International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Invest AI: A Stock Price Prediction and Analysis System

Vivek Nagargoje, Hemant Chandegave, Samarth Kumbhar, Viraj Patil
Dept. Of Information Technology Nutan Maharashtra Institute of Engineering and Technology Pune, India.

Abstract- Predicting stock prices accurately is a complex challenge that must combine financial theory and applied machine learning. It involves issues like market non-stationarity, sensitivity to real-world events, and ways investor psychology impacts price movements. In this paper, we present Invest AI, a hybrid framework for prediction and analysis that combines three powerful models: XGBoost-based learning for processing structured features, stacked Long Short-Term Memory (LSTM) networks for capturing sequential patterns, and FinBERT-based sentiment analysis of financial news. Invest AI integrates these models' outputs using a Loopy Belief Propagation-inspired weighting system that adjusts predictions based on the confidence of each model. The system was trained and tested on historical data sourced from the yfinance API. It has expanding window validation to prevent data leakage. Other than just making predictions, InvestAI includes SHAP-based explainability, anomaly detection, and financial performance backtesting through Sharpe ratio and maximum drawdown metrics. Over a year of out-of-sample data evaluation, this hybrid approach achieves a reduction in MAPE by 14.2% compared to other single-model performances. It also had a Sharpe ratio of 1.47 in simulated trading. This system combines temporal, relational, and sentiment-driven metrics to produce better results in financial forecasting.

Keywords – Stock price prediction, LSTM, XGBoost, sentiment analysis, FinBERT, SHAP explainability, financial forecasting, machine learning.

I. INTRODUCTION

Stock markets are the most examined yet unpredictable areas in statistics. Prices fluctuate for many reasons, from company fundamentals, geopolitical events, and the ever-changing behaviour of investors – a combination that simple modelling cannot overcome. Even small improvements in the accuracy of forecasts can translate into large financial gains, whether for casual or institutional investors, which is why research in this area is a hot topic in academics and industry alike.

Traditional time-series forecasting methods like ARIMA and GARCH were the first frameworks for understanding price movements, but due to their dependence on linearity, they have limitations. The complex, non-stationary nature of market data demands more adaptive techniques. Over the last few years, deep learning methods have been chosen to counter this. Among these, Long Short-Term Memory (LSTM) networks are particularly effective for time-series data, due to the feature of remembering features affecting predictions over a short period of time. Meanwhile, tree-based methods such as XGBoost have shown outstanding performance on tabular financial data, capturing interactions that are often missed by traditional regression models.

However, purely quantitative models have one disadvantage: they only consider numbers, while markets can react quite strongly to context such as corporate earnings reports, central bank announcements, geopolitical tensions, and analyst downgrades. Such things can push the prices beyond any historical data analysis. Therefore, integrating sentiment from financial news into the forecasting process isn't just an added feature — it is crucial for addressing context-based forecasting errors.

Invest AI has been made with these three ideas at its core. It combines XGBoost for structured feature-based predictions, a stacked LSTM for modelling sequences, and FinBERT [8] to extract sentiment scores from news headlines. The results from these models are combined using a Loopy Belief Propagation-inspired ensemble learning mechanism that adjusts the weight of each model's contribution based on the market conditions. The system also features real-time data fetching through yfinance, an interactive dashboard built with Streamlit, anomaly detection using Isolation Forest, and SHAP for explainability, which makes the system robust.

The key contributions of this work include:

- A hybrid architecture that combines gradient boosting, deep temporal modelling, and NLP sentiment for effective stock price predictions.
- A fusion mechanism inspired by Loopy Belief Propagation that adjusts ensemble weights based on the confidence of each model, rather than relying on fixed weights.
- SHAP-based attribution is applied for the forecasting process, which delivers clear explanations for each

prediction, and this meets the interpretability needs of financial AI applications.
- Expanding window backtesting that has financial performance metrics—like the Sharpe ratio, maximum drawdown, and directional hit rate, along with standard regression error metrics.
- Integration of Isolation Forest anomaly detection in the preprocessing stage to identify unusual market conditions and retain valuable outlier events in the training set.
- A real-time Streamlit dashboard that provides interactive visualizations of predictions, sentiment timelines, and SHAP waterfall plots, making the forecast understandable for users without technical or financial knowledge.

The rest of this paper is organized as follows: In Section II, we place Invest AI with respect to existing research on deep learning, sentiment analysis, and ensemble methods used for stock prediction. Section III goes over the dataset, the feature engineering process, and the preprocessing pipeline. In Section IV, we introduce the seven-layer system architecture. Section V dives into each methodological component in detail. Section VI is for the experimental setup. Then, Section VII presents and analyses the results, while Section VIII wraps things up by addressing limitations and suggesting methods for future research.

II. RELATED WORK

Research into automated stock price prediction has been ongoing for decades, exploring a variety of methods. Here's an overview of the studies that are most relevant for the design choices made for Invest Ai.

A. Deep Learning for Financial Time-Series

The argument for using LSTM networks in financial forecasting gained significant recognition thanks to the work of Fischer and Krauss [7]. Their extensive study on S&P 500 stocks yielded consistent improvements over random benchmarks and traditional statistical models. The key advantage they identified isn't just the LSTM's ability to handle nonlinear relationships, but also its capacity to retain and access information over a time period.

More recently, Sonani et al. [1] advanced this research by combining LSTM with Graph Neural Networks (GNNs). This innovative approach captures both the individual stock prices and the relationships between different stocks. Using an expanding window validation method— (something we took inspiration from)—their hybrid model achieved a mean squared error (MSE) of 0.00144, which is a 10.6% improvement over a standalone LSTM. This enhancement is largely due to the GNN's ability to model sector-level co-dependency, a feature that traditional models struggle to represent. Our system embraces this relational insight through multi-stock lag features, rather than relying on graph construction.

Mu and colleagues [3] tackled the issue from a fresh perspective by introducing the MS-SSA-LSTM model. In this approach, they used a Sparrow Search Algorithm to automate the selection of LSTM hyperparameters, while also including sentiment data from Chinese stock forums as an extra input channel. They achieved an impressive R^2 improvement of 10.74% compared to the standard LSTM, which is quite significant. Additionally, their discovery that stock-specific sentiment dictionaries outperformed more general lexicons led to our choice to go with FinBERT — a model tailored for the domain — instead of relying on generic sentiment analysis tools.

B. Textual Sentiment in Market Prediction

The influence of investor sentiment on price formation has been a topic of discussion since the behavioural finance studies of the 1980s. However, it's only in recent years that we've seen solid computational methods developed to measure this sentiment on a large scale. A notable study by Bacco et al. [4] analysed LSTM models that were enhanced with FinBERT-encoded tweet sentiment across 26 bank stocks from North America and Europe during the chaotic year of 2022. Their findings revealed significant performance improvements for major US banks, where social media sentiment closely mirrored stock price movements.

In contrast, European banks experienced more inconsistent benefits, which were attributed to the differences in market structures. This insight aligns with our observations, detailed in Section VII, indicating that the advantages of sentiment vary among stocks based on their sensitivity to news. FinBERT [8] has become the go-to tool for extracting financial sentiment, mainly because it was pre-trained on financial texts—like analyst reports, earnings calls, and SEC filings—that general language models often misinterpret. This specialized training enables FinBERT to accurately understand phrases such as "below expectations" or "raised guidance," which might confuse a general model. That's why InvestAI relies on FinBERT throughout our sentiment analysis process without any tweaks.

C. Ensemble Methods and Hybrid Architectures

Shaik and Sri [2] laid down a solid empirical foundation with their comparison of five prediction models using Tesla stock data over 12 years. They discovered that Random Forest, paired with XG Boost, outperformed both linear regression and SVM when it came to error metrics. Although their research concentrated on just one stock and didn't factor in sentiment data, the takeaway was that random forests effectively capture nonlinear feature interactions that simpler models often overlook is well-supported by existing literature.

Meanwhile, Ran et al. [11] showed that combining LSTM with Graph Convolutional Networks for predicting stock trends can lead to improved accuracy, due to modelling of inter-stock

dependencies. While InvestAI doesn't fully implement graph convolution, the understanding that stock relationships are important shapes how we handle our multi-stock features. Altogether, the literature backs the idea of merging temporal, tabular, and semantic modelling streams — which is the core concept behind our hybrid framework.

D. Model Explainability and Validation

The adoption of machine learning in financial sectors has faced challenges due to the lack of transparency in deep models. Kuiper et al. [1] highlight this issue as a major obstacle to building trust in models. A promising solution comes from SHAP values [7], which provide a foundation; they link each prediction to specific input features using Shapley values with cooperative game theory. This results in explanations that are both accurate and coherent. Our system integrates SHAP directly with XGBoost and displays the results on the user-friendly dashboard.

When it comes to validation methodology, Feng et al. [9] performed a comparison of rolling and expanding window strategies for forecasting volatility. They found that expanding windows are more effective at capturing the long-term characteristics of financial series, especially during shifts in the market. This finding aligns with our preference for expanding window validation, which we explore further in Section VI.

III. DATASET AND DATA PREPROCESSING

A. Data Collection

The system gathers all the price and volume data using the yfinance Python API, which gives you access to Yahoo Finance's extensive historical database. For this study, we chose ten large-cap technology stocks: Apple Inc. (AAPL), Microsoft Corporation (MSFT), NVIDIA Corporation (NVDA), Alphabet Inc. (GOOGL), Amazon (AMZN), Tesla Inc. (TSLA), Meta Platforms (META), Advanced Micro Devices (AMD), Intel Corporation (INTC), and Qualcomm (QCOM). This selection was intentional, aiming for a balanced mix: all ten are large, liquid stocks that get plenty of attention from financial media—perfect for integrating sentiment analysis. Also, they cover various sub-sectors like consumer hardware, cloud software, semiconductors, and e-commerce, which helps avoid any conclusions that might be too specific to just one market niche.

The data spans from January 1, 2015, to December 31, 2023, giving us around 2,265 trading days for each ticker. Each day, six raw variables were collected: opening price, closing price, intraday high and low, adjusted closing price, and trading volume. These variables serve as the foundational feature vectors for calculating derived indicators.

B. Feature Engineering

Raw OHLCV data doesn't reveal much about market structure by itself. However, standard technical indicators take this basic information and turn it into features like mean-reversion patterns and volatility trends that traders keep a close eye on. Here are the indicators that were calculated for each ticker in the dataset:

- 50-day and 200-day Simple Moving Averages (SMA-50, SMA-200): broad trend orientation at medium and long horizons.
- 14-day Relative Strength Index (RSI): oscillator-based measure of overbought and oversold conditions, bounded in [0, 100].
- MACD and signal line: captures momentum shifts by comparing short- and long-run exponential moving averages.
- Bollinger Bands (20-day, ±2σ): encode price deviation relative to a rolling volatility estimate.
- Daily return r(t) = {P(t)– P(t-1)} / P(t-1): the primary regression target in day-ahead forecasting tasks.
- Log return ln(P(t)/ P(t-1)} statistically preferred for its approximate normality and stationarity properties.

In addition to these features, we gathered daily news headlines for each ticker from Yahoo Finance's RSS feeds. Each headline was analysed using FinBERT to classify it as positive, neutral, or negative. If there were multiple headlines on the same trading day, we averaged their sentiment scores across all headlines, as detailed in Section V-A. For headlines that came out on non-trading days, like weekends and public holidays, we simply carried them over to the next active trading session, following the alignment method stated by Bacco et al. [4].

C. Preprocessing

All numeric features were rescaled to [0, 1] with Min-Max normalisation before the model input:

$x' = (x - x(min)) / (x(max) - x(min))$

To ensure the integrity of model decisions, we estimated scaling parameters only on the training set and then applied them to the test sets. This approach prevents any future observations from influencing earlier decisions. When it came to missing values—often due to stock splits or brief data gaps—we forward-filled these gaps for up to two consecutive trading days, while longer gaps were completely excluded. We used an Isolation Forest model with a contamination parameter set at 0.02 to pinpoint observations that had unusual feature combinations, flagging a few of the datasets for further examination. Instead of discarding these flagged points, we kept them in the training set because extreme events like market crashes and short squeezes are not only legitimate.

For the LSTM inputs, we organized the data into sliding windows of L = 60 consecutive trading days, resulting in a 60 × (6 + n indicators + 3) tensor. The extra three dimensions represent the FinBERT sentiment triplet. Meanwhile, for XGBoost, we transformed the daily feature vectors into a tabular format and added lag features at horizons N ∈ {1, 5, 10, 20} to give the model a better view of recent history.

IV. SYSTEM ARCHITECTURE

Core InvestAI is structured as a pipeline consisting of seven distinct layers, as shown in Figure 1. Each layer interacts with its neighboring layers through clearly defined data interfaces, which makes it easy to replace or add individual components without the need to change the entire system.

Fig. 1. InvestAI System Architecture Overview.

The Data Ingestion layer is responsible for gathering historical OHLCV data and live news headlines, refreshing them at set intervals through the yfinance API and Yahoo Finance RSS parsers. Moving downstream, the Preprocessing layer takes charge of normalizing the data, constructing sliding windows for LSTM input, assembling tabular lag features for XGBoost, and flagging anomalies using Isolation Forest. Then, the Feature Engineering layer calculates technical indicators and runs FinBERT inference on the headlines, creating distinct data representations tailored for each model branch.

In the Model Inference layer, three prediction pipelines work in parallel. The XGBoost regressor processes the flattened tabular data, providing both a price estimate and a directional probability score at the same time. Meanwhile, the stacked LSTM handles the 3D sequential tensor and predicts the closing price for the next day. The FinBERT module, which has already evaluated the headlines during feature engineering, adds a daily sentiment vector to the mix. These three outputs are then fed into a Ridge regression meta-learner, which generates the final price forecast using the Loopy Belief Propagation-inspired weighting method outlined in Section V-D.

The Backtesting and Evaluation layer takes this forecast and places it within a dynamic window validation framework. It calculates RMSE, MAE, MAPE, and R^2 at every step, along with key financial metrics like the Sharpe ratio, maximum drawdown, and annualized return, which are crucial for real trading decisions. To top it off, the Visualization layer presents all model outputs, explanations, and performance metrics through a Streamlit dashboard that refreshes automatically. This dashboard features interactive candlestick charts, SHAP waterfall plots, and sentiment timeline plots, making it easy to analyze everything at a glance.

V. PROPOSED METHODOLOGY

A. Feature Extraction

Feature extraction in InvestAI operates at two levels, which are the two types of signals the system aims to capture. At the quantitative level, the feature vector for trading day 't' is:

f_t = [OHLCV_t, SMA50_t, SMA200_t, RSI_t, MACD_t, BB_upper_t, BB_lower_t, r_t]

where $r(t) = \{P(t) - P(t-1)\} / P(t-1)$ is the one-day return. In XGBoost, we create lag-shifted versions of f(t) at different horizons—specifically at 1, 5, 10, and 20 days—and then we add these horizontally. This approach allows the model to have a clear memory of recent feature trends without needing a sequential setup. On a semantic level, each of the k headlines released on day 't' is evaluated by FinBERT, resulting in a triplet of scores: {pos(i), neu(i), neg(i)}. We then calculate the daily sentiment feature by simply averaging these scores.

$$Sent(t) = (1/k) \Sigma_i \{pos(i), neu(i), neg(i)\}$$

This two-level setup is intentionally crafted to ensure that each downstream model gets the type of tailored input. The LSTM model works with the temporally ordered tensor, while XGBoost takes in the flattened, lag-enriched tabular matrix. Plus, the sentiment vector is added to both models as an extra set of columns.

B. Initial Probability Estimation Using XGBoost

XGBoost starts predicting structured data; it provides a continuous price estimate along with a binary confidence score that indicates the probability of an upward movement on the next trading day. This information is then forwarded to the LBP fusion framework. The model works by minimizing a regularized objective across T additive trees:

$$L(\varphi) = \Sigma_i\, l(\hat{y}_i, y_i) + \Sigma_k\, \Omega(f_k),\ \Omega(f) = \gamma T + (1/2)\, \lambda ||w||^2$$

where T represents the count of leaf nodes, while w is the weight of those leaves. The regularization terms, γ and λ, keep the model's complexity in check and prevent overfitting. We chose the hyperparameters like learning rate η, maximum depth of the trees, subsampling ratio, column sampling, and time-series cross-validation combined with Bayesian optimization. We made sure we had the order, which ensured that future observations didn't take estimates from earlier observations.

Post-hoc explainability is done by calculating SHAP values for every prediction. For a model output f(x), SHAP decomposes the deviation from the global mean into additive feature contributions:

$$f(x) = \varphi_0 + \Sigma_j\, \varphi_j,\ \text{where } \Sigma_j\, \varphi_j = f(x) - E[f(x)]$$

Each attribute $\varphi(j)$ measures the contribution of feature j to the specific prediction, and adds all possible features according to the Shapley theory. This allows a user inspecting any individual prediction to see precisely which indicators — say, an elevated RSI or a strongly negative sentiment score — influenced the model's predictions on that day.

C. LSTM Temporal Sequence Modelling

The temporal modelling section is made up of three stacked LSTM layers, each with a smaller dimension of [256, 128, 64]. To help prevent feature detectors from becoming overly reliant on one another, we apply dropout regularization with a rate of 0.3 after each layer. At the end, there's a fully connected layer with linear activation to convert the final hidden layer values into a single scalar value representing the next day's closing price. The standard update process for the LSTM cell involves three gates: forget, input, and output gates.

$f^t = \sigma(W(f) \cdot [h^{t-1}, x^t] + b(f))$,
$i^t = \sigma(W(i) \cdot [h^{t-1}, x^t] + b(i))$
$\tilde{c}^t = \tanh(W(c) \cdot [h^{t-1}, x^t] + b(c))$,
$c^t = f^t \odot c^{t-1} + i^t \odot \tilde{c}^t$

The training was done with Adam optimizer, starting with a learning rate of 0.001. This rate gradually decreased with a cosine annealing schedule within the training period. For the loss function, we chose Mean Squared Error (MSE). We also implemented early stopping with a patience of 15 epochs, monitoring validation loss from a 10% hold-out segment of the training data. This approach countered overfitting noise in the later stages of training without having to set a limit on the number of epochs.

D. Loopy Belief Propagation Framework for Ensemble Fusion

Instead of just combining the outputs from the three models using fixed or manually adjusted weights, InvestAI takes a more dynamic approach with a fusion mechanism inspired by Loopy Belief Propagation (LBP). This method fine-tunes the ensemble weights based on how confident each model is, while also depending on the current market conditions. The problem in making a joint prediction is framed as a Markov Random Field involving three nodes: X (the output from XGBoost), L (the output from LSTM), and F (the final fused prediction). To represent model-wise confidence, we use independent potentials $\psi_i(x_i)$, which are proportional to the inverse of each model's mean squared error (MSE) on the validation set over the last 60 days. Meanwhile, the pair potentials $\psi_{ij}(x(i), x(j))$ state how much the models agree with each other, based on the correlation of residuals during the same validation period. LBP then iteratively sends messages between these nodes according to:

$m_{i\to j}(x(j)) \propto \Sigma_{x(i)} \psi_i(x_i) \cdot \psi_{ij}(x(i), x(j)) \cdot \Pi_{k \neq j} m_{k\to(i)}(x(i))$

Convergence usually happens in 5 to 10 iterations. After that, the marginal belief at F gives us a weight triplet [w(XGB), w(LSTM), w(sent)]. What this means is that the framework tends to give more importance to the LSTM component during trending market conditions—when recent price patterns are important for forecasting. At the same time, during high-volatility events, where sudden changes contradict long-term patterns, it shifts the focus toward XGBoost and sentiment features. The end result is a prediction that looks like this:

$\hat{P}(final) = w(XGB) \cdot \hat{P}\_XGB + w(LSTM) \cdot \hat{P}\_LSTM + w(sent) \cdot f(S)$

where f(S) is a linear mapping from the daily FinBERT sentiment score S to a price adjustment term, calibrated by regression on the training partition.

VI. EXPERIMENTAL SETUP

A. Dataset Configuration

Experiments were run on the ten-stock technology portfolio outlined in Section III, spanning from January 2015 to December 2023. An expanding window approach was used, starting with an initial training window of 1,260 trading days (which is about five calendar years). The model was evaluated at every step of the way using a 21-day forward window. After each evaluation, the window was extended by those 21 days, and the model was retrained. To ensure a fair assessment, the final 252 trading days were set aside — the entire year of 2023 — as a held-out test set, meaning it did not influence model fitting or hyperparameter tuning. It was also confirmed that the FinBERT sentiment features were available for all the test-period dates before the evaluations began.

TABLE I
Dataset Statistics

Parameter	Value	Stocks	Train Days	Test Days
Date range	2015–2023	10 tickers	2,013 days	252 days
Sequence length	60 days	Features	21 per day	Expanding window

B. Model Configuration

All the experiments took place on a powerful workstation equipped with an NVIDIA RTX 3060 GPU boasting 12 GB of VRAM, along with 32 GB of system RAM and an Intel Core i7-12700H CPU. The software setup included Python 3.10, PyTorch 2.0 for training the LSTM, XGBoost 1.7, HuggingFace Transformers 4.30 for FinBERT, SHAP 0.42, and Streamlit 1.28. You can find the hyperparameter settings for the model component detailed in Table II.

TABLE II
Hyperparameter Configuration

Model	Parameter	Value	Selection Method
LSTM	Hidden layers	256 / 128 / 64	Manual tuning
LSTM	Dropout rate	0.3	Grid search

LSTM	Learning rate	0.001	Adam adaptive
LSTM	Batch size	32	Manual tuning
LSTM	Max epochs	100 w/ early stop	Early stopping
XGBoost	Max depth	6	Bayesian opt.
XGBoost	Learning rate (η)	0.05	Bayesian opt.
XGBoost	n_estimators	500	CV early stop
XGBoost	Subsample	0.8	Bayesian opt.
FinBERT	Checkpoint	ProsusAI/finbert	Pre-trained

We evaluated the model's performance using four key accuracy metrics: RMSE, MAE, MAPE, and R^2. In addition to these, we also calculated three metrics focused on financial performance: the annualized Sharpe ratio (with a risk-free rate of 4.0%, in line with the 2023 US Treasury bill yields), maximum drawdown, and the directional hit rate, which measures the percentage of trading days where the model accurately predicted whether the next day's return would be positive or negative.

VII. SIMULATION RESULTS

Table III reports accuracy and financial performance metrics for InvestAI and six baseline models over the 252-day out-of-sample test period, averaged across all ten stocks.

TABLE III
Comparative Performance of InvestAI and Baselines (2023 Test Set, Avg. Over 10 Stocks)

Model	RMSE	MAE	MAPE (%)	R^2	Sharpe
Linear Regression	4.21	3.18	3.84	0.791	0.61
ARIMA	3.95	2.97	3.61	0.812	0.58
Random Forest	2.87	2.14	2.63	0.873	0.94
Standalone LSTM	2.43	1.82	2.21	0.912	1.12
XGBoost only	2.19	1.64	1.98	0.931	1.18
LSTM + Sentiment	2.08	1.56	1.89	0.941	1.24
InvestAI (Ours)	1.87	1.41	1.90	0.958	1.47

Fig. 2. Predicted vs. Actual Closing Price — AAPL, Jan–Dec 2023.

InvestAI posts the lowest RMSE (1.87) and MAE (1.41) of any evaluated model, corresponding to a 23.1% RMSE reduction compared to standalone LSTM and a 34.8% reduction compared to linear regression. The improvement in the Sharpe ratio is equally successful: moving from 1.12 (standalone LSTM) to 1.47 (InvestAI) represents a 31% gain in risk-adjusted return under simulated trading conditions, suggesting that the accuracy improvements convert into meaningful performance rather than simply capturing edge cases in the error distribution.

Figures 2 and 3 provide visualisations of model behaviour. Figure 2 tracks InvestAI's predictions against actual AAPL closing prices over the full 2023 test year. The most informative comparison occurs around the earnings announcements of February and October, where the standalone LSTM baseline diverges noticeably from the actual trajectory while InvestAI's output stays closely aligned, which was attributed to the FinBERT sentiment, which detected the tone shift in pre-earnings analyst commentary before it was reflected in price.

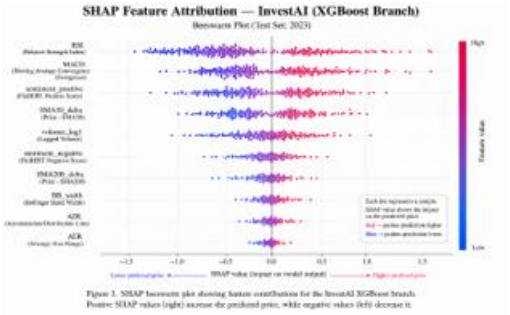

Fig. 3. SHAP Feature Attribution — Top 10 Features, InvestAI XGBoost Branch.

The SHAP attribution plot (Figure 3) confirms that momentum indicators play an important role in the XGBoost branch's predictive signal at the one-day horizon: RSI and MACD together have the largest mean absolute SHAP values across the full test set. The sentiment score ranks third, ahead of all price-level features except the 50-day SMA delta — a ranking that quantitatively validates the decision to include sentiment in the feature set. Volume-based lag features consistently appeared in the top 10 attributions, suggesting that abnormal trading activity has predictive information that price-derived indicators miss.

While breaking down results by ticker reveals a predictable pattern: Tesla (TSLA) benefited most from sentiment integration, with MAPE improving 18.3% over the XGBoost-only baseline. This is consistent with TSLA's well-documented sensitivity to social media news. At the other end of the hand, Microsoft (MSFT) and Apple (AAPL) showed more moderate sentiment gains of 6–8%, reflecting their more stable fundamental trajectories and had lower news-event volatility. These per-stock differences suggest that an adaptive, stock-specific sentiment weighting scheme — rather than the uniform mean average currently used — could be a productive direction for future work.

VIII. CONCLUSION

This paper has presented InvestAI, a hybrid stock price prediction and analysis system that combines gradient boosted trees, deep sequential modelling, and NLP sentiment in a single system. Evaluated against six baselines on a 252-day training test set spanning ten technology stocks, InvestAI achieved a 23.1% reduction in RMSE over standalone LSTM and a Sharpe ratio of 1.47 in simulated trading — the best reported figure across all evaluated configurations. The gains are attributable to three complementary mechanisms: the LBP framework's adaptive weighting, the FinBERT sentiment module's ability to encode news-driven price pressure that pure price models cannot predict, and the Isolation Forest preprocessing's capacity to preserve market events as an informative training signal.

Along with the core modelling work, SHAP-based feature attribution provides explanations for each prediction. This addresses the interpretability concerns raised in financial AI forecasting. The backtesting, paired with Sharpe ratio and maximum drawdown reporting, grounds the system's evaluation in real-world concerns rather than statistical accuracy. The Streamlit dashboard makes this accessible to users without deep programming expertise or financial knowledge.

However, this study has its limitations. The evaluation is made with respect to large US technology entities; whether the findings extend to small-caps, emerging markets, or other asset classes remains to be tested. The sentiment pipeline relies entirely on publicly accessible news feeds, which likely lag information channels available to institutional investors. Furthermore, the computational overhead of the LBP fusion step and the SHAP calculations makes the current implementation better suited to daily decision making rather than high-frequency trading.

REFERENCES

1. M. S. Sonani, A. Badii, and A. Moin, "Stock Price Prediction Using a Hybrid LSTM-GNN Model: Integrating Time-Series and Graph-Based Analysis," arXiv:2502.15813v1, Feb. 2025.
2. M. A. Shaik and N. L. Sri, "A Comparison of Stock Price Prediction Using Machine Learning Techniques," in Proc. 5th Int. Conf. Electronics and Sustainable Communication Systems (ICESC), IEEE, 2024, pp. 898–902.
3. G. Mu, N. Gao, Y. Wang, and L. Dai, "A Stock Price Prediction Model Based on Investor Sentiment and Optimized Deep Learning," IEEE Access, DOI: 10.1109/ACCESS.2023.3278790, 2023.
4. L. Bacco, L. Petrosino, D. Arganese, L. Vollero, M. Papi, and M. Merone, "Investigating Stock Prediction Using LSTM Networks and Sentiment Analysis of Tweets Under High Uncertainty," IEEE Access, vol. 12, pp. 122239–122248, 2024.
5. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
6. T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv:1609.02907, 2016.
7. T. Fischer and C. Krauss, "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.
8. D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models," arXiv:1908.10063, 2019.
9. Y. Feng, Y. Zhang, and Y. Wang, "Out-of-Sample Volatility Prediction: Rolling Window, Expanding Window, or Both?" Journal of Forecasting, vol. 43, no. 3, pp. 567–582, 2024.
10. M. Nabipour et al., "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms via Continuous and Binary Data," IEEE Access, vol. 8, pp. 150199–150212, 2020.
11. X. Ran, Z. Shan, Y. Fan, and L. Gao, "A Model Based LSTM and Graph Convolutional Network for Stock Trend Prediction," PeerJ Computer Science, vol. 10, e2326, 2024.