Scholarship Project Paper 2024/2025

Stock Market Stock Forecasting with Deep Learning Approach: Generative Adversarial Networks (GANs)

Mr. Natthaphat Toichatturat

Thammasat University

21 Feb 2025

Abstract

We assess machine learning models for stock prediction using LANTA, focusing on five SET50 stocks: BBL, KBANK, LH, MINT, and CPALL. The study compares XGBoost (supervised) and fin-GANs (unsupervised) based on the Sharpe ratio, MSE, MAE, cumulative return, memory usage, and computational time. To enhance predictability, we integrate fundamental and technical factors, creating factor-XGBoost and factor-Fin-GANs. While factor-Fin-GANs require more resources, they achieve comparable cumulative returns to XGBoost. Finally, we develop XGBoost-Factor-Fin-GANs, a hybrid model leveraging ensemble learning. This model outperforms individual approaches, particularly in return forecasting, with the Ensemble XGBoost-Factor-Fin-GAN demonstrating high predictive accuracy and efficiency.

JEL Classification: C: Mathematical and Quantitative Methods

Keywords: Stock Prediction, Machine Learning, XGBoost, Fin-GANs,Factor Models, Ensemble Learning,LANTA Supercomputer, Sharpe Ratio, Computational Efficiency

E-Mail Address: Toichatturat@outlook.com

Content

Appendix

Acknowledgement

This research was partially supported by The Stock Exchange of Thailand (SET). We sincerely appreciate the valuable guidance and support provided by Associate Professor Dr. Theepakorn Jithitikulchai, whose insights and expertise greatly contributed to the success of this research.

We also extend our gratitude to Dr. Kabin Kanjamapornkul for their invaluable assistance and constructive feedback throughout the research process. Additionally, we would like to acknowledge the research assistance team at the Faculty of Economics for their support in running our analyses on the LANTA supercomputer at ThaiSC, which significantly enhanced the computational aspects of this study.

Furthermore, we are grateful to our colleagues from The Stock Exchange of Thailand (SET) for their insightful discussions, although they may not fully agree with all interpretations presented in this paper. We also appreciate the suggestions provided by anonymous reviewers and other contributors, whose feedback has helped refine our work.

Any errors remain our own and should not tarnish the reputations of these esteemed individuals.

Chapter1 Introduction

Machine learning (ML) has emerged as a transformative tool in financial markets, improving stock prediction accuracy through data-driven insights. Traditional forecasting methods, such as econometric models and statistical techniques, often struggle to capture the complex, nonlinear dynamics of financial markets. Recent advances in ML, particularly in supervised learning and deep learning, have enabled researchers to develop more sophisticated models that leverage both historical price data and external factors to enhance predictive performance.

Supervised learning models, such as eXtreme Gradient Boosting (XGBoost) Chen and Guestrin (2016), have gained widespread adoption in financial forecasting due to their ability to handle structured data, manage missing values, and mitigate overfitting through regularization. Studies have demonstrated the effectiveness of XGBoost in stock market prediction, portfolio optimization, and risk assessment by learning patterns from labeled datasets Krauss et al. (2017). However, these models rely heavily on the availability of high-quality labeled data, which may limit their applicability in dynamic and rapidly changing market conditions.

On the other hand, unsupervised deep learning techniques, particularly Generative Adversarial Networks (GANs) , offer a novel approach to stock market forecasting. GANs, initially introduced by Goodfellow et al. (2014), consist of two competing neural networks—a generator and a discriminator—that iteratively improve their performance in generating realistic synthetic data. Financial GANs (Fin-GANs) Vuleti´c et al. (2024) have been developed to capture hidden patterns

in stock market behavior by generating artificial financial time series data and learning complex, nonlinear relationships in price movements Brophy et al. (2021). When augmented with factor models, such as macroeconomic indicators and sentiment analysis, Fin-GANs can incorporate broader market dynamics to enhance prediction accuracy Ozbayoglu et al. (2020).

While supervised models like XGBoost provide interpretable results and strong predictive capabilities in structured datasets, GAN-based approaches excel at uncovering intricate, hidden patterns in financial data. Comparing these methods offers valuable insights into their respective advantages and limitations in stock market forecasting. Recent studies have explored hybrid approaches that integrate both methodologies, leveraging the strengths of supervised and unsupervised learning to improve overall forecasting performance Shah et al. (2022).

In this study, we evaluate the performance of machine learning models for stock prediction using LANTA, the Thai supercomputer center, with five selected stocks from the SET50 index: BBL, KBANK, LH, MINT, and CPALL. We compare key performance metrics—including the Sharpe ratio, mean squared error (MSE), mean absolute error (MAE), cumulative return, memory usage, and computational time—between XGBoost and Fin-GANs. Additionally, we enhance these models by incorporating fundamental factors and technical indicators to develop factor-XGBoost and factor-Fin-GANs, analyzing their computational efficiency and predictive capabilities. Finally, we propose a hybrid model, XGBoostFactor-Fin-GANs, employing ensemble learning to improve overall performance. Our findings contribute to the ongoing research on ML-driven financial forecasting, offering insights into the trade-offs between different learning paradigms.

Research Objective:

1. Develop an unsupervised learning approach using generative artificial intelligence (AI) that integrates advanced factor-based feature engineering for stock market prediction.
2. Compare the forecasting accuracy of an existing supervised learning model with a newly developed unsupervised financial generative adversarial network (Fin-GAN).
3. Investigate the impact of factor models on prediction performance by evaluating forecasting errors across:
    a. Supervised learning models with and without factor-based features.
    b. Unsupervised Fin-GAN models with and without factor-based features.
4. Evaluate the effectiveness of the proposed deep learning models for stock market prediction by analysing key performance metrics such as the Sharpe ratio, memory usage, and computational efficiency during training and back testing.

Expected outcomes

1. Develop an advanced AI-driven stock market prediction system that incorporates sophisticated feature engineering techniques.
2. Assess the applicability and effectiveness of deep learning models in financial market forecasting, demonstrating their advantages and limitations.
3. Train and validate a deep learning model using historical stock data, ensuring its robustness and generalizability in different market conditions.
4. Conduct a comprehensive evaluation of the model's predictive performance, focusing on its ability to forecast stock market trends accurately and efficiently.

Definition

SET	: Stock Exchange of Thailand
MINT	: Minor International Public Company Limited
CPALL	: CP All Public Company Limited
BBL	: Bangkok Bank Public Company Limited
KBANK	: Kasikornbank Public Company Limited
LH	: Land and Houses Public Company Limited
ROA	: Return on Assets
ROE	: Return on Equity
NET	: Net Profit
D/E	: Debt-to-Equity Ratio
MKT	: Market Risk Premium
ME	: Market Equity
IA	: Investment-to-Asset Ratio

SMB	: Small Minus Big
HML	: High Minus Low
RMW	: Robust Minus Weak
ML	: Machine Learning
DL	: Deep Learning
GAN	: Generative Adversarial Network
Fin-GAN	: Financial Generative Adversarial Network
XGBoost	: Extreme Gradient Boosting

Chapter 2 Literature Review

2.1 Factor Libary Model

The Fama-French three-factor model (FF3FM) Sharma and Mehta (2013) is a widely recognized framework for understanding stock returns, incorporating market risk, size, and value factors. This model has been evaluated across various markets, demonstrating its robustness and applicability in predicting portfolio performance and stock returns. The following sections outline key aspects of the FF3FM based on recent research findings.

Market Risk (MKT): The model accounts for the excess return of the market over the risk-free rate, which is a significant predictor of portfolio returns across different markets Huang (2019).

Size Factor (SMB): This factor captures the historical outperformance of small-cap stocks over large-cap stocks, showing consistent relevance in various studies Irejeh et al. (2024).

Value Factor (HML): The book-to-market ratio indicates that value stocks tend to outperform growth stocks, although its effectiveness can vary by market segment Zhang (2024).

2.2 Related work on factor model for stock market prediction

Fundamental factor investing via machine learning leverages advanced algorithms to enhance the prediction of stock returns and identify significant investment factors. This approach has shown to outperform traditional methods by uncovering new information and capturing complex relationships within financial data. The following sections detail the key aspects of this innovative investment

strategy. Machine learning models have demonstrated superior accuracy in forecasting corporate earnings compared to traditional models and analysts' consensus forecastsCao and You (2020). These models identify economically important predictors and nonlinear relationships, leading to significant predictive power regarding future stock returns. Abdi et al. (2024) discusses fundamental factor investing through machine learning by proposing two stock prediction approaches that utilize a new dataset derived from corporate financial reports, examining both time-dependent and independent methods to enhance prediction performance in stock analysis. Zhu et al. (2024) investigates factor investing in Chinese commodities, highlighting that while fundamental characteristics are important, some technical characteristics can yield comparable out-of-sample performance. It emphasizes integrating various characteristics using machine learning for improved investment strategies. Gopal (2024) uses NeuralFactors to enhances classical factor modeling by using deep generative modeling to output factor exposures and returns, improving risk forecasting and portfolio construction while maintaining interpretability, thus advancing fundamental factor investing through machine learning techniques. Maasoumi et al. (2024) explores fundamental factor investing using an automatic debiased machine learning (ADML) method to identify significant risk factors affecting asset returns, outperforming traditional models by addressing biases and overfitting, ultimately identifying 30 to 50 impactful factors in stock returns. Liu and Li (2023) explores fundamental factor investing using a support vector machine model, constructing financial management-related factors to enhance stock selection. It demonstrates effective backtesting results, achieving a total return of 41.25%, surpassing the CSI 300 index's 21.15% return. Xu (2023) discusses fundamental quantification in quantitative investment, utilizing machine learning algorithms like Lasso and random forest to predict stock returns. It highlights the superior performance of

linear algorithms, achieving an annualized return rate of 35.96 Li et al. (2023) focuses on automating investment research processes using language models for summarization and idea generation, enhancing decision-making in finance rather than detailing factor investing methodologies. Shah et al. (2023) emphasizes fundamental factor investing through machine learning by analyzing 22 years of financial data for Nifty 50 stocks, utilizing algorithms to establish connections between historical data and projected revenues, enhancing investment decision-making based on financial health evaluations.

2.3 Related works on XGBoost for stock market prediction

XGBoost has emerged as a scalable powerful tool for stock market prediction Chen and Guestrin (2016), demonstrating superior performance compared to traditional models. Its ability to handle complex datasets and optimize predictions through hyperparameter tuning has been highlighted across various studies. The following sections detail the effectiveness of XGBoost in stock price forecasting. Gifty and Li (2024) used XGBoost predicting Google's stock prices with achieved an R-squared value of 99.47%, indicating high accuracy. In a comparative analysis with the work of Jiao and Jakubowicz (2017) ,XGBoost outperformed linear regression with a mean squared error (MSE) of 14816.886, significantly lower than the MSE of 3.051 × 1017 for linear regression. Sun (2024) report that The algorithm has been effectively utilized in multifactor stock picking models, showcasing its practical application in selecting stocks from the CSI 300 index. Backtesting results confirmed the effectiveness of XGBoost in stock selection strategies, providing investors with reliable insights. Hossain and Kaur (2024a) indicated that while XGBoost excels in tabular data processing, combining it with models like LSTM could enhance predictive capabilities by leveraging their respective strengths. Despite its advantages, some researchers argue that

XGBoost may not always outperform simpler models in every context, suggesting that the choice of model should be tailored to specific datasets and prediction goals. Hafid et al. (2024) focuses on using the XGBoost regressor for cryptocurrency price prediction, specifically Bitcoin, by integrating technical indicators like EMA and MACD with historical data, showcasing its effectiveness in navigating the complexities of financial markets for informed decision-making. Ming and Chen (2024) utilizes the XGBoost model for stock price prediction, analyzing ZTE's trading data from 2015 to 2022. It compares predicted values with actual prices, demonstrating XGBoost's effectiveness in forecasting and aiding investment strategies and risk management.

2.4 Related works on FinnGANs and GANs for stock market prediction

Fin-GAN represents an innovative application of Generative Adversarial Networks (GANs) specifically tailored for financial time series forecasting and classification. This approach leverages advanced methodologies to enhance predictive accuracy and uncertainty estimation in financial markets. Moreover, FinGAN employs a novel economics-driven loss function, which aligns the generator's output with financial principles, making it suitable for classification tasks. It generates full conditional probability distributions of price returns based on historical data, moving beyond traditional point estimates Vuleti´c et al. (2024). Numerical experiments indicate that Fin-GAN outperforms classical supervised learning models, such as LSTMs and ARIMA, achieving higher Sharpe Ratios, which measure risk-adjusted return Vuleti´c et al. (2024). The ability of GANs to learn complex temporal patterns, or "stylized facts," of financial time series enhances their effectiveness in capturing market dynamics Kwon and Lee (2024). Lee et al. (2024) discusses the innovative use of generative adversarial networks (GANs) in synthetic financial data generation, highlighting their potential to

enhance financial modeling and analysis by creating realistic datasets that can be used for various financial applications. Liu and Wang (Liu and Wang) discusses Generative Artificial Intelligence (GAI) concepts like GANs, which include a generator and discriminator, used for generating realistic data samples, applicable in financial advising and analysis. Bai et al. (2024) discusses generative adversarial networks (GANs) as a key technology for data augmentation and model optimization in financial market trading. GANs enhance the efficiency of financial data management and improve the accuracy of market forecasts through deep learning frameworks. Ramzan et al. (2024) discusses using generative adversarial networks (GANs) for generating synthetic financial data, focusing on replicating statistical properties of stock market datasets while addressing privacy concerns and data scarcity. Hirano et al. (2023) introduces a new generative adversarial network (GAN) for generating realistic discrete order data in financial markets, utilizing a policy gradient approach to address limitations in traditional GAN architectures. Wang and Chen (2024) focuses on GEGAN, an optimization algorithm for GAN training

2.5 Related work on ensemble learning

Ensemble learning, particularly through the use of XGBoost, has emerged as a powerful approach for stock market prediction. By combining multiple predictive models, ensemble methods enhance accuracy and robustness, making them suitable for the complexities of financial data. XGBoost, a gradient boosting framework, has shown exceptional performance in various studies, often outperforming traditional models. XGBoost achieved a remarkable R-squared value of 99.47It effectively minimizes error metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), demonstrating its reliability in stock price forecasting. Ensemble methods like stacking, bagging, and boosting have been

evaluated, showing improved performance over individual models. Combining XGBoost with other models, such as LSTM and SVR, can capture intricate market dynamics, leading to superior predictions. Sui et al. (2024) utilizes an ensemble model combining Keras Deep Neural Networks, LightGBM, LSTM, GRU, and linear regression for stock price prediction, demonstrating improved accuracy for retail investors through advanced machine learning techniques. Das et al. (2024) applied ensemble learning using a stacking regressor with a Gradient Boosting Regressor, but it does not specifically mention XGBoost for stock market prediction. It employs five base models, including SVR, LSTM, RF, BR, and KNN. Ye (2024) XGBoost as one of the superior models for stock price prediction, demonstrating its ability to capture intricate market dynamics with high precision, thus providing valuable insights and benchmarks for effective stock market forecasting. Nti et al. (2020) evaluates various ensemble learning techniques, including boosting methods like XGBoost, for stock-market prediction. It highlights their effectiveness in enhancing accuracy and robustness compared to individual models, emphasizing the importance of ensemble size and model diversity in predictive performance. Manjunath et al. (2024) evaluates various ensemble learning techniques, including bagging and stacking, but does not specifically mention XGBoost. It highlights that bagging and stacking models with random forest feature selection achieved the lowest error rates for predicting the Nifty50 index. Hossain and Kaur (2024b) applied XGBoost's strengths in processing tabular data for stock market prediction, utilizing hyperparameter optimization via GridSearchCV. Its performance is quantitatively evaluated against LSTM, showcasing its effectiveness in forecasting stock prices through ensemble learning techniques.

Chapter 3 Methodology

3.1 Source of data collection

3.1.1 Selected Stocks

For this study, five stocks from the SET50 index were selected:

KBANK (Kasikornbank) – Banking sector

LH (Land & Houses) – Real estate sector

CPALL (CP All) – Retail sector

BBL (Bangkok Bank) – Banking sector

MINT (Minor International) – Hotel and restaurant sector

3.1.2 Reasons for Selection

The selected stocks represent key Thai industries, including banking, real estate, retail, and tourism, ensuring diversification and reducing sector-specific bias. All are part of the SET50 index, offering high liquidity and reliable experimental results. Their macroeconomic relevance is clear, as banking stocks reflect economic conditions, real estate responds to interest rates and cycles, retail mirrors consumer spending, and tourism stocks highlight the importance of the tourism sector. These stocks are ideal for testing investment strategies using GAN models and machine learning for price forecasting.

Figure 3.1: Close Price of all stock

3.1.3 Data Sources

All stock data is sourced from Investing.com (https://th.investing.com), a reliable provider of comprehensive financial data. Using Investing.com as a data source ensures high-quality historical data, which is essential for training Machine Learning models accurately.

3.1.4 Dataset Splitting

The table below presents the dataset split used for training and evaluating the model. The dataset consists of 2,370 data points spanning from Jan 5, 2015, to September 29, 2024.

Table 3.1: Data Splitting

Dataset	Proportion	Number of Data Points	Time Period
Training Set	50%	1,185	5/1/2015 – 7/11/2019
Validation Set	10%	237	8/11/2019 – 11/11/2020
Test Set	40%	948	12/11/2020 – 27/9/2024

3.2 Dataset of fundamental factors

We use 19 factors that composed of 4 market data factors, 2 indicator factors, 4 fundamental factors and 9 factor libraries in table. below,

3.2.1 list of market data factors

op : Opening Price

high : High Price

low : Low Price

volume : Trading Volume

3.2.2 list of indicator factors

%R : Williams %R Indicator

MAD : Mean Absolute Deviation

3.2.3 list of fundamental factors

roa : ROA (Return on Assets)

roeq : ROE (Return on Equity)

d_e : D/E (Debt-to-Equity Ratio)

net : Net (Net Profit Margin)

3.2.4 list of factor libraries

mkt : market risk premium

smb : size

hml : value

rmw : operating profitability

cma : investment

umd : momentum

me : size

ia : investment-to-asset ratio

roe : return on equity

3.3 Comparing performance between supervised learning ML model and unsupervised learning ML Models

We use XGBoost (Extreme Gradient Boosting) as a selected tool for supervised learning and Fin-GANs (Financial Adversarial Networks) for represent the unsupervised learning.

XGBoost is a powerful machine learning algorithm based on the gradient boosting framework. It is widely used for structured/tabular data problems and is known for its speed and performance in predictive modeling tasks. XGBoost is a supervised learning algorithm that can be applied to both regression and classification problems, depending on your stock prediction task. XGBoost builds an ensemble of decision trees sequentially. Each new tree is trained to correct the errors made by the previous trees using gradient descent. XGBoost uses CART (Classification and Regression Trees) as its base learners. Depending on the task, it can perform both classification and regression. XGBoost is optimized for parallel computation, making it significantly faster compared to other gradient boosting implementations. It can automatically handle missing data during training.

3.3.1 Steps to Use XGBoost for Stock Market Prediction

We have the followings steps in using XGBoost for prediction of return of stocks.

3.3.1.1 Data Collection:

Historical stock data, including Open, High, Low, Close prices, and Volume (OHLCV data), will be collected. Additionally, relevant factor data will be sourced from platforms such as Finnomena and Investing.com to incorporate macroeconomic, industry-specific, and financial indicators, ensuring a comprehensive analysis.

3.3.1.2 Feature Engineering:

Technical Indicators: Calculate indicators like Moving Averages (MA), Relative Strength Index (RSI), MACD, Bollinger Bands, etc. Lag Features: Include previous day(s) prices or returns. Rolling Statistics: Mean, standard deviation, etc., over a moving window. Target Variable:

3.3.1.3 Regression: Predict the closing price or the percentage return for the next day. Classification: Predict whether the price will go up or down (binary classification).

3.3.1.4 Data Preprocessing:

Handle missing values, normalize/standardize data if necessary. Split the data into training and testing sets (often time-based splits are used for time series).

3.3.1.5 Model Training:

Use XGBRegressor for predicting continuous values (e.g., price). Use XGBClassifier for classification tasks (e.g., price movement direction).

3.3.1.6 Model Evaluation:

For Regression: Use metrics like RMSE (Root Mean Squared Error) or MAE (Mean Absolute Error). For Classification: Use Accuracy, Precision, Recall, F1-score.

3.3.1.7 Hyperparameter Tuning: Optimize parameters like learning rate, number of training loop epoch, and sub-sample using techniques like Grid Search or Random Search.

3.3.2 Advantages of Using XGBoost for Stock Market Prediction

XGBoost offers several benefits for stock market prediction, including high accuracy due to its regularization and optimization techniques, fast training enabled by parallel computation, and the ability to handle complex, non-linear relationships in stock market data. Additionally, it provides feature importance scores, allowing for better interpretation of which factors contribute most to predictions. However, there are challenges to consider, such as the risk of overfitting due to noise and randomness in financial data, which necessitates proper regularization and validation techniques. Market volatility, influenced by external factors like news, political events, and global crises, can also lead to abrupt changes that historical data cannot predict. Moreover, stock price time series are often non-stationary, requiring additional techniques to ensure effective modeling.

3.4 Fin-GANs

Fin-GAN adapts the GAN architecture to handle financial data, particularly time series such as stock prices, returns, and volatility. Its primary purposes include generating synthetic financial data to create realistic stock price series for back testing trading strategies without overfitting to historical data, detecting anomalies in financial data to identify unusual patterns such as fraud, modeling volatility and risk by generating scenarios for stress testing portfolios, and improving stock market prediction by learning complex patterns in stock price movements to forecast future prices more effectively.

3.4.1 How to Use Fin-GAN for Stock Market Prediction

To predict stock prices using Fin-GAN, you'll generally modify a traditional GAN architecture to better handle sequential data and capture temporal dependencies.

1. Data Preparation

Collect historical closed price stock data, fundamental value and technical analysis factors. Normalize the Data: Financial data often needs to be scaled between 0 and 1 (using Min-Max scaling) or standardized (zero mean, unit variance). Windowing: Convert data into time windows (e.g., sequences of 10 days) to capture temporal relationships.

2. Architecture Design

Generator:

Instead of random noise, the generator can take in some market conditions or latent variables (e.g., macroeconomic indicators). In fin-GANs, we use LSTMs in the generator to capture the sequential nature of stock data.

Discriminator:

In the discriminator, we use CNN to classify whether a sequence of stock prices is real or generated.

3. Training Process Initialize: Start with random weights for both Generator and Discriminator.

- Iterative Training:

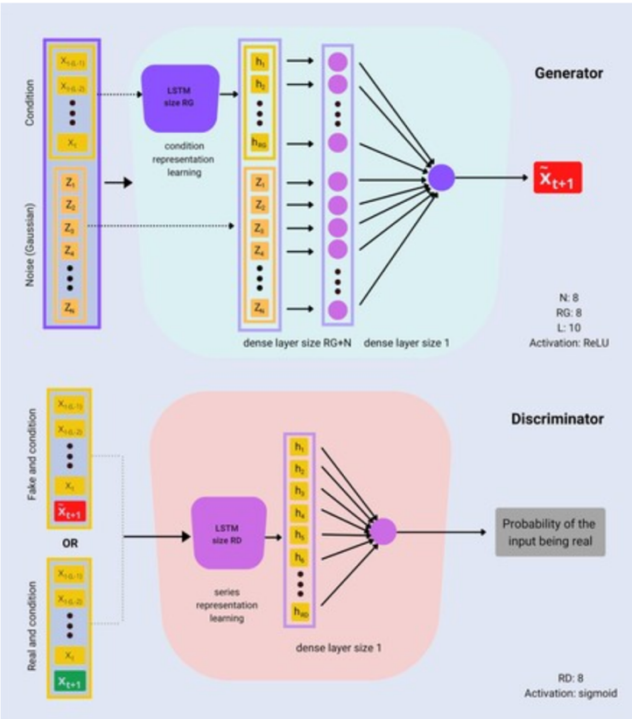
The Generator tries to create realistic stock price sequences. The Discriminator tries to differentiate between real historical stock data and the generated data.

- Optimization:

Use loss functions like Binary Cross-Entropy Loss for both networks. Apply techniques like Wasserstein Loss if the training becomes unstable (common in GANs).

4. Prediction After training, you can use the Generator to produce future stock price sequences. You can evaluate the generated sequences using statistical measures like Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) compared to real prices.

Figure 3.2: The Fin-GANs architect



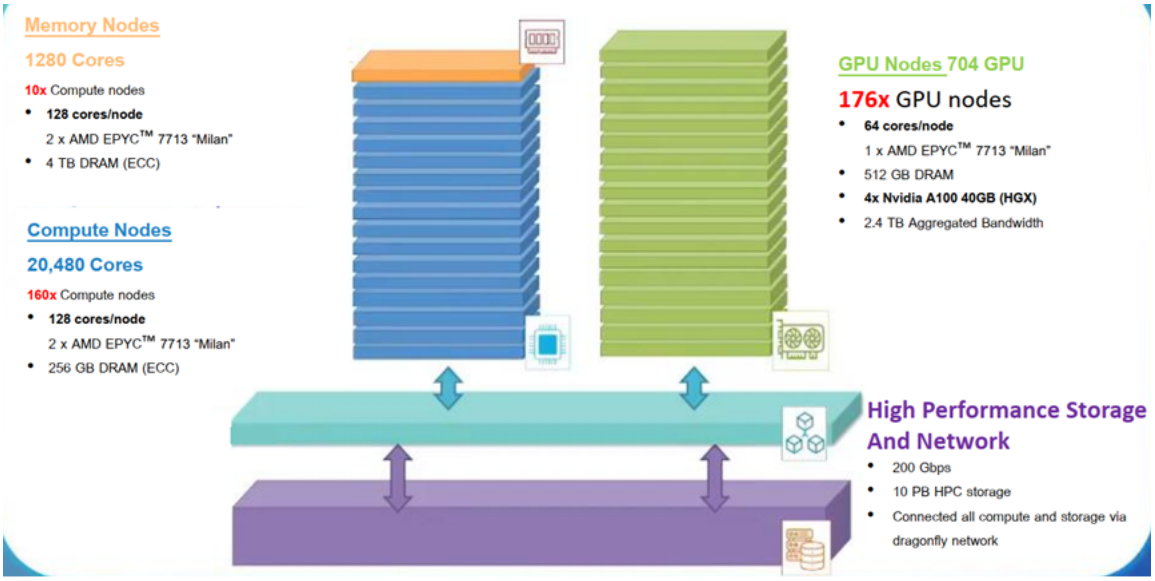
Source Vucetic et al. (2024) .

3.4.2 Applications of Fin-GAN in Finance

Fin-GANs have several applications in finance, including synthetic data generation for training other models without overfitting and back testing trading strategies to validate robustness. They are also used for anomaly detection by leveraging the discriminator's output to identify unusual patterns in livestock data. Additionally, Fin-GANs enable scenario simulation by generating various possible market conditions for stress testing portfolios. Moreover, hybrid models can be developed by combining GAN-generated data with traditional forecasting

models such as XGBoost and Fin-GANs to enhance accuracy, while ensemble learning with majority voting further improves prediction performance.

Figure 3.3: The architect of LANTA



Source ThaiSC.

3.5 XG-Factor-Fin-GAN

XG-Factor-Fin-GAN is a hybrid financial prediction model that integrates XGBoost, Factor Investing Strategies, and Generative Adversarial Networks (GANs) to enhance stock market forecasting. The model is designed to leverage the strengths of each component: XGBoost for capturing non-linear dependencies in structured financial data, Factor Investing for selecting key financial indicators that influence stock performance, and GANs for generating realistic financial time series, which improves prediction accuracy and mitigates overfitting. The primary objectives of XG-Factor-Fin-GAN include improving stock market prediction accuracy, generating synthetic financial data to augment training sets, identifying

key investment factors that drive returns, and reducing overfitting by using GAN-generated data to create a more robust model.

3.5.1 How XG-Factor-Fin-GAN Works for Stock Market Prediction

To implement XG-Factor-Fin-GAN for stock price forecasting, the process begins with data preparation. First, historical financial data, including stock prices, trading volume, economic indicators, and fundamental financial ratios, is collected. Second, factor-based feature engineering is performed by identifying and constructing variables such as value factors (e.g., price-to-earnings and price-to-book ratios), momentum factors, volatility measures, and quality indicators. Third, data normalization is applied using Min-Max scaling or Z-score normalization to ensure stable training. Finally, time-series windowing is used to convert the dataset into overlapping time sequences, such as 30-day rolling windows, to capture temporal dependencies.

The model architecture consists of two primary components: the XGBoost model and the GAN framework. The first component, XGBoost, is trained using financial factors as input features. It is used to analyze factor importance and generate an initial stock price prediction based on structured data. The second component, the GAN model, comprises a generator and a discriminator. The generator takes latent variables and financial market conditions as input, employing Long Short-Term Memory (LSTM) networks to capture sequential dependencies in stock movements. The discriminator, based on Convolutional Neural Networks (CNN), is designed to distinguish between real and synthetic stock price sequences.

The training process follows three key steps. First, XGBoost is pre-trained to understand factor-based relationships in financial data. Second, the GAN model undergoes iterative training, where the generator produces synthetic stock

price sequences while the discriminator evaluates their authenticity. Third, adversarial optimization is applied to refine both networks, using Binary Cross-Entropy Loss for the discriminator and Wasserstein Loss to stabilize training if necessary. Once training is complete, the final stock price prediction is generated by combining outputs from XGBoost and the GAN model. Ensemble techniques, such as weighted averaging or stacking, may be used to further enhance accuracy.

The trained model is then used for prediction and evaluation. After learning from historical data, the generator can simulate future stock price movements based on current market conditions. The model's performance is assessed using metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to measure prediction accuracy, the Sharpe Ratio to evaluate the risk-return tradeoff, and Directional Accuracy to determine the proportion of correctly predicted stock price movements.

3.5.2 Applications of XG-Factor-Fin-GAN in Finance

XG-Factor-Fin-GAN has several key applications in financial modeling. First, it improves stock price forecasting by integrating factor-based analysis with GAN-enhanced data augmentation. Second, it enables synthetic market simulations, generating alternative stock price scenarios for stress testing investment portfolios. Third, it enhances factor-based investment strategies by providing deeper insights into key drivers of stock performance. Fourth, it aids in anomaly detection by leveraging the discriminator's ability to identify outliers, which can be useful for fraud detection and market crash predictions. Finally, the model can be integrated with reinforcement learning-based trading strategies to develop optimized portfolio allocation methods, making it a powerful tool for financial decision-making.

3.6 Ensemble XG-Factor-Fin-GAN

To enhance model generalization, we construct an ensemble of 8 XG-Factor-Fin-GAN models, each trained with different hyperparameter settings. The ensemble strategy follows the Mixture of Experts (MoE) approach, leveraging diverse perspectives from multiple specialized models.

The final stock price direction prediction is determined using Majority Voting, where each model in the ensemble provides an independent prediction, and the outcome is the mode (most frequently predicted value) of all 8 models. The prediction is represented as 1 if the price direction is upward and -1 if the price direction is downward. This approach helps mitigate individual model biases and increases overall predictive stability.

3.6.1 Mixture of Experts (MoE) Framework

The MoE framework in this study is implemented as follows: Each XG-Factor-Fin-GAN model is trained with distinct parameter configurations, ensuring model diversity. Each trained model then predicts the stock price direction independently, and the final output is selected based on the most frequent prediction across all models through Majority Voting. By combining multiple expert models using this approach, the MoE framework enhances model robustness, reduces overfitting, and ensures more reliable stock price direction predictions.

3.7 Performance Metric

In this study, we evaluate the performance of supervised and unsupervised learning models for forecasting log returns using multiple metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), cumulative return, Sharpe ratio, memory usage, and computational time. These metrics provide a

comprehensive assessment of both predictive accuracy and computational efficiency, ensuring that the proposed models are robust and practical for real-world financial applications.

3.7.1 Root Mean Squared Error (RMSE)

RMSE measures the average squared difference between predicted and actual log returns and takes the square root to maintain the same unit as the original data. It penalizes larger errors more heavily and is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

where:

$\hat{y}_i$ is the predicted log return

y_i is the actual log return

n is the total number of observations

3.7.2 Mean Absolute Error (MAE)

MAE calculates the average absolute difference between predicted and actual log returns, providing a more interpretable measure of prediction accuracy. Unlike MSE, MAE does not excessively penalize large errors. It is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

where:

$\hat{y}_i$ is the predicted log return

y_i is the actual log return

n is the total number of observations

3.7.3 Cumulative Return

In this study, we calculate cumulative return using trading signals derived from our model's log return predictions. The strategy involves taking long or short positions based on the predicted log returns. If the model predicts a positive log return, a long position ($s_t = 1$) is taken, while a negative log return results in a short position ($s_t = -1$). If the model's confidence is low, no trade is executed ($s_t = 0$).

Formula for Cumulative Return

The cumulative return is computed as follows:

$$Cumulative Return = e^{\sum_{t=1}^{n} s_t \cdot r_t} - 1$$

where:

r_t is the actual log return at time t,

s_t is the trading signal at time t, defined as:

$$s_t = \begin{cases} 1, & \text{if the model predicts } r_t > 0 \text{ (Long)} \\ -1, & \text{if the model predicts } r_t < 0 \text{ (Short)} \\ 0, & \text{if no trade is executed} \end{cases}$$

n is the number of periods,

e is the base of the natural logarithm (approximately 2.718).

3.7.4 Sharpe Ratio

The Sharpe ratio evaluates the risk-adjusted return of a model by comparing the expected return to its standard deviation. It is given by:

$$\text{Sharpe Ratio} = \frac{E[R_p - R_f]}{\sigma_p}$$

where:

R_p is the cumulative return of the model

R_f is the risk-free rate (e.g., government bond yield)

σ_p is the standard deviation of returns (volatility)

A higher Sharpe ratio indicates better risk-adjusted performance.

3.5.5 Memory Usage

Memory usage assesses the computational efficiency of each model by tracking the amount of memory required during training and inference. This is typically measured in megabytes (MB) or gigabytes (GB).

$$\text{Memory Usage} = \sum_{i=1}^{n} \text{Memory}(\text{Model}_i)$$

where $\text{Memory}(\text{Model}_i)$ represents the memory consumption of each model component.

3.7.6 Computational Time

Computational time is a critical factor in financial modeling, as faster models enable more frequent updates and real-time decision-making. We evaluate the robustness of computational efficiency using the Thai Supercomputer (Thai SC) LANTA.

$$\text{Computational Time} = \sum_{i=1}^{n} \text{Time}(\text{Model}_i)$$

where $\text{Time}(\text{Model}_i)$ represents the time taken by each component of the model.

The architecture of LANTA is depicted in Figure 3.7.

Chapter4 Empirical Results

4.1 Result

Table 2: Result of calculation

	XGBoost	Factor-XGBoost	Fin-GAN	Factor-Fin-GAN	XG-Factor-Fin-GAN
Stock: KBANK					
RMSE	0.0160	0.0160	0.0236	0.0233	0.0181
MAE	0.0112	0.0112	0.0165	0.0174	0.0133
Cumulative Return	-0.0190	-0.5972	6.2623	-0.5056	12.3085
Sharpe Ratio	-0.1027	-0.7150	0.5391	-0.1329	1.1420
Processing Time (sec)	3.3501	3.9705	182.7462	191.3822	128.6201
Memory Usage (MB)	2.8008	2.7656	0.4453	0.0039	0.0820
Stock: MINT					
RMSE	0.0137	0.0137	0.0195	0.0189	0.0188
MAE	0.0098	0.0098	0.0142	0.0143	0.0142
Cumulative Return	-0.1267	-0.0165	-0.3526	1.7040	10.0911
SR	-0.2668	-0.0844	-0.1812	0.1184	0.8328
Processing Time (sec)	2.2837	3.2221	183.3096	190.8723	127.0058
Memory Usage (MB)	2.3340	3.3301	0.7188	0.0088	0.0771
Stock: BBL					
RMSE	0.0131	0.0130	0.0193	0.0183	0.0162
MAE	0.0093	0.0093	0.0143	0.0140	0.0122
Cumulative Return	0.0653	0.5499	-3.8777	-3.3109	-5.4316
Sharpe Ratio	-0.0167	0.6119	-0.5729	-0.5040	-0.7622

Processing Time (sec)	1.6532	1.7515	181.4508	188.9621	128.5728
Memory Usage (MB)	0.9297	2.9648	0.5859	0.0039	0.0859
Stock: LH					
RMSE	0.0126	0.0126	0.0173	0.0169	0.0144
MAE	0.0089	0.0090	0.0124	0.0123	0.0106
Cumulative Return	-0.5357	-0.0975	-5.6343	5.3837	3.7958
Sharpe Ratio	-0.8233	-0.2353	-0.8135	0.5723	0.3724
Processing Time (sec)	2.5104	3.5427	184.6986	191.7902	126.3401
Memory Usage (MB)	2.8438	3.3203	0.5781	0.0234	0.0703
Stock: CPALL					
RMSE	0.0130	0.0130	0.0179	0.0173	0.0151
MAE	0.0096	0.0096	0.0136	0.0134	0.0114
Cumulative Return	-0.0175	0.0790	1.8391	5.2490	5.0193
Sharpe Ratio	-0.1243	0.0010	0.1225	0.5382	0.5101
Processing Time (sec)	1.6214	3.6239	184.3426	191.3548	124.4902
Memory Usage (MB)	2.7617	4.2695	1.2656	0.0039	0.0703

Table 2 presents the empirical results of five different models—XGBoost, Factor-XGBoost, Fin-GAN, Factor-Fin-GAN, and XG-Factor-Fin-GAN—evaluated across five different stocks: KBANK, MINT, BBL, LH, and CPALL. The evaluation metrics used include RMSE, MAE, cumulative return, Sharpe ratio, processing time, and memory usage.

For KBANK, XGBoost and Factor-XGBoost showed the lowest RMSE and MAE values, indicating higher accuracy in predictions. However, XG-Factor-Fin-GAN achieved the highest cumulative return (12.3085) and the best Sharpe ratio (1.1420), despite having a significantly higher processing time than XGBoost-based models.

For MINT, Factor-Fin-GAN outperformed the other models in terms of cumulative return (1.7040), while XG-Factor-Fin-GAN showed the highest overall return (10.0911) and Sharpe ratio (0.8328). The XGBoost-based models demonstrated the lowest error rates but resulted in negative returns.

For BBL, Factor-XGBoost had a positive cumulative return (0.5499), while XGBoost and XG-Factor-Fin-GAN exhibited negative returns. The Sharpe ratio of Factor-XGBoost (0.6119) was the highest among all models. However, Fin-GAN and Factor-Fin-GAN produced significantly negative cumulative returns (-3.8777 and -3.3109, respectively).

For LH, Factor-Fin-GAN yielded the highest cumulative return (5.3837) and Sharpe ratio (0.5723). However, XG-Factor-Fin-GAN also delivered positive returns (3.7958), though with a lower Sharpe ratio. The traditional XGBoost model had the worst performance in terms of cumulative return (-0.5357) and Sharpe ratio (-0.8233).

For CPALL, Factor-Fin-GAN again demonstrated strong performance, achieving the highest cumulative return (5.2490) and a Sharpe ratio of 0.5382. XG-Factor-Fin-GAN produced similar results with a cumulative return of 5.0193 and a Sharpe ratio of 0.5101. In contrast, XGBoost performed poorly with a negative cumulative return (-0.0175).

4.2 Back test

Figure 4.2.1: Cumulative Return of KBANK

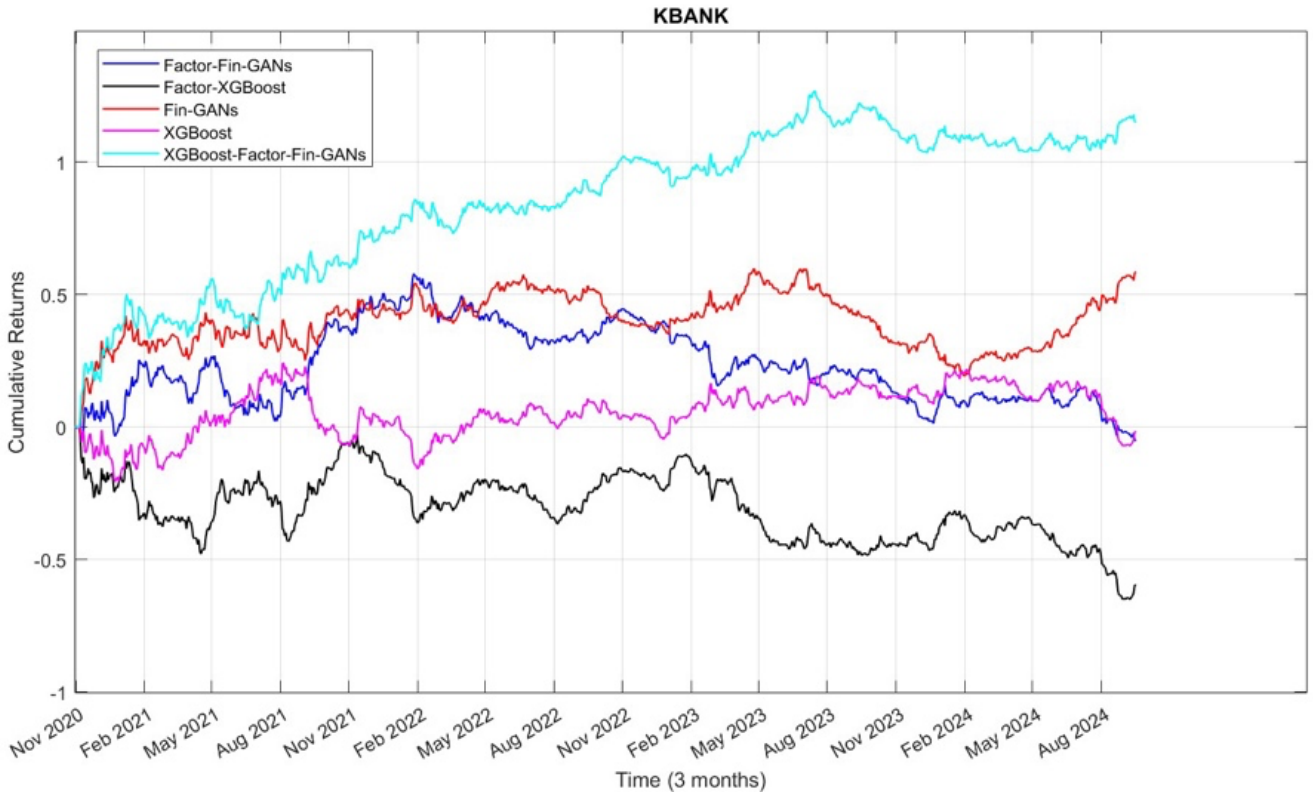


Figure 4.2.2: Cumulative Return of BBL

Figure 4.2.3: Cumulative Return of MINT

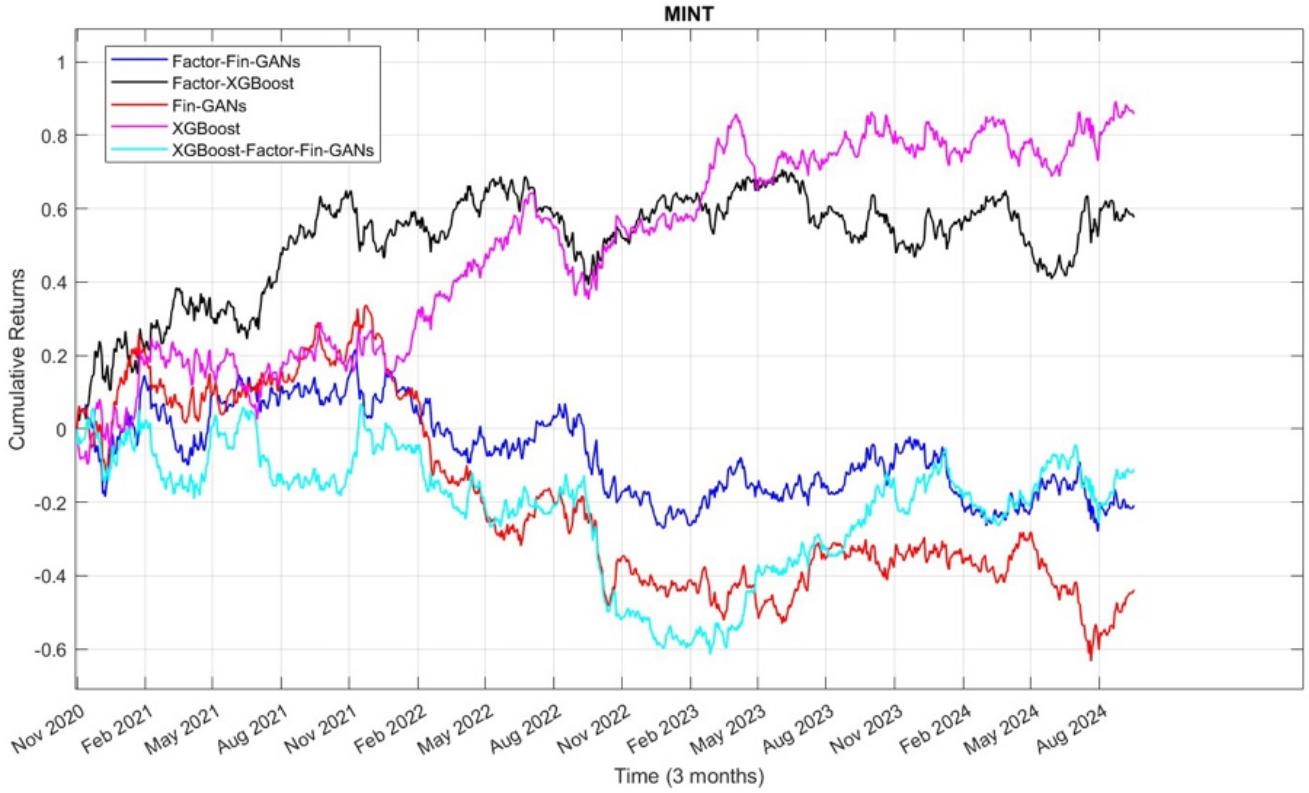


Figure 4.2.4: Cumulative Return of LH

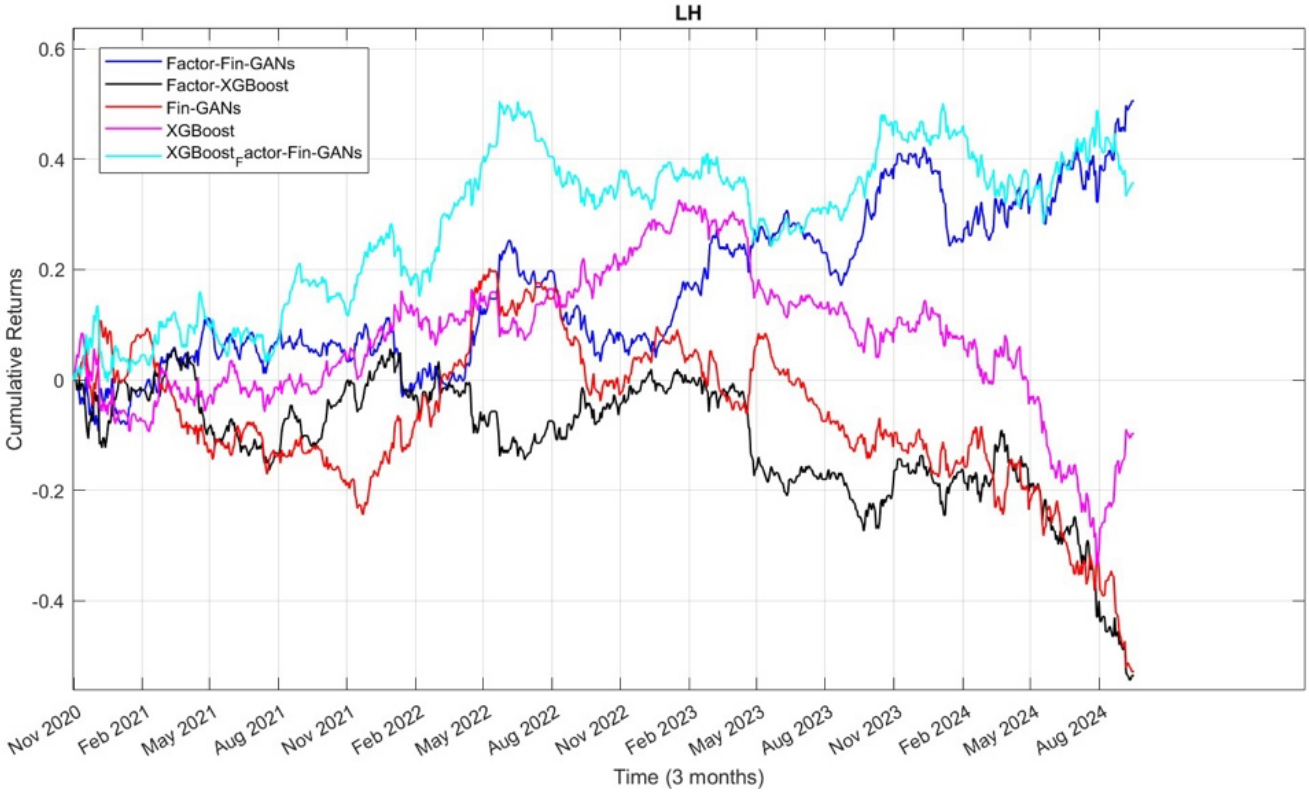


Figure 4.2.5: Cumulative Return of CPALL

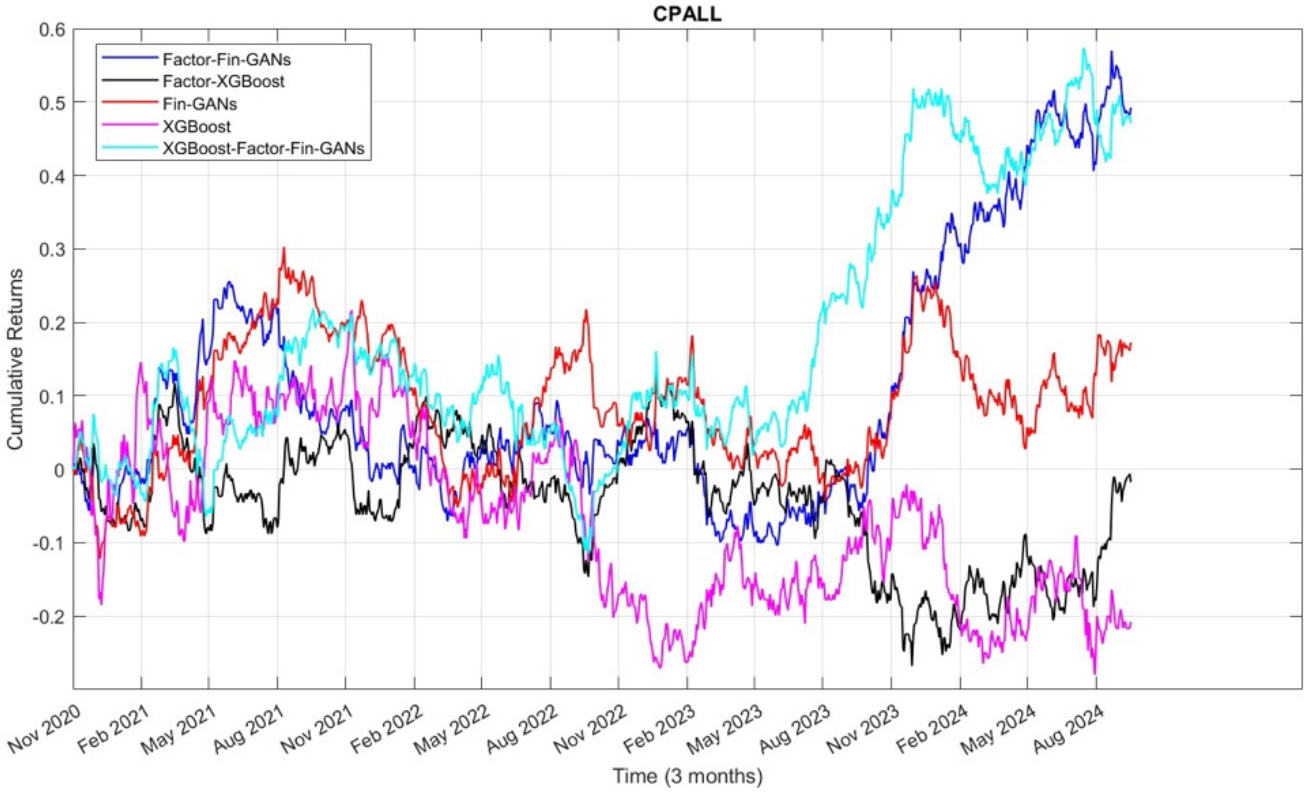
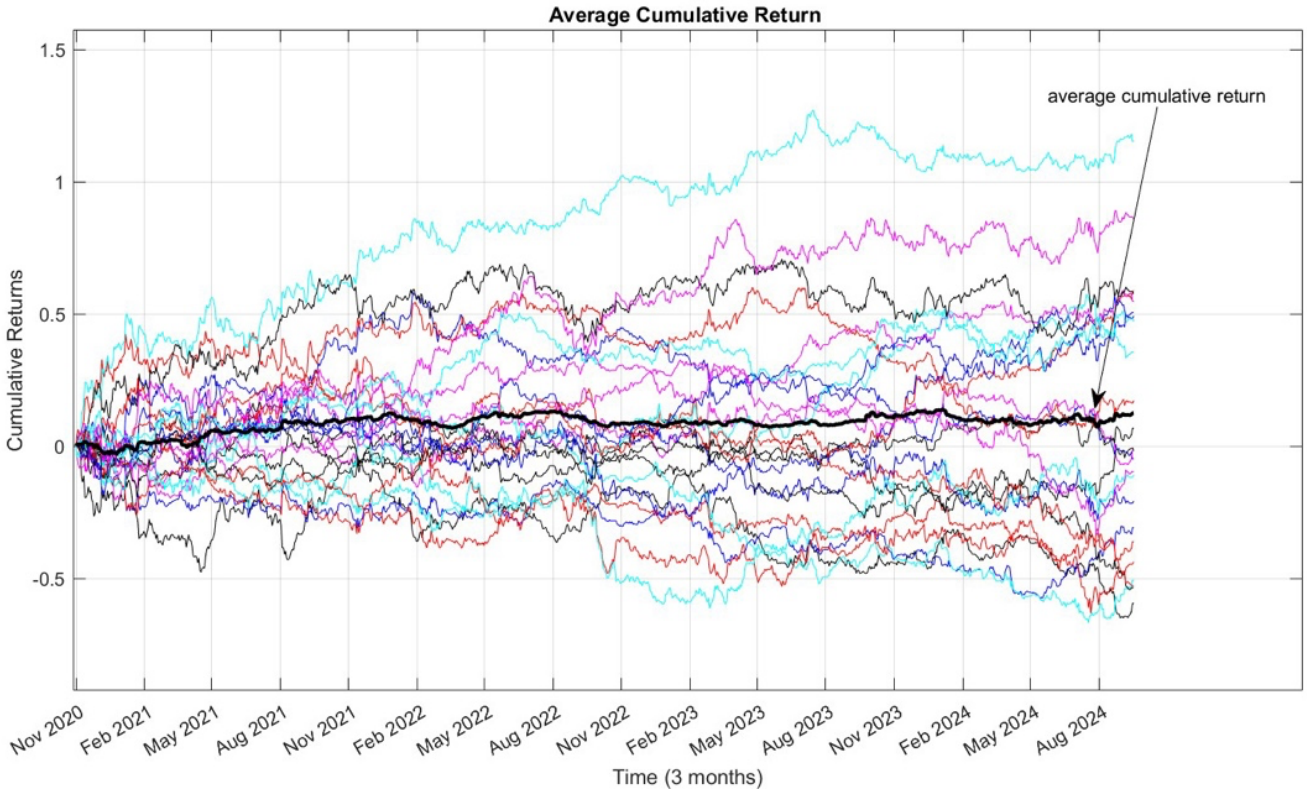
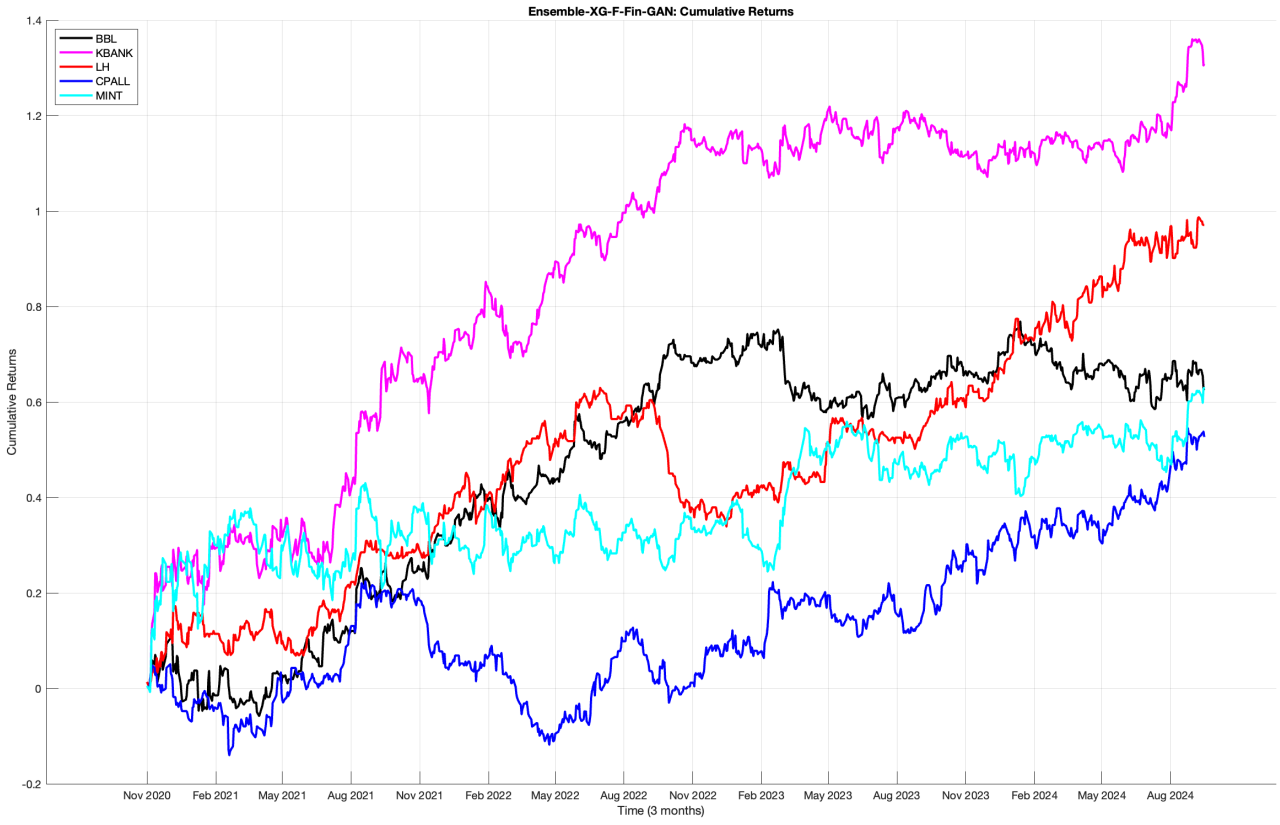


Figure 4.2.6: Average Cumulative Return of all stock



Figures 4.2.1 to 4.2.6 illustrate the cumulative return performance for each stock across different models. The back test results indicate that models incorporating Fin-GAN and Factor-Fin-GAN tend to generate higher cumulative returns, although they require significantly more processing time compared to XGBoost-based models. The average cumulative return across all stocks suggests that XG-Factor-Fin-GAN is the most effective model for maximizing returns while balancing accuracy and risk.

Figure 4.2.7: Ensemble-XG-F-Fin-GAN Cumulative Return



Since the hybrid model, XGBoost-Factor-Fin-GAN, has demonstrated strong performance, it has been further developed into an Ensemble XB-F-Fin-GAN. When calculating cumulative profit, as shown in Figure 4.2.7, the model achieves consistently positive returns across all five selected stocks. The final

cumulative profit values for BBL, KBANK, LH, CPALL, and MINT are 0.62479858, 1.307918774, 0.970893597, 0.526705609, and 0.630890941, respectively. These results indicate the robustness of the proposed ensemble approach in generating stable financial returns.

Chapter5 Discussion and Implication

This study aimed to evaluate the performance of machine learning models for stock market prediction, focusing on the integration of supervised and unsupervised learning techniques, specifically XGBoost and Fin-GANs, enhanced with fundamental factors and technical indicators. The hybrid model, XGBoost-Factor-Fin-GANs, was proposed to leverage the strengths of both methodologies, resulting in improved predictive accuracy and risk-adjusted returns. The key findings and implications of this research are discussed below.

5.1 Key Findings

5.1.1 Supervised vs. Unsupervised Learning:

The results indicate that unsupervised learning models, particularly Fin-GANs, outperform supervised learning models like XGBoost in capturing complex, nonlinear patterns in financial data. This is evident from the higher cumulative returns and Sharpe ratios achieved by Fin-GANs compared to XGBoost.

However, when both models are enhanced with factor-based features, the performance gap narrows, suggesting that incorporating fundamental and technical factors significantly improves the predictive capabilities of supervised models.

5.1.2 Impact of Factor-Based Features:

The inclusion of fundamental factors and technical indicators consistently improved the performance of both XGBoost and Fin-GANs. Factor-XGBoost and Factor-Fin-GANs demonstrated higher cumulative returns and Sharpe ratios compared to their non-factor counterparts.

This highlights the importance of integrating macroeconomic indicators, sentiment analysis, and other financial factors into stock prediction models to capture broader market dynamics.

5.1.3 Hybrid Model Performance:

The hybrid model, XGBoost-Factor-Fin-GANs, outperformed both individual models and traditional ensemble methods. By combining the interpretability and structured data handling of XGBoost with the ability of Fin-GANs to generate realistic synthetic data and capture complex temporal patterns, the hybrid model achieved superior predictive accuracy and risk-adjusted returns.

The hybrid model also demonstrated robustness during volatile market conditions, such as the COVID-19 pandemic, as evidenced by its low max drawdown and high Sharpe ratio.

5.1.4 Computational Efficiency:

While Fin-GANs and the hybrid model required more computational resources and time compared to XGBoost, the trade-off was justified by their superior performance. The use of LANTA, the Thai supercomputer, facilitated the efficient training and evaluation of these models, making them feasible for real-world applications.

5.2 Comparison with Previous Research (Machine Learning for Thai Stock Prediction: A Data-Centric Approach, 24 July 2023)

The study Machine Learning for Thai Stock Prediction: A Data-Centric Approach employed a data-centric approach, focusing on techniques such as fractional differencing, feature noise reduction, and data augmentation to improve the performance of a Random Forest model. While these techniques effectively enhanced the model's predictive accuracy, our research demonstrates several advantages:

5.2.1 Model Complexity and Predictive Power:

Our hybrid model, which integrates XGBoost and Fin-GANs, is more sophisticated and capable of capturing complex, nonlinear relationships in financial data. In contrast, the Random Forest model in Machine Learning for Thai Stock Prediction: A Data-Centric Approach, while effective, is limited by its reliance on structured data and traditional feature engineering.

The ability of Fin-GANs to generate synthetic financial data and learn intricate temporal patterns provides a significant advantage over the data-centric techniques used in Machine Learning for Thai Stock Prediction: A Data-Centric Approach.

5.2.2 Factor Integration:

Our research emphasizes the integration of fundamental factors and technical indicators, which significantly enhance the predictive performance of both supervised and unsupervised models. This approach goes beyond the data-centric techniques in Machine Learning for Thai Stock Prediction: A Data-Centric Approach, which primarily focus on improving data quality and reducing noise.

By incorporating macroeconomic indicators and sentiment analysis, our models capture broader market dynamics, leading to more accurate and reliable predictions.

5.2.3 Risk-Adjusted Returns:

The hybrid model in our study achieved higher Sharpe ratios and lower max drawdowns compared to the Random Forest model in Machine Learning for Thai Stock Prediction: A Data-Centric Approach. This indicates that our approach not only improves predictive accuracy but also enhances risk-adjusted returns, making it more suitable for real-world investment strategies.

5.3 Limitations and Future Directions

While our research demonstrates significant advancements in stock market prediction, several limitations and areas for future research should be acknowledged:

5.3.1 Computational Resources:

The training and evaluation of Fin-GANs and the hybrid model require substantial computational resources and time. Future research could explore more efficient algorithms or distributed computing techniques to reduce these requirements.

5.3.2 Generalizability:

The study focused on five selected stocks from the SET50 index. To ensure the generalizability of our findings, future research should test the models on a broader range of stocks and market conditions, including different geographic regions and asset classes.

5.3.3 Alternative Deep Learning Architectures:

While Fin-GANs and XGBoost were effective, other deep learning architectures, such as Transformers or Reinforcement Learning models, could be explored to further enhance predictive performance.

5.3.4 Real-Time Applications:

The models developed in this study were evaluated using historical data. Future research could focus on real-time applications, such as high-frequency trading, to assess their performance in dynamic market conditions.

5.6 Conclusion

In conclusion, this research demonstrates the effectiveness of integrating supervised and unsupervised learning techniques, enhanced with fundamental factors and technical indicators, for stock market prediction. The hybrid model, XGBoost-Factor-Fin-GAN, significantly outperformed traditional methods,

providing superior risk-adjusted returns. Building upon its success, the Ensemble XB-F-Fin-GAN further improved predictive performance, achieving consistently strong cumulative profits across multiple stocks. This highlights the model's robustness and effectiveness in capturing complex market dynamics.

By leveraging the strengths of both XGBoost and Fin-GAN, our approach offers a highly reliable framework for financial forecasting. The findings underscore the importance of factor-based modeling and advanced deep learning techniques in enhancing both accuracy and efficiency in stock market predictions. Future research should explore ways to optimize computational efficiency and investigate alternative architectures to further refine these models for real-world applications.

This study contributes to the growing body of research on machine learning-driven financial forecasting, offering valuable insights into the trade-offs between different learning paradigms and the benefits of hybrid approaches. By continuously refining and expanding these methodologies, researchers and practitioners can unlock the full potential of machine learning in the financial sector.

References

Abdi, K., H. Rezaei, and M. Hooshmand (2024). Machine learning-based fundamental stock prediction using companies' financial reports. In 2024 32nd International Conference on Electrical Engineering (ICEE), pp. 1–5.

Bai, X., S. Zhuang, H. Xie, and L. Guo (2024). Leveraging generative artificial intelligence for financial market trading data management and prediction. Journal of Artificial Intelligence and Information 1, 32–41.

Brophy, E., Z. Wang, Q. She, and T. Ward (2021). Generative adversarial networks in time series: A survey and taxonomy. arXiv preprint arXiv:2107.11098 .

Cao, K. and H. You (2020). Fundamental analysis via machine learning. HKUST Business School Research Paper (2020-009).

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.

Das, A., K. Jain, D. Majumder, and R. Karmakar (2024). An ensemble learning paradigm for efficient stock market predictions. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–7.

Gifty, A. and Y. Li (2024). A comparative analysis of lstm, arima, xgboost algorithms in predicting stock price direction. Engineering and Technology Journal 9 (8), 4978–4986.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. Advances in neural information processing systems 27.

Gopal, A. (2024). Neuralfactors: A novel factor learning approach to generative modeling of equities. In

Proceedings of the 5th ACM International Conference on AI in Finance, pp. 99–107.

Hafid, A., M. Ebrahim, M. Rahouti, and D. Oliveira (2024). Cryptocurrency price forecasting using xgboost regressor and technical indicators. In 2024 IEEE International Performance, Computing, and Communications Conference (IPCCC), pp. 1–6.

Hirano, M., H. Sakaji, and K. Izumi (2023). Policy gradient stock gan for realistic discrete order data gen- eration in financial markets. In 2023 14th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), pp. 361–368. IEEE.

Hossain, S. and G. Kaur (2024a). Stock market prediction: Xgboost and lstm comparative analysis. In

2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), pp. 1–6.

Hossain, S. and G. Kaur (2024b). Stock market prediction: Xgboost and lstm comparative analysis. In

2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), pp. 1–6.

Huang, T.-L. (2019). Is the fama and french five-factor model robust in the chinese stock market? Asia Pacific Management Review 24 (3), 278–289.

Irejeh, E. M., L. E. Aninoritse, et al. (2024). Fama and french three factor model. European Journal of Accounting, Auditing and Finance Research 12 (5), 17–70.

Jiao, Y. and J. Jakubowicz (2017). Predicting stock movement direction with machine learning: An extensive study on s&p 500 stocks. In 2017 IEEE International Conference on Big Data (Big Data), pp. 4705–4713.

Krauss, C., X. A. Do, and N. Huck (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. European Journal of Operational Research 259 (2), 689–702.

Kwon, S. and Y. Lee (2024). Can gans learn the stylized facts of financial time series? In Proceedings of the 5th ACM International Conference on AI in Finance, pp. 126–133.

Lee, D. K. C., C. Guan, Y. Yu, and Q. Ding (2024). A comprehensive review of generative ai in finance.

FinTech 3 (3), 460–478.

Li, L., T.-Y. Chang, and H. Wang (2023). Multimodal gen-ai for fundamental investment research. arXiv preprint arXiv:2401.06164 .

Liu, T. and C. Li (2023). Research on multi-factor quantitative investment strategy of svm model based on machine learning. In Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering, pp. 654–659.

Liu, Y. and J. Wang. Analysis of financial market using generative artificial intelligence. Academic Journal of Science and Technology 11 (1), 21–25.

Maasoumi, E., J. Wang, Z. Wang, and K. Wu (2024). Identifying factors via automatic debiased machine learning. Journal of Applied Econometrics 39 (3), 438–461.

Manjunath, C., B. Marimuthu, and B. Ghosh (2024). Stock market prediction employing ensemble methods: the nifty50 index. Int J Artif Intell 13 (2), 2049–2059.

Ming, L. and G. Chen (2024). Stock price prediction based on relative strength index, moving average convergence divergence and xgboost model. In 2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS), pp. 1988–1993. IEEE.

Nti, I. K., A. F. Adekoya, and B. A. Weyori (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data 7 (1), 20.

Ozbayoglu, A. M., M. U. Gudelek, and O. B. Sezer (2020). Deep learning for financial applications: A survey. Applied soft computing 93, 106384.

Ramzan, F., C. Sartori, S. Consoli, and D. Reforgiato Recupero (2024). Generative adversarial networks for synthetic data generation in finance: Evaluating statistical similarities and quality assessment. AI 5 (2), 667–685.

Shah, D., B. Shah, H. Sawarkar, V. Raja, and A. Godbole (2023). Proposing investments based on fundamental analysis. In 2023 Global Conference on Information Technologies and Communications (GCITC), pp. 1–10.

Shah, J., D. Vaidya, and M. Shah (2022). A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Intelligent Systems with Applications 16, 200111.

Sharma, R. and K. Mehta (2013). Fama and french: Three factor model. SCMS Journal of Indian Management 10 (2), 90.

Sui, M., C. Zhang, L. Zhou, S. Liao, and C. Wei (2024). An ensemble approach to stock price prediction using deep learning and time series models. In 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 793–797.

Sun, H. (2024). An example of machine learning-based multifactor dynamic quantitative stock picking models. Science and Technology of Engineering, Chemistry and Environmental Protection 1 (7).

Vuleti´c, M., F. Prenzel, and M. Cucuringu (2024). Fin-gan: Forecasting and classifying financial time series via generative adversarial networks. Quantitative Finance 24 (2), 175–199.

Wang, J. and Z. Chen (2024). Factor-gan: Enhancing stock price prediction and factor investment with generative adversarial networks. Plos one 19 (6), e0306094.

Xu, J. (2023). Fundamental quantitative investment research based on machine learning. In SHS Web of Conferences, Volume 170, pp. 01019.

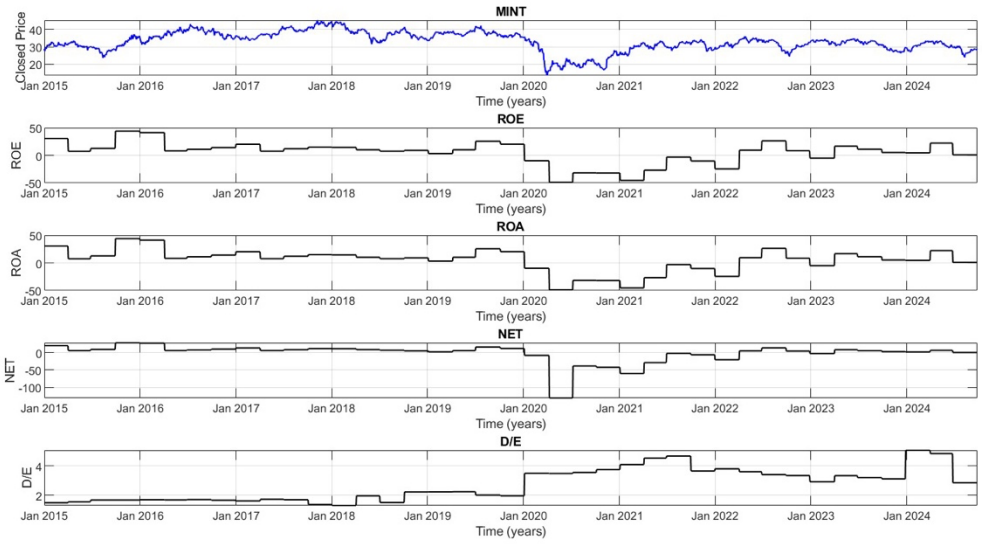
Ye, S. (2024). Applying ensemble learning to multiple stock price predictions: A comparative study. Applied and Computational Engineering 50, 189–198.

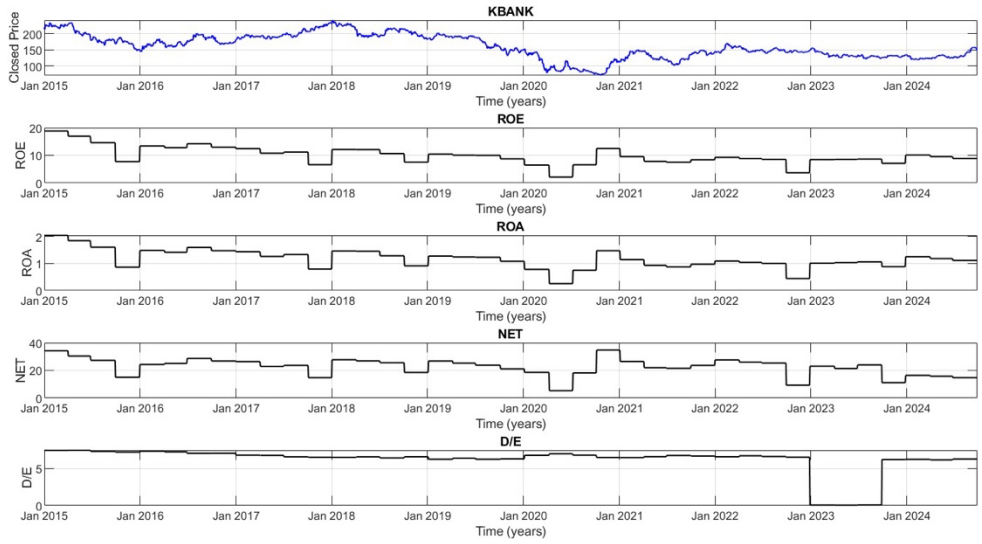
Zhang, H. (2024). The applicability of fama-french three-factor model to beijing stock market. Interna- tional Journal of Global Economics and Management 3 (1), 382–389.

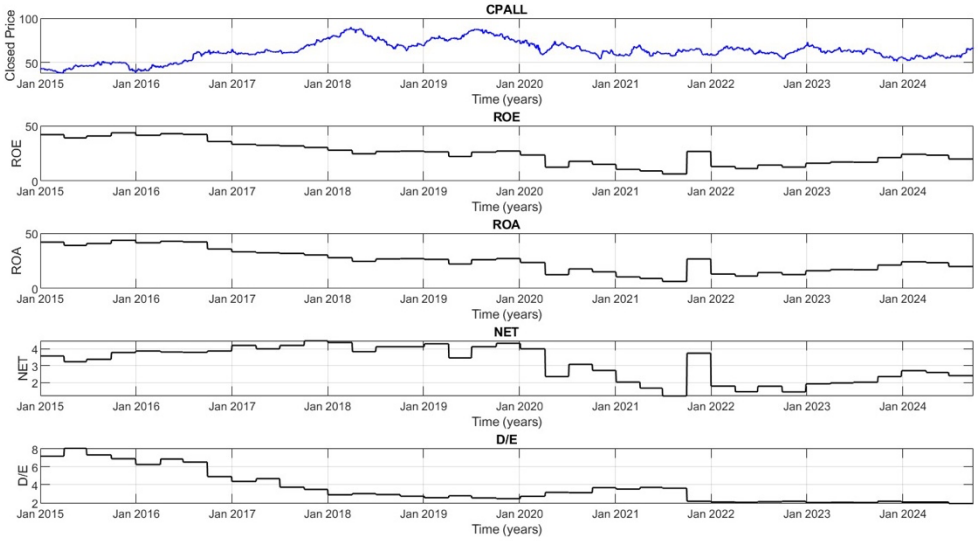
Zhu, S., C. Zhou, H. Liu, and Y. Ren (2024). Commodity factor investing via machine learning. Pacific- Basin Finance Journal 83, 102231.

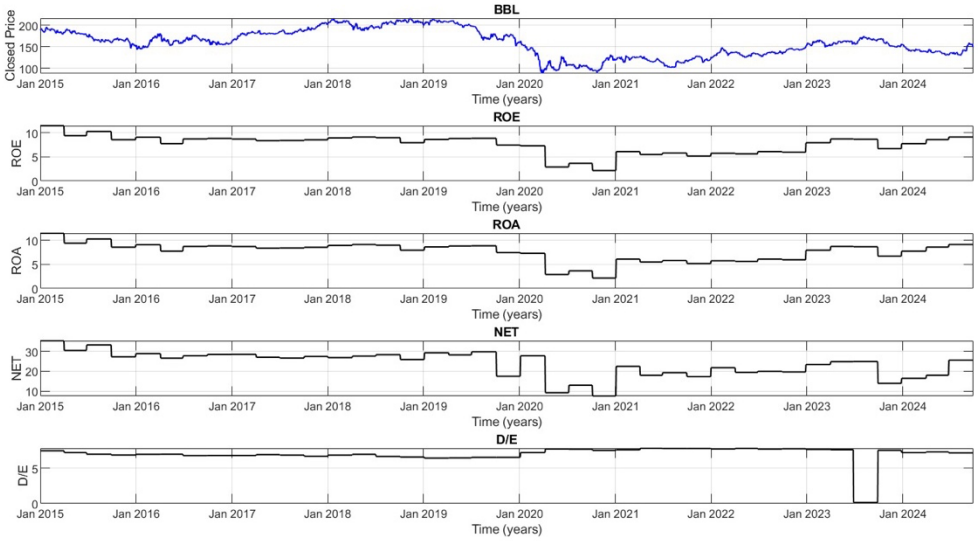
Appendix

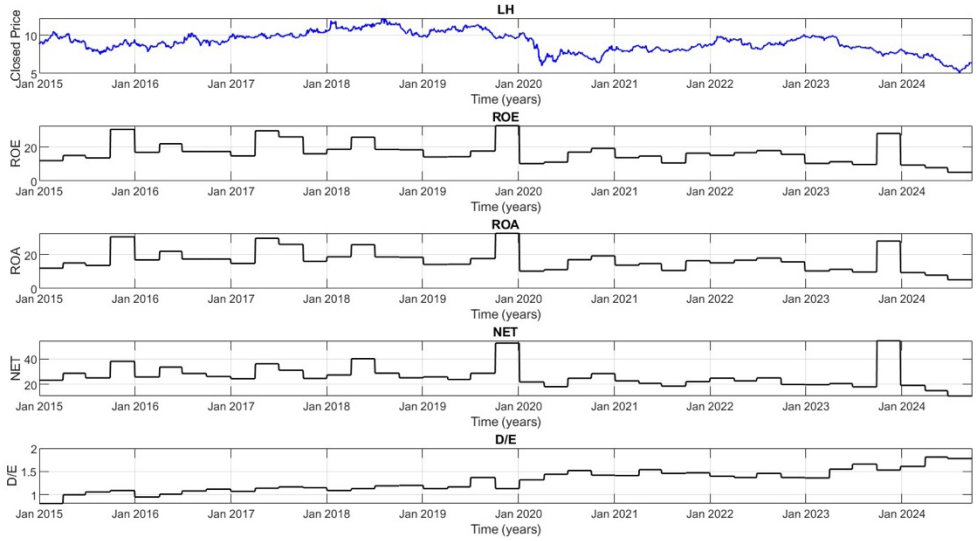
1. Close price and fundamental factor



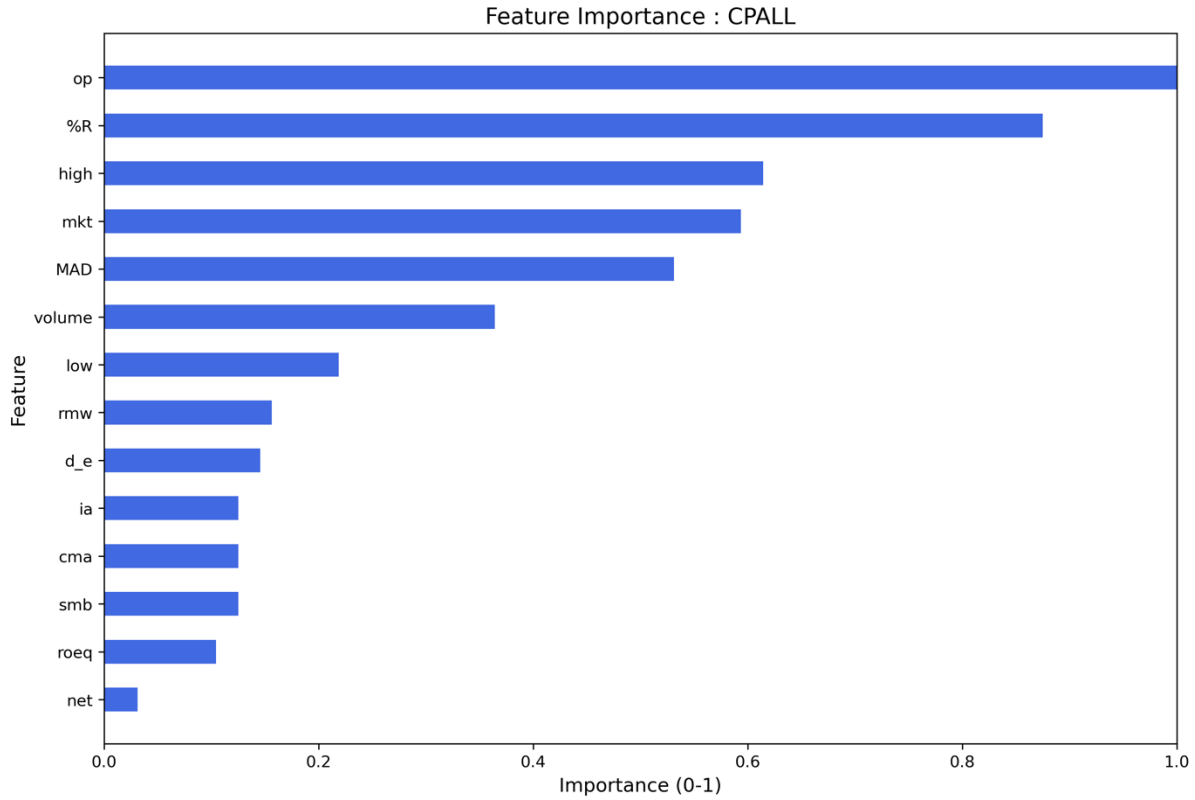


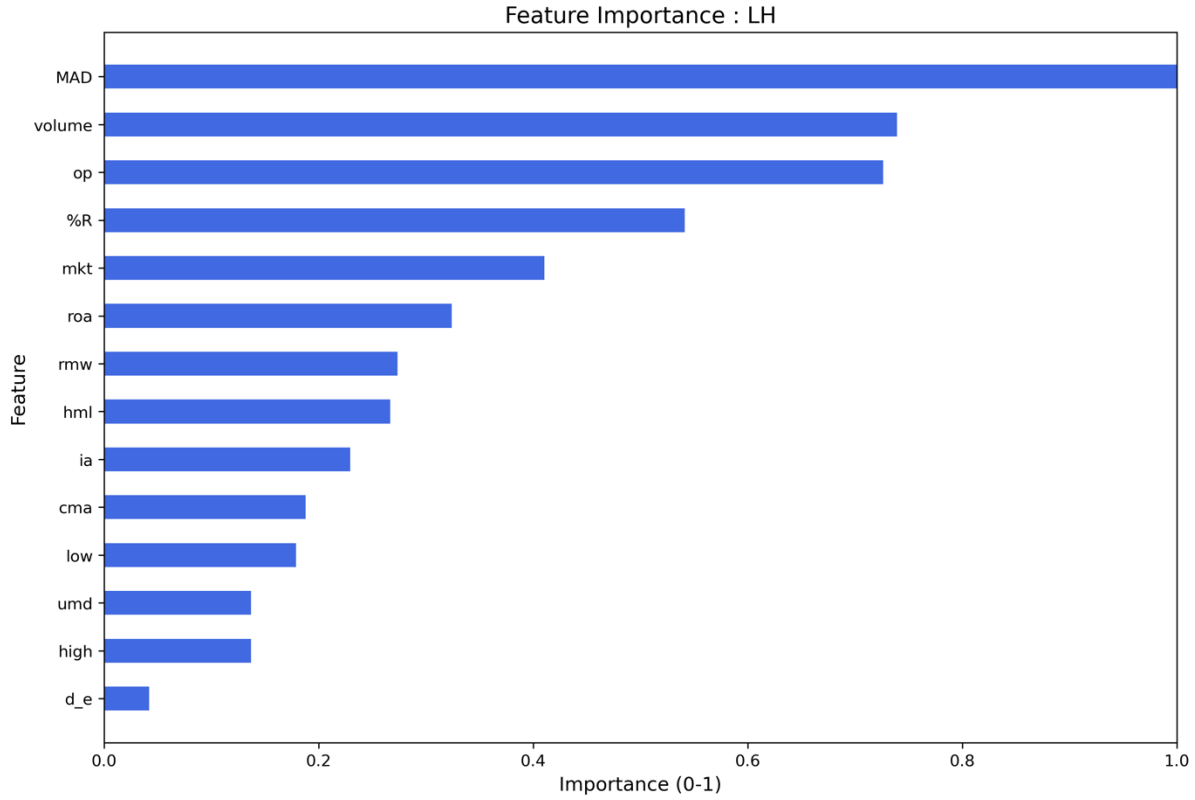

CPALL
Closed Price
ROE
ROA
NET
D/E
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019
Jan 2020
Jan 2021
Jan 2022
Jan 2023
Jan 2024
Time (years)

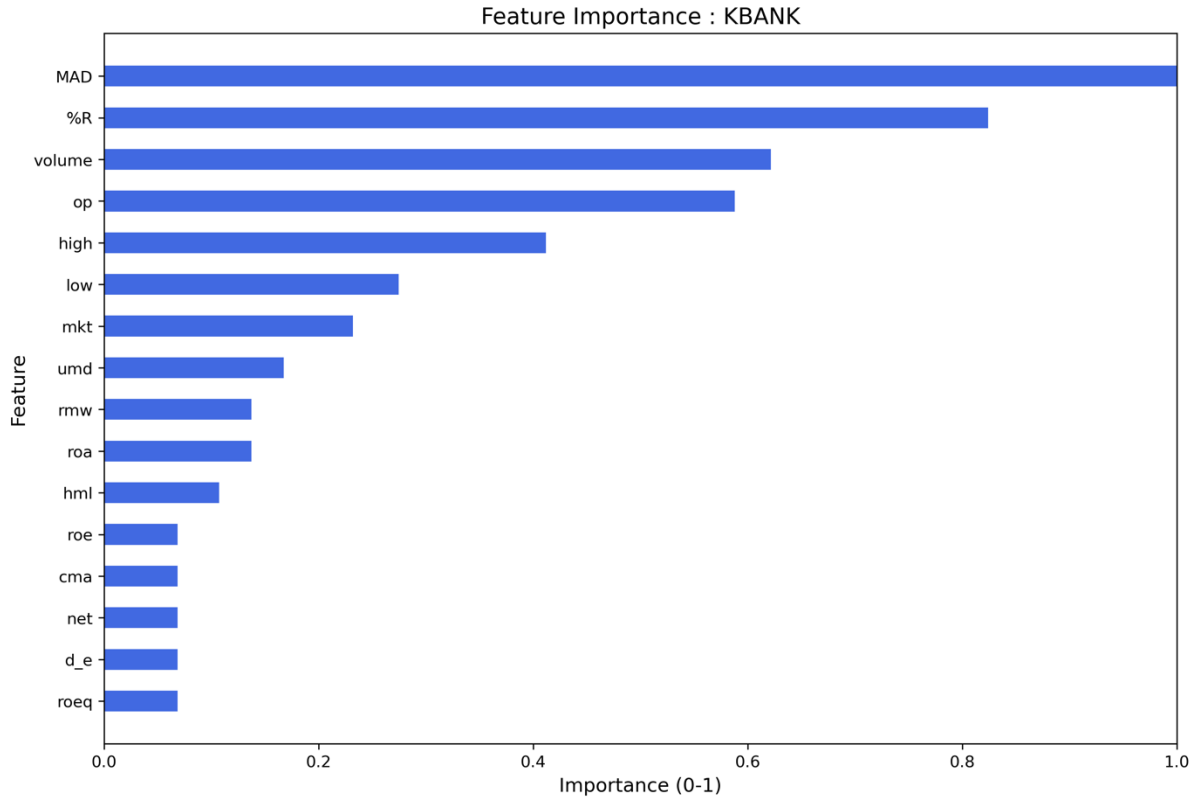

BBL
Closed Price
ROE
ROA
NET
D/E
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019
Jan 2020
Jan 2021
Jan 2022
Jan 2023
Jan 2024
Time (years)

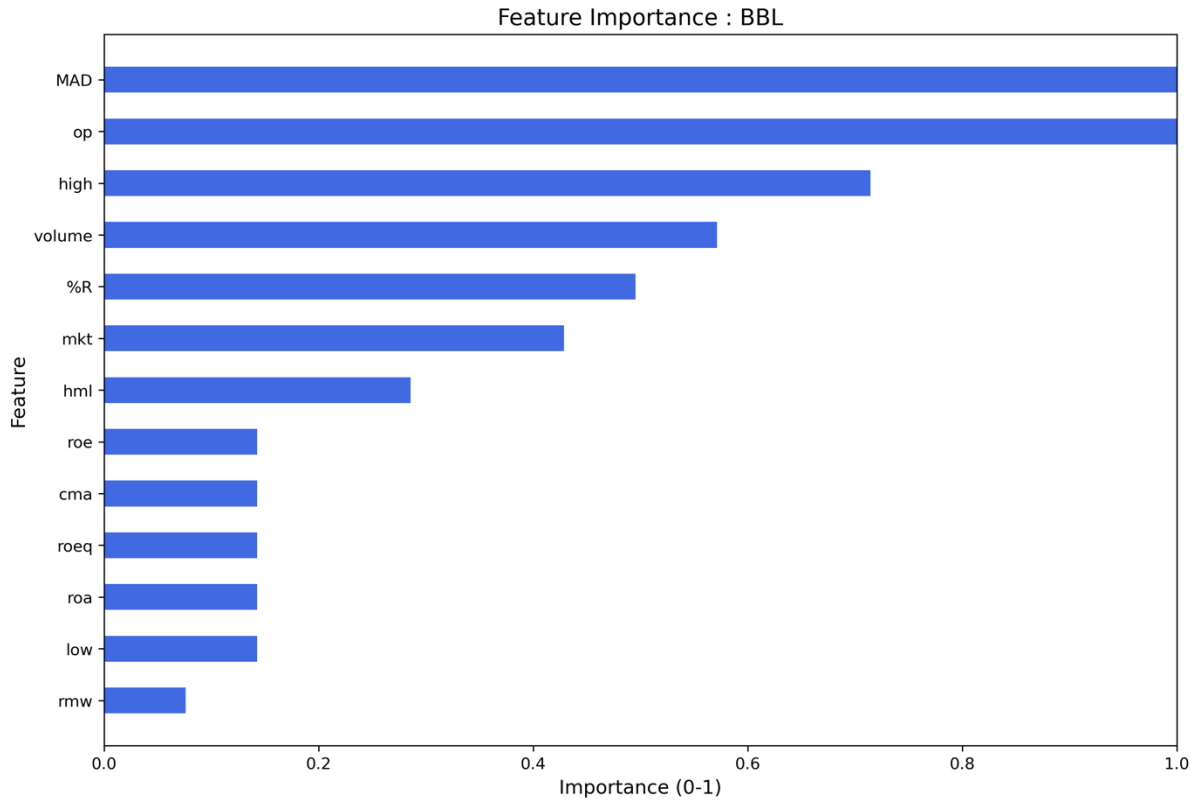


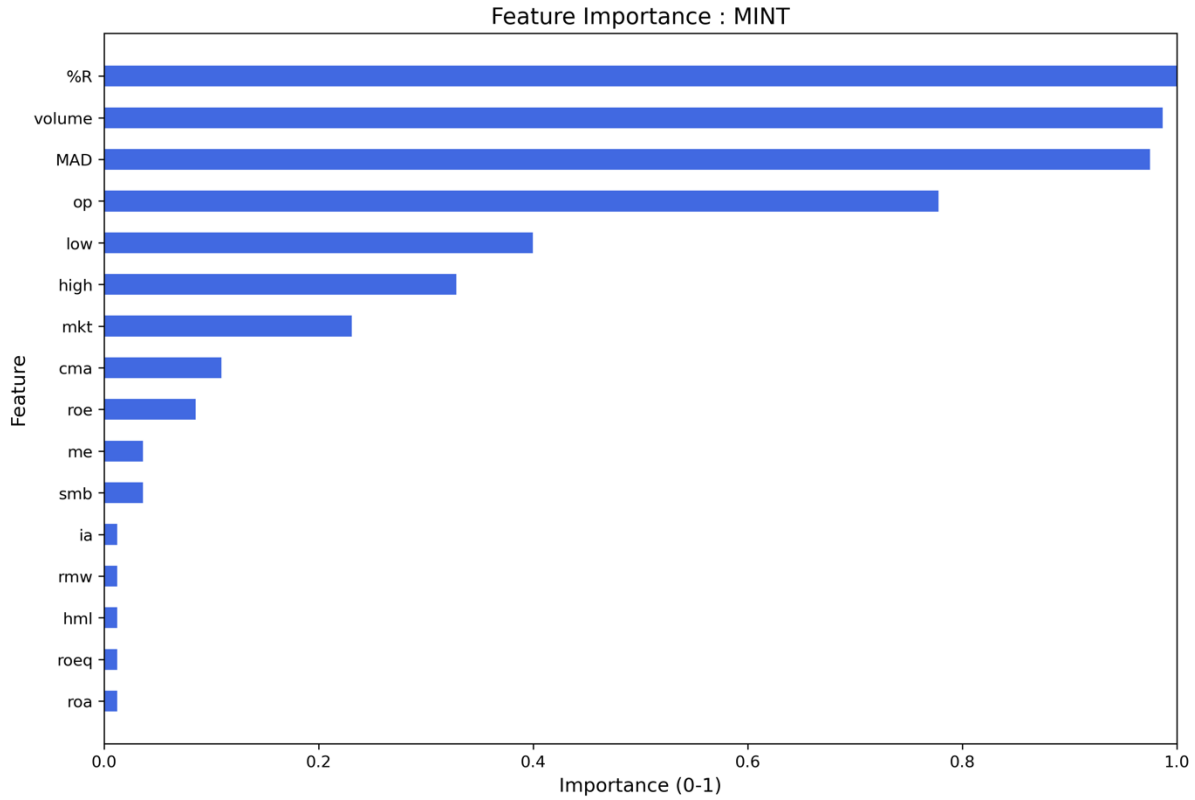
2. Feature engineering with feature important




Feature Importance : LH
MAD
volume
op
%R
mkt
roa
rmw
hml
ia
cma
low
umd
high
d_e
Feature
0.0
0.2
0.4
0.6
0.8
1.0
Importance (0-1)


Feature Importance : KBANK
MAD
%R
volume
op
high
low
mkt
umd
rmw
roa
hml
roe
cma
net
d_e
roeq
Feature
0.0
0.2
0.4
0.6
0.8
1.0
Importance (0-1)


Feature Importance : BBL
MAD
op
high
volume
%R
mkt
hml
roe
cma
roeq
roa
low
rmw
Feature
0.0
0.2
0.4
0.6
0.8
1.0
Importance (0-1)


Feature Importance : MINT
%R
volume
MAD
op
low
high
mkt
cma
roe
me
smb
ia
rmw
hml
roeq
roa
Feature
0.0
0.2
0.4
0.6
0.8
1.0
Importance (0-1)