QF604: Econometrics of Financial Market

Group Project Report

Prepared by:

Yu Lingfeng

Wang Wenjie

Wang Haotong

Chen Liangrui

Liu Jing

Prepared for:

Professor LIM Kian Guan,

Lee Kong Chian School of Business,

Singapore Management University

February 2024

1. Abstract

The paper we replicated is titled "Using machine learning to predict clean energy stock prices: How important are market volatility and economic policy uncertainty?" by Perry Sadorsky. It addressed the critical need for forecasting clean energy stock prices amid the disruptive impacts of climate change. The study employed machine learning techniques for "green" stock price prediction. The paper concluded that Random Forests, Extremely Randomized Trees, Gradient Boosting and Support Vector Machine outperform Lasso and Naïve Bayes in terms of prediction accuracy.

We have reconstructed all the models and forecast data mentioned in the reports and made some improvements. By employing a Linear Regression Model, we enhanced the accuracy of our predictions. Prior to this, we conducted Feature Engineering to prepare and preprocess the required data. The author of the paper we tried to replicate taking all models as classifier model, to retain more data information, we changed some of the models to become a regressor model. We believe that using linear regression rather than linear classification greatly improves the accuracy of our predictions of stock returns over the next 20 days.

2. Data

2.1 Green Energy Targets

The information utilized for this analysis is sourced from various platforms, encompassing a range of financial instruments and indicators related to the clean energy sector and broader economic factors. The period of interest is between 1st,July, 2008 and 30th, June, 2022.

The Green Energy Targets are obtained from three Exchange Traded Funds (ETFs): Invesco Wilder Hill Clean Energy ETF (PBW), iShares Global Clean Energy ETF (ICLN), and First Trust NASDAQ Clean Edge Green Energy ETF (QCLN). These ETFs provide a comprehensive view of the clean energy market.

As the paper concluded that all three targets produced comparable results, we selected PBW as the only replication target to avoid redundancy. In order to conclude more precisely, we have added in a final test on out-of-sample data set from 1st, Jul 2022 to 1st, Feb 2024

2.2 Original Feature Data

The original feature data includes information from Yahoo Finance and Bloomberg. From yfinance, we gather data on market volatility represented by VIX (S&P 500), VXN (NASDAQ 100) and OVX (Crude Oil). Bloomberg contributes data on Economic Policy Uncertainty (EPU), Economic Market Uncertainty (EMU), and the Infectious Disease Tracker (IDT), providing insights into broader economic and global health factors. We noticed that for EPU, EMU, and IDT data, there are around 50% more valid date tags, as those data are not only recorded in trading days. Thus, enormous difference might imply a mismatch of the feature.

2.3 Technical Indicators

Technical indicators such as Moving Averages (MA50 & MA200) and the Williams Accumulation/Distribution (WAD) are computed using stock price data. Research suggests that these indicators are relevant for price prediction, thus we made no significant changes (Bustos & Quimbaya, 2020) (Wang, Li, & Wu, 2020) (Yin & Yang, 2016). Other technical indicators are not to be adopted, as the hyperparameters are not specified in the paper.

2.4 Additional Feature Data for Linear Regression

Incorporating further features for a comprehensive analysis, yfinance contributes details on TNX (yield on U.S. Treasury bond index), SPX (S&P 500 index), N225(Nikkei 225 index), JPY=X (Japanese Yen to USD exchange rate), and EUR=X (Euro to USD exchange rate). From Bloomberg, we obtain data on the Hang Seng Index (HSI), and the Shanghai Composite Index (SCI), providing insights into global financial markets.

3. Machine Learning Methods

3.1 Random Forest and Extremely Randomized Trees

Random forests and Extremely Randomized Trees are ensemble methods based on decision trees. Decision trees, while accurate, can suffer from high variance. Ensembles address this by creating multiple decorrelated trees, with random forests using bootstrapping and feature randomness. Extra Trees, similar to random forests, select random features but also employ randomly chosen split values. This method does not use bootstrapping and creates a tree ensemble with less correlation.

3.2 Gradient Boosting

Boosting, on the other hand, sequentially adjusts a decision tree by fitting new trees to the residuals. This process continues until a stopping criterion is met, improving the fit gradually. Boosting involves tuning parameters like the number of trees, shrinkage parameter, and interaction depth.

3.3 Support Vector Machine

Support Vector Machine (SVM) partitions data into distinct groups using a hyperplane. SVM aims to find the maximum margin hyperplane that provides the greatest separation between data groups. Kernels are employed to map data into higher-dimensional space, potentially revealing nonlinear relationships.

3.4 Naïve Bayes and Lasso

Naïve Bayes is a probability-based classifier utilizing Bayes theorem, often employed as a benchmark. Lasso, a regression method, incorporates regularization and variable selection. It is similar to ridge regression but may set certain parameter values to zero, resulting in a sparse model that reduces variance and bias. As the report specifies, these method results serve as control group, for concluding on the efficiency of prediction models.

3.5 Linear Regression

This model establishes a linear relationship between the target variable and selected features. It assumes a straightforward connection between these features and the direction of stock price changes. The model's simplicity allows for easy interpretation and provides insights into underlying patterns. Feature choices significantly impact prediction accuracy, and different combinations are explored. This is our model of choice in predicting the asset price movement, due to the consistency in performance and potential in feature engineering.

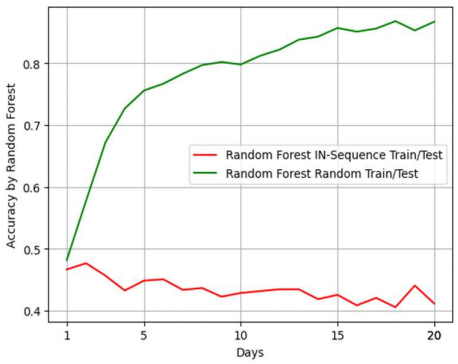
4. Result Replication

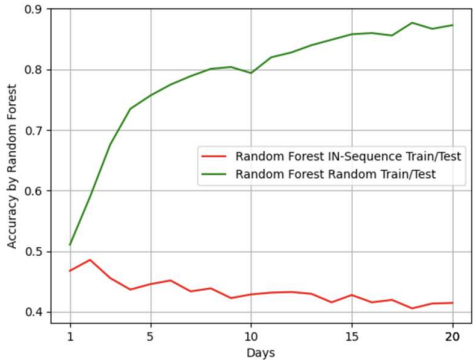
According to author's description, "the data were randomly split into a training set consisting of 70% of the data and a testing set consisting of 30% of the data". Although we do not agree with such Train/Test partition for time series, we decided to replicate the procedure first. Overall, we obtained similar level of prediction accuracy as that of the paper, which is around 90%, as shown by the green plots.

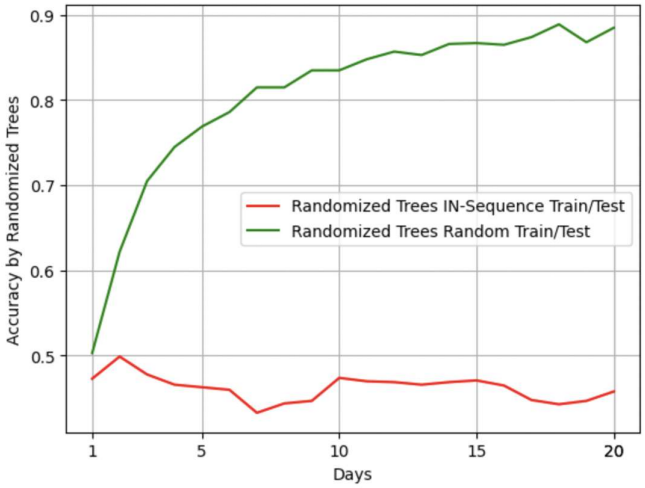
For comparison, we also used the same model parameters on an in-sequence train/test data partition, and the accuracy is much lower, as shown by the red plots. We can also see that all the Machine Learning models result in generally worse predictions than Naïve Bayes predictions.

4.1 Random Forest and Extremely Randomized Tree

We have set the number of trees in the forest to 100 in the first call and 500 in the second call. The randomness of the estimator, or the seed, was set to 1 to ensure result reproducibility. The number of features and the maximum depth of the tree were set to None.

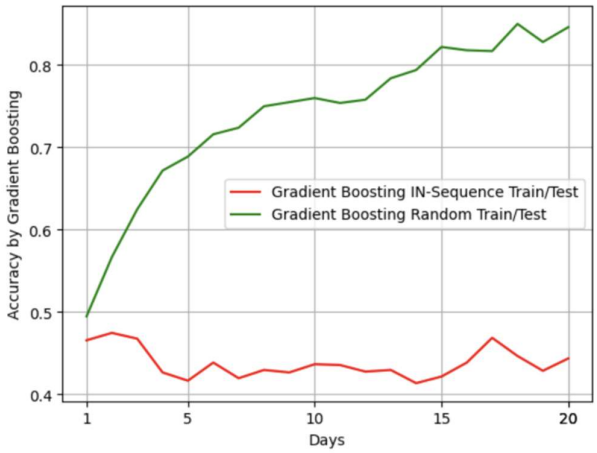






4.2 Gradient Boosting

For the gradient boosting mode, the number of boosting stages is set at 500, the square root of the total number of features will be considered at each split, learning rate as 0.05 and the maximum depth of the tree set to 4 is imposed on individual trees to limit their complexity and prevent overfitting to the training data.



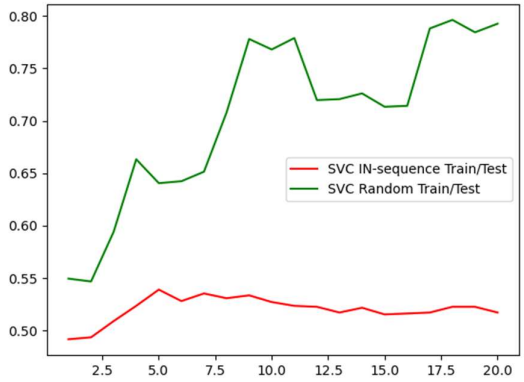
4.3 Support Vector Machine

For SVM, the author has used ten-fold cross validation with 10 repeats. The SVM model utilized a radial basis function ("rbf" kernel) and was fine-tuned with two parameters: cost and gamma. A grid search method was applied to determine the optimal values for each forecast. The grid for the cost parameter encompassed a range of values such as 0.1, 1, 10, 100, and 1000, while the grid for the gamma parameter spanned values including 0.2, 0.5, 1, 2, 3, and 4. The replicated accuracy is around 80%, which is slightly lower than the author's but there's meaningful resemblance.

However, we know that cross-validation on time series information is not the best practice. Sequential partition, or making sure every test data is after all of the training

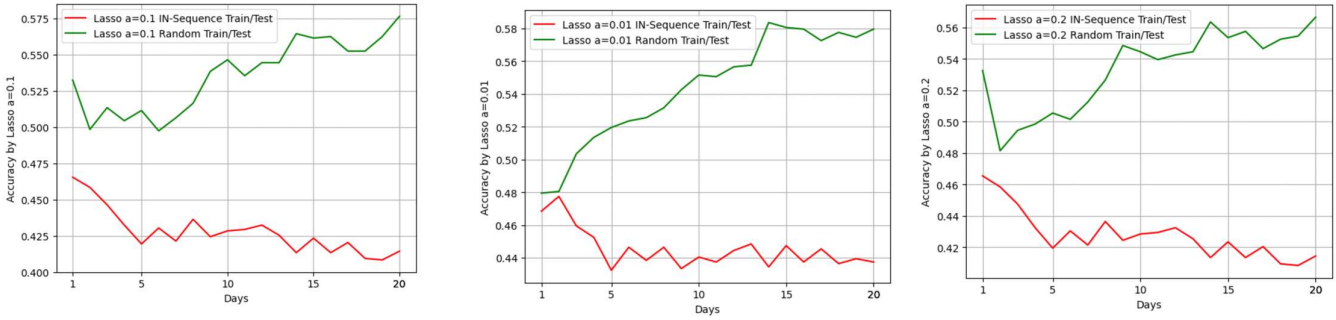
data, is a more theoretically sound procedure, The red plot is obtained is conducted with exactly the same steps, except how train and test data is obtained.

The above replication shows that, while it is possible to obtain a result similar to that of the paper, the accuracy would be much lower if the author used the correct method (in-sequence train/test split)



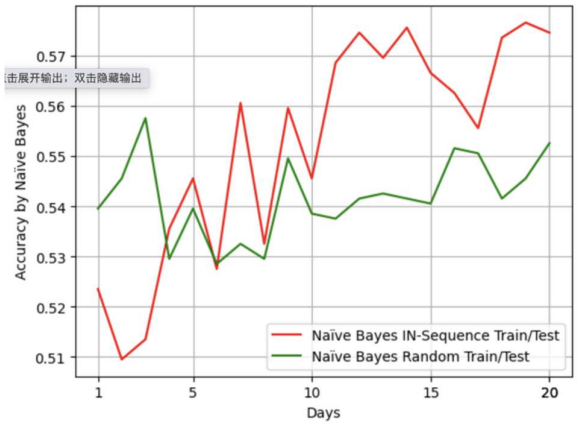
4.4 Lasso

To assess the performance of the Lasso regression model with different values of the regularization parameter (alpha). The Lasso model is instantiated three times with different values of alpha: 0.01, 0.1, and 0.2.



4.5 Naïve Bayes

Naïve Bayes algorithms are specifically designed for classification tasks, and it is important to note that Naïve Bayes classifiers work with binary labels. Therefore, the target variable needs to be converted into binary labels, typically represented as 1 and -1. However, when we did the conversion, some log return of stock price (in decimal) may be considered "lost" because the model's output consists of a probability distribution rather than deterministic predictions. This means that the model's output is not entirely definitive but rather an estimate based on observed data and prior probabilities. Consequently, while Bayesian classifiers provide probability distributions of class labels as outputs, there is also the potential for information to be "lost," resulting in reduced predictive accuracy of the model.



5. Feature Engineering for Both Replication and Improvement:

5.1 Original Feature and Data Preparation:

The original feature combination is based on our best guess. Before we replicated the result of the paper, we treated the original feature data. The treatments include merging multiple datasets, handling missing values, splitting the data into features and targets, defined regression models instantiated each model with specific hyperparameters.

5.2 Extended Feature Engineering:

EX1 to EX7 represent the extended engineered features we created to enhance the Linear Regression Model's predictive capabilities. EX1 captures the lagged percentage changes in stock prices. EX2 calculates the log returns of the PBW adjusted closing prices. EX3 represents the second moment of the lagged percentage changes captured in EX1. EX4 mirrors EX3 but focuses on the log return of the PBW stock. EX5 characterizes the lagged percentage changes in volatility levels. EX6 is the second moment of the volatility series. EX7 mirrors EX6 but focuses on the lagged percentage changes in volatility levels.

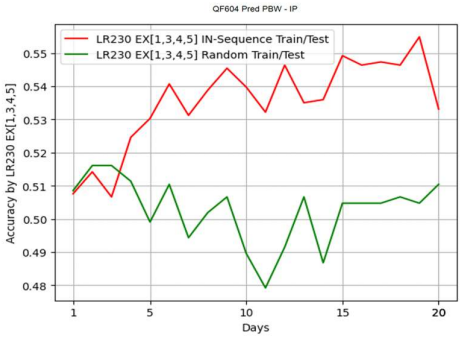
6. Improvement by Linear Regression

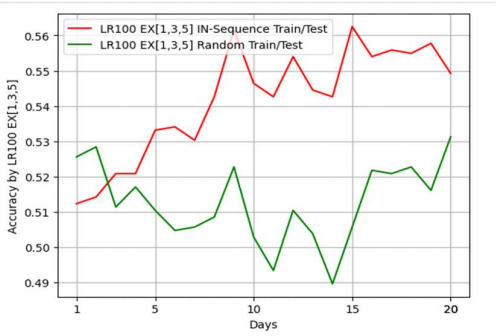
6.1 Linear Regression Tuning

A comprehensive analysis of the Linear Regression model's performance was conducted by exploring different combinations of 9 features and selecting the optimal combination. Note that EPU, EMU, and IDT data were excluded due to the improper match on date tags.

The dataset is divided into training and test sets, and the initial performance of the Linear Regression model using all available variables is evaluated. Also, correlation analysis on the training data to identify effective predictors, a mean correlation cut off is determined for selecting the most influential variables. Therefore, this methodology enables a comprehensive assessment of the Linear Regression models predictive capabilities across diverse input variable combinations, aiding in the identification of the most influential predictors for the target variables. We have generated 511 combinations in total and the best two performances of features combinations as shown

below:



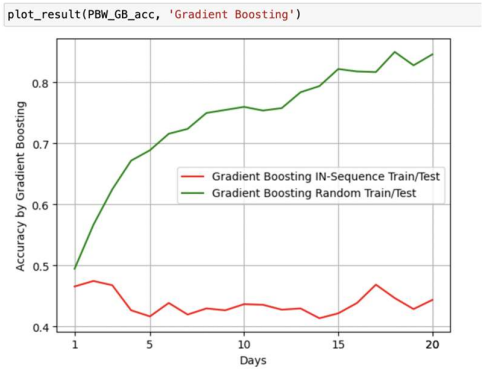


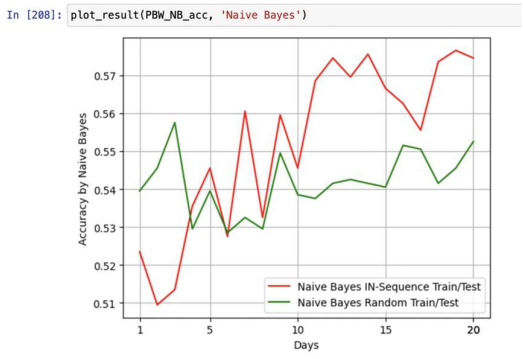
7. Conclusion:

7.1 Train/Test Effects

The author's choice of train/test split with time series proves to be a crucial factor influencing the performance of machine learning models. Most of the green plots have much higher prediction accuracy than the red plots. But we question the theoretical soundness of such practice.

For example, comparing Gradient Boosting and Naive Bayes Models, we can clearly see that when we split the training and testing datasets according to sequential order, the performance of Naive Bayes Model (red plot on the right) is significantly better than that of Gradient Boosting (red plot on the left). If we take the Naïve Bayes model as a benchmark stated by the author, these plots fully demonstrate the instability of the Gradient Boosting model under the effects of training/ testing set splits.





7.2 Features Set Effects

The selection of features in Linear Regression also emerged as a pivotal factor contributing to prediction accuracy. The careful consideration and tuning of features are essential for improving model performance.

The extension of the final model testing beyond the report period aimed to observe its effects over time. However, it is noted that the last testing window exhibits a strong downward trend, potentially limiting the generalization of testing results.

The best linear regression model, number 230, with feature combinations EX1, EX3, EX4, EX5, generally demonstrates improved accuracy. Although the accuracy does not always surpass 50%, a peak around the 15 days forecast window achieves an accuracy of 55%, aligning with in-sample testing.

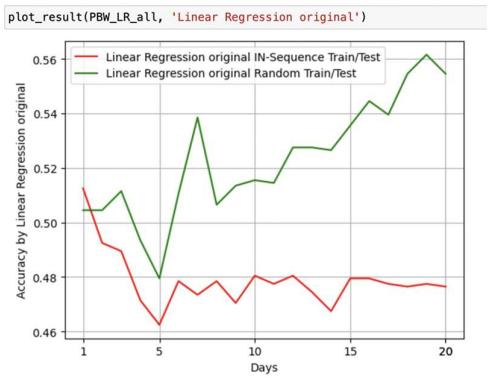
7.3 Overall

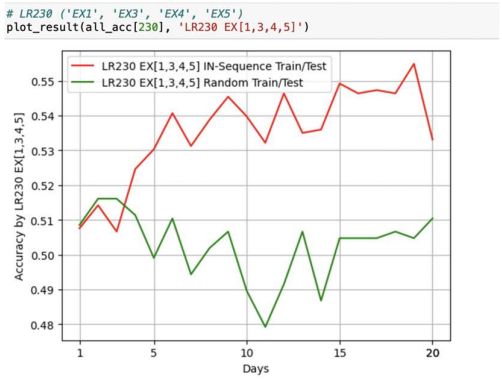
The conclusion of the paper we replicate regarding high performance is deemed inaccurate due to the improper partitioning of training and testing dataset. At the same time, the assessment of feature significance remains inconclusive, reflecting the improper training of the model.

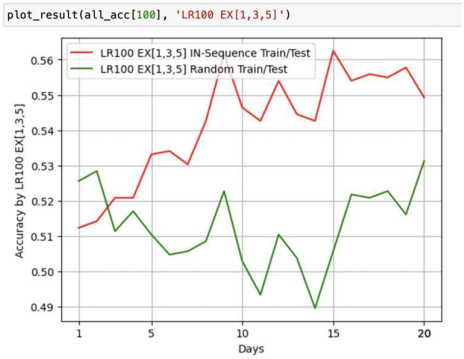
Due to the variations in date coverage, certain features such as EPU, EMU, IDT exhibit improper significance. The feature IDT is susceptible to look-forward bias.

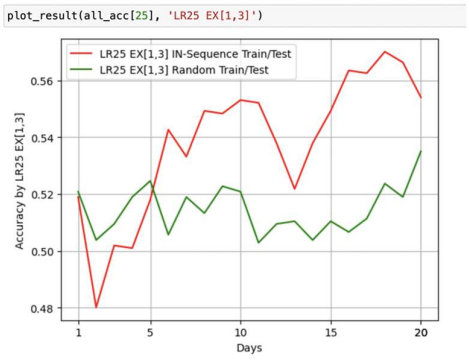
Most Technical Indicator Features have a high degree of freedom, and are subject to dividend effects, thus could not be replicated with existing information. This is especially noteworthy, since TA may be subject to overfitting.

From the plots below, in the Linear Regression model, the prediction accuracy based on the features originally selected by the author is much lower than the additional features we included.



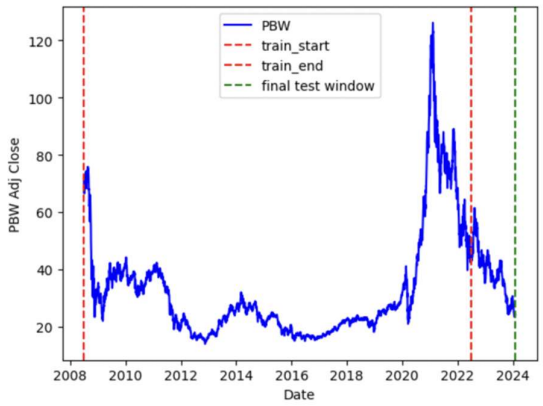






7.4 Final Model Testing with Out-of-Sample testing (2022-07-01 to 2024-02-01):

We have imported and set the stock price from 2022-07-01 to 2024-02-01 as out-of-sample for final model testing.



Naïve Bayes is not perform as well as it did earlier, it serves as a benchmark for comparison with other models.

Gradient Boosting stands out as one of the top performing machine learning algorithms, unlike the claim in (Sadorsky, 2021), which suggests Random Forest to be the top performing model.

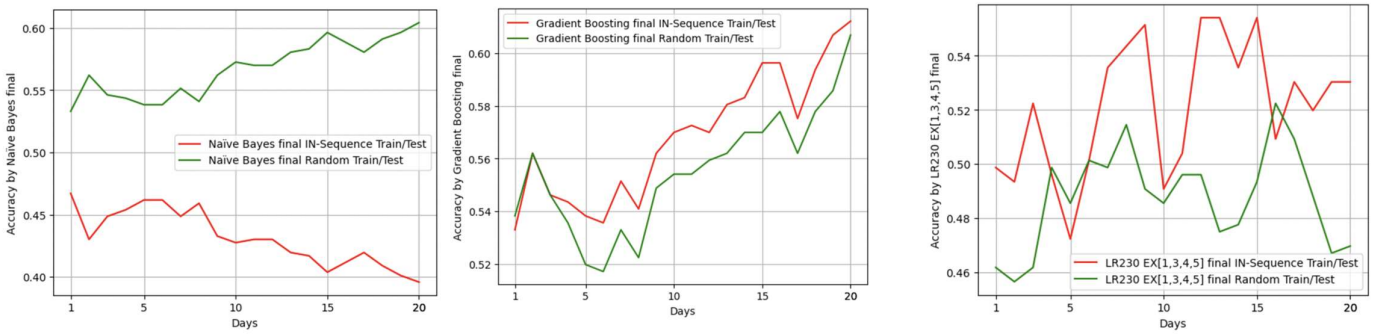
In-depth analysis reveals:

a. The In-Sample test with random partition gives 80% accuracy; in-sequence partition gives around 45%.

b. The Out-of-Sample test with random partition gives 60% accuracy; in-sequence partition gives around 60%.

c. The performance dropped significantly, and implied inconsistency.

d. The result is better than final model testing, yet this could be due to the strong trend within the testing window, or in another words, market regime bias.



In summary, by replicating the authors' approach and improving on it we find that the authors have a problem with the division of the dataset, resulting in the test set containing a large amount of information from the future, which leads to an exceptionally high accuracy in the final prediction, which is somewhat out of line with reality.

We believe that the correct approach should be to divide the dataset in chronological order, select stable models such as linear regression, SVM for prediction, choose appropriate feature engineering and conduct out of sample test.

8. References

Bustos, O., & Quimbaya, A. (2020). Stock Market Movement Forecast: A Systematic Review. *Expert Systems with Applications*, 156.

Sadorsky, P. (2021). A random forests approach to predicting clean energy stock prices. *Journal of Risk and Financial Management*, 14.

Wang, Y., Li, L., & Wu, C. (2020). Forecasting commodity prices out-of-sample: Can technical indicators help. *International Journal of Forecasting*, 666.

Yin, L., & Yang, Q. (2016). Predicting the oil prices: Do technical indicators help. *Energy Economics*, 56.