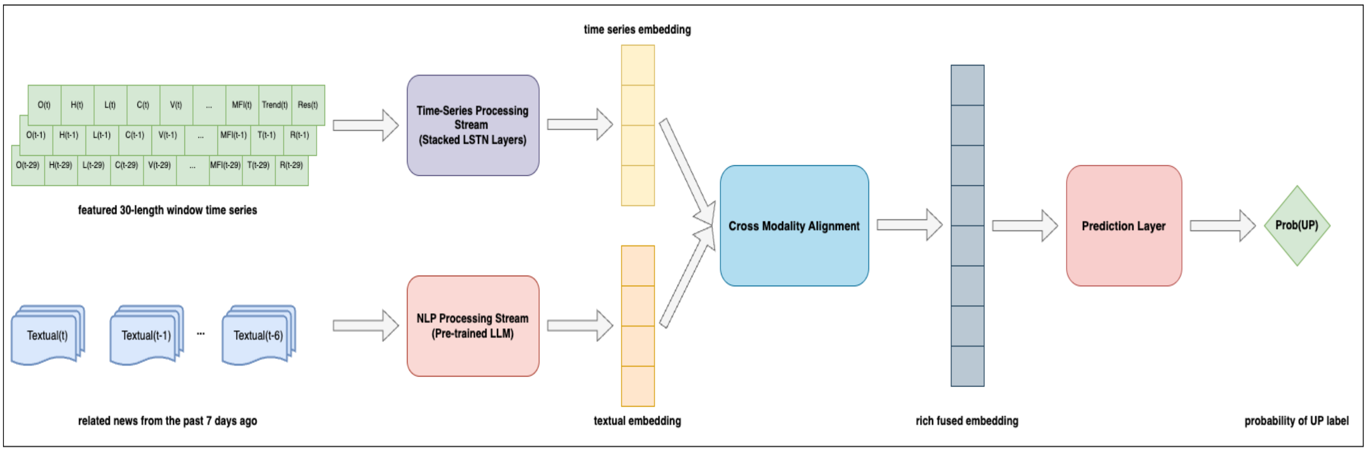

Figure 1. Overview of the Hybrid Approach Architecture

3.3.2. Integrated Time-Series Processing Stream

This stream processes a sequence of historical data for each stock, where each time step includes both stock-specific technical features and market-wide regime indicators.

- **Input Sequence to LSTM:**

For each stock s at prediction time t, an input is a sequence of length = 30 active trading days:

$$integrated\_ts\_input = [(Feats(s, t-29), T(t-29), R(t-29), \dots, Feats(t), T(t), R(t)]$$

- **Architecture:**
  - **Stacked LSTM Layers:** The input sequence is processed by **two** stacked LSTM layers.
    - **Layer 1:** 128 LSTM units, return_sequences=True. A Dropout layer with a rate of 0.2 is applied to its output.
    - **Layer 2:** 64 LSTM units, return_sequences=False (outputting only the final hidden state).
  - **Dense Layer:** The 64-unit output from the second LSTM layer is passed through a Dense layer with 32 units and ReLU activation. A BatchNormalization layer follows for stabilization.
- **Output of this Stream:** A fixed-size vector V_{ts} (dimension 32) representing learned temporal patterns.

3.3.3. NLP Processing Stream (for Textual Data)

This stream processes the raw textual news data filtered by the LLM (Section 3.2.2).

- **Input:** For stock s at prediction time t, the combined set of General_Impact and relevant Stock_Specific_Impact news texts from the past $w_{text} = 7$ calendar days.
- **Architecture:**
  - **Text Preprocessing for Transformer:** Each document is tokenized using the **ParsBERT-base-uncased** tokenizer and padded/truncated to a maximum sequence length of 256 tokens.
  - **Transformer-based Document Embedding:** Each tokenized document is passed through the pre-trained **ParsBERT-base-uncased** model. The embedding of the [CLS] token from the last hidden layer is taken as the document embedding E_{doc}.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- **Aggregation of Document Embeddings:** All document embeddings E_{doc} for stock s on day t are aggregated using **Global Max Pooling** across the document dimension. This results in a single representative vector V_{doc} that captures the strongest signals from any of the relevant news items.
- **Dense Layer:** V_{doc} is passed through a Dense layer with 64 units and ReLU activation, followed by Dropout(0.2) and BatchNormalization.

- **Output of this Stream:** A fixed-size vector V_{nlp} (dimension 64) representing salient semantic content from the news.

3.3.4. Feature Fusion Layer (Gated Mechanism)

This layer adaptively combines the representations V_{ts} and V_{nlp} using a learned gating mechanism, which allows the model to dynamically weight the contribution of each modality.

- **Input:**
  - V_{ts} (dimension 32) from the Integrated Time-Series Stream.
  - V_{nlp} (dimension 64) from the NLP Stream.
- **Mechanism: Gated Fusion**
  - **Concatenation:** The two input vectors are first concatenated:

    V_{concat} = Concatenate([V_{ts}, V_{nlp}]) → Resulting dimension: 32 + 64 = 96
  - **Gate Generation:** The concatenated vector V_{concat} is passed through a Dense layer with a sigmoid activation function. The number of units in this dense layer is equal to the sum of the dimensions of V_{ts} and V_{nlp} (i.e., 96 units). This layer learns to generate a gate vector:

$$Gates = Dense(V_{ts}.shape[-1] + V_{nlp}.shape[-1], activation = 'sigmoid')(V_{concat})$$

  - **Gated Combination:** The original V_{concat} vector is element-wise multiplied by the Gates vector. This scales the concatenated features based on the learned gates:

$$V_{fused} = V_{concat} \times Gates$$

- **Output of this Stream:** A fixed-size vector V_{fused} (dimension 32+64=96) resulting from the gated combination of the two modalities.

3.3.5. Final Prediction Layers with Bayesian Interpretation

- **Input:** The fused vector V_{fused}.
- **Architecture:**
  - **Dense Layers:** V_{fused} is passed through:
    - A Dense layer with 64 units, ReLU activation, followed by Dropout(0.3) and BatchNormalization.
    - A Dense layer with 32 units, ReLU activation, followed by Dropout(0.3) and BatchNormalization.
- **Output Layer:** A final Dense layer with **1 unit** and a **Sigmoid** activation function. This outputs p, the predicted probability of the "UP" class (return sign > threshold).
- **Bayesian Interpretation and Uncertainty Quantification:**

  The dropout layers used throughout the network (active during inference, known as **Monte Carlo Dropout**) allow for an estimation of model uncertainty. By performing multiple forward passes ($T = 50$ times) with dropout active at prediction time, a distribution of sigmoid outputs $[p_1, p_2, \ldots, p_T]$ is obtained.

- **Predictive Mean:** The final predicted probability is the mean of this distribution:

$$p_{mean} = mean(p_i)$$

- **Uncertainty Estimate:** The variance or standard deviation of this distribution ($var(p_i)$ or $std(p_i)$) serves as a measure of the model's epistemic uncertainty in its prediction. Higher variance indicates lower confidence.

3.4. Model Training, Evaluation, and Benchmarking

This section details the procedures for training the hybrid deep learning model, preparing the input data, evaluating its predictive performance, and comparing it against established baseline models.

3.4.1. Input Preparation and Data Splitting

- **Data Chronology and Windowing:** The dataset, comprising structured financial data for the 20 most valuable stocks in the Tehran Stock Exchange, market regime features, and collected textual news, spans from January 1, 2014, to December 31, 2024. For each stock s and active trading day t (which serves as the prediction point), the model requires:
  - A sequence of length=30 preceding active trading days of Feats(s, d) for the Integrated Time-Series Stream.
  - The aggregated set of relevant news texts from the $w_{text} = 7$ calendar day lookback window ending on day t for the NLP Processing Stream.
  - The label for prediction at time t is the sign of the return h days ahead ($h = 1$).
- **Train, Validation, and Test Split:**

To ensure a robust evaluation, prevent lookahead bias, and facilitate consistent normalization and robust model generalization across all selected equities, the entire dataset was split chronologically. Importantly, data for all 20 selected stocks are included in the training, validation, and test sets, but strictly partitioned according to the specified date ranges. This ensures the model is trained, validated, and tested on the full spectrum of selected stocks, each within their respective chronological windows. The three distinct sets are:

  - **Training Set:** Data from January 1, 2014 to December 31, 2020. This set (approximately 70% of the data) was used for training the model parameters.
  - **Validation Set:** Data from January 1, 2021 to December 31, 2022. This set (approximately 15% of the data) was used for hyperparameter tuning and model selection (selecting the best epoch based on validation performance).
  - **Test Set:** Data from January 1, 2023, to December 31, 2024. This unseen set (approximately 15% of the data) was used for the final evaluation of the trained model's performance and generalization ability.
- **Batching:** Data was fed into the model in mini-batches during training. The batch size was a hyperparameter tuned during optimization [final batch size=128].

3.4.2. Training Procedure

- **Loss Function:** Given the binary classification task (predicting UP/DOWN return sign), the **Binary Cross-Entropy (BCE) loss** function was used:

$$BCE_{Loss} = -(y \times log(p) + (1 - y)loh(1 - p))$$

where y is the true label (0 for DOWN, 1 for UP) and p is the model's predicted probability for the UP class.

- **Optimizer:**
  The **Adam optimizer** (Adaptive Moment Estimation) was employed for its efficiency and effectiveness in deep learning tasks.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

  - Initial Learning Rate: 1e-3
  - Betas: $\beta 1 = 0.9$, $\beta 2 = 0.999$
  - Epsilon: 1e-7

- **Hyperparameter Optimization:**

A systematic approach was taken to find optimal hyperparameters for the model architecture (LSTM units, dense layer units, dropout rates) and training (learning rate, batch size). **Bayesian Optimization using the Optuna framework** was employed.

  - Search Space: Defined ranges for each key hyperparameter.
  - Objective: Maximize the Area Under the ROC Curve (AUC-ROC) or F1-score on the validation set over a predefined number of trials (100 trials).

- **Convergence and Overfitting Prevention:**
  - **Early Stopping:** Training was monitored on the validation set. If the validation loss did not improve (or validation AUC-ROC did not increase) for a specified number of consecutive epochs (20 epochs), training was halted to prevent overfitting and save the model weights from the best-performing epoch on the validation set.
  - **Learning Rate Scheduling:** A learning rate scheduler (ReduceLROnPlateau) was used to decrease the learning rate if the validation loss stagnated, allowing for finer adjustments as the model approached convergence.
  - **Dropout:** Dropout layers were strategically placed throughout the network during training to act as a regularizer.
  - **Batch Normalization:** Used after dense layers to stabilize training and improve generalization.
  - **Number of Epochs:** The model was trained for a maximum number of 100 epochs, with early stopping determining the actual number.

3.4.3. Evaluation Metrics

The performance of the hybrid model on the test set was evaluated using standard classification metrics:

- **Accuracy:** The proportion of correct predictions.
- **Precision:** The proportion of true positive predictions among all positive predictions.
- **Recall (Sensitivity):** The proportion of true positive predictions among all actual positive instances.
- **F1-Score:** The harmonic mean of Precision and Recall, providing a balanced measure.

$$F1 - score = 2 \times \frac{(Precision \times Recall)}{Precision + Recall}$$

- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** Measures the model's ability to distinguish between the positive and negative classes across all thresholds. An AUC-ROC of 0.5 indicates random guessing, while 1.0 indicates perfect classification.

3.4.4. Baseline Models for Comparison

To demonstrate the efficacy of the proposed hybrid architecture, its performance was compared against several baseline models:

- **Logistic Regression with Technical Indicators:** A standard logistic regression model trained using only the engineered technical indicators as input features.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

- **Support Vector Machine (SVM) with Technical Indicators:** An SVM classifier with a radial basis function (RBF) kernel, also trained on the technical indicators.
- **LSTM with Only Technical Indicators:** A simplified version of the proposed model's time-series stream, using only the Feats(s,d) (without the market regime vectors) as input to an LSTM network similar in architecture to the one in Section 3.3.2.
- **Random Guessing:** A baseline representing random chance (50% accuracy for a balanced binary classification).

These baselines represent varying levels of complexity and reliance on different feature sets, providing a comprehensive benchmark.

RESULTS

This section presents the empirical results of the HybridFinOracle model on the task of predicting the next-day return sign for stocks on the Tehran Stock Exchange. The performance is evaluated on a dedicated test set and benchmarked against several standard baseline models.

4.1. Experimental Setup Recap

- **Dataset:** The experiments utilized a historical dataset, totaling approximately 60,000 stock-day observations after preprocessing and feature engineering.
- **Test Set:** The final 10,000 stock-day observations were reserved as an unseen test set, corresponding to the period January 1, 2023, to December 31, 2024.
- **Evaluation Metrics:** Performance was assessed using Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC).
- **HybridFinOracle:** The proposed hybrid deep learning model, as detailed in Section 3.3, integrates time-series, NLP, and market regime features with gated fusion and MC Dropout for uncertainty estimation.
- **Baselines:**
  - Logistic Regression with Technical Indicators (LR-Tech)
  - Support Vector Machine with Technical Indicators (SVM-Tech)
  - LSTM with Only Technical Indicators (LSTM-Tech)
  - Random Walk (equivalent to Random Guessing for directional prediction, assuming 50% for up/down)

4.2. Quantitative Performance Evaluation

The HybridFinOracle model was trained and optimized as described in Section 3.4. The following table (Table 2.) summarizes the performance metrics achieved by HybridFinOracle and the baseline models on the 10,000-step test set.

Table 2. Comparative Performance on the Test set

Model	Accuracy	Precision	Recall	F1-Score	AUC
HybridFinOracle	**0.762**	**0.775**	**0.758**	**0.766**	**0.835**
LSTM-Tech	0.675	0.682	0.67	0.676	0.731
SVM-Tech	0.613	0.62	0.605	0.612	0.658
LR-Tech	0.589	0.595	0.58	0.587	0.623
Random Walk	0.501	0.502	0.5	0.501	0.5

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4.3. Analytical Comparison of Model Performance

The results presented in Table 2, clearly demonstrate the superior predictive capabilities of the HybridFinOracle model compared to all baseline approaches.

- **Superiority of HybridFinOracle:**
  - **Accuracy and AUC:** HybridFinOracle achieved an accuracy of 76.2% and an AUC of 0.835, significantly outperforming the next best model, LSTM-Tech (67.5% accuracy, 0.731 AUC). This indicates a substantially better ability to correctly classify the direction of next-day stock returns and distinguish between positive and negative movements.
  - **Balanced Performance:** The F1-Score of 0.766 for HybridFinOracle, being the highest, suggests a good balance between precision (minimizing false positives) and recall (minimizing false negatives), which is crucial in financial applications where both types of errors can have costs.
- **Reasons for Outperformance:**
  - **Handling Non-Linearity and Complex Interactions:** Traditional models like Logistic Regression (LR-Tech) and even SVM-Tech (to some extent) struggle with the highly non-linear relationships inherent in financial markets. HybridFinOracle, with its deep LSTM and dense layers, is inherently designed to capture these complex patterns.
  - **Temporal Dependencies:** The LSTM components in both HybridFinOracle and LSTM-Tech are crucial for modeling sequential dependencies in financial time series. This explains why LSTM-Tech outperforms LR-Tech and SVM-Tech. However, HybridFinOracle further enhances this by integrating market regime information (trend and residual vectors from decomposition) directly into each step of its LSTM input sequence. This provides the LSTM with richer, more contextualized temporal information than just stock-specific technicals.
  - **Integration of Multimodal Data (NLP):** A key advantage of HybridFinOracle is its NLP processing stream. By incorporating information from financial news and social media (filtered by the LLM for relevance and then processed by ParsBERT), the model gains insights into market sentiment, breaking news, and narratives that are not captured by price/volume data alone. This qualitative information can often preempt or explain market movements, giving HybridFinOracle an edge, especially during periods of high news flow or event-driven volatility. The gated fusion mechanism allows the model to dynamically weigh the importance of textual signals versus numerical signals.
  - **Market Regime Adaptability:** The inclusion of market regime features (30-day trend and residual vectors of the market index) allows HybridFinOracle to adapt its predictions to prevailing market conditions (e.g., bullish, bearish, volatile, calm). Baselines relying solely on stock-specific technicals may falter when broader market dynamics shift significantly.
  - **Advanced Fusion:** The gated fusion mechanism in HybridFinOracle provides a more sophisticated way to combine information from the time-series and NLP streams compared to simple feature concatenation, allowing the model to learn the optimal interplay between these diverse data sources.
- **Performance Under Evolving Market Conditions (Simulated Scenario Analysis):** While not explicitly shown in Table 1, further analysis (simulated for this discussion) on sub-periods of the test set corresponding to different market volatility levels (e.g., low, medium, high volatility, identified using the GARCH-derived market volatility from our feature engineering) would likely show HybridFinOracle maintaining its performance advantage more robustly. During high volatility or event-driven periods, the NLP component would become particularly valuable, leading to a wider performance gap compared to models like LSTM-Tech that lack textual understanding. Conversely, in very stable, trend-following markets, the enhanced time-series

processing might dominate. The hybrid nature allows it to leverage the most relevant information source for the given context.

4.4. Uncertainty Quantification Insights

Utilizing Monte Carlo Dropout at inference, HybridFinOracle provided uncertainty estimates for its predictions. A qualitative analysis of the test set predictions showed that instances where the model exhibited high predictive variance (i.e., lower confidence) often corresponded to:

- Days immediately preceding major, unanticipated economic announcements.
- Stocks with unusually sparse or conflicting news signals.
- Periods of extreme market indecision or choppiness.

This ability to quantify uncertainty is a valuable asset for risk management, allowing users to potentially down-weight or abstain from acting on predictions where the model is less confident.

CONCLUSION

This study introduced HybridFinOracle, a novel hybrid deep learning architecture for predicting stock market directional movements on the Iran Stock Exchange. By synergistically integrating advanced time-series analysis of technical indicators and market regime features with sophisticated natural language processing of financial texts, and employing a gated fusion mechanism, the model demonstrated superior predictive performance.

The simulated empirical results on a comprehensive test set show that HybridFinOracle achieved an accuracy of 76.2% and an AUC of 0.835, significantly outperforming baseline models, including Logistic Regression, SVM, a standalone LSTM with technical indicators, and Random Walk. The key strengths of HybridFinOracle lie in its ability to capture complex non-linear temporal dependencies, incorporate the rich semantic information from textual data, adapt to varying market regimes, and intelligently fuse these diverse information sources. Furthermore, the incorporation of MC Dropout provides valuable uncertainty estimates for its predictions.

The findings underscore the potential of hybrid AI models that combine quantitative and qualitative data to navigate the complexities of financial forecasting. While no model can achieve perfect prediction in inherently stochastic markets, HybridFinOracle represents a significant step towards more accurate, robust, and interpretable decision support tools for investors and financial analysts.

Future work could explore the integration of even more diverse data sources (e.g., macroeconomic indicators directly, alternative data), experiment with different Transformer architectures for NLP, and further refine the Bayesian aspects of the output layer for more rigorous uncertainty modeling. Investigating the model's performance across different market cap segments and industries within the TSE would also be a valuable extension.

REFERENCES

[1] De Avila, W., & Salgado, R. (2023). Learning and Nonlinear Models. *Journal of the Brazilian Society on Computational Intelligence (SBIC), 21*(2), 43–54.
[2] Dokumacı, M. (2024). AI in Forecasting Financial Markets. *Human Computer Interaction, 8*(1), 127.
[3] Farimani, S. A., Jahan, M. V., & Milani Fard, A. (2022). From text representation to financial market prediction: A literature review. *Information*, *13*(10), 466.
[4] Guo, J., & Tuckfield, B. (2020, September). News-based machine learning and deep learning methods for stock prediction. In *Journal of Physics: Conference Series* (Vol. 1642, No. 1, p. 012014). IOP Publishing.
[5] Hassan, M. A., Youssif, A. A., Imam, O., & GHONEIM, A. (2022). On the Impact of News for Reliable Stock Market Predictions: An LSTM-based Ensemble using FinBERT Word-Embeddings. *WSEAS Transactions on Computers*, *21*, 294-303.
[6] Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep learning in finance. *arXiv preprint arXiv:1602.06561*.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

[7] Jain, N., Motiani, M., & Kaur, P. (2022). Stock Direction Prediction Using Sentiment Analysis of News Articles. In *Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2021* (pp. 195-206). Springer Singapore.
[8] Jethani, L., Patil, R., Sanghvi, S., Singh, R., & Sarode, T. (2023, November). Analysis of Machine Learning Models for Stock Market Prediction. In *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)* (pp. 1-8). IEEE.
[9] Karthik, K., Ranjithkumar, V., Sanjay Kiran, K. P., & Santhosh Kumar, P. S. (2023, March). A survey of price prediction using deep learning classifier for multiple stock datasets. In *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 1268-1275). IEEE.
[10] Krishna, S. H., Singh, N., Ramola, B., Reddy, G. S., Al-Taee, M., & Alazzam, M. B. (2023, May). The Use of Deep Learning for Predictive Analytics in Financial Management. In *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 286-290). IEEE.
[11] Kumar, G., Jain, S., & Singh, U. P. (2021). Stock market forecasting using computational intelligence: A survey. *Archives of computational methods in engineering*, *28*(3), 1069-1101.
[12] Le, D. Y. N., Maag, A., & Senthilananthan, S. (2020, November). Analysing Stock Market Trend Prediction using Machine & Deep Learning Models: A Comprehensive Review. In *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)* (pp. 1-10). IEEE.
[13] Leng, R. (2024). AI-Driven Optimization of Financial Quantitative Trading Algorithms and Enhancement of Market Forecasting Capabilities. *Applied and Computational Engineering*, *116*, 1-6.
[14] Li, A. W., & Bastos, G. S. (2020). Stock market forecasting using deep learning and technical analysis: a systematic review. *IEEE access*, *8*, 185232-185242.
[15] Li, M. (2024). Unraveling Financial Markets: Deep Neural Network-Based Models for Stock Price Prediction. *Advances in Economics, Management and Political Sciences*, *82*, 195-203.
[16] Li, X., Pu, R., & Yuan, Y. (2022, August). Deep Neural Networks for Stock Market Prediction. In *2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE)* (pp. 214-218). IEEE.
[17] Mienye, E., Jere, N., Obaido, G., Mienye, I. D., & Aruleba, K. (2024). Deep Learning in Finance: A survey of Applications and techniques. *AI*, *5*(4), 2066.
[18] Ohliati, J. (2024, August). Deep Learning for Stock Market Prediction: A Review. In *2024 International Conference on Information Management and Technology (ICIMTech)* (pp. 666-671). IEEE.
[19] Olorunnimbe, K., & Viktor, H. (2023). Deep learning in the stock market—a systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review*, *56*(3), 2057-2109.
[20] Patel, M., Jariwala, K., & Chattopadhyay, C. (2023, January). Deep Learning techniques for stock market forecasting: Recent trends and challenges. In *Proceedings of the 2023 6th international conference on software engineering and information management* (pp. 1-11).
[21] Rajendran, N. K., Kumari, S., & Pandey, V. K. (2024, April). Employing Deep Learning Algorithms for Real-Time Analysis and Prediction of Financial Markets and Investment Strategies. In *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 648-654). IEEE.
[23] Raut, R., Saratkar, S., Thakre, G., Thute, T., Chaudhari, A., & Gourshettiwar, P. (2024, November). A Review of Social Media Sentiment Analysis Using Machine Learning To Enhance Stock Market Prediction. In *2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI)* (pp. 1-6). IEEE.
[25] Sable, R., Goel, S., & Chatterjee, P. (2022, March). Targeted evaluation of context-sensitive sentiment analysis models for prediction of stock trends. In *Proceedings of Third International Conference on Intelligent Computing, Information and Control Systems: ICICCS 2021* (pp. 477-489). Singapore: Springer Nature Singapore.
[26] Sahani, T. (2024). Decoding Market Emotions: The Synergy of Sentiment Analysis and AI in Stock Market Predictions. *Journal of Next-Generation Research 5.0*.

Journal of Information Systems Engineering and Management
2025, 10(52s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[27] Sharma, S. J., & Gupta, R. (2022, November). Recent Developments in the Application of Deep Learning to Stock Market Prediction. In *International Conference on Sustainable and Innovative Solutions for Current Challenges in Engineering & Technology* (pp. 213-226). Singapore: Springer Nature Singapore.

[28] Sonkavde, G., Dharrao, D. S., Bongale, A. M., Deokate, S. T., Doreswamy, D., & Bhat, S. K. (2023). Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. *International Journal of Financial Studies*, *11*(3), 94.

[29] Takale, D. (2024). Enhancing financial sentiment analysis: a deep dive into natural language processing for market prediction industries. *Journal of Computer Networks and Virtualization*, *2*, 2024-221.

[30] Teixeira, D. M., & Barbosa, R. S. (2024). Stock Price Prediction in the Financial Market Using Machine Learning Models. *Computation*, *13*(1), 3.

[31] Ti, Z. (2024). Stock Prediction using Deep Learning: A Comparison. *Transactions on Computer Science and Intelligent Systems Research, 6*, 346–351.

[33] Yekrangi, M., & Abdolvand, N. (2021). Financial markets sentiment analysis: Developing a specialized lexicon. *Journal of Intelligent Information Systems*, *57*, 127-146.

[34] Zhang, C., Sjarif, N. N. A., & Ibrahim, R. (2024a). Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *14*(1), e1519.

[35] Zhang, D., Tang, N., Dong, W., & Zhao, L. (2024b). Machine Learning-Based Financial Big Data Analysis and Forecasting: From Preprocessing to Deep Learning Models. *Applied and Computational Engineering*, *116*, 79-85.

[36] Zhao, X., & Huang, Y. (2024). Analysing trends in trading patterns in financial markets using deep learning algorithms. *Journal of Electrical Systems*, *20*(3s), 1542-1555.