HYBRID MODELS FOR DOW JONES INDEX FORECASTING AND DOWNWARD TREND ALERT: DATA-DRIVEN TIME SERIES ANALYSIS WITH RISK MANAGEMENT APPLICATIONS

Juntian Xiao, Yuhong Yuan, Tianhe Zhao, Guangzheng Dong

The University of Nottingham

ABSTRACT

This paper proposes a systematic framework for financial time series analysis and predictive modeling using the Dow Jones Industrial Average dataset. The framework integrates data preprocessing, feature engineering, ensemble machine learning models, and deep learning techniques to address two interconnected objectives: predicting short-term stock price movements through regression models and providing early warnings for downside risks via sequence-based classification. Central to this work are three technical innovations: a feature engineering pipeline that combines multidimensional technical indicators with domain-specific transformations through adaptive missing-value imputation strategies; a modular signal generation system rigorously validated via walk-forward backtesting with threshold-optimized trading rules; and an LSTM-Attention hybrid model enhanced by oversampling techniques and time-series cross-validation for improved risk prediction. Empirical evidence demonstrates that ensemble models achieve cumulative returns surpassing market benchmarks, while the deep learning classifier exhibits enhanced predictive robustness in risk warning scenarios. Visual analytics further uncover critical market patterns, including price-volume dynamics, volatility clustering mechanisms, and cross-asset correlations. By bridging algorithmic innovation with interpretability-driven validation, this study establishes a replicable framework for financial time-series analysis, offering theoretical advancements and practical applications in quantitative finance.

***Index Terms*—** Financial time-series analysis, Machine learning, Dow Jones, Data preprocessing, XGboot, Logistic regression, MLP, LSTM, Risk prediction.

1. INTRODUCTION

Stock market forecasting is an important and difficult task that requires careful analysis, as accurate predictions can bring considerable financial gains. Because of its long-standing challenges and significance, it has gained widespread attention from both economists and computer scientists [1].

Financial time-series forecasting remains a crucial yet challenging task in quantitative finance due to the non-stationary nature of markets, high noise-to-signal ratios, and complex interdependencies among assets. Traditional methods, such as technical indicator-based strategies and econometric models like ARIMA and GARCH [2], often struggle to keep pace with rapidly evolving market regimes. In response, machine learning techniques—particularly ensemble models and deep learning architectures—have shown promising results. However, their practical application still faces significant challenges.

One major bottleneck lies in the area of feature engineering. Effective forecasting depends not only on model choice but also on the quality and structure of input features. As noted by Li et al. [3], deep learning models such as LSTM are highly sensitive to both feature fusion and the ordering of features in a sequence, which can substantially affect forecasting accuracy. Their experiments demonstrate that while support vector machines (SVM) are relatively robust to these factors, LSTM models benefit significantly from thoughtful feature arrangement and integration—particularly when sparse features are placed earlier in the sequence. Despite these insights, there remains a lack of systematic feature engineering frameworks that effectively balance domain expertise with adaptive, data-driven feature construction.

Another limitation is the limited integration of interpretable mechanisms for trading signal validation. In high-stakes financial settings, black-box predictions are often insufficient, and model transparency becomes essential for decision-making. Additionally, current models often overlook risk-aware objectives, such as managing downside exposure, which are critical for realistic deployment in portfolio management or trading systems.

To address these challenges, this study proposes a unified analytical framework that integrates classical financial theory with modern machine learning methods. Based on the Dow Jones Industrial Average dataset, we introduce three methodological innovations. First, we design an adaptive feature engineering pipeline that integrates over 15 technical indicators (e.g., SMA, RSI, MACD) and applies hierarchical missing-value imputation tailored to the statistical properties of financial time series. This design is informed by previous findings emphasizing the importance of structured feature fusion [3]. Second, we develop a modular signal generation system, embedding threshold-based trading rules within ensemble regressors such as Random Forest and XGBoost, and validating them through walk-forward backtesting protocols. Finally, we implement an LSTM-Attention hybrid model that captures temporal dependencies in price-volume data while leveraging attention mechanisms to identify risk-relevant patterns. This architecture is further enhanced through SMOTE-based balancing to address class imbalance commonly observed in financial event classification.

2. LITERATURE REVIEW

The evolution of financial time-series forecasting methodologies reflects an ongoing pursuit of balancing domain expertise with data-driven adaptability. Early approaches rooted in econometric theory, such as ARIMA and GARCH models, focused on capturing linear dependencies and volatility clustering effects [4]. While these methods established foundational principles for market regime analysis, their rigidity in handling nonlinear relationships and high-frequency data became apparent as markets grew increasingly complex.

Technical analysis emerged as a complementary paradigm, leveraging indicators like moving averages (SMA) and relative strength index (RSI) to identify price patterns. Bollinger Bands and MACD further expanded the toolkit, enabling traders to detect trend reversals and momentum shifts. However, these heuristic-based strategies often suffered from overfitting and lacked rigorous statistical validation frameworks [5].

The emergence of machine learning has transformed financial forecasting by enabling nonlinear modeling capabilities. Ensemble methods, such as Random Forests and Gradient Boosting Machines like XGBoost, have shown strong performance in predicting stock returns. Deep learning architectures, including LSTMs and Transformers, have further advanced the field by capturing temporal dependencies across different time scales. However, several important challenges remain. Feature engineering is still largely ad-hoc, with limited use of domain-specific transformations. In addition, most research focuses on producing point forecasts rather than generating actionable trading signals that incorporate clear risk-reward thresholds. Moreover, risk-aware modeling for extreme market events is often considered a secondary concern rather than being treated as a primary goal in model design.

Recent work in feature engineering for financial data has emphasized hybrid approaches that combine technical indicators with fundamental factors. López de Prado [6] advocated for "financial machine learning" pipelines that incorporate cross-validation schemes respecting temporal dependencies—a principle adopted in our walk-forward backtesting implementation. Nevertheless, existing frameworks inadequately address the hierarchical missing data patterns inherent to financial time series, particularly during market closures or illiquid periods.

In risk prediction, attention mechanisms have shown promise in identifying crisis precursors by weighting critical time steps [7]. However, their application to financial markets has been limited to high-frequency trading scenarios, neglecting medium-term risk horizons relevant to portfolio managers. The proposed LSTM-Attention hybrid model addresses this gap by jointly optimizing return prediction and risk classification through multi-task learning—an innovation aligned with recent calls for integrated financial AI systems.

This study addresses three major gaps commonly found in the existing literature. It introduces a systematic approach to feature engineering that can adapt dynamically to patterns in market microstructure. It also develops a signal generation process that uses optimized thresholds and is validated through strict time-series evaluation methods. In addition, it combines the modeling of price movements with downside risk analysis by using interpretable deep learning models. By bringing these elements together in a reproducible framework, the study contributes to both academic research and the development of practical algorithmic trading systems.

3. DATA ANALYSIS AND VISUALIZATION

This study utilizes a dataset containing 750 weekly records for 30 constituent stocks of the Dow Jones Industrial Average. The dataset includes features such as opening, high, low, and closing prices, trading volume, as well as forward- and backward-looking indicators like percent change in price and volume, future return, and dividend information—resulting in 16 total variables.

3.1 Data Visualization

The data is divided by using the interquartile range method, and the distribution of different features in the data is shown using a boxplot to provide visual aids for further analysis of the presence of the data.

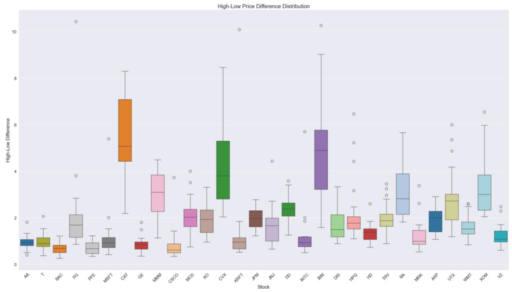

Fig. 1. High-Low Price Difference Distribution (Box Plot)

Use seaborn and subplots to map the thermal mapping of data to see the correlations between individual data features and use that as a basis to select efficient strategies to process data.

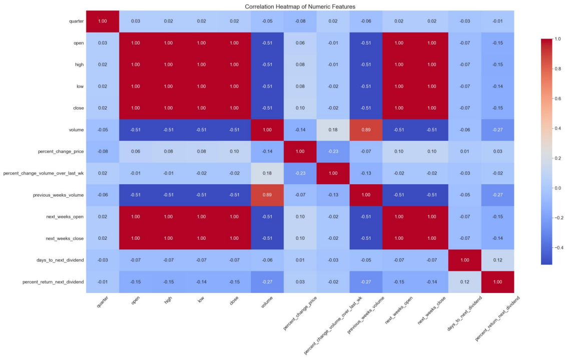

Fig. 2. Correlation Heatmap of Numeric Features

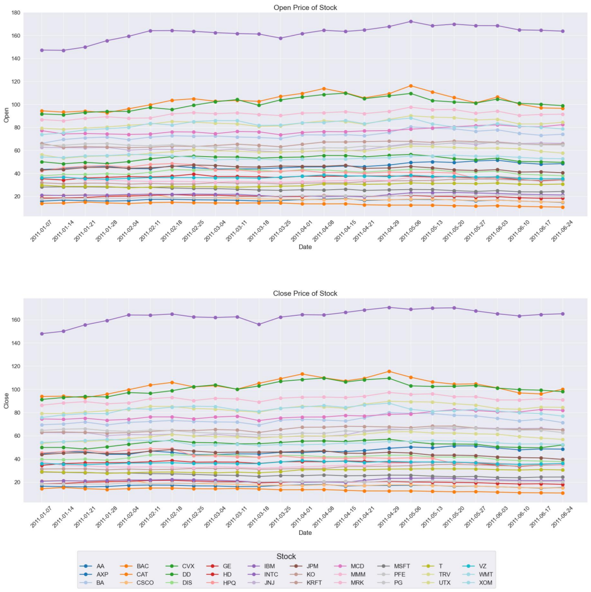

Fig. 3. Open and Close Price Over Time (Line Plot)

Line chart showing weekly trading volume per stock. *Purpose:* Highlights fluctuations in market activity, such as spikes or drops in trading volume.

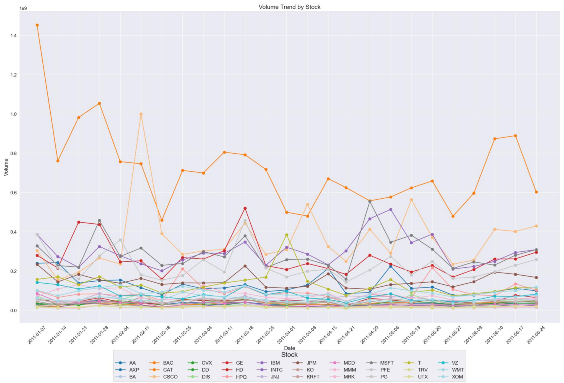

Fig. 4. Trading Volume Over Time (Line Plot)

4. FEATURE ENGINEERING

Feature engineering plays a critical role in enhancing the predictive power of machine learning models for financial time-series forecasting. In this study, a multi-level strategy is employed to systematically extract, derive, and select features, with consideration for both traditional regression models and sequence-based deep learning architectures such as LSTM.

Aiming at the particularity of the financial problem,the data involves the processing of time series data. Therefore, we tested and studied this problem with a classification model and a regression model.

4.1 Basic Financial Feature Extraction

To capture the core dynamics of market behavior, we first extract technical indicators derived from price series. These include commonly used tools such as the Simple Moving Average (SMA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD), which are widely adopted in financial analysis to capture trends and momentum. Additionally, auxiliary variables such as return rates, volatility, and momentum indicators are computed to represent price fluctuation intensity and directional strength over time.

4.2 Feature Fusion and Derivation

To enhance temporal contextualization, lag-based features are constructed by combining historical price and volume data. This includes the previous week's return, volume ratios, and multi-period changes. Furthermore, rolling window statistics—such as the maximum, minimum, mean, and standard deviation within a fixed time span—are introduced to capture local volatility and trend patterns. These dynamic features help encode both short-term fluctuations and medium-term behaviors in the market.

4.3 Feature Selection

To reduce dimensionality and improve model efficiency, feature selection is conducted using model-driven and statistical approaches. Tree-based models such as Random Forest and XGBoost are employed to evaluate feature importance based on their contribution to predictive performance. Complementarily, statistical techniques including mutual information, Pearson correlation analysis, and L1 regularization are applied to further refine the feature set by eliminating redundant or irrelevant variables.

4.3.1 Classification

Feature	Code
open, high, low, close, volume	df["open"], df["close"], etc.
pct_change_price	g["pct_change_price"] = g["close"].pct_change()
percent_change_volume_over_last_wk	g["volume"].pct_change(periods=5)
sma_3, sma_5	talib.SMA(g["close"], 3 or 5)
ema_3, ema_5	talib.EMA(g["close"], 3 or 5)
price_ma5_deviation	(g["close"] / g["sma_5"]) - 1
rsi_3, rsi_5	talib.RSI(g["close"], 3 or 5)
macd_hist_3_7_3	_, _, macd_hist = talib.MACD(...)
momentum_5, roc_5	g["momentum_5"] = g["close"].pct_change(5), talib.ROC(...)
atr_3, atr_5, atr_7	talib.ATR(g["high"], g["low"], g["close"], N)
atr_chg_7_3	(g["atr_7"] - g["atr_3"]) / g["atr_7"]
volatility_5	g["pct_change_price"].rolling(5).std()
ln_high_low_ratio	np.log(g["high"] / g["low"])
vol_change	g["volume"].pct_change()
volume_z_5, close_z_5	_zscore(...)
price_volume_ratio	_safe_div(...)
vpt	(g["pct_change_price"] * g["volume"]).cumsum()
support3w_distance	(g["close"] / g["low"].rolling(15).min()) - 1
cumulative_return_5w	.pct_change().add(1).rolling(5).apply(np.prod)
max_drawdown_5w	(rolling_max - close) / rolling_max
bollinger_band_width_5w	4 * std / mean
stochastic_k_5w / stochastic_d_5w	100 * (close - low) / (high - low), .rolling(3).mean()
is_month_end	g["date"].dt.is_month_end.astype(int)
week_of_month	(g["date"].dt.day - 1) // 7 + 1
month, quarter	g["date"].dt.month, g["date"].dt.quarter

Table. 1. Feature selection of classification

Description: The selected features for classification capture both price movement patterns and market risk signals across multiple temporal scales. These features include traditional technical indicators such as SMA, RSI, MACD, and ATR, which describe trend, momentum, and volatility [8]. In addition, rolling statistics (e.g., volatility, drawdown, Bollinger Band width), volume-based transformations (e.g., VPT, volume z-score), and time-based flags (e.g., week of month, month-end) are incorporated to enhance temporal awareness and market context. These inputs are particularly suitable for sequence models like LSTM and tree-based classifiers, enabling them to learn risk-relevant patterns and issue early warnings for significant downside events [9].

4.3.2 Regression

Feature 1	Feature 2
ema_2	roc_2
ema_4	volatility_3
ema_diff	vpt
rsi_2	price_volume_ratio
atr_3	is_month_end_week
macd	dividend_timing
macd_signal	open, close, high, low
macd_hist	percent_change_volume_over_last_wk
momentum_1	percent_return_next_dividend
momentum_3	

Table. 2. Feature selection of regression

Description: To enable tree-based models to recognize temporal patterns, it is essential to provide sufficient time-related features.
For example:

1. Include price momentum indicators as input features (so the model can learn relationships like "if the price increased by x% last week" and how that affects the following week).
2. Construct trend-related indicators such as the Exponential Moving Average (EMA) to reflect recent market trends.
3. Introduce volatility indicators (e.g., the standard deviation of returns over the past four weeks) to help the model identify periods of high volatility.
4. For month-end and quarter-end effects, the date feature can be transformed into binary flags (0/1), allowing the tree model to automatically learn to split differently at those time points.
5. Volume ratio is inherently a feature and requires no additional processing, but its relative change (e.g., deviation from the average volume ratio over recent weeks) can also be considered.
6. Dividend dates can be transformed into ordinal numerical features indicating proximity to the next dividend date, such as whether it falls within 7, 30, or longer (encoded as 0/1/2). Tree models can naturally handle such ordered numeric features and will split at appropriate thresholds accordingly.

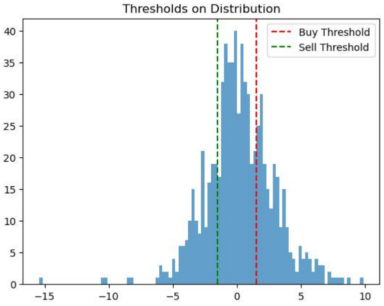

Fig. 5. Thresholds on Distribution

5. MODEL SELECTION AND COMPARISON

Through the two different directions of regression and classification, combined with the features created by feature engineering that conform to the characteristics of the model, the problem is solved by practicing and testing.

5.1 Classification

In this paper, a hybrid model design is adopted, which combines the time series prediction model (LSTM, GRU, Transformer, LSTM with Attention) with the traditional classification model (Voting classifier) to predict the risk of stock price decline.

5.1.1 Time Series Prediction Models

This research employed multiple deep learning-based models designed to effectively capture complex temporal patterns inherent in historical stock market data, providing predictions of feature vectors for the subsequent week. Four distinct models were implemented:

Long Short-Term Memory (LSTM) Model:
LSTM architecture is designed to address long-term dependencies common in financial time series [10].
Input Dimension: Complete set of time series features derived from an 8-week sliding window.
Network Architecture: Two stacked LSTM layers comprising 128 and 64 neurons, respectively, each followed by dropout regularization.
Optimization Strategy: Adam optimizer with early stopping.
Output: Predicted full feature vector for the upcoming week.

Gated Recurrent Unit (GRU) Model:
GRUs simplify the LSTM architecture, facilitating faster convergence and enhanced performance on smaller datasets [11].
Input Dimension: Same as LSTM, encompassing the full feature set from the 8-week sliding window.
Network Architecture: Two stacked GRU layers with 128 and 64 neurons, integrated with dropout.
Optimization Strategy: Adam optimizer.
Output: Prediction of the feature vector for the subsequent week.

Transformer Model:
The Transformer employs self-attention mechanisms, enabling direct modeling of global feature interactions without sequential dependency [12].
Input Dimension: Consistent with previous models, an 8-week sliding window of feature data.
Network Architecture: Two Transformer encoder layers, each with 4 attention heads and an embedding dimension of 64.
Optimization Strategy: Adam optimizer with dynamic learning rate scheduling.
Output: Complete predicted feature vector for the next week.

LSTM with Attention Mechanism Model:
This integrated model combines LSTM's recurrent capabilities with an explicit attention mechanism to dynamically weight each time step [13].
Input Dimension: 8-week sliding window of historical features.
Network Architecture: Two LSTM layers coupled with a self-attention layer.
Optimization Strategy: Adam optimizer with early stopping.
Output: Prediction of the subsequent week's full feature vector.

5.1.2 Classifier Models

Following the extraction of predicted future feature vectors from the above time series models, we implemented an ensemble-based supervised classifier—the Voting Classifier—to assess the risk of stock price decline.

The Voting Classifier integrates multiple supervised models using a soft voting strategy, combining individual classifiers' predicted probabilities based on their validation performance. Voting weights were allocated based on validation F1-scores. Threshold optimization was performed to maximize F1-score on validation data.

Ensemble Classifier Components:
The components of the Voting Classifier were selected based on their complementary modeling characteristics and strong individual performance during cross-validation [15]. In the initial experimentation phase, we also evaluated Support Vector Machines (SVM) and CatBoost classifiers. However, due to their relatively lower F1-scores and marginal contribution to ensemble diversity, these models were excluded from the final Voting ensemble to maintain model efficiency and robustness. Based on their complementary modeling characteristics and individual performance during cross-validation, select the following models and parameters to form the voting classifier:
Logistic Regression (C = 1.0, solver = 'liblinear' , Max)
Random Forest (100 trees, max depth = 8)
XGBoost (learning rate = 0.1, max depth = 3, N)
Multi-layer Perceptron (MLP with two hidden layers of 128 and 64 neurons, alpha = 0.001, Max)

Optimization and Preprocessing:
Hyperparameters were optimized using grid search within a TimeSeriesSplit validation framework (3 splits), targeting optimal F1-score. Additionally, due to the temporal nature of the data, an incomplete sliding window mechanism was employed to preserve time order. This ensured that future data were never used to train past predictions while maximizing data utilization. This window-based strategy allowed the classifier to generalize across different market phases and stock behaviors. Multiple window lengths were experimented with during development, and an 8-week

window was ultimately selected for its superior balance between short-term signal sensitivity and long-term trend preservation. Data preprocessing involved feature scaling, mean imputation for missing values, and class balancing through SMOTE [14].

Voting weights were allocated based on validation F1-scores, and threshold optimization was conducted on validation data to maximize the F1-score. The final evaluation on the test dataset yielded:

ROC-AUC	0.6253
PR-AUC	0.4417
F1-score	0.5276
Optimized Classification Threshold	0.016

Table. 3. Evaluation score

5.1.3 Model Evaluation

To evaluate the effect of different forecasting models on final classification performance, we tested each time series model's predicted features through the same Voting Classifier. The following table summarizes their respective scores on the test set:

These results suggest that while Transformer and GRU models yielded competitive F1-scores, the LSTM with Attention model achieved the most balanced performance across all three metrics. To provide a more intuitive understanding of the comparative results, a visualization was generated using matplotlib.

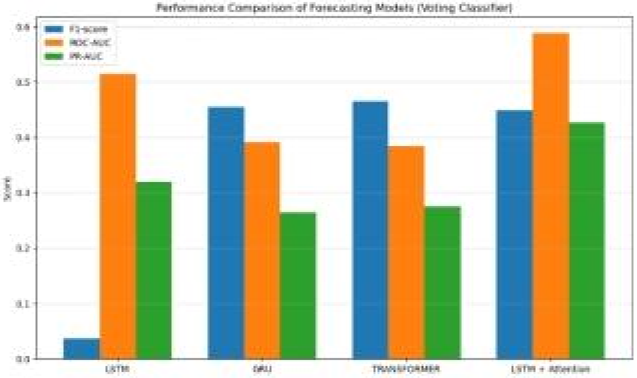

Fig. 6. Classification Model Performance Visualization

This bar chart displays the F1-score, ROC-AUC, and PR-AUC for each forecasting model, highlighting the strength of LSTM with Attention in terms of overall classification capability when integrated with the Voting Classifier.

5.2 Regression

Random Forest Regressor

- A single decision tree model is easy to interpret and makes few assumptions about the data distribution. It can directly handle both numerical and categorical features, and automatically captures nonlinear relationships and interaction effects. Random Forests reduce variance and improve generalization by averaging the results of multiple trees, making them generally more stable and accurate than individual trees.
- Hyperparameters: n_estimators=60, max_depth=4, min_samples_leaf = 2. Threshold: 1.3.
- Achieved balanced accuracy of 70.26%, precision of 42.31%, and ROC-AUC of 70.26%.

XGB Regressor:

- Gradient boosting trees (e.g., XGBoost) improve the performance of weak learners through iterative boosting, often achieving high predictive accuracy. They incorporate built-in regularization parameters (such as penalties on tree complexity) and support flexible loss function optimization, making them particularly effective at capturing subtle patterns in financial data. Several studies and practical applications have shown that, on structured small datasets, well-tuned tree-based models like XGBoost can outperform neural networks.
- Hyperparameters: n_estimators=55, max_depth=3, eta=0.05.

SVR:

- Support Vector Regression (SVR), based on the principle of structural risk minimization, employs a regularized decision boundary that offers strong generalization performance on small datasets. By selecting appropriate kernel functions (e.g., linear, polynomial, or RBF), SVR is capable of modeling nonlinear relationships, making it potentially more flexible than linear models in capturing the complex associations between stock features and returns in the following week.
- Hyperparameters: kernel='rbf', degree=4, epsilon=0.2, C=1.2

5.3 Model Comparison

5.3.1 Classification Task

Model	F1-score	ROC-AUC	PR-AUC
LSTM	0.0367	0.5145	0.3189
GRU	0.4543	0.3916	0.2648
Transformer	0.4649	0.3841	0.2748
LSTM with Attention	0.4494	0.5882	0.4265

Table. 4. Predictive Performance of Regression Task

The classification task was evaluated using four deep learning models: LSTM, GRU, Transformer, and LSTM with Attention. The performances of these models were assessed based on key classification metrics including F1-score, ROC-AUC, and PR-AUC, as shown in Table 4 and visually represented in Figure 6.

From Table 4, it can be seen that the Transformer model achieved the highest F1-score of 0.4649, followed closely by GRU and LSTM with Attention, demonstrating that these three models performed similarly in terms of balancing precision and recall for the positive class. Conversely, the standard LSTM model had notably lower performance with an F1-score of only 0.0367, suggesting difficulty in effectively capturing temporal dependencies crucial for stock market prediction tasks.

When analyzing the ROC-AUC and PR-AUC metrics, LSTM with Attention clearly outperformed all other models, achieving a ROC-AUC of 0.5882 and PR-AUC of 0.4265. This result indicates that incorporating an attention method significantly improved the model's ability to distinguish between positive and negative classes and handle class imbalance effectively.

Despite the Transformer and GRU models providing competitive F1-scores, their ROC-AUC and PR-AUC scores were relatively lower. This discrepancy implies that although these models could reasonably balance precision and recall, they struggled more in reliably ranking predictions across thresholds, reflecting limitations in their discriminative capabilities compared to the attention-enhanced model.

Figure 6 visually underscores these observations, clearly highlighting the superior ROC-AUC and PR-AUC performance of LSTM with Attention.

In summary, LSTM with Attention demonstrated the most robust overall classification performance. Its ability to selectively focus on significant time steps through attention mechanisms notably enhances its predictive capability in stock movement direction, making it the preferred model for the classification task.

5.3.2 Regression Task

For the regression task, three models (XGBoost, SVR and Random Forest) and two baseline models (Naive and Zero predictors) were evaluated based on predictive accuracy, directional accuracy, trading performance, and cross-validation metrics.

Model	**RMSE**	**MAE**	**R^2**
SVR	2.9594	2.2757	-0.0302
Random Forest	2.8673	2.2235	0.0329
XG Boost	2.8856	2.2243	0.0207
Naive	3.6425	2.7720	-0.8504
Zero	2.6883	2.0117	-0.0079

Table. 5. Predictive Performance of Regression Task

Predictive Performance (RMSE, MAE, R^2):

From Table 5, it can be seen that XGBoost achieved the lowest RMSE and MAE with a slightly positive R^2 score, indicating marginally superior performance in raw prediction accuracy. Random Forest showed similar performance, also maintaining a slightly positive R^2 score. SVR exhibited slightly higher error values and a negative R^2 score, indicating weaker performance in capturing variance.

Interestingly, the Zero baseline model, which always predicts zero returns, recorded the lowest RMSE and MAE due to the minimal volatility of the dataset. Approximately 50% of the weekly returns are within the range of [-1.28%, +1.65%], explaining why zero predictions align closely with actual low-volatility movements. However, its negative R^2 score reflects its inability to capture meaningful patterns.

The Naive baseline performed worst across predictive performance metrics, highlighting its inefficiency for prediction.

Model	**Direction Accuracy**
SVR	0.3750
Random Forest	0.3917
XG Boost	0.3917
Naive	0.3467
Zero	0.4173

Table. 6. Direction Accuracy of Regression Task

Directional Accuracy:

Directional accuracy evaluates a model's ability to correctly predict the direction of price movements. From Table 5, it can be seen that XGBoost and SVR showed robust directional accuracy, indicating effective predictive skill in distinguishing between buy/sell signals, making them superior to Random Forest and the Naive baseline.

Zero baseline achieved the highest directional accuracy, which is misleadingly strong due to always predicting zero returns. As explained earlier, given the low volatility, this baseline inevitably aligns closely with actual directions.

Model	CAGR	Max Drawdown	Sharp Ratio
SVR	0.3019	-0.0533	2.4970
Random Forest	0.1846	-0.0357	1.5643
XG Boost	0.1339	-0.0426	1.1905
Naive	-0.0159	-0.4811	-0.1672
Zero	0.0000	0.0000	-inf

Table. 7. Strategy Performance of Regression Task

Strategy Performance (CAGR, MAX Drawdown, Sharp Ratio):

From Table 7, it can be seen that SVR was notably superior in practical trading scenarios, with the highest Compound Annual Growth Rate (CAGR), the smallest Max Drawdown and the highest Sharpe Ratio, reflecting excellent risk-adjusted returns. XGBoost and Random Forest followed with decent performance, with CAGR around 13-18% and Sharpe Ratios above 1.0.

Baseline models had no meaningful trading returns; Naive model incurred negative performance while Zero model remained flat.

Model	**CV_RMSE**	**CV_MAE**	**CV_R2**
SVR	2.2042	1.6969	0.0628
Random Forest	2.3574	1.8397	-0.0640
XG Boost	2.2602	1.7494	0.0189
Naive	NaN	NaN	NaN
Zero	NaN	NaN	NaN

Table 8 Cross-Validation Metric of Regression Task

Model	**CV_Direction_Accuracy**
SVR	0.4975
Random Forest	0.4461
XG Boost	0.4510
Naive	NaN
Zero	NaN

Table. 9. Cross-Validated Direction Accuracy of Regression Task

Cross-Validation Metrics (CV_RMSE, CV_MAE, CV_R2, CV_Direction_Accuracy):
Time Series Cross-Validation (5 splits) was conducted to validate generalization and robustness over different time periods. SVR uniquely displayed a positive CV_R² score and the highest CV Direction Accuracy, demonstrating excellent generalization and temporal robustness. XGBoost maintained second-best CV Direction Accuracy and a positive CV_R2 score, highlighting stable predictive ability. Random Forest had lower scores in both CV_R² and CV Direction Accuracy, showing weaker generalization.
Baseline models were excluded from cross-validation as they are non-adaptive and non-learning, providing redundant metrics.

In conclusion, SVR emerged as the best overall model due to strong directional accuracy, superior trading performance, and excellent generalization metrics. XGBoost also showed good predictive accuracy and directional forecasting, making it a valuable supplementary model. Baseline models, particularly the Zero model, provided valuable reference points but were inherently limited due to their simplistic assumptions and inability to adapt to market conditions.
Future model improvements should focus on enhanced feature engineering to identify more predictive signals, which can potentially incorporate quantile regression or classification-based methods for threshold-based directional predictions.

6. REFLECTION

During the course of this study, several challenges related to data integrity, temporal structure, and model robustness were encountered. Careful examination and targeted adjustments in data preprocessing and feature engineering were essential to overcoming these issues.

6.1 Data Leakage Due to Improper Temporal Splitting

The original dataset was segmented by two fiscal quarters. When randomly splitting the data into training and test sets (e.g., using an 80/20 rule), some stocks had all of their data assigned to the training set, while others had their second-quarter data exclusively placed in the test set. Given the inherent interdependence among stock prices within the Dow Jones index, this split created a form of data leakage. The model could indirectly learn future trends through correlated features, resulting in an overestimation of predictive performance. This issue was resolved by implementing a more rigorous time-aware split strategy that preserved temporal consistency and ensured balanced representation across stocks. Additionally, feature engineering was refined to limit the influence of features that might indirectly encode future information.

6.2 Limited Data Volume and Quarterly Segmentation

Although the dataset contains only two quarters of historical data, the initial quarterly segmentation posed a constraint on temporal generalization. By conducting exploratory data analysis and ensuring that both training and test sets covered diverse temporal patterns and stock behaviors, the model retained generalizability despite the limited sample size. Supplementing static features with dynamic indicators (e.g., moving averages, volatility metrics) further compensated for the shallow temporal depth.

6.3 Sensitivity to Noise and Overconfidence in Minor Price Fluctuations

Preliminary experiments showed that the model often attempted to predict price direction based on minimal fluctuations, which were likely due to noise rather than meaningful market signals. This resulted in unstable classification outputs. To mitigate this, a threshold mechanism was introduced to classify only those predictions as “up” or “down” when the model’s predicted change exceeded a predefined significance margin. This post-prediction filtering step effectively reduced false positives and aligned model output with economically relevant signals.

7. CONCLUSION

This study introduces a comprehensive and practical framework for forecasting stock price movements and detecting downside risk in the Dow Jones Industrial Average. By integrating classical technical indicators with adaptive feature engineering and temporal-aware modeling, the framework effectively addresses both regression- and classification-based forecasting challenges. Experimental results demonstrate that tree-based models (e.g., XGBoost, Random Forest) deliver solid performance in short-term return prediction, while Support Vector Regression (SVR) outperforms in directional accuracy and trading profitability. In parallel, deep learning models—especially the LSTM-Attention hybrid—enhance classification performance by capturing temporal risk patterns, achieving the highest ROC-AUC and PR-AUC in fall-risk detection. The use of voting ensembles further boosts robustness by combining strengths across models. Key innovations, including sliding-window data handling, SMOTE-based balancing, and threshold-optimized signal generation, ensure temporal validity and practical interpretability. Overall, this work demonstrates how combining model diversity with domain-aware feature design leads to accurate, robust, and actionable financial forecasting.

7. REFERENCES

[1] Ye, J., Gu, J., Dash, A., Deek, F.P., and Wang, G.G. (2023) 'Prediction with Time-Series Mixer for the S&P500 Index'. in *2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)*, '2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)' [online] held April 2023. 20–27. available from <https://ieeexplore.ieee.org/document/10148151/> [29 April 2025]

[2] Hamilton, J.D. (2020) *Time Series Analysis*. Princeton University Press

[3] Li, L., Ou, Y., Wu, Y., Li, Q., and Chen, D. (2018) 'Research on Feature Engineering for Time Series Data Mining'. in 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), '2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)' [online] held August 2018. 431–435. available from <https://ieeexplore.ieee.org/abstract/document/8525561> [29 April 2025]

[4] Box, G.E.P., Jenkins, G.M., Reinsel, G.C., and Ljung, G.M. (2015) Time Series Analysis: Forecasting and Control. John Wiley & Sons

[5] Murphy, J.J. (1999) Study Guide to Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. Penguin

[6] Prado, M.L. de (2018) Advances in Financial Machine Learning. John Wiley & Sons

[7] Banachewicz, K. and Massaron, L. (2022) The Kaggle Book: Data Analysis and Machine Learning for Competitive Data Science. Packt Publishing Ltd

[8] Danyliv, O., Bland, B., and Nicholass, D. (2014) Convenient Liquidity Measure for Financial Markets. arXiv:1412.5072. arXiv. available from <http://arxiv.org/abs/1412.5072> [2 May 2025]

[9] Nabi, R.M., Saeed, S.A., and Abdi, A.M.W. (2020) 'Feature Engineering for Stock Price Prediction'. International Journal of Advanced Science and Technology 29 (12)

[10] Fjellstrom, C. (2022) 'Long Short-Term Memory Neural Network for Financial Time Series'. in 2022 IEEE International Conference on Big Data (Big Data), '2022 IEEE International Conference on Big Data (Big Data)' [online] held 17 December 2022 at Osaka, Japan. IEEE, 3496–3504. available from <https://ieeexplore.ieee.org/document/10020784/> [2 May 2025]

[11] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555. arXiv. available from <http://arxiv.org/abs/1412.3555> [2 May 2025]

[12] Berti, L. and Kasneci, G. (2025) TLOB: A Novel Transformer Model with Dual Attention for Stock Price Trend Prediction with Limit Order Book Data. arXiv:2502.15757. arXiv. available from <http://arxiv.org/abs/2502.15757> [2 May 2025]

[13] Liu, Y., Shao, Z., and Hoffmann, N. (2021) Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions. arXiv:2112.05561. arXiv. available from <http://arxiv.org/abs/2112.05561> [2 May 2025]

[14] Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. (2002) 'SMOTE: Synthetic Minority Over-Sampling Technique'. Journal of Artificial Intelligence Research 16, 321–357

[15] Worasucheep, C. (2022) 'Ensemble Classifier for Stock Trading Recommendation'. Applied Artificial Intelligence 36 (1), 2001178