Predicting Next-Day Returns of the S&P 500 Index:

A Comparative Study of Time-Series, Machine Learning, and Deep Learning Models

Author: Kwangmyung Lee

Table of Contents

Abstract

This study tests whether including fundamental, technical, and macroeconomic data can help predict the next-day return of the S&P 500 index.

To ensure a balanced comparison across different approaches, five models were used: a traditional time-series model (ARIMA), an econometric model (VECM), two machine learning models (Random Forest and XGBoost), and a deep learning model (LSTM).

The analysis was conducted using data from January 2020 to December 2024. Among the five models, the LSTM model showed better performance compared to the others, while the Random Forest model turned out to be the weakest in prediction accuracy.

However, even though the results show relatively strong performance for LSTM, it is difficult to conclude that LSTM is always the best model. The results can change significantly depending on the time period or market conditions. In addition, some models simply predicted values close to the average return rather than capturing actual variation, which suggests that short-term return prediction is still very challenging.

Overall, the findings indicate that next-day return forecasting is difficult, and the models used in this study do not provide consistently reliable predictions. However, the experiment was useful in comparing how each model behaves under the same dataset and conditions.

1. Introduction

There has been significant development in the field of stock price prediction. With advancements in computational power, large amounts of data can now be used to build models and evaluate whether those models actually work. Traditionally, technical indicators were used to study this topic. However, more recently, not only technical indicators but also macroeconomic data have been incorporated to improve model accuracy.

The purpose of this project is to use both fundamental data — including technical indicators and macroeconomic variables — to determine whether combining these different types of data actually improves predictive performance. The models tested in this project are the ARIMA model, VECM model, Random Forest, XGBoost, and LSTM.

2. Related work

A few research studies were reviewed to serve as references and to establish a baseline for our experiment. The selected papers include models from traditional statistical methods, machine learning, and deep learning, allowing us to create a balanced comparison across different modeling approaches.

Latif et al. (2025)

- Data Used: Technical indicators + macroeconomic variables (e.g., interest rates, CPI, oil prices)
- Models Applied: Random Forest, Support Vector Machine (SVM), Logistic Regression

Bhandari et al. (2022)

- Data Used: Historical S&P 500 closing prices + technical indicators (e.g., RSI, MACD)
- Models Applied: Long Short-Term Memory (LSTM) neural network

Barlybayev et al. (2025)

- Data Used: Financial data + macroeconomic indicators + technical variables
- Models Applied: ARIMA, Random Forest, Gradient Boosting, LSTM, Adaptive Neuro-Fuzzy models

3. Data Description & Preprocessing

The time horizon of this study is from January 1, 2020, to December 31, 2024. The data were collected from two primary sources.

The first source is Yahoo Finance, from which the following market-related variables were extracted: S&P 500 index, NASDAQ Composite, 10-year Treasury yield, Dollar Index, crude oil futures, gold futures, CBOE Volatility Index (VIX), silver futures, copper futures, and natural gas futures.

The second source is FRED (Federal Reserve Economic Data), from which we collected the Federal Funds Rate, CPI data, unemployment rate, University of Michigan Consumer Sentiment Index, and the St. Louis Financial Stress Index.

Once extracted, the data were merged into a single DataFrame with the S&P 500 index as the main reference. Macroeconomic indicators were forward-filled and shifted by either one week or two weeks to account for reporting delays.

In addition to the fundamental and macroeconomic data, technical indicators were computed using the S&P 500 price series. The technical indicators used in this study include SMA, EMA, RSI, MACD, and Bollinger Bands.

- SMA (Simple Moving Average): 5, 20, 60-day windows
- EMA (Exponential Moving Average): 12, 26-day windows
- RSI (Relative Strength Index): 14-day momentum indicator
- MACD (Moving Average Convergence Divergence): using 12, 26, and 9-day spans
- Bollinger Bands: 20-day mean ± 2 standard deviations

Finally, the target variable is the next-day return of the S&P 500, which was calculated using the percentage price difference from the previous day to compute the profit.

4. Methodology

In this research, five models were used to evaluate the prediction performance.
The ARIMA model was included as a traditional time-series forecasting method, which integrates autoregressive (AR), moving average (MA), and differencing components to model the behavior of the series. To select the optimal hyperparameters, a grid search was performed

based on evaluation metrics such as RMSE and MAE. The ARIMA model uses only the S&P 500 price data to estimate the next day's return.

The VECM model was used to incorporate macroeconomic factors into the prediction. Technical indicators were excluded from this model because the S&P 500 price level is already included, and adding derived technical variables may introduce additional noise. The optimal lag order was selected using the Akaike Information Criterion, and the model was then trained and tested based on the selected lag structure.

The Random Forest model was used to capture the non-linear relationships between variables. The data used in this model included fundamental variables, technical indicators, and macroeconomic factors.
To improve model performance, the hyperparameters were optimized using randomized search with time-series cross-validation. The parameters tuned include the number of trees, maximum depth, minimum samples per split, minimum samples per leaf, and the maximum number of features considered at each split.

The XGBoost model was included as a gradient boosting method, which places more weight on previous errors during training in order to improve prediction accuracy. This approach is different from Random Forest, which is based on bagging (bootstrap aggregation) rather than boosting. The objective function used in this model was squared error, and hyperparameter tuning was performed using randomized search with time-series cross-validation.

The LSTM model was used as the deep learning approach in this study. Based on previous research, LSTM has shown reasonable performance in financial time-series forecasting, which motivated its inclusion in this project. LSTM is particularly suitable for stock market data because its architecture is designed to capture sequential patterns and long-term dependencies, unlike traditional machine learning models. All input features were scaled using MinMaxScaler, and then reshaped into a 3-dimensional array so that the model could learn from 30 days of historical data to predict the next day's return. During early experiments, it was observed that without proper scaling or reshaping, the model performed worse than the other models

The final LSTM architecture used in this study is summarized below:

- LSTM layer (64 units, return_sequences=True)
- Dropout (0.2) for regularization
- LSTM layer (32 units)
- Dense layer (16 units, ReLU activation)
- Output layer (1 unit) for predicting the next-day return

5. Evaluation

To evaluate model performance, 80% of the dataset was used for training and the remaining 20% was used for testing. Since this study is based on time-series data, the train–test split was not done randomly. Instead, the first 80% of the data (earlier period) was assigned to the training set, and the most recent 20% was used as the test set, in order to prevent future information from leaking into past data.

Three evaluation metrics were used: RMSE, MAE, and Directional Accuracy. RMSE (Root Mean Squared Error) was selected because it penalizes large errors more heavily and represents

overall predictive accuracy. MAE (Mean Absolute Error) was used to measure the average magnitude of prediction errors. Directional Accuracy was included to evaluate whether the model correctly predicts the direction of the next day's return (up or down).

6. Results

The performance results of all five models were collected into a single DataFrame and visualized using bar charts for RMSE, MAE, and Directional Accuracy. The results show several notable patterns.

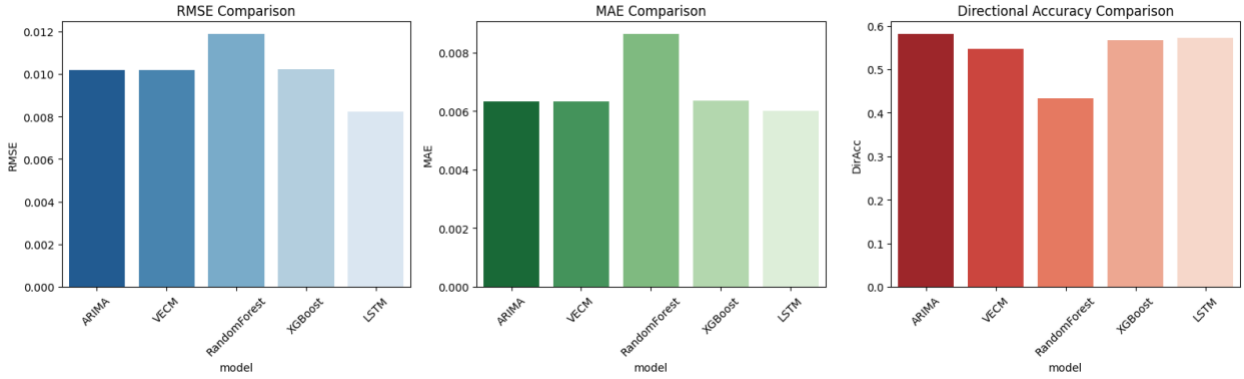


Figure 1. Performance Metrics Results

First, the LSTM model achieved the best overall performance, recording the lowest RMSE and MAE among all models. This indicates that the deep learning approach was able to capture non-linear patterns in the data more effectively than traditional statistical or machine-learning models.

On the other hand, the Random Forest model performed the worst, especially in terms of directional accuracy. Its directional accuracy was below 0.5, meaning the model predicted the wrong direction more often than a coin-flip baseline (50%). This suggests that the Random Forest model struggled to generalize when given a large number of mixed macroeconomic and technical features.

Another interesting observation is that the ARIMA model, which used only S&P 500 price data, performed similarly to the VECM model, even though VECM included additional macroeconomic indicators. This implies that including macroeconomic data does not necessarily guarantee superior performance, unless the model is able to exploit long-range dependencies effectively (as in LSTM).

The XGBoost model performed in between the two extremes. While its accuracy was better than Random Forest, its error levels were similar to the ARIMA and VECM models, suggesting that boosting alone was not sufficient to outperform deep learning in this dataset.

7. Additional Analysis on Model Behavior

To better understand why the models produced different levels of accuracy, an additional analysis was performed by visually comparing the actual next-day returns with the predicted returns from each model.

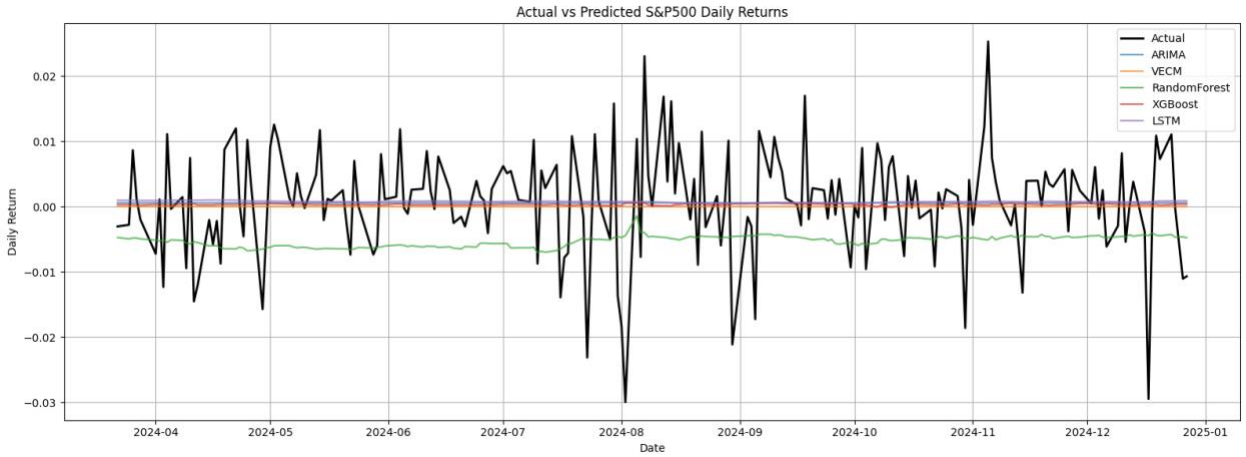


Figure 2. Actual vs Model Predicted

In the first plot, which overlays the actual returns with each model's predictions, it becomes clear that most models predict values very close to 0% daily return, resulting in almost flat prediction lines. This suggests that several models failed to learn meaningful variation from the data and instead converged toward predicting values near zero, which minimizes error but does not offer useful forecasting power. The only exception was the Random Forest model, which showed a strong downward bias.

To isolate the models' behaviors more clearly, the actual return line was removed in the second plot, and only the predicted returns were displayed. This makes the differences among the models more apparent:

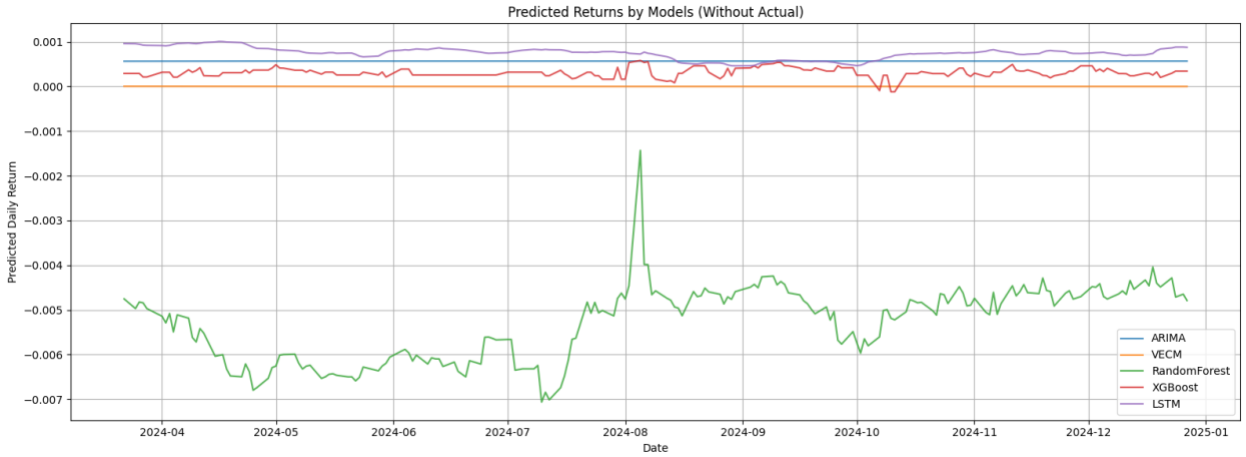


Figure 3. Model Prediction details

ARIMA and VECM generated almost completely flat predictions, meaning they effectively returned a constant value across the entire test period. This indicates that these statistical models were not capable of capturing the short-term volatility of the market under the given configuration.

Random Forest showed a consistently negative prediction pattern, implying that the model failed to balance the data distribution and leaned toward predicting declines, which explains its very low directional accuracy.

XGBoost and LSTM, in contrast, displayed meaningful variation across the prediction window. Although the error between predicted and actual values remained noticeable, these models at least captured some degree of upward and downward movement, which suggests stronger potential for improvement through tuning or additional feature engineering.

This visual inspection supports the numerical evaluation: traditional time-series models tended to predict near-zero returns, tree-based models performed poorly due to over-bias or under-fitting, and deep learning/boosting models demonstrated comparatively higher predictive flexibility.

8. Discussion

This study has several limitations that should be acknowledged.

First, the macroeconomic variables used in the model do not reflect the true dates on which market participants receive the information. Many economic indicators, such as CPI or unemployment data, are released with a delay of one or two weeks. Although this study shifted the data by 7–14 days to approximate the reporting lag, it still does not perfectly replicate how traders react to real-time macroeconomic news.

Second, the technical indicators used in this study were limited to a small set of commonly used features (SMA, EMA, RSI, MACD, Bollinger Bands). Because only a subset of possible indicators was selected, it is not possible to conclude whether technical indicators as a whole are ineffective. A more exhaustive feature set—or feature selection algorithm—may lead to different results.

Third, the study focused only on a single-day-ahead forecast of S&P 500 returns, rather than price levels. Many previous studies forecast the next day's price and report high accuracy, but this is fundamentally different from predicting returns. Price forecasting benefits from natural price continuity (tomorrow's price is usually close to today's), which makes models look more accurate than they actually are. In contrast, return data are stationary and more volatile, which makes prediction significantly harder and reveals the true difficulty of short-term market forecasting.

Additionally, the experiment was limited to five models representing traditional time-series, machine learning, and deep learning approaches. There exist many other models—such as Prophet, Transformer-based architectures, hybrid CNN–LSTM models, and reinforcement learning frameworks—that were not explored here. Therefore, the study does not fully conclude which model class is definitively optimal.

One interesting observation is that simply changing the date range of the experiment could lead to completely different model rankings. This suggests that the performance of forecasting models is strongly dependent on market regime (e.g., COVID crash, bull market recovery, high-interest-rate period). A more systematic approach—such as regime classification and model evaluation within each period—may provide deeper insight into when each model works best.

9. Conclusion

The study examined five different models over the period from January 2020 to December 2024 and found that the deep learning model (LSTM) outperformed both the traditional time-series models and the machine learning models. However, even with this result, we cannot conclude that LSTM is always the best model for predicting next-day returns, because its performance may change depending on the time period or market environment.

One important observation is that LSTM was able to capture non-linear patterns and showed meaningful variation in predictions, while several other models simply predicted values close to the average return and produced almost flat forecasts. Since the gap between the actual returns and the model predictions was still large, it is questionable whether these models can reliably predict next-day returns in real trading conditions.

Therefore, based on the experimental results, it is unlikely that next-day stock returns can be accurately predicted using the models tested in this study. However, the experiment provided useful insight into how each type of model behaves, which can help guide future research with more advanced architectures, better features, or regime-based analysis.

Reference

Latif, S., Aslam, F., Ferreira, P., & Iqbal, S. (2025). Integrating macroeconomic and technical indicators into forecasting the stock market: A data-driven approach. *Economies, 13*(1), 6. https://doi.org/10.3390/economies13010006

Bhandari, H. N., Rimal, B., Pokhrel, N. R., et al. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications, 9*, 100320. https://doi.org/10.1016/j.mlwa.2022.100320

Barlybayev, A., Ongalov, N., Milosz, M., Nazyrova, A., & Sembiyeva, L. (2025). Comparative analysis of classical, machine learning, deep learning, and adaptive neuro-fuzzy models for forecasting the S&P 500 index using financial, macroeconomic, and technical indicators. *International Journal of Innovative Research and Scientific Studies, 8*(6), 3286–3296. https://doi.org/10.53894/ijirss.v8i6.6277

Patsiarikas, M., Papageorgiou, G., & Tjortjis, C. (2025).
Using machine learning on macroeconomic, technical, and sentiment indicators for stock market forecasting. *Information, 16*(7), 584. https://doi.org/10.3390/info16070584

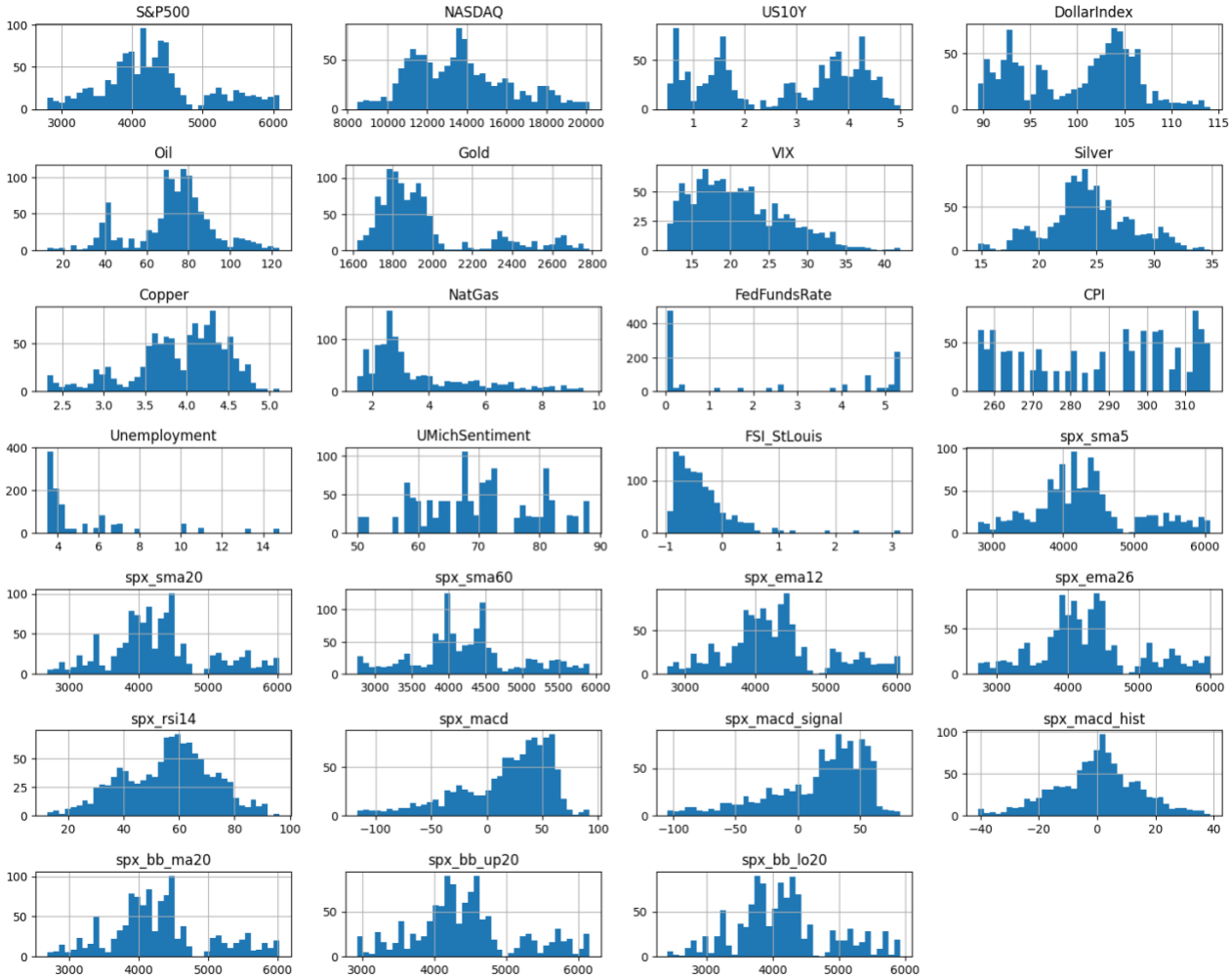
OpenAI. (2025, October). *ChatGPT (October 2025 version)* [Large language model]. OpenAI. https://chat.openai.com

Note: ChatGPT was used for grammar correction, sentence rephrasing, and improving clarity of English expression in this proposal.

Appendix A. Descriptive Statistics of Dataset (Jan 2020 – Dec 2024)

Variable	Count	Mean	Std	Min	25%	50%	75%	Max
S&P500	1122	4316.51	722.88	2797.80	3879.17	4220.28	4567.34	6090.27
NASDAQ	1122	13635.39	2475.86	8494.75	11670.36	13474.11	15120.13	20173.89
US10Y	1122	2.71	1.39	0.52	1.44	2.98	3.98	4.99
DollarIndex	1122	99.91	6.03	89.44	93.63	101.67	104.56	114.11
Oil	1122	72.88	19.21	12.34	66.11	75.31	82.82	123.70
Gold	1122	1958.45	265.35	1623.30	1788.70	1872.60	1978.78	2788.50
VIX	1122	21.02	5.82	11.86	16.46	20.03	24.74	41.98
Silver	1122	24.37	3.60	14.72	22.46	24.09	26.35	34.83
Copper	1122	3.91	0.56	2.32	3.62	4.04	4.33	5.12
NatGas	1122	3.66	1.89	1.48	2.40	2.86	4.57	9.68
FedFundsRate	1122	2.36	2.33	0.05	0.08	1.68	5.06	5.33
CPI	1122	288.03	20.19	255.80	266.63	294.94	306.14	316.45
Unemployment	1122	5.15	2.49	3.50	3.60	4.00	6.10	14.80
UMichSentiment	1122	70.05	9.15	50.00	62.80	69.40	77.20	88.30
FSI_StLouis	1122	-0.36	0.51	-0.96	-0.70	-0.48	-0.17	3.14
spx_sma5	1122	4311.66	721.75	2798.71	3866.56	4217.02	4561.21	6070.92
spx_sma20	1122	4293.34	719.04	2681.57	3879.11	4208.50	4533.64	6027.01
spx_sma60	1122	4245.36	708.23	2758.71	3873.84	4200.30	4527.94	5917.51
spx_ema12	1122	4303.13	719.40	2755.32	3875.79	4217.59	4535.02	6048.19
spx_ema26	1122	4285.99	715.57	2733.34	3887.11	4200.86	4542.07	6003.19
spx_rsi14	1122	56.06	15.92	12.40	44.18	57.48	66.90	96.23
spx_macd	1122	17.14	41.41	-116.36	-5.46	28.03	48.91	92.58
spx_macd_signal	1122	17.10	38.34	-104.48	-3.50	28.10	45.53	81.88
spx_macd_hist	1122	0.04	14.24	-40.85	-8.54	0.69	8.39	38.79
spx_bb_ma20	1122	4293.34	719.04	2681.57	3879.11	4208.50	4533.64	6027.01
spx_bb_up20	1122	4439.71	722.64	2934.82	4012.90	4364.33	4737.36	6159.78
spx_bb_lo20	1122	4146.98	720.65	2417.21	3726.83	4089.02	4405.48	5919.61

Appendix B. Input Data Histogram



Appendix C. Input Data Correlation Heatmap