### 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 back testing 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, Prophet, 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 is 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 by leveraging feature importance rankings and built-in regularization. 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.

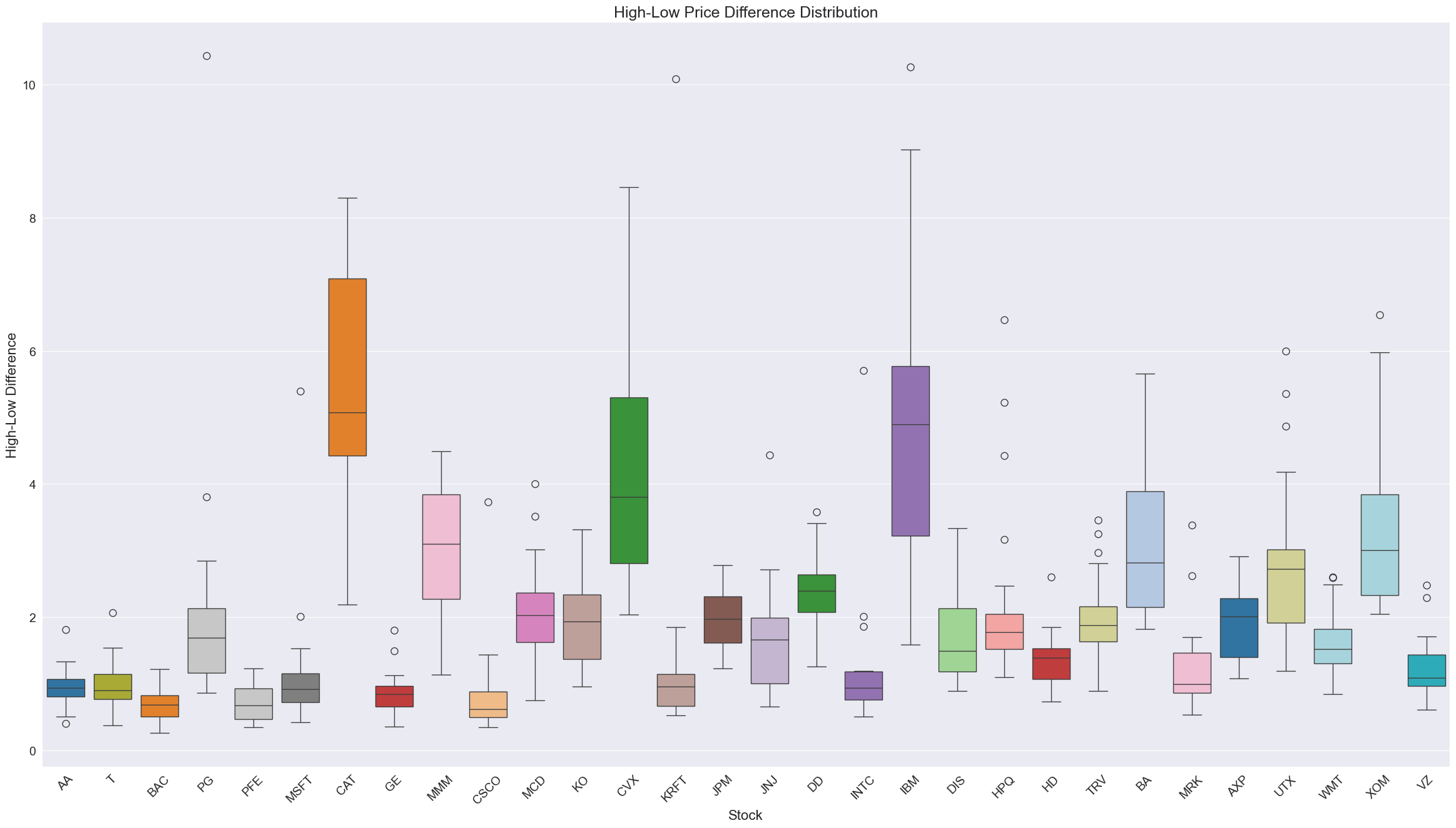


Fig. 1. High-Low Price Difference Distribution (Box Plot)

Use seaborn and subplot 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.

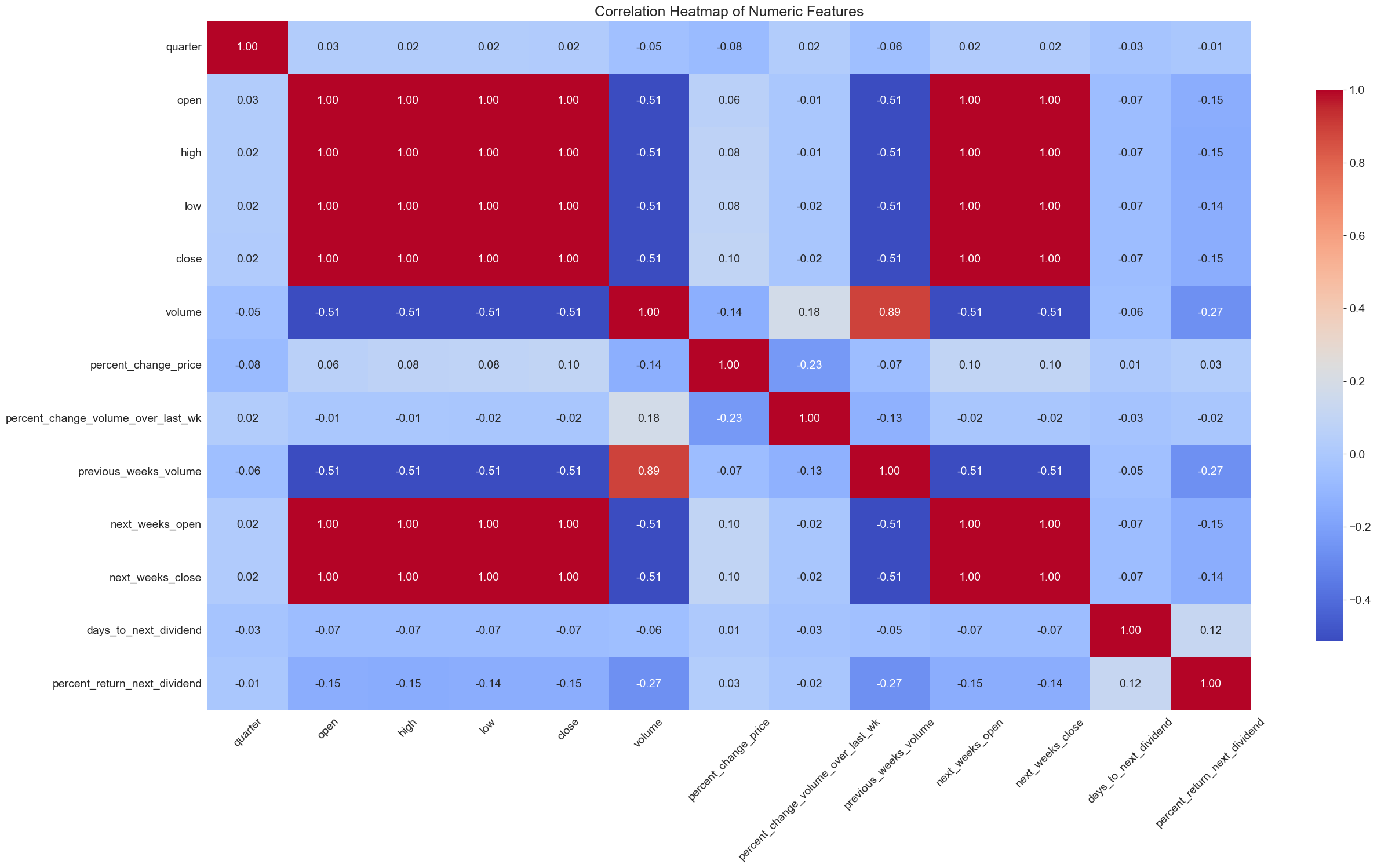


Fig. 2. Correlation Heatmap of Numeric Features

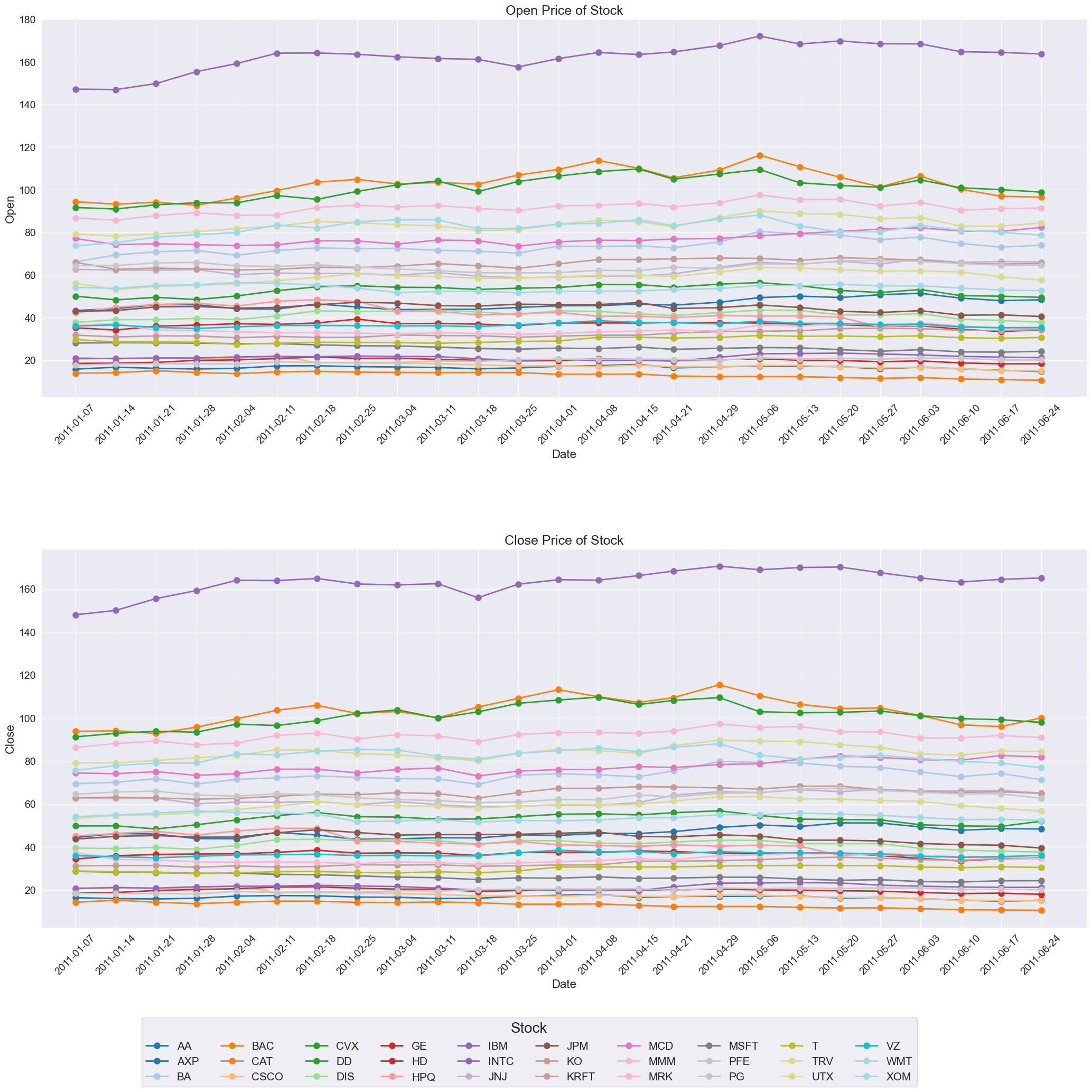


Fig. 3. Open and Close Price Over Time (Line Plot)

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

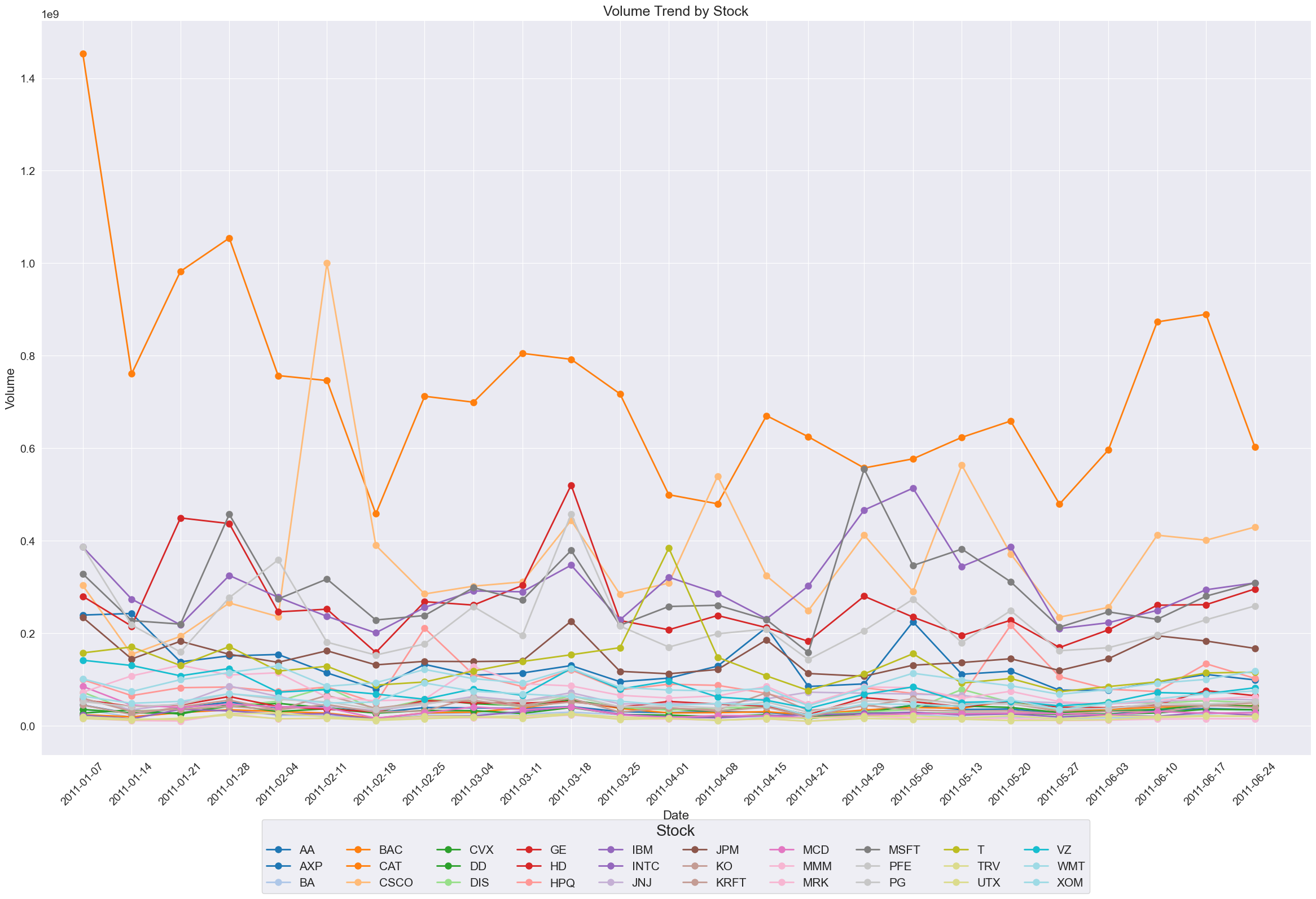


Fig. 4. Trading Volume Over Time (Line Plot)

#### 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 and 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 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 | Function | Code |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table. 1. Feature selection of classification

Description:

###### *4*.*3.2* *Regression*

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Function | Code |  |
| ema\_2 |  |  |  |
| ema\_4 |  |  |  |
| ema\_diff |  |  |  |
| rsi\_2 |  |  |  |
| atr\_3 |  |  |  |
| macd |  |  |  |
| macd\_signal |  |  |  |
| macd\_hist |  |  |  |
| momentum\_1 |  |  |  |
| momentum\_3 |  |  |  |
| roc\_2 |  |  |  |
| volatility\_3 |  |  |  |
| vpt |  |  |  |
| price\_volume\_ratio |  |  |  |
| is\_month\_end\_week |  |  |  |
| dividend\_timing |  |  |  |
| open |  |  |  |
| high |  |  |  |
| low |  |  |  |
| close |  |  |  |
| percent\_change\_volume\_over\_last\_wk |  |  |  |
| percent\_return\_next\_dividend |  |  |  |
|  |  |  |  |

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.

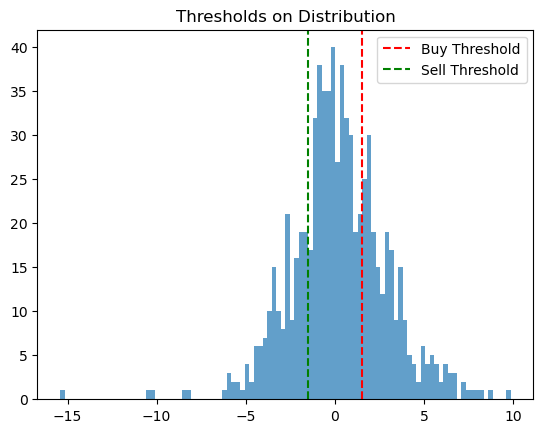


Fig. 5. Thresholds on Distribution

#### 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 practice and testing.

##### 5.1 Classification

L(model):

* Based on its efficiency and ease of interpretation for binary classification tasks.
* Hyperparameters: Regularization strength (C=1.2), solver (liblinear), and maximum iterations (500). Threshold: 0.52.
* Achieved balanced accuracy of 70.26%, precision of 42.31%, and ROC-AUC of 70.26%.

R(model):

* Processes high-dimensional data and provides feature importance assessment.
* Hyperparameters: 75 estimators. Threshold: 0.48.
* Achieved balanced accuracy of 68.12%, precision of 41.67%, and ROC-AUC of 68.12%.

A(model):

* It makes good use of weak classifiers for cascade, and different classification algorithms can be used as weak classifiers, and it also has high accuracy.
* Hyperparameters: SAMME algorithm, learning rate (0.47), and 170 estimators. Threshold: 0.51.
* Achieved balanced accuracy of 75.33%, precision of 44.83%, and ROC-AUC of 75.33%.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | X | Y | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table. 3. Feature selection of classification

Conclusion 1 2 3

##### 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

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | X | Y | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table. 4. Feature selection of regression

Conclusion 1 2 3

##### 5.3 Model Comparison

Lines and description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | M | M | R | R |
| R |  |  |  |  |
| R |  |  |  |  |
| R |  |  |  |  |

Table. 5. Outcome of Comparison

Conclude of this part

#### 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 Incomplete Weekly Data and Predictive Validity Another challenge arose when the data for a given week was incomplete—such as missing closing prices or volume data—yet the week was still used for predicting the subsequent week's price. This violated the principle of causal modeling, where inputs should fully precede outputs in time. To address this, such records were either excluded or imputed using domain-specific strategies (e.g., forward-fill based on prior week trends) to ensure that only complete, causally valid sequences were used for forecasting.

##### 6.4 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 presents a comprehensive framework for financial time-series forecasting that systematically integrates domain-specific feature engineering, ensemble machine learning, and deep learning techniques. The proposed methodology addresses two critical challenges in quantitative finance: accurate prediction of short-term price movements and reliable early detection of downside risks. By developing an adaptive feature engineering pipeline that combines multidimensional technical indicators with dynamic missing-value imputation strategies, we overcome the limitations of conventional technical analysis while preserving interpretability. The modular signal generation system, validated through rigorous walk-forward backtesting protocols, demonstrates the practical viability of threshold-optimized trading rules in diverse market conditions. Furthermore, the novel LSTM-Attention hybrid model establishes a new paradigm for risk-aware forecasting by jointly modeling temporal price dependencies and attention-weighted risk patterns.

#### 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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task and Weighting** | **Data pre-processing (10%)** | **Feature**  **Selection**  **(25%)** | **ML method**  **development**  **(25%)** | **Method**  **Evaluation**  **(10%)** | **Report**  **Writing**  **(30%)** |
| **Juntian Xiao** | **0%** | **0%** | **0%** | **0%** | **50%** |
| **Yuhong Yuan** | **0%** | **0%** | **0%** | **0%** | **10%** |
| **Guangzheng Dong** | **0%** | **0%** | **0%** | **0%** | **5%** |
| **Tianhe Zhao** | **0%** | **0%** | **0%** | **0%** | **5%** |