Essays on forecasting time series with machine learning techniques

by

Ali Lashgari

B.S., Amirkabir University of Technology, 2013
M.S., Amirkabir University of Technology, 2015
M.S., Kansas State University, 2022

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Economics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Abstract

This dissertation consists of three essays on forecasting time series with machine learning techniques, delving into different financial and economic domains. The first essay investigates the forecasting accuracy of the S&P 500 stock market index during the pandemic, utilizing text mining and technical analysis. It finds that the LSTM model, which uses numerical data rather than financial news, offers superior accuracy in predicting price movements, showing the effectiveness of machine learning over traditional analysis methods.

The second essay tests the hypothesis that the Federal Reserve responds to data revisions when setting monetary policy. To deal with the large number of data revisions that the Federal Reserve can potentially respond to, we use four machine learning techniques, Lasso, ridge regression, elastic net, and post-Lasso. When using our preferred method, elastic net, we conclude that the Federal Reserve responds to five types of data revision. We discuss the implications of this finding for theoretical macroeconomic models in the context of the signal extraction problem.

The third essay addresses the predictability of West Texas Intermediate crude oil prices, influenced by macroeconomic factors. By comparing the performance of Long Short-Term Memory (LSTM) and Random Forest (RF) models, they are superior capability in both short and long-term forecasts during significant economic shocks like the 2008 financial crisis and the COVID-19 pandemic. The inclusion of SHAP analysis further enriches the understanding of how historical prices and macroeconomic indicators like the Consumer Price Index and exchange rates play pivotal roles in forecasting.

Essays on forecasting time series with machine learning techniques

by

Ali Lashgari

B.S., Amirkabir University of Technology, 2013
M.S., Amirkabir University of Technology, 2015
M.S., Kansas State University, 2022

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Economics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Co-Major Professor
Lance Bachmeier

Approved by:

Co-Major Professor
Yoonjin Lee

Copyright

Abstract

This dissertation consists of three essays on forecasting time series with machine learning techniques, delving into different financial and economic domains. The first essay investigates the forecasting accuracy of the S&P 500 stock market index during the pandemic, utilizing text mining and technical analysis. It finds that the LSTM model, which uses numerical data rather than financial news, offers superior accuracy in predicting price movements, showing the effectiveness of machine learning over traditional analysis methods.

The second essay tests the hypothesis that the Federal Reserve responds to data revisions when setting monetary policy. To deal with the large number of data revisions that the Federal Reserve can potentially respond to, we use four machine learning techniques, Lasso, ridge regression, elastic net, and post-Lasso. When using our preferred method, elastic net, we conclude that the Federal Reserve responds to five types of data revision. We discuss the implications of this finding for theoretical macroeconomic models in the context of the signal extraction problem.

The third essay addresses the predictability of West Texas Intermediate crude oil prices, influenced by macroeconomic factors. By comparing the performance of Long Short-Term Memory (LSTM) and Random Forest (RF) models, they are superior capability in both short and long-term forecasts during significant economic shocks like the 2008 financial crisis and the COVID-19 pandemic. The inclusion of SHAP analysis further enriches the understanding of how historical prices and macroeconomic indicators like the Consumer Price Index and exchange rates play pivotal roles in forecasting.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my profound gratitude to my co-major professors, Dr. Yoonjin Lee, and Dr. Lance Bachmeier, for their support, encouragement, and guidance throughout this journey. I am also deeply thankful to the rest of my committee members, Dr. Behzad Ghanbarian and Dr. Steven Cassou, for their feedback and support. I extend my heartfelt appreciation to all my colleagues, friends, and the faculty at Kansas State University. Their support has made my graduate school experience more enjoyable.

Lastly, but most importantly, I want to thank my parents for their boundless love, unwavering support, and constant inspiration. Words cannot express my sincere gratitude for all they have done for me throughout my life. My deepest appreciation goes to my amazing wife, Elaheh, for her endless love, support, and patience, and to my sister, who has been my supporter and hero.

Dedication

To my lovely family

Chapter 1 – Forecasting Stock Market Return During COVID-19 Pandemic via Text Mining and Technical Analyses

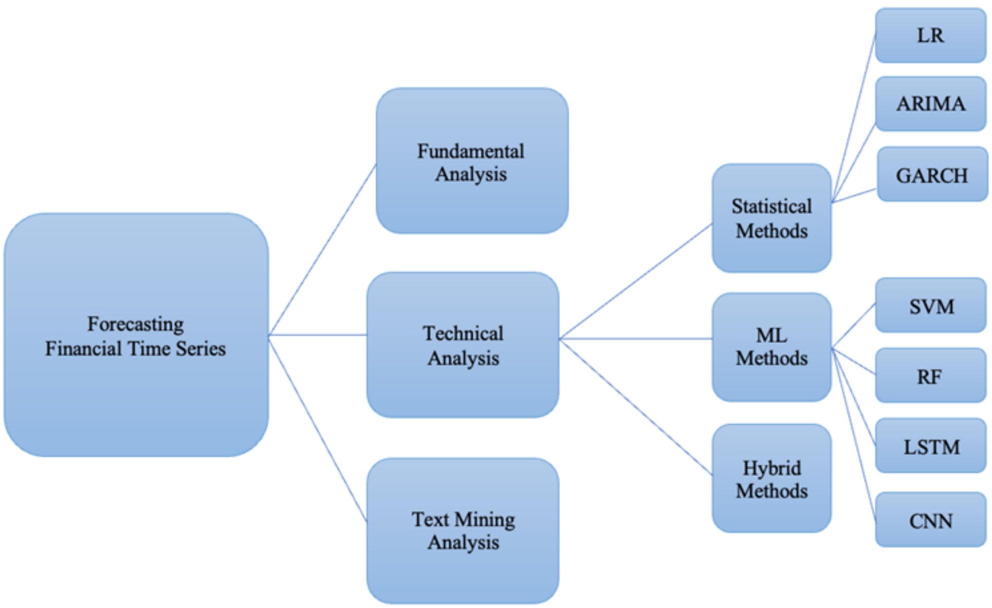
1.1 Introduction

Stock return predictability is deeply explored through two perspectives in finance (Dong et al., 2022.). Cross-sectional analysis examines if characteristics of firms can predict differences in stock returns, with notable studies by Fama & French (2015), Harvey et al. (2016), Mclean & Pontiff (2016), and Hou et al. (2020) identifying various market anomalies. Concurrently, time-series research delves into forecasting the overall market's future performance by evaluating economic and financial indicators like valuation ratios and interest rates, as discussed by Nelson (1976), Fama & French (1989), and Pastor and Stambaugh (2009) to comprehend the equity risk premium's determinants.

Recent advancements in machine learning have led to the development of algorithms like gradient boosted regression trees, artificial neural networks, random forests, and support vector machines. These algorithms excel in identifying intricate patterns and relationships, including non-linear and context-specific interactions that linear approaches often miss. Additionally, they are particularly adept at handling data with multicollinearity, outperforming traditional linear regression models in such scenarios. Machine learning methods are tailored to enhance prediction accuracy outside the sample set in complex data scenarios by preventing overfitting. Moreover, these methods, such as neural networks, are adept at handling nonlinear relationships within the predictive models (Han et al., 2023). In enhancing out-of-sample cross-sectional return forecasts, the adoption of machine learning methodologies is crucial because it is instrumental in mitigating overfitting concerns, underscoring the significance of sophisticated statistical techniques in the analysis of financial data.

Forecasting financial time series is another crucial perspective in finance and economics that predicts potential risks, market trends, and investment timing in financial markets. In order to analyze and predict financial market behavior, fundamental and technical analysis have been used before 2010. Fundamental analysis is a method that uses economic and financial information about a company to predict its future performance. This includes analyzing financial statements, industry trends, and macroeconomic conditions to determine a company's value and potential for growth. Technical analysis, on the other hand, uses historical price and volume data to identify trends and make predictions about future market behavior. In technical analysis, traditional statistical methods like the autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and autoregressive conditional heteroscedastic model (ARCH) have been the mainstay for modeling financial time series. While these approaches have seen widespread use, their interpretability remains challenging, making it difficult for users to grasp the rationale behind their predictions (Xie et al., 2021). Modern neural network models, despite their complexity, offer improved accuracy due to their intricate architectures; however, their "black box" nature impedes understanding of their internal learning processes, posing a dilemma for human investors (Linardatos et al., 2020). Text mining, also known as text data mining, is another way to approach forecasting time series. This method involves the use of Natural Language Processing (NLP) techniques to extract meaningful insights from unstructured text data (Figure 1.1). NLP, as a field of computer science and artificial intelligence, has been around since the 1950s, but it was not until the mid-2010s that NLP algorithms started to achieve significant breakthroughs in processing and understanding human language. It utilizes machine learning and natural language processing (NLP) techniques to analyze vast amounts of textual data, such as news articles, social media posts, and earnings call transcripts. The goal of text mining analysis is to identify relationships between the sentiment expressed in textual data and the movements in the stock market (Yun et al., 2023).

Figure 1.1. Forecasting Financial Time Series Methods



This study investigates the accuracy of different statistical methods and text mining for predicting S&P 500 (time-series category) using financial news and historical price data during COVID-19 pandemic. By comparing these two different methodologies, we can determine which approach is more reliable in real-world problems - historical prices or news.

This essay compares the accuracy of different technical methods and text mining for predicting S&P 500 using financial news and historical price data. One of the text mining methods called BERT, is a language model based on the transformer architecture, and another is FINBERT, which has been pre-trained on a large corpus of financial text and fine-tuned on a dataset of financial news articles, are used in this essay. The results show that LSTM can effectively learn the underlying patterns in financial data and make accurate predictions. The Long Short-Term Memory (LSTM) model has demonstrated its capacity to decipher complex patterns within financial datasets, leading to precise predictions. When assessing its performance through the lens of Root Mean Square Error (RMSE), the LSTM model achieves the minimum average RMSE across various training and testing datasets. This performance is representing a marginal 0.7% improvement over the XGBoost model, and significantly outperforms the Support Vector Machine (SVM) model, translating to a considerable 16% reduction in error.

The essay contributes to the literature on forecasting time series in some dimensions. First, it makes an equal comparison between classical and modern models in forecasting financial time series. Second, it introduces the comparison of BERT and FinBERT models in text mining during COVID-19. Furthermore, comparing these two different methodologies, technical and text mining analysis, it is possible to determine which approach is more reliable in real-world problems - historical prices or news. This comparison can help investors decide which method to use when predicting future values in unpredictable events like covid shock.

The following is how the essay is structured. The related work is described in the section that follows. The proposed methodology, evaluation metrics, and data set are all covered in Section 3. The experimental setup and results are shown in Section 4. Section 5 finally exposes our conclusions and future work.

1.2 Literature Review

Finance literature has deeply investigated stock return predictability through two key lenses. The cross-sectional perspective, highlighted by Fama & French (2015) and others, examines how specific characteristics of firms may influence stock returns, identifying a range of market anomalies. Complementing this, time-series analysis, with seminal contributions from Nelson (1976) and Fama & French (1989) probes the future performance of the overall market by evaluating economic and financial indicators to understand equity risk premiums.

The introduction of machine learning (ML) methods represents a significant advancement in predictive accuracy, particularly for complex data scenarios that traditional models struggle with due to overfitting. These ML methods, such as neural networks, excel in discerning nonlinear relationships within predictive models. Among the various ML methodologies, the Least Absolute Shrinkage and Selection Operator (LASSO), introduced by Tibshirani (1996), is notable for its efficacy in estimating high-dimensional cross-sectional regressions, thereby enhancing predictive accuracy while controlling for overfitting. Beyond LASSO, other techniques like forecast combination, principal component regression, and the innovative C-ENet approach have proven effective in out-of-sample market return forecasts, offering considerable statistical and economic validation. Rapach & Zhou (2020) explore the out-of-sample predictability of the aggregate stock market return, affirming that employing advanced ML methods can significantly improve predictability, even in challenging contexts.

Studies such as those by Dong et al. (2022) bridge the gap between cross-sectional and time-series analyses by using ML to enhance prediction accuracy. Dong and colleagues found that anomaly portfolio returns are significantly predictive of market excess returns, especially during periods of high market volatility. Gu et al. (2020) further emphasize the potential of ML in finance, demonstrating its superiority in measuring equity risk premiums over traditional methods.

To answer the question "Is out-of-sample predictability achievable?" the literature offers diverse viewpoints. Rapach (2022) explores the out-of-sample predictability of the aggregate stock market return by enhancing the traditional predictive regression approach with machine learning techniques. These innovations include methods like forecast combination, principal component regression, and the LASSO to handle large predictor sets and noisy data efficiently. Their findings affirm that employing these advanced methods can significantly improve the predictability of market returns, even in challenging out-of-sample contexts, showcasing the potential of blending financial theory with machine learning for more accurate market forecasts. The study by Welch and Goyal (2008) critically examines the out-of-sample predictability of stock returns. It focuses on comparing forecasts that incorporate new predictor variables against a benchmark forecast. The forecasting process uses historical data to predict future returns without looking ahead, thus avoiding bias. They analyze whether expanding or rolling data windows are more effective for predictions. Their findings suggest that despite intuitive appeal, changing parameters over time can be challenging, often making an expanding window approach more practical due to its balance of bias and efficiency. The study by Martin & Nagel (2021) critically reassesses traditional market efficiency tests in the context of big data's challenges. They emphasize that the vast number of variables available for forecasting has fundamentally altered the landscape, making traditional tests less relevant. By modeling a high-dimensional prediction scenario, they demonstrate that even when investors use machine learning techniques like shrinkage (ridge regression) or sparsity (Lasso) optimally, asset prices still exhibit in-sample predictability due to learning-induced errors and the regularization process. However, this predictability does not extend out-of-sample, suggesting that traditional in-sample efficiency tests may often reject the efficient market hypothesis in high-dimensional settings, even when investors are optimally using information. Their work underscores the importance of out-of-sample tests in evaluating market efficiency in the age of big data, as these tests retain economic meaning by reflecting the genuine unpredictability of returns when investors face a high-dimensional prediction problem.

Cochrane (2011) argues that when faced with numerous predictive variables, traditional regression and portfolio sorting may fall short. Machine learning emerges as a solution, adept at handling vast predictor sets and adapting to varied functional forms through regularization techniques to prevent overfitting and select suitable models. However, the opacity of deep learning models, often labeled as "black boxes," poses a challenge. Karolyi & Van Nieuwerburgh (2020) stress the importance of comprehending the economic underpinnings behind machine learning for achieving reliable out-of-sample predictions.

Han et al. (2023) introduce a machine learning-enhanced Fama-MacBeth regression for cross-sectional return forecasts, addressing big data's challenges. This work has two key economically interpretable components. First, cross-sectional regressions relating individual returns to lagged characteristics have been estimated. Second, the cross-sectional slope coefficient estimates, as characteristic payoffs. By incorporating LASSO and random features, they refine prediction accuracy and include non-linearities. Their approach, tested on over 200 firm characteristics, significantly outperforms traditional methods, offering valuable economic insights.

Perform a comparative analysis of machine learning methods for empirical asset pricing, focusing on equity risk premiums. They test various machine learning models, including trees and neural networks, and find these methods significantly improve risk premium measurement over traditional approaches, doubling the performance of leading regression-based strategies. The study uses a vast dataset of nearly 30,000 stocks over 60 years, with predictors including stock characteristics, macroeconomic variables, and industry sectors. The research underscores machine learning's potential in enhancing understanding and prediction of asset returns, justifying its growing role in finance and fintech industries.

Avramov et al. (2023) explore machine learning for stock return predictability, focusing on deep learning's capacity to aggregate multiple anomalies for identifying mispriced stocks. The paper highlights the economic impact of machine learning on stock return predictability, emphasizing deep learning's ability to identify mispriced stocks, especially in complex market conditions. It notes that while machine learning enhances profitability, its efficacy is challenged by economic constraints and trading costs. The analysis explores a broad range of machine learning models, including neural networks and beta pricing models, to assess stock return predictability. It delves into different machine learning techniques applied to a vast dataset of U.S. stocks spanning from 1987 to 2017, aiming to discern the economic viability of machine learning-based investments under realistic market constraints.

Malladi (2024) states that in traditional programming, the process involves developers writing specific instructions or "rules" in a code form, which are then tested against data to determine their effectiveness. If the rules do not yield accurate results, they are modified or discarded. This approach has given rise to many methods like multiple regression over time. Human intuition plays a crucial role in deciphering because some rules work better than others. In contrast, machine learning (ML) automates the rule-finding process. ML systems learn from provided data and corresponding outcomes, identifying patterns to predict new data points. In ML, especially in unsupervised learning, the system discovers patterns and groupings within the data itself, which is particularly useful for processing videos, images, and text. With supervised learning, human feedback classifies the accuracy of machine predictions, enabling the system to refine its algorithms to reduce errors. However, the rationale behind the machine-generated rules, especially in complex models like Support Vector Machine (SVM) or AdaBoost, may not be readily explainable. Furthermore, there is a significant paradigm shift in allowing machines to self-learn and potentially eclipse human roles in economic and financial forecasting, which presents a conceptual obstacle that can be more daunting than the technical learning curve (Malladi, 2024).

Word embedding is a technique in natural language processing (NLP) for representing words as numerical vectors. The goal of word embedding is to capture the meaning of words in a way that can be used as input to machine learning models. Each word in a text corpus is represented as a high-dimensional vector, where each dimension corresponds to a particular semantic or syntactic property of the word. The values of the vector are learned from the training data, and the vectors are designed to capture the relationships between words in the corpus. For example, words that are semantically similar are expected to have similar vectors, while words that are semantically dissimilar are expected to have dissimilar vectors. There are several types of word embedding techniques that are commonly used in natural language processing (NLP). Some of the most popular ones include:

- Word2Vec: Published in 2013 by Tomas Mikolov et al.
- GloVe: Published in 2014 by Jeffrey Pennington et al.
- FastText: Published in 2016 by Piotr Bojanowski et al.
- ELMo: Published in 2018 by Matthew Peters et al.
- BERT: Published in 2018 by Jacob Devlin et al.
- GPT: Published in 2018 by Radford et al.

The study by Atkins et al. (2018) provides valuable insights into the predictive power of financial news compared to the close price of assets or indexes. The authors use machine learning models to analyze news feeds and predict the direction of asset price movement and asset volatility movement. Their findings reveal that news-derived information is a better predictor of market volatility than the close price of an asset or index, with an average directional prediction accuracy of 56%.

The study by Souma et al. (2019) explores the use of deep learning for sentiment analysis in financial news, with a focus on defining polarity based on stock price returns after the release of news articles. The authors report that their methodology, which utilizes a combination of recurrent neural network with long short-term memory units, shows improved forecasting accuracy when selecting news with the highest positive and negative scores as positive and negative news, respectively. They suggest several avenues for future research, including exploring different methods for defining polarity and using different deep learning methodologies. Overall, the findings of this study have significant implications for the field of financial forecasting and demonstrate the potential of deep learning methods in enhancing sentiment analysis.

The paper by Guo & Tuckfield (2020) investigates the effectiveness of news-based machine learning and deep learning methods in predicting stock indexes and individual stocks. The study compares the performance of machine learning and deep learning approaches using news data processed through natural language processing. The results suggest that deep learning outperforms other machine learning methods in predicting both stock indexes and individual stocks, with a 4.5% improvement for stock indexes and a 3% improvement for individual stocks. The authors discuss the implications of these findings and suggest future areas of research, such as using other types of data and optimizing the model for stronger generalization ability. This study highlights the potential of deep learning in predicting stock market trends and underscores the importance of data processing and model optimization in achieving accurate results.

The paper by Jena & Majhi (2023) investigates the role of Twitter sentiments during the COVID-19 pandemic as a crucial determinant for predicting stock market movements. Employing a machine learning approach, the study utilizes long short-term memory (LSTM) networks and traditional time-series models such as autoregressive moving average (ARMA) and linear regression. Focusing on various sectors in the United States and India, the research demonstrates that LSTM outperforms time-series models in predicting stock values. Noteworthy is the finding that socio-political sentiments, particularly those expressed on Twitter, significantly influence stock market fluctuations. The study highlights variations in performance across sectors, emphasizing the heightened impact of sentiments during the pandemic on the Finance and Consumer Goods sectors. The research underscores the relevance of non-monetary factors, such as social media sentiments, in understanding and predicting stock market behavior, advocating for further exploration of neural network models in diverse sectors for enhanced accuracy and interpretability.

Lin et al. (2022) investigate the factors influencing text mining-based stock prediction, specifically examining text feature representations, machine learning models, and the impact of different news platforms. The study employs eight combinations of two context-free (TF-IDF, Word2Vec) and two contextualized (ELMo, BERT) text feature representations, coupled with three learning techniques (SVM, CNN, LSTM). Results indicate that CNN combined with Word2Vec and BERT performs the best across daily, weekly, and monthly stock predictions, surpassing traditional SVM methods. The research emphasizes the influence of the selected news platform on prediction models and notes the importance of financial word count and sentiment scores in news articles for effective stock prediction. The paper suggests future considerations, including sentiment word nets, industry-specific analyses, integration of image-based features, and investigations into optimal news platforms and contents for constructing more effective prediction models.

This study offers a distinctive contribution by conducting a comprehensive comparison of preceding models encompassing technical approaches and text mining methods to ascertain the methodology yielding the highest accuracy in real-world problem-solving. Moreover, it adopts a dual-pronged approach, employing both regression and classification, to predict the return rate of the S&P 500 during the recent pandemic. Notably, the research delves into the strengths and weaknesses of different models, especially in the context of economic shocks such as the COVID-19 pandemic. To enhance the accuracy of the text mining method, the study incorporates FinBERT by Araci (2019), a pre-trained language model developed by the Bloomberg research team. FinBERT undergoes training on an extensive corpus of financial documents, including news articles, SEC filings, and earnings call transcripts, thereby augmenting its proficiency in financial lexicon, entity recognition, and sentiment analysis.

1.3 Methodology and Data Set

This study compares two popular methods of forecasting the future return rate of the S&P 500: (i) technical analysis, which includes statistical models such as Autoregressive Moving Average (ARMA), Generalized Autoregressive Conditional Heteroskedasticity with Mean Equation (GARCH-M), and also machine learning methods, such as Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), and Long Short-Term Memory (LSTM). (ii) text mining analysis, which utilizes the Bidirectional Encoder Representations from Transformers

(BERT) model as well as FinBERT, a specialized version of BERT designed for financial text analysis.

Both technical analysis and text mining analysis use different techniques to analyze and predict the future return rate of the S&P 500. Technical analysis employs statistical methods and machine learning models to analyze historical data, while text mining analysis uses natural language processing and machine learning techniques to analyze financial news. The data sets we will use come from investing.com and financial headlines from Wall Street Journal and both include 10 years of data before COVID-19. Our methodology involves converting text data into numerical vectors using techniques such as word embedding, and inputting this into machine learning models such as LSTM. On the other hand, technical analysis will involve using traditional methods such as ARMA, GARCH, and some machine learning approaches. Finally, the accuracy of the predictions made by the different methods of forecasting the future price of the S&P 500 can be evaluated using a significant evaluation metrics such as Root Mean Squared Error (RMSE).

Through a comparison of these distinct methodologies, it is possible to ascertain which of the two approaches - historical prices or news - is more dependable in actual real-world situations, thus enabling investors to determine which method to employ when predicting future values in unpredictable events. The matter of whether price or news is a superior forecaster of the future is a multifaceted one and is likely reliant on the particular circumstances. Nevertheless, this study's discoveries can provide useful understandings into the advantages and limitations of these disparate approaches and equip investors with the knowledge to make more informed decisions.

1.3.1 Technical Analysis

1.3.1.1 Statistical Methods

1.3.1.1.1 ARMA

Auto Regressive Moving Average is a statistical model class that is used for time series analysis and forecasting. It is a widely used approach in the field of econometrics for modeling and predicting economic data such as stock prices, GDP, and inflation rates. ARMA models assume that time series data can be described as a combination of autoregressive (AR), moving average (MA), and often assume stationary. The basic mathematical equation for an ARMA(p,q) model:

(1.1)

$$y_t = c + \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p} - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q} + \varepsilon_t$$

where:

y_t is the value of the time series at time t, c is a constant or intercept term, p is the autoregressive (AR) coefficients, and q is the moving average (MA) coefficient.

The process of order selection, which may involve statistical methods like the Akaike information criterion (AIC) or the Bayes information criterion (BIC). Ensuring stationarity in the dataset is paramount for the efficacy of ARMA (AutoRegressive Moving Average) models. The choice of ARMA over ARIMA (AutoRegressive Integrated Moving Average) is motivated by the nature of the data, which involves return rates. The stationary test results serve as a comprehensive evaluation, ensuring that the data exhibits stable mean, variance, and autocorrelation structures. Specifically, in the context of return rates, which inherently involve financial data, achieving stationarity through differencing is often preferred.

1.3.1.1.2 GARCH-M

GARCH-in-Mean (GARCH-M) is an extension of the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model that incorporates the conditional volatility into the mean equation of a time series model. In financial modeling, GARCH-M is commonly used to capture the effect of past volatility on the conditional mean of the time series. The model assumes that high volatility in the past influences the current mean of the time series (see Bollersley, 1986). The mathematical equation for GARCH-M (p, q) can be expressed as follows.

(1.2)

$$y_t = \mu + \lambda\sigma_t + a_t$$

(1.3)

$$a_t = \sigma_t\varepsilon_t$$

(1.4)

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i a_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

In these equations, y_t is the observed value at times, μ is the mean of the time series, a_t is the standardized residual, σ_t is the conditional standard deviation (volatility), ε_t is the standard residuals, $\alpha\ and\ \beta$ are parameters to be estimated. The GARCH-M model introduces the conditional volatility into the conditional mean equation, allowing past volatility to influence the expected value. This makes GARCH-M particularly suitable for financial time series where volatility clustering is observed. The model is estimated using maximum likelihood methods, and diagnostic checks are performed to ensure model adequacy. The key distinction between GARCH-M (GARCH-in-Mean) and ARMA (Autoregressive Moving Average) models lies in their primary objectives within time series analysis. The GARCH-M model is designed to capture the concept that higher risk, as manifested by increased volatility, is expected to be compensated by higher returns in the stock market. It's not just used to model volatility clustering, but rather to explain the relationship between risk and expected returns, implying that returns could be predictable based on the risk level indicated by past volatility. In contrast, ARMA models are versatile tools designed to capture temporal dependencies by incorporating autoregressive (AR) and moving average (MA) terms, which account for linear dependencies on past values and forecast errors. While GARCH-M specializes in addressing volatility effects on the mean, ARMA models serve as general-purpose models applied across diverse fields to characterize the intrinsic dynamics of time series data.

1.3.1.2 Machine Learning (ML) Methods

Machine learning offers a wide array of methods for solving complex problems. In supervised learning, techniques like linear regression provide a foundational approach to predicting numerical outcomes, while logistic regression is employed when dealing with classification tasks. Decision trees and random forests are versatile algorithms used for both regression and classification, and they offer interpretability through their tree-like structures. Support Vector Machines (SVMs) are powerful tools for classification and regression, particularly when dealing with high-dimensional data. k-Nearest Neighbors (k-NN) is a simple yet effective algorithm for classification and regression, relying on the similarity between data points. Naive Bayes methods are great for text classification and other tasks involving probabilities. Unsupervised learning methods include Principal Component Analysis (PCA), which is used for dimensionality reduction and data visualization, and k-Means clustering, which helps group data into clusters based on similarity. Hierarchical clustering methods, such as

Agglomerative and Divisive clustering, organize data hierarchically. Apriori and Eclat are employed in association rule learning to uncover patterns in data. Lastly, in reinforcement learning, Q-learning and Deep Q Networks (DQNs) are used to train agents to make sequential decisions and learn optimal strategies through trial and error. These machine learning methods provide the toolkit for a diverse range of applications across various domains (Cao & Tay, 2000; Chhajer et al., 2022).

Machine learning and deep learning are subsets of artificial intelligence. Machine learning encompasses various techniques where a system is trained to learn and make predictions from data without being explicitly programmed. It includes algorithms like decision trees, random forests, and support vector machines. Deep learning, on the other hand, is a more specialized form of machine learning that focuses on neural networks with multiple layers, known as deep neural networks. Deep learning models, including Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), are neural networks that can automatically learn and make predictions from data. ANNs are inspired by the human brain and are versatile for tasks like image and speech recognition. CNNs are designed for processing grid-like data, such as images, making them essential for computer vision. RNNs excel with sequence data, making them vital for tasks like time series forecasting and natural language processing. Among RNNs, LSTMs stand out for their ability to capture long-range dependencies, facilitating sequence prediction, while GRUs offer computational efficiency compared to LSTMs while maintaining strong performance on sequence-based tasks. These neural network models collectively enable deep learning's wide applicability in complex data analysis and pattern recognition, often outperforming traditional machine learning techniques for tasks involving unstructured data like images, text, and audio (Chhajer et al., 2022).

Ensemble methods by Dietterich (2000) combine the predictions of multiple machine learning models to improve accuracy and robustness, including techniques like bagging and boosting. Reinforcement learning is a type of machine learning where agents learn how to make sequences of decisions by interacting with an environment and receiving rewards. XGBoost (Extreme Gradient Boosting) is primarily categorized as an ensemble method. It belongs to the ensemble learning category because it combines the predictions of multiple decision trees to create a stronger predictive model. In the context of our literature review, the performance of Support Vector Machines (SVM), Long Short-Term Memory (LSTM) networks, and XGBoost, three distinct predictive models will be compared to determine which one provides more accurate predictions for stock returns rate during the pandemic induced market shock (Cakici et al., 2023).

1.3.1.2.1 LSTM

Long Short-Term Memory method by Hochreiter & Schmidhuber (1997) is a type of recurrent neural network that is capable of modeling complex temporal patterns in sequential data, such as stock prices. Unlike regular feed-forward neural networks, which only consider the current input, information in the RNN travels in loops from layer to layer, preserving the context based on previous inputs and outputs. However, RNNs have some limitations such as slow computation time and difficulty retaining information over long periods (Bengio et al., 1994). LSTM overcomes these shortcomings by using a cell to remember information over time intervals and three gates to regulate the flow of information into and out of the cell. The capacity to capture long-term dependencies, versatility in handling different forecasting jobs, and handling missing variables are all advantages of LSTM in forecasting. Disadvantages include complexity in training and optimization, difficulty in interpreting results, sensitivity to hyperparameters, and potential for overfitting (Goodfellow et al., 2016). The specific advantages and disadvantages may vary depending on the use case and dataset.

The ability of LSTMs to selectively remember and forget information over long periods of time makes them well-suited for modeling and predicting data sequences with complex patterns and dependencies. LSTMs accomplish this by employing memory cells, which allow them to keep a long-term memory of previous inputs and selectively update that memory based on new inputs. This makes LSTMs especially useful in applications like language translation, sentiment analysis, and stock price prediction, where understanding and modeling complex dependencies in sequential data is critical. The mathematical model for Long Short-Term Memory (LSTM) networks can be summarized as follows.

(1.5)

$$f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f)$$

(1.6)

$$i_t = \sigma(W_i.[h_{t-1}, X_t] + b_i)$$

(1.7)

$$\widetilde{C_t} = tanh(W_C.[h_{t-1}, X_t] + b_C)$$

(1.8)

$$C_t = f_t.C_{t-1} + i_t.\widetilde{C_t}$$

(1.9)

$$o_t = \sigma(W_o.[h_{t-1}, X_t] + b_o)$$

(1.10)

$$h_t = o_t . \tanh{(C_t)}$$

Where f_t, the forget gate determines what information from the cell state should be discarded, i_t or input gate decides which information should be stored in the cell state, $\widetilde{C_t}$ is the cell state update creating a candidate cell state, C_t updates the cell state, o_t or output gate determines the next hidden state, h_t is the hidden state producing the final hidden state.
In the equations, σ represents the sigmoid activation function, tanh is the hyperbolic tangent activation function, W_f, W_i, W_C, W_o are weight matrices, b_f, b_i, b_C, b_o are bias vectors, and $[h_{t-1}, X_t]$ denotes concatenation. The LSTM's ability to selectively remember and forget information over long sequences makes it effective for modeling sequential data. The LSTM model has three main components: the input gate, the forget gate, and the output gate. The input gate controls the flow of information into the cell state, while the forget gate selectively discards information that is no longer relevant. The output gate controls the flow of information from the cell state to the hidden state. LSTM networks are trained using backpropagation through time (BPTT), which involves calculating the gradients of the loss function with respect to the model parameters at each time step and propagating them backwards through time. This allows the network to learn complex patterns and dependencies in sequential data, making LSTMs a powerful tool for a wide range of applications in natural language processing, speech recognition, and time series analysis.

1.3.1.2.2 SVM

The Support Vector Machine (SVM) is a powerful machine learning algorithm introduced by Vapnik (1999), designed to classify data points into different categories. SVM is particularly suited for both classification and regression tasks, making it versatile for various applications, including stock price prediction. Unlike traditional feed-forward neural networks, SVM is a supervised learning method that doesn't rely on complex neural architectures. SVM operates by identifying a decision boundary, known as a hyperplane, that maximizes the margin between data points of different classes. This margin, called the "support vector," acts as the backbone of the model. By selecting the support vectors, SVM effectively focuses on the most informative data points to make accurate predictions. The ability to find a high-dimensional hyperplane and handle non-linear data through kernel functions is a significant strength of SVM. One of the key advantages of SVM is its generalization power, which allows it to make predictions on unseen data accurately. SVM can efficiently handle high-dimensional data, and with appropriate parameter tuning, it's less prone to overfitting. Furthermore, SVM can deal with non-linear relationships in the data by transforming it into a higher-dimensional space. However, SVM also has limitations, such as the need for careful selection of the kernel function, potential sensitivity to hyperparameters, and challenges with interpreting complex decision boundaries. Moreover, SVM might not perform optimally when the dataset is exceptionally large or noisy (Cristianini & Shawe-Taylor, 2000).

1.3.1.2.3 XGBOOST

XGBoost, or Extreme Gradient Boosting, is a powerful ensemble learning algorithm introduced by Chen and Guestrin (2016) that has gained prominence for its exceptional performance in various machine learning competitions and predictive modeling tasks. XGBoost is particularly suitable for both classification and regression tasks, making it a valuable tool for stock price prediction and many other applications.

Extreme Gradient Boosting belongs to the gradient boosting family of algorithms. It builds predictive models by training an ensemble of decision trees sequentially. Each new tree corrects the errors made by the existing ensemble, allowing XGBoost to iteratively improve its predictive accuracy. XGBoost's key strengths include its ability to handle complex non-linear relationships in data, efficiently manage missing values, and reduce the risk of overfitting through regularization techniques. One of the distinct advantages of XGBoost is its speed and scalability. By optimizing for computational efficiency, XGBoost processes data faster than many other algorithms, which is essential for handling large datasets. Additionally, XGBoost provides feature importance scores, aiding in the interpretation of model predictions. However, like many machine learning methods, XGBoost also has limitations, including the need for fine-tuning of hyperparameters, sensitivity to outliers, and a potential increase in complexity when dealing with numerous features (Chen & Guestrin, 2016).

1.3.2 Text Mining Analysis

1.3.2.1 BERT

Bidirectional Encoder Representations from Transformers is a deep learning model that has been pre-trained by Google AI Language (see Devlin et al., 2018). It has transformed natural language processing tasks by achieving cutting-edge performance on a variety of tasks such as text classification, question answering, and text generation. Unlike traditional language models, which process text sequentially, BERT is a transformer-based model that can consider both the preceding and following words to determine the context of a word or sentence. This enables BERT to capture complex linguistic phenomena such as word sense disambiguation and coreference resolution, resulting in significant accuracy and performance improvements.

The BERT model architecture is based on the transformer architecture, which is made up of a series of encoder and decoder layers that process input sequences in parallel, attention-based fashion. BERT pre-trains the model on a large corpus of text data with a bidirectional transformer encoder, then uses a masked language modeling objective to predict the missing words in a sentence. The pre-trained BERT model can be used as a starting point for a wide range of natural language processing tasks during fine-tuning, allowing it to achieve great performance with relatively little data. BERT is a complex model that involves multiple layers of neural networks, but at its core, it uses a transformer-based architecture, given an input sequence X of tokens, BERT first applies an embedding layer to convert each token into a vector representation. The embedded tokens are then fed into a series of transformer encoder layers, which process the sequence in a parallel, attention-based manner. The output of the final transformer layer is a sequence of context-aware token representations, which can be used for downstream tasks such as text classification or question answering.

1.3.2.2 FinBERT

It is a BERT-based domain-specific language model designed specifically for analyzing financial text data. FinBERT has been pre-trained on a large corpus of financial text data, allowing it to comprehend the subtleties of financial language and terminology (see Araci, 2019). It’s been used for everything from sentiment analysis to financial document classification and financial forecasting. FinBERT has demonstrated cutting-edge performance on a variety of financial text datasets, making it an invaluable tool for financial analysts and researchers. The FinBERT model architecture is similar to the BERT model architecture, but it has been fine-tuned on financial text data to achieve higher accuracy and performance. It encodes the input text data and extracts contextual representations using a deep transformer-based neural network. After that, the contextual representations can be used for a variety of financial text analysis tasks, such as sentiment analysis or document classification. FinBERT can significantly re- duce the amount of manual effort required for financial text analysis while also improving results accuracy.

1.3.3 Dataset

In this study, the financial time series data (S&P 500 price) comes from Investing.com, We use the daily closing price of the index, and text data comes from financial daily headline of Wall Street Journal and both from Jan 3, 2012, to Dec 29, 2022. This extensive dataset offers a robust and comprehensive insight into market dynamics and behaviors, especially during the challenging period marked by the COVID-19 pandemic. The analysis primarily focuses on the return rates of the S&P 500, utilizing this rich dataset to investigate how market trends evolved and responded to various events during this two-year timeframe.

1.3.3.1 Data Preprocessing

In order to achieve more accurate and reliable results when analyzing data, preprocessing is required to ensure the data is clean, complete, and in the proper form for machine learning (Kotsiantis et al., 2006). Based on the dataset, the preprocessing may include working with missing data or outliers, scaling and normalization, feature engineering, data decomposition, temporal aggregation, etc. In real life, the collected data are most likely to have incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data. Data preprocessing is an important procedure to detect and remove these data points to improve the data's accuracy, consistency, and reliability.

For stock dataset analysis, we consider several key steps to ensure the accuracy and reliability of the data. The initial phase involves the detection of incorrect data points, which may include outliers, missing values, or inconsistent entries. Subsequently, the identified incorrect data points are refined through various techniques, such as imputation, removal, or interpolation, to enhance the overall quality of the dataset. Throughout this process, maintaining the original order of the data is crucial to preserve temporal or sequential dependencies. Finally, the refined dataset undergoes thorough validation to ensure it aligns with the specified requirements for subsequent data analysis, providing a robust foundation for extracting meaningful insights and making informed decisions. Data normalization is a process in order to transform data into common scale without changing the difference in the range of the values. Data normalization is useful for transforming data into a proper form for machine learning or other analytical methods. In the literature, there are several data normalization techniques. However, this study uses Mean normalization that is one of the well-known scaling data methods. Mean normalization is an approach to scale and center values, so that the mean and variance are zero and one respectively (Alpaydin, 2020). Mean normalization transform the actual data points into a new dataset where each data points are between minus one and positive one with zero mean. For the text dataset the process involves several key stages to transform raw text into a format suitable for analysis and modeling. Initial steps include cleaning the text by removing special characters, punctuation, and common stopwords, ensuring a consistent and focused dataset. Tokenization breaks down sentences into individual words or tokens, while text normalization further refines the data by applying techniques such as stemming and lemmatization to reduce words to their base forms.

Handling contractions, resolving abbreviations, and removing HTML tags and URLs contribute to refining the text for subsequent analysis. Once the text is preprocessed, feature extraction techniques such as Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and word embeddings are employed to convert text into numerical representations. Imbalanced data concerns are addressed for classification tasks, and the dataset is split into training, validation, and test sets. Vectorization and embedding further refine the text representation, and additional steps like feature scaling may be applied if necessary. For deep learning models, specific tokenizers and embeddings, such as BERT embeddings, are utilized. The comprehensive text processing pipeline ensures that the resulting dataset is conducive to effective analysis and modeling, providing a foundation for extracting valuable insights from textual information.

1.3.4 Hyperparameters Tuning

Hyperparameters are the parameters of a given machine learning that cannot be tuned by the learning process, instead they must be adjusted before training. The accuracy of the machine learning algorithms highly depends on the hyperparameters and tuning them is a critical step.

1.3.4.1 Randomized Search

According to among the hyperparameters tuning techniques, randomized search has shown a is more effective than grid search. From a specific distribution, the hyperparameters of a given algorithm are tuned randomly.

All steps of the randomized search can be summarized as follows,

- Specifying the hyperparameters and their corresponding distribution and ranges.

- Setting the number of iterations
- Evaluate the accuracy of the model by the selected hyperparameters.
- Repeat the previous steps until the best model is found.

1.3.4.2 Bayesian Optimization

LSTM can perform better when its hyperparameters are optimized using the Bayesian optimization technique. It selects the most promising collection of hyperparameters to examine using a probabilistic model and iteratively changes its model as new data come in (J. S. Bergstra et al., 2011). Bayesian optimization can be used to adjust LSTM hyperparameters including batch size, learning rate, learning rate per layer, and number of LSTM layers, among others. Bayesian optimization can help boost the LSTM model's accuracy and generalization performance in a specific forecasting task by identifying the best hyperparameters.

1.3.5 Cross-Validation

When analyzing time series data, where observations are arranged chronologically and may be connected with one another, common cross-validation procedures assume that data points are independently and identically distributed (Hyndman & Athanasopoulos, 2018). Utilizing a rolling window is one popular method for the time series cross-validation, where the training set is made up of all data up to a specific time point and the test set is made up of the following data window. By using rolling window cross-validation, it enables the model to consider how the underlying patterns of the data change over time, which is crucial for time series forecasting. The cross-validation on a rolling basis starts from a small subset or window of the train data and forecasts the next data point(s). The forecasted data point(s) is transferred to the next window as a new train data to forecast the new data point.

1.3.6 Evaluation Metrics

Evaluation metrics are used to measure the performance of a machine learning model. In the context of predicting the behavior of the stock market during the COVID-19 pandemic, two evaluation metrics are used in this study. Mean squared error (MSE) measures the average squared difference between the predicted and actual values. A lower MSE indicates better prediction accuracy (Alpaydin, 2020). It should be noted that using multiple evaluation metrics can provide a completer and more accurate picture of machine learning model performance. Different evaluation metrics show different aspects of model performance and can reveal model strengths and weaknesses that may not be evident with only one criterion.

RMSE is chosen as an evaluation metric for financial time series forecasting due to its sensitivity to errors, straightforward interpretability in financial units, the squaring effect that emphasizes outliers, promotion of continuous and smooth predictions, suitability for model comparison, well-defined mathematical properties, and its close connection to residual analysis.

The mean squared error (MSE) served as a fundamental forecasting error metric, quantifying the variance between predicted and actual values through the square of their differences and subsequent averaging. Consequently, MSE's sensitivity depends on the magnitude of prediction errors. To address this, we also utilized the root mean squared error (RMSE), which is the square root of MSE. RMSE, being less sensitive, provides a more intuitive understanding of error magnitude. Additionally, the mean absolute error (MAE) averaged the absolute errors, demonstrating reduced sensitivity to error size and a lower susceptibility to outliers compared to MSE. A lower MSE, RMSE, and MAE collectively indicate improved performance, signifying predictions that closely align with actual values. In the evaluation of models, evaluation metrics are used to assess the performance and effectiveness of a model in making predictions or classifications. These metrics provide insights into how well a model generalizes to new, unseen data and can guide the selection of the most suitable algorithm for a specific task. In the context of predicting the behavior of the stock return rate during the COVID-19 pandemic, three evaluation metrics are used in this study as follows.

1.3.6.1 Mean Squared Error (MSE)

The Mean Squared Error mentioned by Alpaydin (2020) is a fundamental metric for quantifying the average of the squared differences between predicted and actual values in a regression model. It assesses the precision of predictions, with lower MSE values indicating better model accuracy. Mathematically, MSE is calculated as the sum of squared residuals divided by the number of data points. The Root Mean Squared Error (RMSE) is derived from the MSE and is particularly useful for interpreting prediction errors in the original unit of the target variable. It's the square root of the MSE and provides a more interpretable measure of prediction accuracy (Hastie et al., 2009). The equations are as follows:

(1.11)

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

(1.12)

$$RMSE = \sqrt{MSE}$$

where y_i represents actual values, $\hat{y}_i$ is predicted values, and $\hat{y}_i$ is the number of data points.

1.4.6.2 Mean Absolute Error (MAE)

The Mean Absolute Error mentioned by James, Witten, Hastie, & Tibshirani (2013) evaluates model accuracy by calculating the average of the absolute differences between predicted and actual values. MAE is less sensitive to outliers compared to MSE and is often used when outlier impact needs to be minimized. Mathematically, equation 3 show the MAE for predicted values $\hat{y}_i$ and actual values y_i.

(1.13)

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

1.3.7 Deibold-Mariano Test

The Diebold-Mariano test, introduced by Diebold and Mariano in 1995, stands as a statistical tool in econometrics and finance for rigorously comparing the forecasting accuracy of two models. Its primary objective is to assess whether one model significantly outperforms another, often aiding in the evaluation of new models against established benchmarks. The test's null hypothesis posits no difference in forecast accuracy between the models, with the alternative hypothesis suggesting a significant difference. The test statistic, derived from the mean squared forecast errors, follows a standard normal distribution under the null hypothesis. A significant result implies discernible differences in accuracy, guiding researchers and practitioners in model selection and decision-making. The Diebold-Mariano test statistic (DM) is calculated as the ratio of the average differences in forecast errors to the root mean squared forecast errors. The test statistic is evaluated against a standard normal distribution to determine its significance. A positive or negative result indicates superior forecast accuracy in favor of the corresponding model, aiding in robust model comparison.

1.4 Results and Discussion

Seven different models were used to forecast the price of the S&P 500 in this study. The first two ARMA and GARCH-M from statistical method, three methods LSTM, SVM, and XGBOOST from machine learning method, and the last two BERT and FinBERT are text mining methods. The dataset was divided into training and test sets, crucial for determining optimal configurations for each model in response to the shocks induced by COVID-19 cases. Statistical analysis reveals substantial shifts in confirmed cases reported by the WHO Coronavirus dashboard in January 2021 and January 2022. Correspondingly, the S&P 500 return rate exhibits marked volatility in reflection of these changes. Hence, this study strategically selects three pivotal time points for training and testing: January 2020 (Train %70- Test %30), marking the commencement of the pandemic and significant market transformations, and subsequently, January 2021 (Train %80- Test %20) and January 2022 (Train %90- Test %10), aligning with periods of heightened confirmed cases, inducing further shocks to the market. Different model parameters and settings were experimented with to optimize and find the configuration with the lowest error on the test set. The evaluation metrics of each model was compared to assess its forecasting performance.

1.4.1 Technical Analysis

1.4.1.1 Statistical Methods

In statistical models, the best ARMA model for the return data has been found using three commonly used criteria: Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Mean Squared Error (MSE) in Table 1.1.

Table 1.1. ARMA Model

Train/Test size	*p*	*d*	*q*	*AIC*	*BIC*	*MSE*
Train %70- Test %30	1	0	1	46699.8	4722.1	2.37
Train %80- Test %20	4	0	3	6315.92	6367.25	1.45
Train %90- Test %10	2	0	2	7010.61	7045.53	2.24

Results from GARCH model estimation reveal significant autoregressive and moving average parameters (p and q) at lag one. Subsequently, employing the GARCH-M model for prediction yields optimal Mean Squared Error (MSE) values of 3.81, 6.10, and 3.73 across three distinct cases. Figures 1.2 and 1.3 visually represent the disparities in predictive performance between the ARMA (AutoRegressive Moving Average) and GARCH-M models. While ARMA models are dedicated to modeling mean structures, GARCH-M models stand out for their ability to simultaneously capture mean and dynamic volatility.

1.4.1.2 Machine Learning Methods

In machine learning models, first an LSTM model is used to forecast time series data. The Keras library is used to create and train the model, with the input layer consisting of two LSTM layers followed by a Dense output layer. After that, the model is compiled with the 'adam' optimizer and the 'mean squared error' loss function. The provided code implements a Long Short-Term Memory (LSTM) neural network for time series forecasting, specifically applied to financial data. The preprocessing stage involves scaling the closing prices of the dataset using Min-Max scaling and creating sequences of historical prices paired with corresponding return values. Subsequently, the data is split into training and testing sets, and an LSTM model is constructed with two layers, incorporating dropout regularization to mitigate overfitting. The model is trained on the training set, and predictions are made on the test set. The inverse transformation is applied to bring the predicted and actual values back to their original scale, and common evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are calculated to assess the model's performance.

Figure 1.2. GARCH-M Model

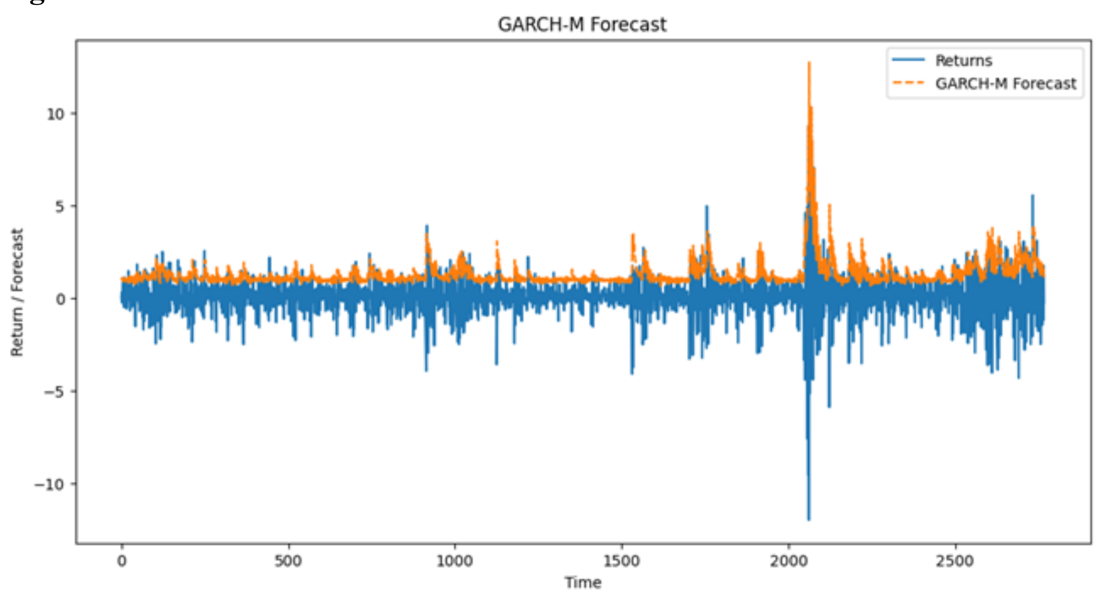
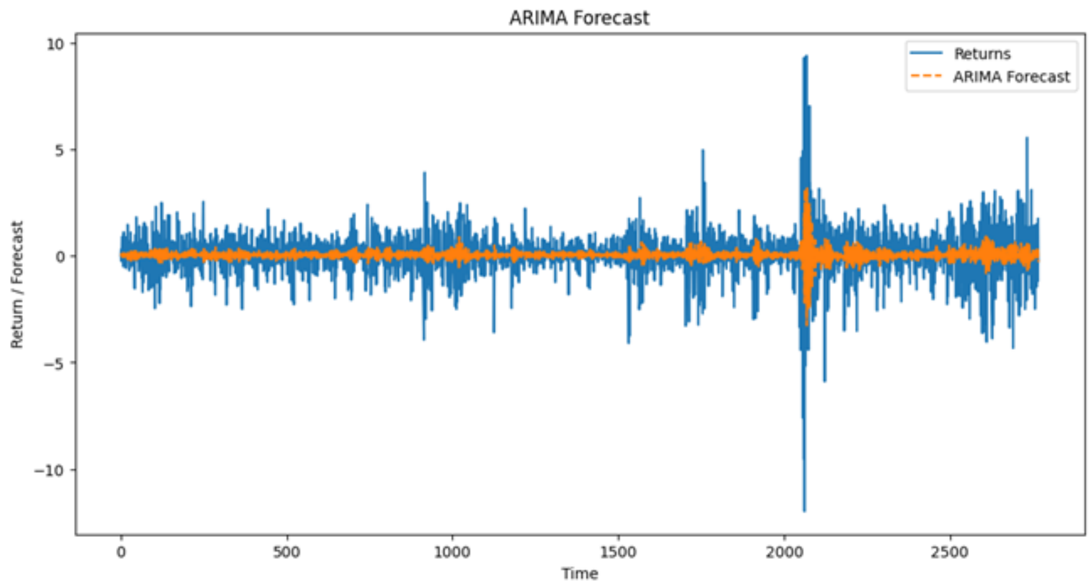


Figure 1.3. ARMA Model



s

This LSTM architecture with dropout layers aims to capture intricate patterns in financial time series data, and the model is trained for 200 epochs with a batch size of 64, as Baysian optimization used. This work aligns with contemporary practices in time series forecasting using deep learning, contributing to the ongoing exploration of neural network models in financial research. The assessment of overfitting during the LSTM model training process incorporates two crucial considerations. Firstly, the absence of a substantial gap between the evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), for the training and test datasets provides an indication of model generalization. The goal is to ensure that the model performs consistently across seen and unseen data. Secondly, monitoring the behavior of the loss function throughout different epochs is instrumental in detecting overfitting. The gradual decrease in the loss function over successive epochs implies that the model is learning and adapting to the training data. These combined approaches offer a comprehensive evaluation of overfitting, reflecting the model's ability to generalize to new data while also observing its convergence during the training process.

This study uses a Support Vector Machine (SVM) regression model tailored for time series forecasting, with a primary focus on assessing its performance under varying training sizes. The dataset undergoes stratified division into distinct training sizes (70%, 80%, and 90%). For each partition, an SVM regressor employing a linear kernel is meticulously trained on the return rate to anticipate corresponding returns. Robust evaluation metrics, encompassing Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Root Mean Squared Logarithmic Error (RMSLE), are meticulously computed to discern the accuracy of predictive outcomes. The ensuing juxtaposition of actual versus predicted values is visually depicted, affording an all-encompassing comprehension of the model's efficacy across diverse data subsets. This systematic evaluative methodology, coupled with visual representations, yields invaluable insights into the adaptability and performance of the SVM model for intricate time series forecasting scenarios marked by diverse training data proportions. Notably, observations reveal that the SVM model tends to approximate a moving average in instances where the test size decreased, particularly in scenarios characterized by heightened volatility exceeding 2% in the return rate (Figures 1.4 to 1.6).

This result presents an in-depth exploration of the application of an XGBoost regression model for time series forecasting, specifically concentrating on evaluating its performance across varying training sizes. The dataset is systematically divided into distinct training proportions (70%, 80%, and 90%), with each subset used to train an XGBoost regressor on the return rate, predicting subsequent returns. The model is meticulously optimized through the adjustment of hyperparameters, including the learning rate, maximum depth, number of estimators, subsampling rate, gamma, reg_alpha (L1 regularization), and reg_lambda (L2 regularization). A comprehensive assessment is conducted using key evaluation metrics, such as Mean Squared

Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Root Mean Squared Logarithmic Error (RMSLE), providing a thorough insight into the accuracy of the forecasting outcomes. The findings are visually represented through plots illustrating the actual versus predicted values for each distinct training size, offering a holistic comprehension of the XGBoost model's efficacy across diverse data subsets. The study concludes that the XGBoost model consistently demonstrates robust performance, adeptly capturing intricate patterns and delivering precise forecasts, even under varying training scenarios. The accompanying Figures (Figures 1.7 to 1.9) further illustrate that, despite occasional lag, the model generally exhibits proficiency in forecasting directional changes in return rates as the test size decreases.

Figure 1.4. SVM Forecast (Test Size: 70%)

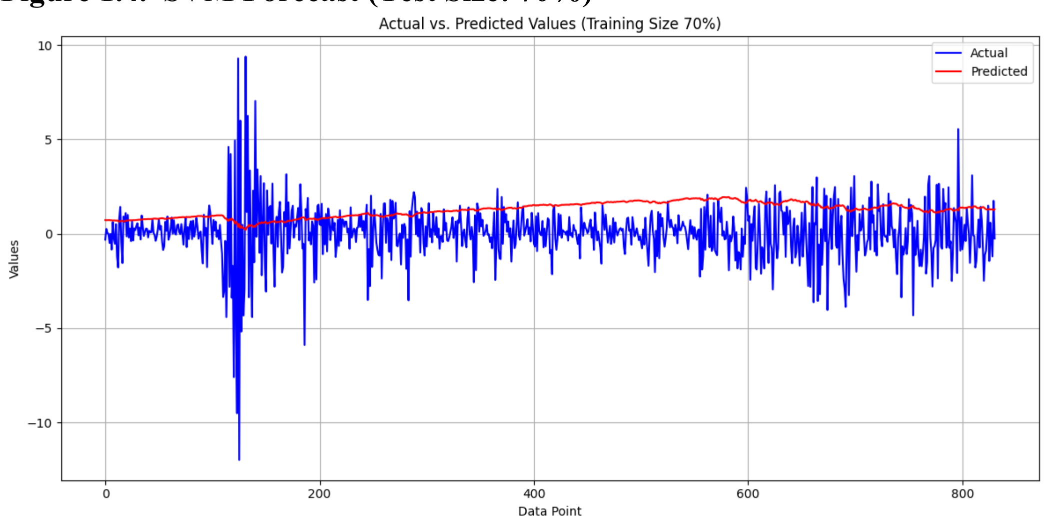


Figure 1.5. SVM Forecast (Test Size: 80%)

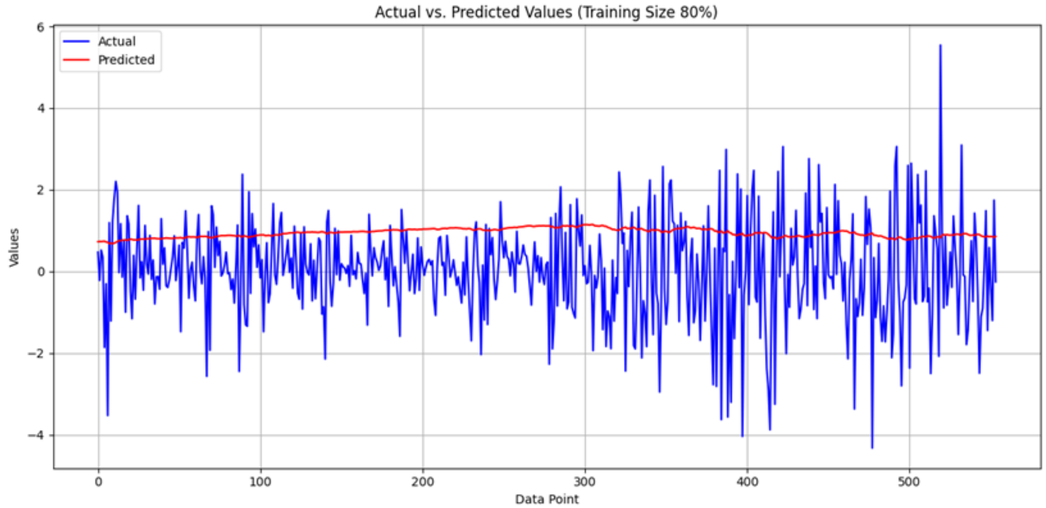


Figure 1.6. SVM Forecast (Test Size: 90%)

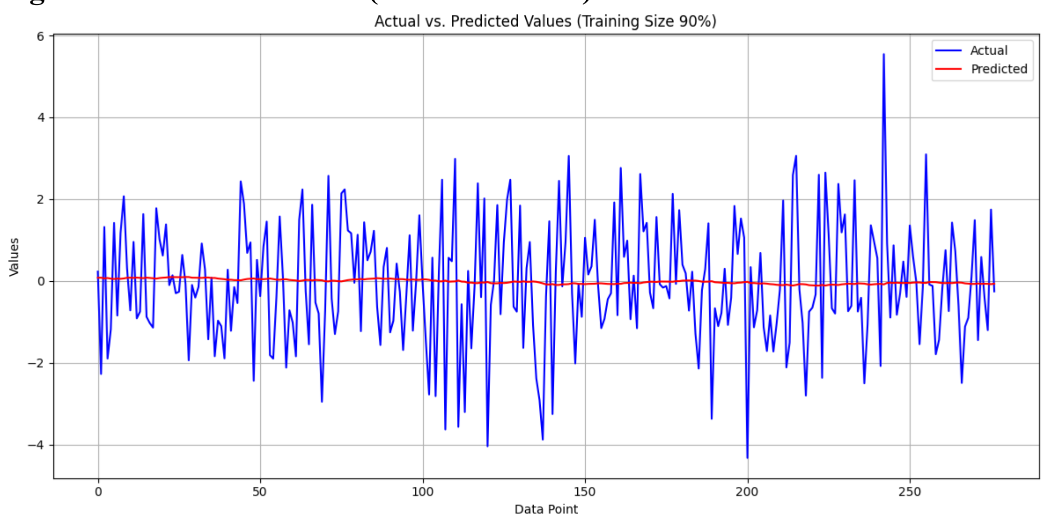


Table 1.2 provides a comparative analysis of the performance metrics for three models, LSTM (Long Short-Term Memory), SVM (Support Vector Machine), and XGBoost models in the context of time series forecasting. The metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). In terms of MSE, LSTM exhibit comparable and relatively lower values compared to XGBoost. This suggests that LSTM models yield more accurate predictions with lower squared errors. The standard deviations presented below the mean values indicate the consistency or variability in model performance across different trials. In summary, the results suggest that, based on the provided metrics, LSTM

models are more effective in time series forecasting, consistently achieving lower errors compared to XGBoost and SVM. The standard deviations offer insights into the stability and robustness of each model's performance.

Table 1.2. Performance Metrics Comparison for Machine Learning Models

	Train and test size	LSTM	SVM	XGBOOST
MSE	70-30	2.37	4.07	2.37
	80-20	1.45	2.33	1.45
	90-10	2.23	2.25	2.24
RMSE	70-30	1.54	2.02	1.54
	80-20	1.21	1.53	1.21
	90-10	1.49	1.5	1.5
MAE	70-30	1	1.55	1
	80-20	0.9	1.22	0.9
	90-10	1.18	1.18	1.19

Figure 1.7. XGBoost Forecast (Test Size: 70%)

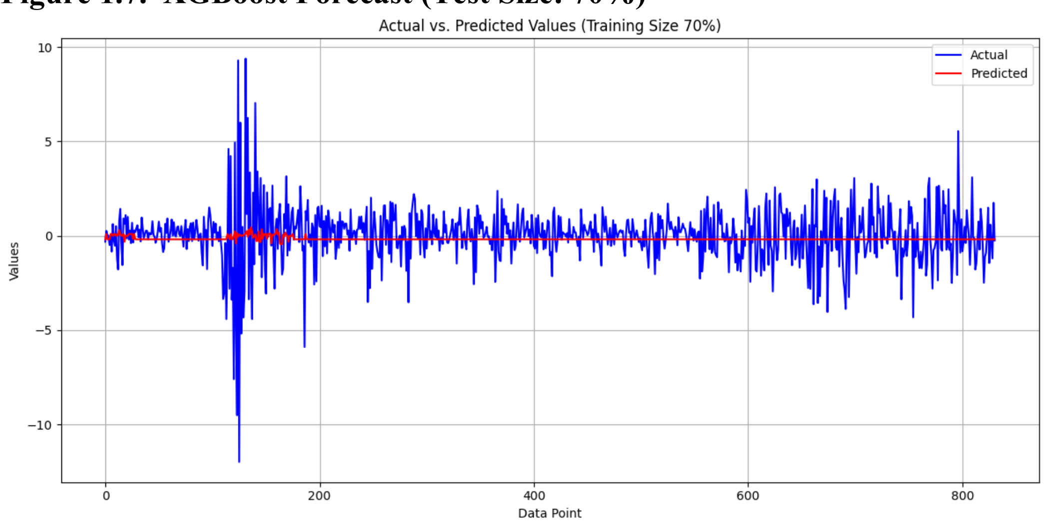


Figure 1.8. XGBoost Forecast (Test Size: 80%)

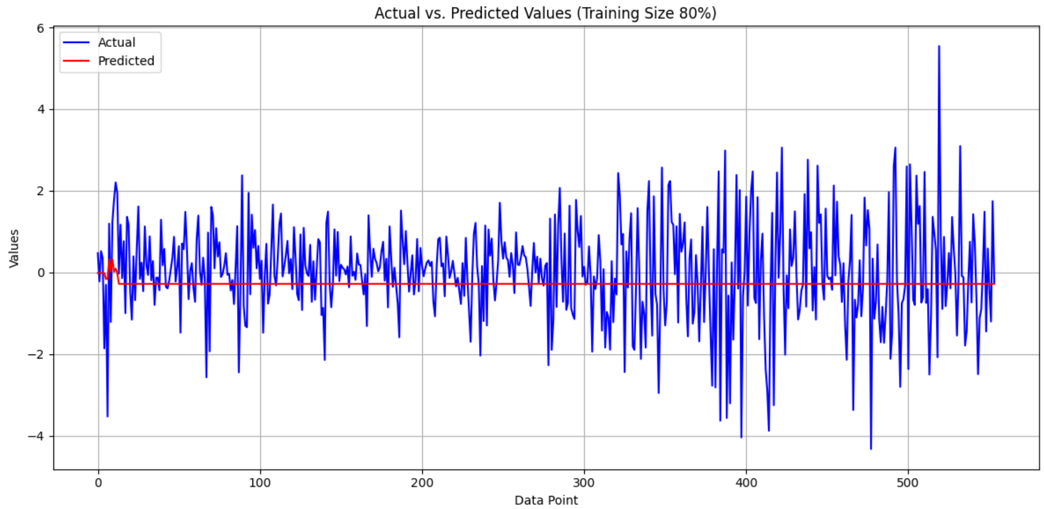
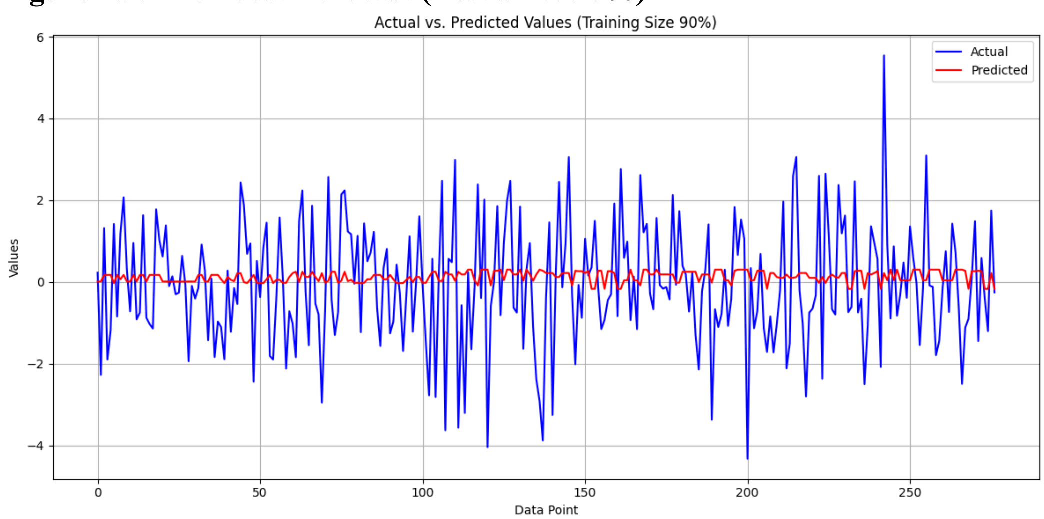


Figure 1.9. XGBoost Forecast (Test Size: 90%)



1.4.2 Text Mining Analysis

The comparative analysis between BERT and FINBERT models in forecasting the return rate of S&P 500 based on a decade of financial news data from the Wall Street Journal reveals intriguing insights. Despite expectations, BERT consistently outperforms FINBERT across varied train/test configurations and forecast horizons, displaying lower forecast errors. This unexpected result prompts a nuanced exploration into the interplay between model performance and the nature of the financial news dataset. BERT's superior performance may stem from its pre-training on a diverse corpus, allowing it to capture more generalized linguistic patterns within financial headlines. In contrast, the domain-specific fine-tuning of FINBERT for financial sentiment analysis might encounter limitations if the Wall Street Journal dataset lacks the diversity present in BERT's pre-training data. This observation underscores the importance of aligning model selection with the characteristics and diversity of the dataset in question.

The observed phenomenon extends to the task-specific strengths of each model. BERT, as a transformer-based language model, excels in numerical regression tasks by leveraging its capacity to capture intricate contextual information. Conversely, FINBERT, tailored for financial sentiment analysis and classification, outshines in predicting the directional aspect of financial returns. This strength aligns with the model's focus on discerning nuanced sentiment cues in financial language, contributing to more accurate directional predictions. The divergence in performance underscores the necessity of task-specific model design, wherein BERT's generic language understanding proves advantageous for numerical predictions, while FINBERT shines in tasks emphasizing nuanced sentiment and directional forecasting based on news sentiment. These findings contribute to a nuanced understanding of model performance in the context of financial data and underscore the significance of aligning model characteristics with the intricacies of the dataset at hand. FinBERT's specialized training on financial language allows it to be sensitive to the nuances of market-relevant jargon, which might explain the greater volatility in its predictions — it's reacting to the specific financial indicators in the text. BERT, however, with its more generalized training, might not give as much weight to these financial terms. Consequently, its predictions could follow a more consistent trajectory, potentially leading to fewer large errors across various test cases (Table 1.3).

Table 1.3. BERT and FinBERT Results

	Train/Test size	h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8	h=9	h=10
Actual	train 70 test 30	-0.31	0.26	0.03	0	-0.49	-0.01	-0.84	0.62	-0.24	-0.53
	train 80 test 20	0.47	-0.22	0.52	0.34	-1.86	-0.3	-3.53	1.19	-1.21	1.23
	train 90 test 10	0.23	-2.27	1.32	-1.9	-1.18	1.42	-0.84	1.17	2.07	0.31
BERT	train 70 test 30	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	train 80 test 20	0.28	0.16	0.28	0.16	0.18	0.31	0.23	-0.02	0.15	0.15
	train 90 test 10	0.13	0.06	-0.02	0.26	0.06	0.19	-0.09	0.01	0.4	0.21
FINBERT	train 70 test 30	0.13	0.14	0.15	-0.03	0.22	0.11	0.12	0.14	0.09	-0.03
	train 80 test 20	0.00	0.1	0.14	0.1	0.14	0.12	0.07	0.27	0.16	0.03
	train 90 test 10	0.13	0.38	0.14	0.3	0.15	0.12	0.2	0.23	0.18	0.11

1.5 Conclusion and Future Work

This essay delved into forecasting the return rate of the S&P 500, juxtaposing the efficacy of text mining against other models. Leveraging datasets from investing.com and financial headlines from the Wall Street Journal, we harnessed word embedding techniques to translate textual data into numerical vectors for regression modeling. Simultaneously, technical analysis involved the application of established methods like ARMA and GARCH-M, alongside machine learning paradigms such as SVM, XGBoost, and the LSTM model. The evaluation of forecasting methods centered on three key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and MSE.

In assessing the predictive accuracy of LSTM and ARMA models for stock market return rates during the COVID-19 pandemic in the U.S., both models underwent hyperparameter tuning, resulting in comparable error levels. However, the Diebold-Mariano test was employed to statistically compare their performance. The test revealed a low p-value (all twelve times less than 3.50e-50), indicating that LSTM outperforms ARMA in terms of forecast accuracy for quarterly data after Jan 2020, but it is more than 0.111 for yearly data (three times) and shows we cannot reject the null hypothesis and it is no significant difference in predictive accuracy between ARMA and LSTM. Therefore, based on this analysis, the LSTM model demonstrates superior accuracy compared to the ARMA model in predicting stock market return rates during the challenging conditions imposed by the COVID-19 pandemic in quarterlies. The outcomes underscored the supremacy of the LSTM model, demonstrating superior accuracy in predicting the S&P 500's future return rate compared to BERT or ARMA models. This advocates for the LSTM model as a robust tool for forecasting S&P 500 prices. Notably, during the examined period, the study illuminated that, contrary to expectations, financial news data did not offer a discernible advantage over price data, especially in the context of the COVID-19 pandemic. Textual data analysis, however, exhibited a heightened capacity to capture uncertainty and unravel complexities during this tumultuous period compared to machine learning methods.

The FinBERT model presents a transformative potential in financial analysis and prediction, promising novel insights and discoveries. Yet, recognizing the nascent stage of this model, further research is imperative to fully exploit its capabilities and assess its effectiveness across diverse financial datasets. This research paves the way for a nuanced understanding of the intricate dynamics between traditional and modern forecasting methods, enriching the discourse on the evolving landscape of financial prediction.

Chapter 2 – Does the Federal Reserve Respond to Data Revisions?

2.1 Introduction

Data revisions are an important consideration any time macroeconomic variables are used for prediction. As explained by Landefeld et al. (2008), while the Bureau of Economic Analysis (BEA) makes a first release of GDP about one month after the quarter has ended, it is in reality an "advance estimate" of GDP due to the incomplete nature of the data. It is a composite of survey data, other data, and extrapolations of data from the early part of the quarter to the end of the quarter (for which no data is yet available). One month later, a "preliminary estimate" of GDP is released, of which 23 percent is still extrapolated data. After another month has passed, and three months after the quarter has ended, the BEA releases its "final estimate" of GDP for the quarter. The final estimate is largely, but not entirely, based on survey and other data. The BEA may continue to make changes to its estimate of GDP for that quarter in the future.

The details vary for releases of data on other macroeconomic variables, such as employment and the trade deficit, but they are also subject to substantial revisions. These revisions are of sufficient importance that they receive coverage in the popular press. One example is a story on cnn.com by Mena (2023), "The US economy grew faster in the first quarter than previously reported", and it included the passage “Gross domestic product, the broadest measure of economic output, increased at an annualized rate of 1.3% in the first quarter, up from an initial estimate of 1.1% reported last month. The change was mostly driven by an upward revision to private inventory investment, which includes finished goods, materials, and works in progress being saved for a later date. That means inventory investment had less of a drag on GDP earlier this year.”

A large literature related to data revisions followed the publication of Orphanides (2001). He showed that what had previously been interpreted as serious errors by the Federal Reserve in setting monetary policy, based on analysis using final revised macroeconomic data, could instead by blamed on the poor quality of the earlier releases of data the Federal Reserve was using to guide its decisions. See (Croushore, 2011) for a survey on the published literature up to that point.

The motivation for this essay begins with the observation that data revisions are likely to cause the Federal Reserve to reverse some of the changes it makes to the target federal funds rate. For example, a strong GDP release may result in a tightening of monetary policy, one that is reversed if GDP is subsequently revised downward in the following releases. At the same time, one of the established facts in the monetary policy literature is the degree of inertia in the Federal Reserve's target policy rate. Most commonly attributed to a preference for interest rate smoothing, it is common to include lags of the target federal funds rate in estimated Taylor rules to account for this inertia (Coibion and Gorodnichenko, 2012; Clarida et al., 2000).

These two observations imply the presence of a signal extraction problem as in Keen (2010). The effect of monetary policy on the economy changes, relative to the conventional New Keynesian model, if agents do not know whether a change in the policy rate target is persistent, due to interest rate smoothing and other inertial factors, or transitory, due to data revisions. Furthermore, the effect of monetary policy is time-varying if the share of variation in the policy rate explained by data revisions fluctuates. Data revisions would be a component of the "nonpolicy shock" in the model of Keen (2010). The other monetary policy shocks in that paper are extremely persistent changes to the coefficients in the monetary policy rule.

2.2 Methodology and Dataset

2.2.1 Empirical Approach

Our primary concern in this essay is the response of the change of federal funds rate to macroeconomic data revisions. If there is evidence of such an effect, theoretical models of monetary policy will need to account for the corresponding signal extraction problem. In principle, this is a straightforward empirical exercise, involving OLS regressions and hypothesis testing. In practice, the Federal Reserve takes many variables into consideration when making monetary policy decisions (Bernanke & Boivin, 2003; Bernanke et al., 2005), dozens of macroeconomic variables are subject to revision, and there may be a lag between the observed data revision and the response of monetary policy. In the case of a small number of revisions, we could estimate the OLS regression

(2.1)

$$\Delta ffr_t = \alpha + \sum_{h,j} \beta_{hj} x_{t-h|t-j}^{rev} + \varepsilon_t$$

where Δffr_t is the change in federal funds rate and $x_{t-h|t-j}^{rev}$ is the revision of the estimated value of macroeconomic variable x in time period $t-h$, with the revision occurring in time period $t-j$. Since the revision has to occur at a later date than the time period being measured, $t-j > t-h$. The null hypothesis of no response to data revisions can be tested with the F-test

(2.2)

$$H_0 : \beta_{00} = \beta_{01} = \cdots = 0$$

The assumption of a "small number of revisions" is critical. The regression is only feasible if there are more observations than available revisions. Even if the regression is feasible, the F-test will only be powerful if the number of observations is large relative to the number of regressors.

A standard response to concerns about the inclusion of irrelevant regressors is to use a model selection criterion such as the Akaike information criteria or Schwarz information criteria, where all potential models are estimated and the one with the lowest associated information criteria value is treated as the correct specification. Estimation of all possible models is only an option if the number of revisions is small. That can be addressed by using upward stepwise model selection (James et al., 2013), but it is only a partial solution because (i) most potential models are not considered, and (ii) path dependence may lead to the selection of suboptimal models. Finally, there is little evidence that it is an effective tool for model selection when there are a large number of potential models.

One approach that would have power to reject the null hypothesis would be to run separate regressions for each revision:

(2.3)

$$\Delta ffr_t = \alpha + \beta x^{rev}_{t-h|t-j} + \varepsilon_t$$

and then test the null hypothesis $\beta = 0$. If $\beta = 0$ is rejected for any of the revisions, there is evidence that the federal reserve responds to revisions. It is well known that t- and F-statistics used for inference do not have standard critical values when doing multiple hypothesis tests. Even if proper critical values are known, the inclusion of many irrelevant variables will raise the critical value substantially, reducing the power of the test (Corradi & Swanson, 2013).

In light of these concerns, we take a more modern approach, applying methods for handling a large number of regressors that originated in the machine learning literature. One method is regularization, and in particular, the Lasso for an introduction (Belloni et al., 2014; James et al., 2013; Tibshirani, 1996). By imposing a penalty for non-zero coefficients, the Lasso sets some of the coefficients to zero, and shrinks others in the direction of zero. Unlike OLS regression, the penalty imposed by the Lasso allows estimation even when there are more regressors than observations.

The second method starts with the observation that we are using macroeconomic variable revisions as an instrument for the federal funds rate, in order to identify the part of the federal funds rate driven by data revisions. Chernozhukov et al. (2015) introduce a method for constructing an optimal instrument when there are many potential instruments. In this framework, the question is whether it is possible to construct an instrument out of all revisions, or if they collectively do not provide any useful information.

2.2.2 Regularization

Regularization is a technique used to enhance the performance of regression models by adding a penalty term to the loss function. This helps prevent overfitting, ensuring that the model generalizes well to new, unseen data. The three most common regularization methods are Lasso, Ridge, and Elastic Net regression.

2.2.2.1 Lasso (Least Absolute Shrinkage and Selection Operator)

Lasso regression uses L1 regularization, which penalizes the absolute values of the coefficients. The objective function for Lasso is:

(2.4)

$$min_{\beta}\left\{\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\}$$

where y_i is the response variable, x_i is the vector of predictor variables, β is the vector of coefficients, λ is the regularization parameter, and p is the number of predictors. Lasso can shrink some coefficients to exactly zero, effectively performing variable selection and regularization simultaneously (Tibshirani, 1996).

2.2.2.2 Ridge Regression

Ridge regression, also known as Tikhonov regularization, uses L2 regularization, which penalizes the squared values of the coefficients. The objective function for Ridge regression is:

(2.5)

$$min_{\beta}\left\{\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\}$$

Ridge regression shrinks the coefficients towards zero but does not set any coefficients to zero, meaning all predictors remain in the model. This is particularly useful in the presence of multicollinearity among the predictors (Hoerl & Kennard, 1970).

2.2.2.3 Elastic Net

Elastic net regression combines both L1 and L2 regularization penalties. The objective function for elastic net is:

(2.6)

$$min_{\beta}\frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda[(1-\alpha)\|\beta\|_2^2/2 + \alpha\|\beta\|_1]$$

Where:

- β is the vector of the coefficients.
- n is the number of observations.
- λ is the regularization parameter that controls the overall strength of the penalty.
- α is the mixing parameter that determines the balance between $L1$ and $L2$ penalties.
- $||\beta||_2^2$ represents the $L2$ norm (squared magnitude) of the coefficients, which corresponds to Ridge regression.
- $||\beta||_1$ represents the $L1$ norm (sum of absolute values) of the coefficients, which corresponds to Lasso regression.

The point is λ (regularization parameter) controls the overall penalty strength. A higher λ means more shrinkage. α (mixing parameter) controls the balance between $L1$ and $L2$ penalties. When $\alpha = 1$, Elastic Net becomes Lasso; when $\alpha = 0$, it becomes Ridge (Zou & Hastie, 2005). Generally, Lasso regression performs variable selection by applying an L1 penalty, which can shrink some coefficients to exactly zero, leading to sparse models; however, it may struggle with highly correlated predictors, potentially selecting one predictor and ignoring others. Ridge regression, on the other hand, uses an L2 penalty that shrinks the coefficients but does not set any to zero, effectively handling multicollinearity by reducing the impact of correlated predictors, yet it retains all variables in the model. Elastic Net regression combines both L1 and L2 penalties, merging the strengths of Lasso and Ridge by performing variable selection and effectively handling multicollinearity.

2.2.3 Post-Lasso

Post-Lasso is a two-step procedure that enhances the Lasso regression by addressing the issue of model selection consistency. Lasso regression performs variable selection by shrinking some coefficients to zero, but it can sometimes select an incorrect set of variables, especially in finite samples. The post-lasso approach refines this by running a standard regression on the set of variables selected by Lasso. This second step helps in achieving more accurate coefficient estimates. In the context of high-dimensional data, where the number of potential predictors is large relative to the number of observations, post-lasso can be particularly useful. It involves the following steps:

- Lasso Regression: Perform Lasso regression to select a subset of relevant predictors.
- OLS Regression: Conduct ordinary least squares (OLS) regression using the predictors selected by Lasso.

By performing OLS on the selected subset, post-lasso addresses potential biases introduced by the regularization in the Lasso step, leading to more reliable and efficient estimates (Belloni et al., 2014).

2.2.4 Dataset

The data set used in this empirical analysis was obtained from the Philadelphia federal reserve's real-time data set for macroeconomists. This comprehensive data set includes quarterly observations of various economic variables, reflecting the real-time macroeconomic data as it appeared at each point in time, known as vintages.

The term "vintage" is used to describe each specific date, reflecting how the data appeared at that particular time. For example, consider the ROUTPUT index, which measures real GDP. For the first quarter of 1966, the initial report listed real output for that quarter as 633.8, released in the second quarter of 1966 (vintage of real-time data set). By the second quarter of 2023, this value for the first quarter of 1966 is updated to 4409.5. It is possible to select any vintage from the second quarter of 1966 to the second quarter of 2023 and examine the data set to see the recorded value of real output for the first quarter of 1966 as it appeared in each vintage.

The dataset includes 31 quarterly macroeconomic variables such as dividends, various measures of personal and government expenditures, incomes, taxes, investments, imports, exports, and price indices, providing a comprehensive overview of economic activity and policy impacts (Appendix A).

The analysis covers data from the fourth quarter of 1994 (1994: Q4) to the first quarter of 2022 (2022: Q1), focusing on a period of increased transparency in U.S. monetary policy.

The Wu-Xia Shadow Federal Funds Rate data set, extracted from the Federal Reserve Bank of Atlanta, provides estimates of the federal funds rate during zero lower bound (ZLB) periods (December 2008 to December 2015, and from March 2020 to Feb 2022). This data helps for understanding monetary policy when traditional interest rate tools are limited. The shadow rate constructed using one-month forward rates from yield curve data, updates based on the Gurkaynak, Sack, and Wright estimates. The data is available from January 1990 to February 2022, reflecting the last business day of each month but just the data has been used at the end each of quarter between from the fourth quarter of 1994 (1994: Q4) to the first quarter of 2022 (2022: Q1).

2.3 Results and Discussion

The analysis begins with fitting an AR (1) model to the federal funds rate (FFR) series to examine its autoregressive properties. The estimation results indicate a highly persistent process, with the AR (1) coefficient estimated at 0.9877 (standard error: 0.0117) and the intercept estimated at 3.8631 (standard error: 2.5196). These results suggest that the FFR series is close to a random walk, prompting further investigation into its stationarity. To determine the stationarity of the FFR series, several unit root tests were conducted. The Augmented Dickey-Fuller (ADF) test yielded a test statistic of -2.8288 with a p-value of 0.2326, failing to reject the null hypothesis of a unit root. Similarly, the Phillips-Perron (PP) test produced a Dickey-Fuller Z(alpha) value of -10.32 with a p-value of 0.5186, also failing to reject the null hypothesis of a unit root. Additionally, the KPSS test, which tests the null hypothesis of stationarity, reported a KPSS level of 1.5813 with a p-value of 0.01, rejecting the null hypothesis. These results collectively indicate that the FFR series is non-stationary.

Given the non-stationarity of the FFR series, differencing was applied to achieve stationarity. The differenced series was then subjected to the same battery of tests. The ADF test on the differenced series reported a test statistic of -3.8028 with a p-value of 0.02133, rejecting the null hypothesis of a unit root. The PP test yielded a Dickey-Fuller Z(alpha) value of -51.152 with a p-value of 0.01, confirming the rejection of the null hypothesis. The KPSS test reported a KPSS level of 0.12112 with a p-value of 0.1, failing to reject the null hypothesis of stationarity.

2.3.1 Regularization

In the estimated equation for FFR, the instruments could be estimated by Lasso and Ridge regression, table 2.1 shows different values of alpha and lambda in elastic net equation. Lasso regression, with a high α value, performs strong variable selection, effectively reducing model complexity by retaining only the most significant predictors. Elastic Net, with a mix of α=0.5, balances between Lasso and Ridge, selecting a moderate number of variables while still performing some level of shrinkage. Ridge regression (α=0) retains all variables, applying shrinkage to coefficients without eliminating any, which can be useful when multicollinearity is a concern. The choice of model and regularization parameters (λ and α) significantly impacts which variables are retained and the strength of their coefficients, highlighting the trade-off between model complexity and interpretability.

Table 2.1. Different Values of Alpha and Lambda in Elastic Net Equation

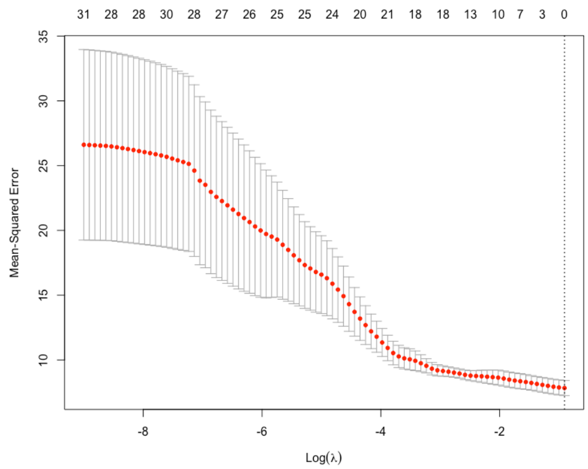
Variable	Ridge	Lasso	Elastic Net	Post-Lasso
	α =0	α =1	α =0.5	
	λ=0.5	λ=0.5	λ=0.5	
(Intercept)	2.35E+00	2.2797	2.3166	-0.063
DIV	7.83E-03	.	.	0.0029
NCON	8.61E-03	.	.	.
NDIP	1.22E-03	.	.	.
NOUTPUT	-3.80E-03	.	.	.
NPI	-2.00E-05	.	.	.
NPSAV	-2.08E-04	.	.	.
OLI	-6.65E-03	.	.	.
P	-6.06E-02	.	.	.
PCON	-6.78E-02	.	-0.003	.
PINTPAID	1.42E-02	.	.	.
PROP	2.03E-02	.	.	.
PTAX	-6.61E-03	.	-0.003	.
RATESAV	-3.02E-01	.	.	.
RCON	1.06E-03	.	.	.
RCOND	-1.92E-03	.	.	.
RCONS	1.49E-04	.	.	.
RENTI	-8.19E-03	.	.	.
REX	9.13E-03	.	.	.
RG	2.36E-04	.	.	.
RGF	3.06E-03	.	.	.
RGSL	-7.76E-04	.	.	.
RIMP	-1.08E-03	.	.	.
rinvbf	-1.07E-02	.	-0.002	.
rinvchi	-1.08E-02	.	.	.

rinvresid	-3.07E-02	.	-5E-04	.
RNX	1.46E-02	.	0.0038	.
ROUPUT	9.25E-05	.	.	.
SSCONTRIB	-4.83E-03	.	.	.
TRANPF	4.19E-02	.	.	.
TRANR	-8.54E-03	.	.	.
WSD	-1.58E-03	.	.	.
# of variables	31	0	5	1
T-test or F-test	17.35 (6.243e-05) ***	-	4.281 (0.0409) *	2.122 (0.0361) *

Note: Significance level (***: p < 0.001, **: p < 0.01, *: p < 0.05, : p < 0.1)

The graph below displays the mean squared error (MSE) against the logarithm of the regularization parameter λ during cross-validation for Lasso. As λ increases (moving left to right), the MSE initially decreases, and the number of predictors are in above of the graph 2.1. The optimal λ, which minimizes the MSE, is approximately 0.5. At this value, the Lasso shrinks the model, and all coefficients of revisions go to zero. But Table 2.1 shows that for post-Lasso, the variable DIV remains included in the model, indicating the dividends revision data is useful for forecasting federal funds rate if we select this method. In addition, we can use 5 revisions as predictors of federal funds rate when we employed elastic net method. Five macroeconomics variables are Price Index for Personal Consumption Expenditures, Personal Tax & Nontax Payments, Real Gross Private Domestic Investment: Nonresidential, Real Gross Private Domestic Investment: Residential, and Real Net Exports of Goods and Services, and the F-test value of 4.281 with a p-value of 0.0409 indicates that the model is statistically significant at the 95% confidence level.

Figure 2.1. Cross-Validation Mean Squared Error for Lasso with Optimal Lambda



2.4 Conclusion

In economic analysis, the accurate assessment of policy decisions depends on the quality and relevance of the data employed. Economists often encounter difficulties when analyzing historical policy actions based on present-day data. This issue is known as data vintage, which refers to the temporal context in which information was available to policymakers and agents at a specific time.

The Federal Reserve's role in shaping the United States' monetary policy is aimed at achieving broad objectives such as maximum employment, stable prices, and moderate long-term interest rates. To ensure effective policy implementation, the Federal Open Market Committee (FOMC) continuously assesses the state of the economy and sets the federal funds rate.

In this essay by employing machine learning methods showed that how many regressors (macroeconomic variable revisions) are useful for forecasting federal funds rate. The results from this investigation underscore the complex dynamics of policy formulation and implementation, emphasizing the critical role of data vintage in policy analysis. This underscores the necessity of utilizing real-time data from the past to accurately understand and predict the decisions made by agents and policymakers.

Chapter 3 – Forecasting Oil Prices During Economic Shocks via Machine Learning Methods: LSTM vs. RF

3.1 Introduction

Crude oil prices, specifically West Texas Intermediate (WTI), have historically exhibited significant volatility influenced by various economic, geopolitical, and environmental factors. One of the notable periods of price shocks was the global financial crisis of 2008-2009, which profoundly impacted the oil market. During this period, the financial crisis led to reduced oil demand due to worldwide economic slowdowns, resulting in a sharp decline in WTI crude oil prices (Kilian, 2009; Hamilton, 2009). Additionally, the price collapse in 2014-2015 was driven by an oversupply of crude oil in the global market, with increased production, particularly from shale oil in the United States and non-OPEC countries. The decision by OPEC not to cut production levels further exacerbated the oversupply situation, leading to a significant drop in oil prices (Baumeister & Kilian, 2016).

Moreover, the year 2020 presented another major challenge for the crude oil market due to the unprecedented COVID-19 pandemic. Lockdowns, travel restrictions, and economic slowdowns worldwide drastically reduced oil demand. The price war between OPEC and Russia, coupled with a supply glut, added further pressure on crude oil prices during this period (Narayan, 2020; Wheeler et al., 2020).

Considering these recurring price shocks, accurate forecasting models are important for understanding crude oil price behavior. The objective of this project is to develop a predictive model for forecasting Texas crude oil price using advanced machine learning algorithms, specifically Long Short-Term Memory (LSTM) and Random Forest (RF). This study aims to leverage historical data to find accurate and reliable predictions. The project aims to assist stakeholders in the oil industry, financial markets, and energy-dependent sectors to make informed decisions and develop strategies in response to oil price shocks in recent years. It also aims to evaluate the prediction accuracy of these two machine learning methods over short and long terms following the COVID-19 pandemic and the global financial crisis, determining if there are differences in accuracy between the two methods. This analysis will provide insights into the effectiveness of LSTM and RF models in capturing the complexities of crude oil price dynamics under different economic conditions.

3.2 Literature Review

Crude oil prices are influenced by a multitude of economic variables, including GDP growth, inflation, exchange rates, interest rates, and market indices. Understanding these relationships is critical, particularly during periods of economic upheaval. Research by Madiha Riaz et al. (2020) analyzed the impact of oil price changes on macroeconomic variables of major oil-exporting countries from 1970 to 2019, identifying significant long-term relationships between oil prices, GDP deflator, and real interest rates, with short-term impacts on inflation and interest rates. Kamah & S. Riti (2020) highlighted the dependency between changes in crude oil prices and 15 macroeconomic variables in Nigeria, emphasizing the need for consistent monitoring and economic diversification. Alper and Torul (2008) and Doroodian & Boyd (2003) examined the impact of oil prices on Turkey's economic activity and the U.S. economy, respectively, highlighting the mitigating effects of global liquidity conditions and technological advancements. Fan & Xu (2011) focused on structural changes in the oil market, identifying different periods dominated by market fundamentals and speculative events. Horn (2004) analyzed OPEC's strategy to stabilize oil prices, underscoring the cartel's influence on global oil prices and the challenges of maintaining consistent member interests. These studies collectively underscore the significance of GDP, inflation, interest rates, exchange rates, and market indices in influencing oil prices, highlighting the complex interplay of these variables in shaping oil market dynamics.

Over the past two decades, various machine learning methods have been employed to predict oil prices, effectively addressing the complex and nonlinear nature of the data. Xie et al. (2006) proposed a method using Support Vector Machines (SVM) for forecasting crude oil prices, demonstrating superior performance compared to ARIMA and Back Propagation Neural Network (BPNN) models. The study showcased SVM's robustness in handling nonlinear and complex data.

Herrera et al. (2019) compared traditional econometric models with machine learning methods, specifically Artificial Neural Networks (ANN) and Random Forests (RF), for forecasting key energy commodities. Their findings indicated that RF outperformed both traditional models and ANN in terms of predictive accuracy and ability to predict turning points, suggesting RF's value for long-term energy price forecasting.

An et al. (2019) used a modified linear regression approach incorporating economic indicators to forecast oil prices, showing improved accuracy over traditional methods. Similarly, Ding (2018) combined Akaike's Information Criterion (AIC) with ANN for crude oil price forecasting, achieving superior prediction accuracy by integrating EEMD into the ANN model.

Sen & Choudhury (2024) utilized deep learning models, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), optimized with Particle Swarm Optimization (PSO) to forecast crude oil prices. Their study demonstrated high accuracy, highlighting the effectiveness of combining deep learning models with optimization techniques.

Wang et al. (2018) introduced internet concern data into an extreme learning machine (ELM) model, significantly improving short-term prediction accuracy. Li et al. (2019) integrated online media text mining with deep learning techniques, showing that qualitative information from news text enhances forecast accuracy.

The literature reveals that machine learning methods such as SVM, ANN, RF, LSTM, ELM, and hybrid models have significantly advanced crude oil price forecasting. These methods provide valuable tools for stakeholders in the oil market to navigate its volatility and make informed decisions.

Crude oil prices have been subject to significant volatility due to various economic, geopolitical, and environmental factors. When focusing on the impact of two major economic shocks—the Global Financial Crisis (2008-2009) and the COVID-19 Pandemic (2020)—on crude oil prices, numerous studies highlight the use of different machine learning methods to analyze and forecast these prices. Several studies specifically examine these periods to understand the effects of machine learning and statistical modeling on oil prices and the efficacy of various predictive models.

He et al. (2012) proposed a wavelet decomposed ensemble (WDE) model to improve the accuracy of crude oil price forecasting by analyzing the dynamic market microstructure at finer time scales. Using daily closing prices of West Texas Intermediate (WTI) and Brent crude oil from January 2, 2002, to August 26, 2011, obtained from the Energy Information Administration (EIA), their model combines wavelet analysis and ensemble techniques, following the Heterogeneous Market Hypothesis (HMH) to address the non-stationarity and dynamic nature of market data. The wavelet analysis decomposes the data into simpler components, and the ensemble model reduces estimation bias by averaging forecasts from different wavelet families based on in-sample performance. Empirical studies demonstrated that the WDE model outperforms traditional benchmark models like Random Walk (RW) and Autoregressive Moving Average (ARMA) in both level and directional predictive accuracy. The study concluded that the WDE model effectively captures the time-varying heterogeneous market structure, leading to significant improvements in crude oil price forecasting.

Mingming & Jinliang (2012) proposed a multiple wavelet recurrent neural network (MWRNN) model for crude oil price forecasting, integrating wavelet analysis with recurrent neural networks (RNN) to capture multiscale data characteristics. Using data from Brent and West Texas Intermediate (WTI) crude oil prices from 1946 to 2010, they utilized wavelet analysis to decompose the data into trend and random components. The RNNs forecasted prices at different scales, and a back propagation neural network (BPNN) combined these forecasts into an optimal prediction. The model demonstrated high prediction accuracy, with an average error of 4.06% for testing and 3.88% for training data. This approach effectively addressed the nonlinear and complex nature of crude oil prices, offering a robust tool for forecasting in the energy market.

Panella et al. (2012) proposed a Mixture of Gaussian (MoG) neural network model to forecast energy commodity prices, such as crude oil, coal, natural gas, and electricity. The study used daily price data from 2001 to 2010 for both the European and U.S. markets, focusing on the U.S. market for detailed analysis. The commodities included were coal, Henry Hub natural gas, crude oil, and electricity. The model applied maximum likelihood estimation for parameter calibration and used a hierarchical constructive procedure to optimize model complexity, enhancing prediction accuracy and preventing overfitting. The approach demonstrated superior performance compared to traditional models, providing accurate forecasts and capturing the complex dynamics of energy prices, including volatility clustering and nonlinear relationships.

Zhang & Hong (2022) developed an LSTM neural network model to forecast crude oil prices using data from February 1986 to May 2021, with 75% of the data used for training and 25% for testing. The study compared the LSTM model's performance with ARIMA and ANN models, finding that LSTM demonstrated superior forecasting accuracy and stability, particularly in short-term and long-term predictions, outperforming both ARIMA and ANN models. In medium-term forecasting, ANN showed slightly better performance than LSTM. The authors concluded that the LSTM model has strong generalization ability and is highly effective in capturing the nonlinear and volatile nature of crude oil prices, making it a robust tool for economic analysis and decision-making.

Gao et al. (2022) developed an explainable machine learning framework for forecasting crude oil prices during the COVID-19 pandemic. They compared six advanced machine learning models, including extreme gradient boosting (XGB) and light gradient boosting machine (LGBM), using novel data sources such as user search trends, digital currencies, and COVID-19 case data. The study found that the LGBM model outperformed other models in terms of predictive accuracy. The framework also provided an interpretable mechanism using SHAP values to explain the importance of various features affecting crude oil prices. This approach aims to offer practical guidance for market participants, enhancing decision-making under market volatility during the pandemic.

Tian et al. (2023) examined the predictability of crude oil prices before and after the COVID-19 outbreak using data from January 1990 to July 2022. They used a comprehensive set of 15 macroeconomic factors and 18 technical indicators for prediction. The macroeconomic factors included the treasury bill rate, long-term yield, inflation, stock return variance, Kilian's index, money supply, industrial production index, unemployment rate, Chicago Fed's national activity index, capacity utilization, economic policy uncertainty, geopolitical risk, growth of crude oil production, growth of crude oil stock, and growth of crude oil import. The study found that combination models outperformed shrinkage methods during the COVID-19 period due to the altered correlations between predictors and crude oil prices. The research underscores the need for adaptive forecasting models in volatile markets, providing valuable insights for investors during periods of extreme events like the COVID-19 pandemic.

Iglesias & Rivera-Alonso (2022) analyzed the volatility patterns of Brent and WTI oil prices during major crises, including the COVID-19 pandemic. They identified two distinct volatility patterns: (a) erratic spikes during supply/demand disruptions and (b) higher volatility persistence during economic/financial crises. Their results, obtained using GARCH-type models, indicated that supply/demand disruptions led to more immediate and sharp volatility spikes, whereas financial crises resulted in prolonged periods of high volatility. The study emphasizes the importance of understanding these patterns for investors and different triggers result in different types of market uncertainty.

Bourghelle et al. (2021) examined the impact of the COVID-19 pandemic on oil price volatility, focusing on both demand and supply shocks. The pandemic led to a demand shock by reducing global demand for crude oil and increasing uncertainty, and a supply shock due to an oil trade war between Saudi Arabia and Russia. Using a VAR model and impulse-response functions, the study analyzed daily data from January 2014 to April 2020 to assess the effects of these shocks on West Texas Intermediate (WTI) oil price volatility. The results showed significant increases in oil price volatility driven by heightened uncertainty and investor anxiety during the COVID-19 crisis. The study highlights that greater uncertainty leads to more volatility, and these findings remain robust even after various tests. This research provides valuable insights for investors and policymakers to better understand and manage oil price dynamics during similar crises.

This project contributes to the existing literature by analyzing crude oil price in the United States forecasting during time periods with large economic disruptions, including the COVID-19 pandemic and the Global Financial Crisis. By employing two distinct models, Long Short-Term Memory (LSTM) and Random Forest (RF), and comparing their performance, and provide valuable insights into which model is more effective in predicting oil price behavior during economic turbulence. Both Random Forest (RF) and Long Short-Term Memory (LSTM) models have distinct advantages and limitations in the context of time series forecasting. The results of utilizing these methods between the short-term shock and the long-term shock resulting from a financial crisis are useful for investors. Additionally, the use of SHAP for explaining the impact of specific economic variables enhances the interpretability of these machine learning models, offering practical guidance for investors, policymakers, and stakeholders in managing oil market volatility.

3.3 Methodology and Dataset

3.3.1 Machine Learning Method

The study implements two kind of machine learning methods to predict the crude oil price (WTI) in two different economics shocks. Long Short-Term Memory (LSTM) neural networks, a type of recurrent neural network (RNN) renowned for its ability to capture long-term dependencies in sequential data, such as oil prices. LSTM models, developed by Hochreiter & Schmidhuber (1997), effectively address the limitations of standard RNNs by using cells to retain information over extended periods and gates to regulate information flow (Goodfellow et al., 2016). This capability makes LSTM particularly suitable for time-series forecasting where patterns over time are crucial.

In addition to LSTM, the study employs the Random Forest (RF) model, an ensemble learning method that aggregates multiple decision trees to enhance prediction accuracy and robustness (Breiman, 2001). Random Forest is known for handling complex relationships within the data, reducing overfitting through averaging, and providing insights into feature importance, making it advantageous for forecasting tasks (Pentreath, 2015).

Both models have been chosen for their specific strengths in forecasting complex datasets like crude oil prices or stock market prices. The LSTM model excels in capturing temporal dependencies and handling sequences with missing data, while the Random Forest model offers high accuracy and robustness to outliers and missing values, along with feature importance insights (Herrera et al., 2019, and Zhang & Hong, 2022).

Hyperparameter tuning is critical for optimizing both models. For the LSTM model, selecting the number of LSTM units, learning rate, and dropout rate are essential for balancing model complexity and performance. For the Random Forest model, determining the optimal number of trees, maximum depth, and the number of features considered for each split is crucial for enhancing predictive accuracy while managing computational efficiency.

These models will be evaluated and compared using statistical evaluation metrics approach to ensure robustness and accuracy in predicting oil prices. The comprehensive evaluation aims to provide insights into which model performs better under varying economic conditions, offering valuable guidance for researchers and decision-makers in the energy sector.

3.3.1.1 LSTM

Long Short-Term Memory method mentioned by Hochreiter & Schmidhuber (1997) is a type of recurrent neural network that is capable of modeling complex temporal patterns in sequential data. Unlike regular feed-forward neural networks, which only consider the current input, information in the RNN travels in loops from layer to layer, preserving the context based on previous inputs and outputs. However, RNNs have some limitations such as slow computation time and difficulty retaining information over long periods (Bengio et al., 1994). LSTM overcomes these shortcomings by using a cell to remember information over time intervals and three gates to regulate the flow of information into and out of the cell. The capacity to capture long-term dependencies, versatility in handling different forecasting jobs, and handling missing variables are all advantages of LSTM in forecasting. Disadvantages include complexity in training and optimization, difficulty in interpreting results, sensitivity to hyperparameters, and potential for overfitting (Goodfellow et al., 2016). The specific advantages and disadvantages may vary depending on the use case and dataset.

The ability of LSTMs to selectively remember and forget information over long periods of time makes them well-suited for modeling and predicting data sequences with complex patterns and dependencies. LSTMs accomplish this by employing memory cells, which allow them to keep a long-term memory of previous inputs and selectively update that memory based on new inputs. This makes LSTMs especially useful in applications like language translation, sentiment analysis, and stock price prediction, where understanding and modeling complex dependencies in sequential data is critical. The mathematical model for Long Short-Term Memory (LSTM) networks can be summarized as follows.

(3.1)

$$f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f)$$

(3.2)

$$i_t = \sigma(W_i.[h_{t-1}, X_t] + b_i)$$

(3.3)

$$\widetilde{C_t} = \tanh(W_C.[h_{t-1}, X_t] + b_C)$$

(3.4)

$$C_t = f_t.C_{t-1} + i_t.\widetilde{C_t}$$

(3.5)

$$o_t = \sigma(W_o.[h_{t-1}, X_t] + b_o)$$

(3.6)

$$h_t = o_t.\tanh(C_t)$$

Where f_t, the forget gate determines what information from the cell state should be discarded, i_t or input gate decides which information should be stored in the cell state, $\widetilde{C_t}$ is the cell state update creating a candidate cell state, C_t updates the cell state, o_t or output gate determines the next hidden state, h_t is the hidden state producing the final hidden state.

In the equations, σ represents the sigmoid activation function, tanh is the hyperbolic tangent activation function, W_f, W_i, W_C, W_o are weight matrices, $b_f, b_i,\ b_C, b_o$ are bias vectors, and $[h_{t-1}, X_t]$ denotes concatenation. The LSTM's ability to selectively remember and forget information over long sequences makes it effective for modeling sequential data.

The LSTM model has three main components: the input gate, the forget gate, and the output gate. The input gate controls the flow of information into the cell state, while the forget gate selectively discards information that is no longer relevant. The output gate controls the flow of information from the cell state to the hidden state. LSTM networks are trained using backpropagation through time (BPTT), which involves calculating the gradients of the loss function with respect to the model parameters at each time step and propagating them backwards through time. This allows the network to learn complex patterns and dependencies in sequential data, making LSTMs a powerful tool for a wide range of applications in natural language processing, speech recognition, and time series analysis.

3.3.1.2 Random Forest

Random Forest is an ensemble learning method that leverages multiple decision trees to make predictions for both classification and regression tasks. This method, proposed by Breiman (2001), is based on the concept of creating a 'forest' of decision trees, where each tree is built using a random subset of the data and features. The final prediction is derived by aggregating the predictions of all individual trees, thus reducing overfitting and improving generalization (Breiman, 2001).

The Random Forest algorithm consists of some key steps: Bootstrap Sampling (a random sample of the data is taken with replacement, creating multiple subsets), Tree Construction (decision trees are constructed for each subset), and Aggregation (the predictions from all decision trees are aggregated to produce the final output). The advantages of Random Forest include its high accuracy in handling complex and non-linear datasets, robustness to outliers and missing values, and the ability to provide insights into feature importance. However, it can be computationally intensive and may require significant memory, especially with large datasets (Pentreath, 2015).

The mathematical model for Random Forest (RF) for a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i represents the feature vector and y_i represents the target variable, the Random Forest algorithm can be mathematically expressed as follows:

Bootstrap Sampling:

Generate B bootstrap samples D_b from the original dataset D.

Tree Construction:

For each bootstrap sample D_b:

- Grow a decision tree T_b using the bootstrap sample D_b
- At each node, select the best split among a random subset of features m (where m is less than the total number of features M).

Aggregation:

- For regression tasks, the final prediction $\hat{y}$ for an input x is given by the average of predictions from all trees:

(3.7)

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

- For classification tasks, the final prediction is the mode of predictions from all trees:

(3.8)

$$\hat{y} = mode\{T_b(x)\}_{b=1}^{B}$$

Where:

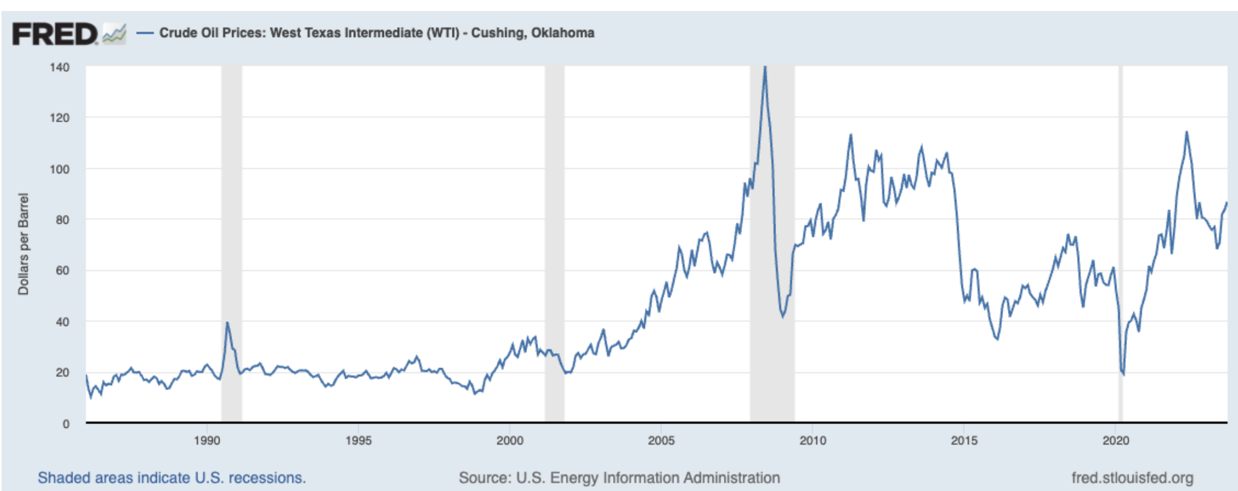
- B is the number of trees in the forest.

- $T_b(x)$ is the prediction from the b-th tree for the input x.

3.3.2 Dataset

This research uses spot prices of West Texas Intermediate (WTI) crude oil, focusing specifically on the periods of the global financial crisis and the COVID-19 pandemic (Figure 1). West Texas Intermediate is chosen as it is a benchmark for crude oil prices in the United States, representing the most important and liquid market for domestic crude oil. Most U.S. crude oil grades are typically priced against WTI's calendar monthly average. Unlike Brent and other crude oil types, which are primarily traded as waterborne cargoes, WTI operates as a pipeline market with near-constant flow rates into its delivery point in Cushing, Oklahoma. Another key reason for selecting WTI is its extensive historical data availability, with daily series starting from January 1986. Our dataset spans from January 2, 1986, to January 7, 2021, for daily prices, and from April 1986 to October 2023 for monthly data. This data is sourced from the Federal Reserve Economic Data (FRED) website.

Figure 3.1. Crude Oil Price, West Texas Intermediate (WTI)



The collected data will undergo preprocessing, including handling missing values, scaling, and normalization, to ensure it is suitable for input to the machine learning models.

Numerous studies have identified a range of macroeconomic variables that significantly influence crude oil prices, emphasizing the complexity and interconnectedness of the global economy. For example, Madiha Riaz et al. (2020) highlighted the long-term relationships between oil prices and the GDP deflator and real interest rates, while short-term dynamics showed immediate impacts on inflation and interest rates. Kamah & S. Riti (2020) found that changes in crude oil prices significantly predicted various macroeconomic variables in Nigeria, including the all-share index, exchange rate, and GDP. Alper and Torul (2008) analyzed Turkey's economic activity, incorporating global liquidity conditions, and emphasized the role of GDP growth and inflation. Fan & Xu (2011) investigated structural changes in the oil market, identifying distinct periods where different market forces played significant roles.

This study focuses on using Industrial Production (percentage change) as a measure of US output, primarily because monthly GDP data for the US is not readily available. CPI (Consumer Price Index) is a key measure of inflation, FFR (Federal Funds Rate) is a measure of monetary policy, and the NASDAQ is a representative measure of the performance of the stock market in the United States, especially for the technology sector and growth stocks. The US to EU Exchange Rate is significant during the global financial crisis and representative of trading activities. These variables are essential for understanding oil price fluctuations during economic shocks because they reflect broader economic health, inflation trends, monetary policy, market performance, and international trade dynamics. Furthermore, AR models indicate that the first lag of WTI is crucial, reinforcing the importance of temporal dependencies in oil price forecasting.

3.3.3 Hyperparameter Tuning

Hyperparameter tuning involves selecting the optimal set of hyperparameters for a machine learning model to improve its performance. Hyperparameters are the configurations that are set before the learning process begins, unlike parameters that are learned during training. Optimizing the performance of Long Short-Term Memory (LSTM) networks for time series forecasting involves careful tuning of several hyperparameters. The most critical hyperparameters include:

Number of Epochs determines how many times the learning algorithm will work through the entire training dataset. Selecting the appropriate number of epochs is essential to ensure the model learns the underlying patterns without overfitting or underfitting the data. Typically, a higher number of epochs may lead to better learning but can also increase the risk of overfitting if not controlled properly (Goodfellow et al., 2016).

Batch Size specifies the number of training samples to work through before updating the internal model parameters. A smaller batch size often provides a more precise estimate of the gradient, while a larger batch size may speed up the training process by making more significant updates to the model parameters. Balancing batch size is crucial for optimizing computational efficiency and model performance (Hochreiter & Schmidhuber, 1997).

Number of Neurons indicates the number of units or neurons within each LSTM layer. The number of neurons controls the model's capacity to capture complex temporal patterns in the data. However, increasing the number of neurons also increases the computational complexity and the risk of overfitting. Determining the optimal number of neurons is vital for maintaining the balance between model complexity and generalization (Gers et al., 2000).

Learning Rate controls how much the model's weights are adjusted with respect to the loss gradient during training. A smaller learning rate ensures stable convergence but may require more training time, whereas a larger learning rate can speed up training but risks overshooting the optimal solution. Adaptive learning rates, such as those provided by the Adam optimizer, can be particularly useful (Kingma & Ba, 2014).

Dropout is a regularization technique that helps prevent overfitting by randomly setting a fraction of input units to zero at each update during training. The dropout rate specifies this fraction, with typical values ranging from 0.2 to 0.5. Proper tuning of the dropout rate is essential to ensure model generalization (Srivastava et al., 2014).

Careful tuning of these hyperparameters—number of epochs, batch size, and number of neurons-is critical for optimizing the performance of LSTM models in capturing long-term dependencies in sequential data. Random Forest is also a versatile and robust ensemble learning method that combines multiple decision trees to enhance predictive accuracy. The key hyperparameters include:

Number of Trees specifies the number of decision trees in the forest. More trees generally improve accuracy by reducing variance, but also increase computational cost. Finding an optimal number of trees is essential for balancing performance and efficiency (Breiman, 2001).

Maximum Depth of Trees controls the depth of each tree, limiting the number of splits. Preventing overfitting by restricting tree depth is crucial, as deeper trees can capture more intricate patterns but may also capture noise (Liaw & Wiener, 2002).

Minimum Samples Split indicates the minimum number of samples required to split an internal node. Adjusting this parameter helps manage the trade-off between model complexity and generalization. A higher value can prevent the model from learning overly specific patterns that do not generalize well to new data (Geurts et al., 2006).

Tuning these hyperparameters for LSTM and Random Forest models is vital for optimizing their performance in predicting complex datasets, such as oil prices during economic shocks.

3.3.4 Grid Search

Grid Search is a systematic approach to hyperparameter optimization in machine learning, widely used to enhance model performance by exhaustively searching through a specified parameter space. This technique involves defining a set of hyperparameters and their possible values, then evaluating the model's performance for every combination of these hyperparameters within a predefined grid. By training and validating the model across all possible combinations, Grid Search ensures the identification of the optimal configuration. For example, in Random Forest models, it explores variations in parameters such as the number of trees, maximum depth of each tree, and minimum samples required to split an internal (Breiman, 2001; Geurts et al., 2006; Liaw & Wiener, 2002).

Similarly, for Long Short-Term Memory (LSTM) networks, Grid Search includes parameters like the number of epochs, batch size, and the number of neurons in each LSTM layer (Hochreiter & Schmidhuber, 1997; Gers et al., 2000; Goodfellow et al., 2016). Despite being computationally intensive, as it requires multiple model trainings for each combination of parameters, Grid Search is highly effective in ensuring that the best possible set of hyperparameters is identified, thereby improving the model's accuracy and robustness. For example, if we consider the parameters epochs: [50, 100], batch size: [32, 64], and neurons: [50, 100], Grid Search would evaluate each combination, training and validating the model eight times (50 epochs with 32 batch size and 50 neurons, 50 epochs with 32 batch size and 100 neurons, etc.). The combination yielding the best validation metrics would be selected as the optimal set, ensuring enhanced performance.

3.3.5 Evaluation Metrics

Evaluation metrics are used to measure the performance of a machine learning model. In the context of predicting the behavior of the stock market during the COVID-19 pandemic, two evaluation metrics are used in this study. Mean squared error (MSE) measures the average squared difference between the predicted and actual values. A lower MSE indicates better prediction accuracy as mentioned by Alpaydin (2020). It should be noted that using multiple evaluation metrics can provide a completer and more accurate picture of machine learning model performance. Different evaluation metrics show different aspects of model performance and can reveal model strengths and weaknesses that may not be evident with only one criterion.

RMSE is chosen as an evaluation metric for financial time series forecasting due to its sensitivity to errors, straightforward interpretability in financial units, the squaring effect that emphasizes outliers, promotion of continuous and smooth predictions, suitability for model comparison, well-defined mathematical properties, and its close connection to residual analysis. The mean squared error (MSE) served as a fundamental forecasting error metric, quantifying the variance between predicted and actual values through the square of their differences and subsequent averaging. Consequently, MSE's sensitivity depends on the magnitude of prediction errors. To address this, we also utilized the root mean squared error (RMSE), which is the square root of MSE. RMSE, being less sensitive, provides a more intuitive understanding of error magnitude. Additionally, the mean absolute error (MAE) averaged the absolute errors, demonstrating reduced sensitivity to error size and a lower susceptibility to outliers compared to MSE. A lower MSE, RMSE, and MAE collectively indicate improved performance, signifying predictions that closely align with actual values. In the evaluation of models, evaluation metrics are used to assess the performance and effectiveness of a model in making predictions or classifications. These metrics provide insights into how well a model generalizes to new, unseen data and can guide the selection of the most suitable algorithm for a specific task. In the context of predicting the behavior of the stock return rate during the COVID-19 pandemic, three evaluation metrics are used in this study as follows.

3.3.5.1 Mean Squared Error (MSE)

The Mean Squared Error mentioned by Alpaydin (2020) is a fundamental metric for quantifying the average of the squared differences between predicted and actual values in a regression model. It assesses the precision of predictions, with lower MSE values indicating better model accuracy. Mathematically, MSE is calculated as the sum of squared residuals divided by the number of data points. The Root Mean Squared Error (RMSE) is derived from the MSE and is particularly useful for interpreting prediction errors in the original unit of the target variable. It's the square root of the MSE and provides a more interpretable measure of prediction accuracy (Hastie et al., 2009). The equations are as follows:

(3.9)

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

(3.10)

$$RMSE = \sqrt{MSE}$$

where y_i represents actual values, $\hat{y}_i$ is predicted values, and $\hat{y}_i$ is the number of data points.

3.3.5.2 Mean Absolute Error (MAE)

The Mean Absolute Error mentioned by James, Witten, Hastie, & Tibshirani (2013) evaluates model accuracy by calculating the average of the absolute differences between predicted and actual values. MAE is less sensitive to outliers compared to MSE and is often used when outlier impact needs to be minimized. Mathematically, equation 3 show the MAE for predicted values $\hat{y}_i$ and actual values y_i.

(3.11)

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

3.3.6 SHAP

SHapley Additive exPlanations (SHAP) is a unified framework for interpreting machine learning models, grounded in cooperative game theory. It provides an explanation for the output of a model by attributing each feature's contribution to the prediction, ensuring consistency and accuracy (Lundberg & Lee, 2017). SHAP assigns each feature an importance value for a particular prediction by calculating Shapley values, which are derived from cooperative game theory. These values represent the average marginal contribution of a feature across all possible combinations of features, ensuring a fair and interpretable measure of feature importance.

The SHAP framework is particularly advantageous because it maintains a balance between local and global interpretability. Locally, it explains individual predictions by showing how each feature contributes to a single prediction. Globally, it aggregates these local explanations to provide a comprehensive view of feature importance across the entire dataset (Lundberg et al., 2020). This dual perspective makes SHAP a robust tool for understanding complex machine learning models, especially in high-stakes applications like economics, finance, healthcare, and critical infrastructure where interpretability and trust are paramount.

3.4 Results

3.4.1 Dataset and Preprocessing

The daily dataset reveals significant declines in July 2008 and January 2020, corresponding to the global financial crisis and the COVID-19 pandemic, respectively. To evaluate the effectiveness of different machine learning models in predicting post-crisis crude oil prices, the analysis will include calculating evaluation criteria for one day, one week, one month, and one year after these shocks.

For the monthly analysis, all relevant macroeconomic variables, including GDP, CPI (Consumer Price Index), FFR (Federal Funds Rate), NASDAQ, and the US to UK Exchange Rate, sourced from the FRED website, will be utilized. The dataset spans from January 1, 1986, to January 1, 2024, for studying the COVID-19 impact, and from January 1, 1986, to April 1, 2012, for assessing the global financial crisis, extending the evaluation period to four years post-shock.

The data preparation involves detecting and handling missing values and correcting date formats to ensure the data is suitable for machine learning applications. Following data cleaning, mean normalization is applied to scale the data appropriately, enhancing the performance and accuracy of the machine learning models.

3.4.2 Machine Learning for Daily Dataset

3.4.2.1 Financial Global Crisis

Table 3.1 compares the actual WTI crude oil prices with the predicted prices using Random Forest (RF) and Long Short-Term Memory (LSTM) models for the period from July 11, 2008, to August 22, 2008 (till one month after the shock). The table includes specific dates, actual WTI prices, and the respective predictions from both models. The actual prices represent the recorded market prices, while the predicted prices show the forecasts made by each model.

Table 3.1. Actual vs Predicted Oil Prices after Financial Globe Crisis

Date	Actual WTI	Predicted WTI (Random Forest)	Predicted WTI (LSTM)
7/11/2008	144.96	138.172613	134.755646
7/14/2008	145.16	138.648579	136.888443
7/15/2008	138.68	138.77764	138.666458
7/16/2008	134.63	136.788861	137.552612
7/17/2008	129.43	135.630902	134.89621
7/18/2008	128.94	135.096114	131.03096
7/21/2008	131.43	134.371336	128.159576
7/22/2008	127.25	133.334811	127.39267
7/23/2008	123.73	127.162124	125.962189
7/24/2008	124.62	125.508798	123.762749
7/25/2008	122.59	125.288445	122.551765
7/28/2008	124.72	125.339425	121.214027
7/29/2008	122.21	125.234131	121.052574
7/30/2008	126.74	125.250622	120.382744
7/31/2008	124.17	126.071126	121.456512
8/1/2008	125.03	125.312759	121.531555
8/4/2008	121.45	125.265965	121.744453
8/5/2008	118.71	125.169189	120.616325
8/6/2008	118.57	119.30852	118.748726
8/7/2008	119.84	118.125324	117.271996
8/8/2008	115.42	121.6949	116.897881
8/11/2008	114.44	117.719426	115.322815
8/12/2008	113.1	116.966009	113.824966
8/13/2008	115.96	116.105362	112.411644
8/14/2008	115.05	116.438803	112.485107
8/15/2008	113.46	116.438803	112.488503
8/18/2008	112.92	116.022055	112.023178
8/19/2008	114.39	116.022055	111.428864
8/20/2008	115.48	116.438803	111.529289
8/21/2008	121.23	116.438803	112.049507
8/22/2008	114.48	122.619684	114.500114

Figure 3.2. Actual vs Predicted Oil Prices LSTM after Financial Globe Crisis

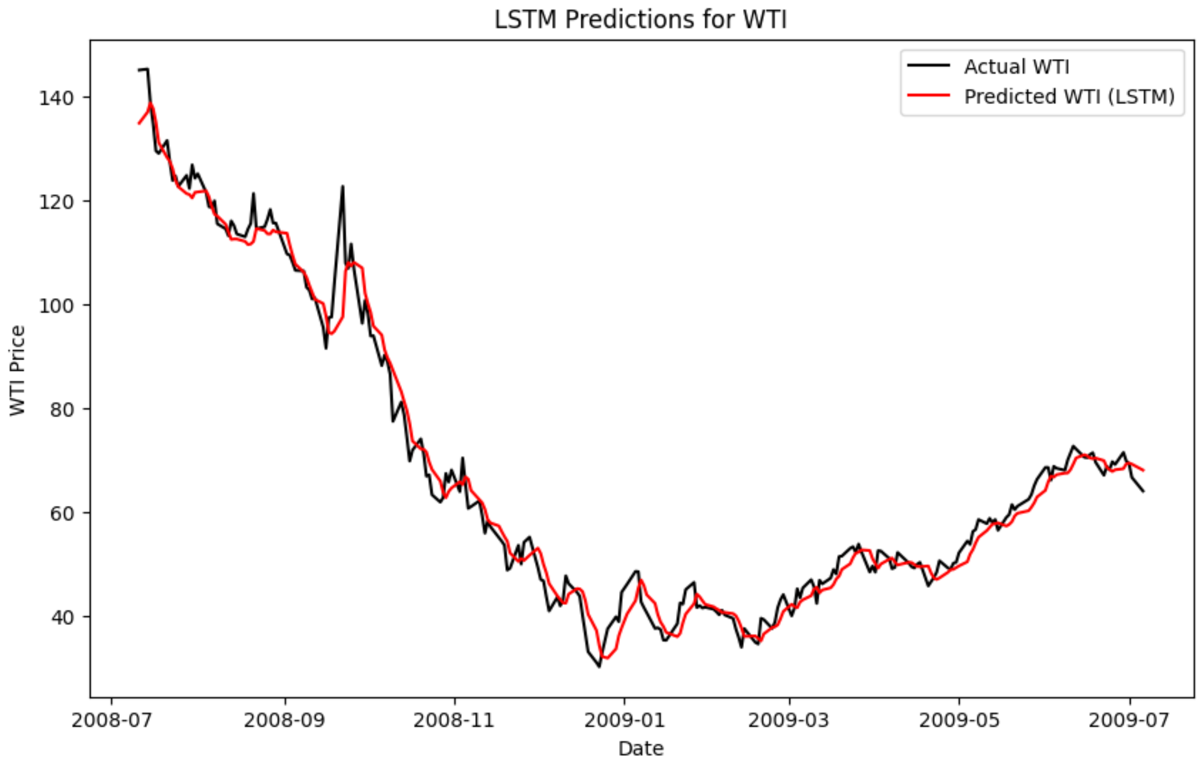


Figure 3.3. Actual vs Predicted Oil Prices RF after Financial Globe Crisis

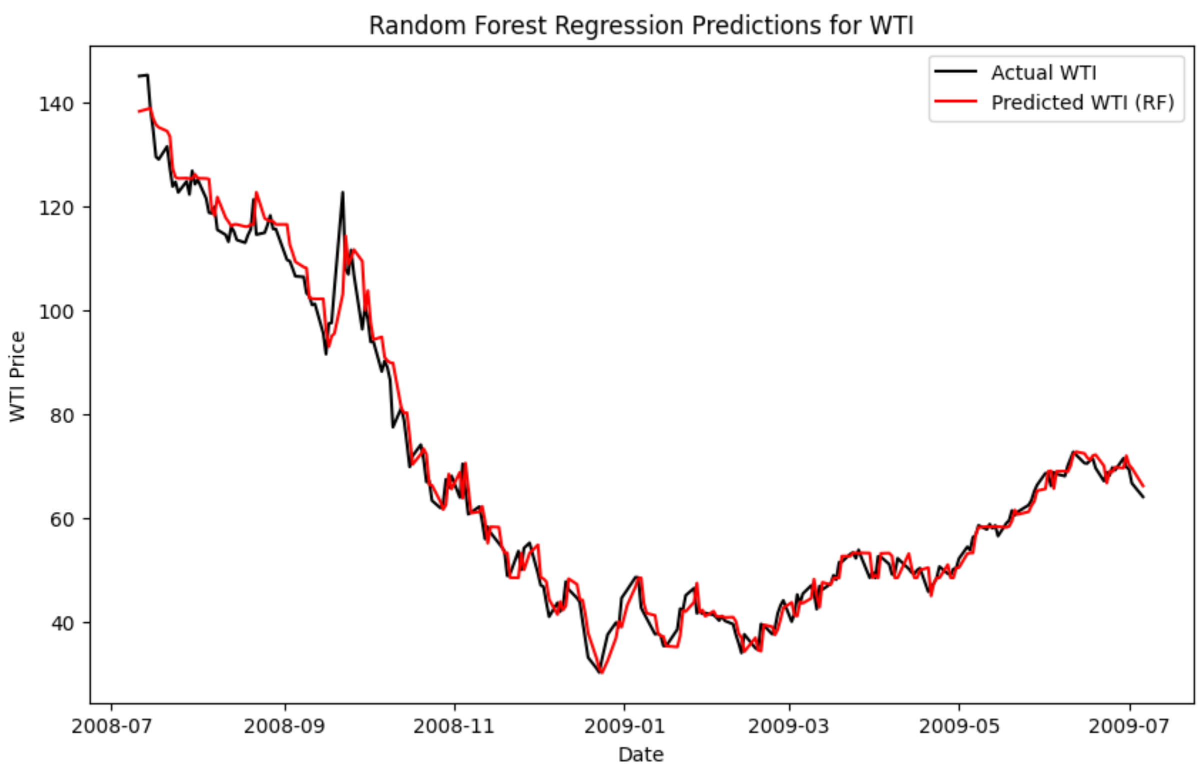


Table 3.2 presents an evaluation of Long Short-Term Memory (LSTM) and Random Forest (RF) models for predicting WTI crude oil prices following the global financial crisis. The metrics used are Mean Absolute Error (MAE) and Mean Squared Error (MSE) across different time horizons: one day, one week, one month, and one year ahead. The LSTM model shows moderate short-term accuracy with an MAE of 10.2 and an MSE of 104.12 for one day ahead. It improves significantly for one week ahead (MAE of 4.6, MSE of 32.29) and one month ahead (MAE of 2.37, MSE of 14.47), maintaining robust long-term forecasts with an MAE of 2.63 and MSE of 12.8881 for one year ahead.

Table 3.2. Evaluation Metrics after Financial Globe Crisis

	evaluation	one day ahead	one week ahead	one month ahead	one year ahead
LSTM	MAE	10.2	4.6	2.37	2.63
	MSE	104.12	32.29	14.47	12.8881
RF	MAE	6.78	4.4	3.24	2.45
	MSE	46.06	25.44	15.75	11.49

Conversely, the RF model exhibits superior short-term prediction accuracy with an MAE of 6.78 and MSE of 46.06 for one day ahead. It continues to perform well over one week (MAE of 4.4, MSE of 25.44) and maintains competitive accuracy in the medium term (MAE of 3.24, MSE of 15.75). In the long term, RF also provides highly accurate predictions with an MAE of 2.45 and MSE of 11.49 for one year ahead. These results indicate that while RF is more effective for immediate short-term forecasts, LSTM offers strong medium to long-term forecasting capabilities, making both models valuable for predicting crude oil prices during periods of economic instability.

3.4.2.2 COVID-19

Table 3.3 presents a comparison between the actual WTI crude oil prices and the predicted prices using two machine learning models: Random Forest and Long Short-Term Memory (LSTM). The data spans from January 13, 2020, to February 26, 2020. The predictions for Random Forest and LSTM are shown alongside the actual prices, allowing for a direct comparison of the models' accuracy. Both models show an approximation of the actual prices.

Figures 3.4 and 3.5 provide a visual comparison of the actual and predicted WTI crude oil prices for the same period. Figure 3.4 shows the predictions using the Random Forest model, while the second figure displays the predictions from the LSTM model. In both graphs, the black line represents the actual WTI prices, and the red line represents the predicted prices. The figures demonstrate that both models closely follow the trend of actual prices, with the LSTM model showing slightly better alignment during sharp price drops, particularly noticeable around the dramatic price dip in April 2020.

Table 3.3. Actual vs Predicted Oil Prices after COVID-19

Date	Actual WTI	Predicted WTI (Random Forest)	Predicted WTI (LSTM)
1/13/2020	58.17	59.104351	59.801311
1/14/2020	58.34	58.230669	59.076965
1/15/2020	57.86	58.230669	58.889507
1/16/2020	58.52	58.169547	58.616341
1/17/2020	58.55	58.261127	59.0219
1/21/2020	58.25	58.261127	59.216106
1/22/2020	56.76	58.230669	58.979904
1/23/2020	55.51	56.78129	57.659019
1/24/2020	54.09	55.731427	56.299255
1/27/2020	53.09	53.854151	54.922127
1/28/2020	53.33	53.562111	53.847591
1/29/2020	53.29	53.600932	53.815636
1/30/2020	52.19	53.600932	53.918785
1/31/2020	51.58	52.396517	53.005447
2/3/2020	50.06	51.766968	52.158405
2/4/2020	49.59	49.932272	50.850876
2/5/2020	50.87	49.631595	50.12529
2/6/2020	50.94	50.865047	51.070492
2/7/2020	50.34	50.917329	51.580704
2/10/2020	49.59	50.350069	51.009533
2/11/2020	50	49.631595	50.146259
2/12/2020	51.13	49.932272	50.334103
2/13/2020	51.41	51.373904	51.437031
2/14/2020	52.03	51.474832	51.986141
2/18/2020	52.1	52.305766	52.452389
2/19/2020	53.31	52.371875	52.622654
2/20/2020	53.77	53.600932	53.609932
2/21/2020	53.36	53.816183	54.303074
2/24/2020	51.36	53.600932	54.056274
2/25/2020	49.78	51.474832	52.256058
2/26/2020	48.67	49.726049	50.493015

Figure 3.4. Actual vs Predicted Oil Prices RF after COVID-19

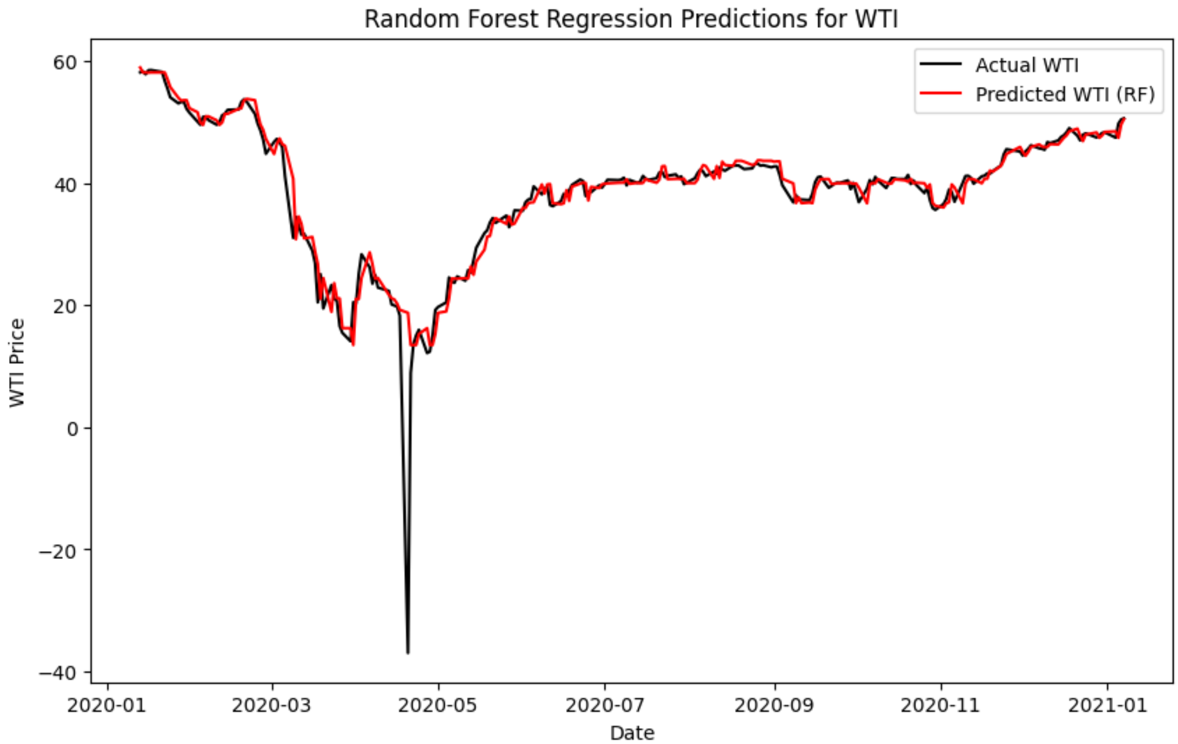


Figure 3.5. Actual vs Predicted Oil Prices LSTM after COVID-19

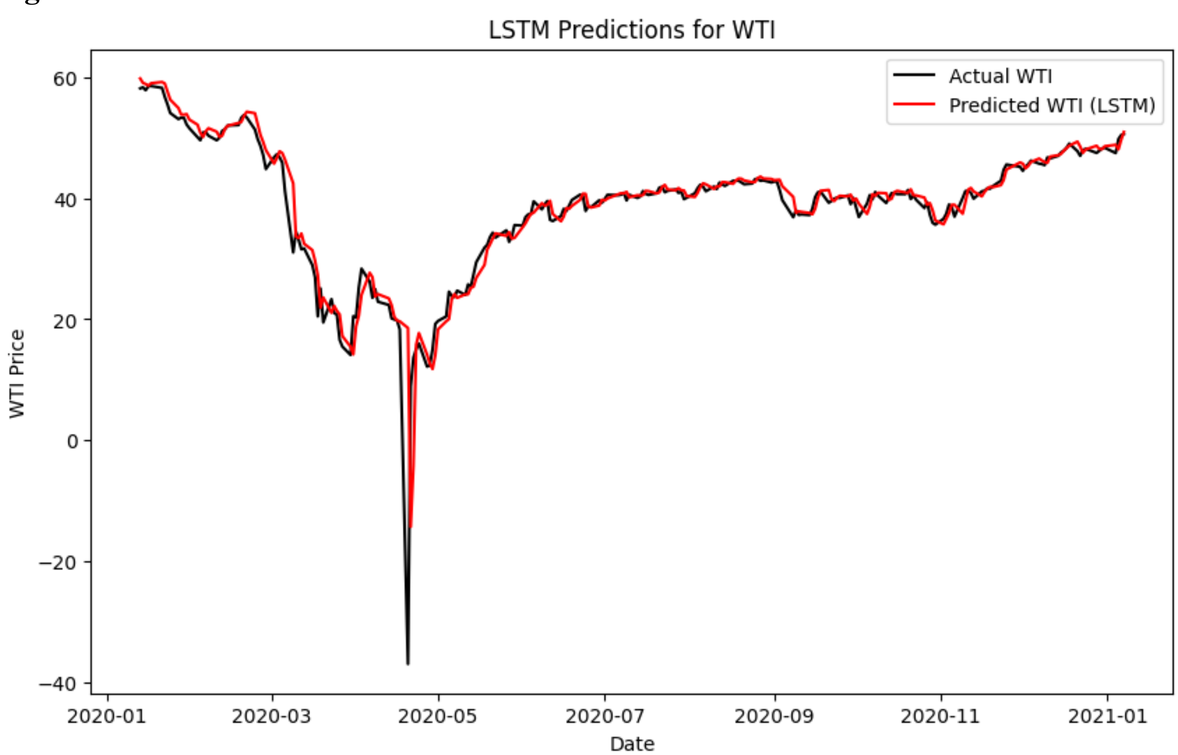


Table 3.4 displays the predictive performance of LSTM and Random Forest (RF) models in forecasting WTI crude oil prices after the COVID-19 shock. The evaluation metrics—Mean Absolute Error (MAE) and Mean Squared Error (MSE)—are reported for predictions made one day, one week, one month, and one year ahead.

The results indicate that RF consistently outperforms LSTM across all time horizons. For instance, one day ahead predictions by RF have an MAE of 0.93 and an MSE of 0.87, compared to LSTM's MAE of 1.63 and MSE of 2.66. This trend continues with RF achieving lower error metrics for one week, one month, and one year ahead predictions. Notably, both models show increased error metrics as the prediction horizon extends to one year, with the LSTM model's MSE surging significantly to 18.74, suggesting potential overfitting or difficulty in capturing longer-term trends under pandemic-induced volatility. These results underscore RF's robustness and superior performance in short-term and long-term crude oil price forecasting during the COVID-19 economic disruption.

Table 3.4. Evaluation Metrics after COVID-19

	evaluation	one day ahead	one week ahead	one month ahead	one year ahead
LSTM	MAE	1.63	1.02	1.11	1.51
	MSE	2.66	2.47	1.86	18.74
RF	MAE	0.93	0.5	0.76	1.4
	MSE	0.87	0.48	0.92	15.45

3.4.3 Machine Learning for Monthly Dataset

3.4.3.1 Financial Global Crisis

We extend our analysis to forecast WTI crude oil prices over different periods following two significant economic shocks (the global financial crisis in 2008 and the COVID-19 pandemic in 2019). The forecasting horizon spans 49 months post-shock to assess the robustness and accuracy of our predictive models. The training data encompasses WTI prices from January 1, 1986, until the onset of each respective shock—85% monthly data for the global financial crisis and 89% monthly data for the COVID-19 pandemic. The analysis includes five key macroeconomic variables: Industrial Production as a representative of GDP, CPI, FFR, NASDAQ, and US to EU exchange rates, alongside the lagged WTI price due to its significance indicated by ARIMA models.

For the rolling window approach, we employed a window size that incorporates these six variables. The predictive models, LSTM and Random Forest (RF) were evaluated using mean absolute error (MAE) and mean squared error (MSE) for forecasting one month ahead, one year ahead, and four years ahead. This comprehensive evaluation allows us to gauge the performance of each model in predicting crude oil prices across different temporal horizons, providing insights into the effectiveness of using macroeconomic indicators in conjunction with historical WTI prices for forecasting during economic shocks.

Table 3.5 provides a comparison of actual WTI prices with predictions from LSTM and RF models across several months following the global financial crisis. The data illustrates that while both models capture the general trends, LSTM consistently shows closer alignment with actual prices, particularly evident in months with significant deviations, such as in early 2009 (Figures 3.6 and 3.7). The ability of LSTM to adapt to nonlinear patterns and longer-term dependencies likely contributes to its superior performance in these periods.

Table 3.6 summarizes the performance metrics (MAE and MSE) for both LSTM and RF models over various forecasting horizons. The metrics reveal that LSTM outperforms RF in terms of lower MAE and MSE for all predictions. However, over longer horizons (4 years), RF shows improved performance with lower MAE even that the first month ahead, suggesting that RF may better capture long-term trends when given sufficient data. The consistent performance of RF in long-term forecasting can be attributed to its ensemble learning approach, which reduces overfitting by averaging multiple decision trees.

Table 3.5. Actual vs Predicted Oil Prices after Financial Global Crisis for Future Months

Date	Actual WTI	Predicted WTI (LSTM)	Predicted WTI (RF)
4/1/2008	112.580455	109.507919	98.533949
5/1/2008	125.397619	119.637581	99.78005
6/1/2008	133.88	136.178589	98.993838
7/1/2008	133.370909	142.293518	99.351661
8/1/2008	116.66619	134.139175	94.910708
9/1/2008	104.114286	110.757042	94.972827
10/1/2008	76.608696	100.374435	69.074666
11/1/2008	57.309474	65.061714	62.815574
12/1/2008	41.121818	49.818119	51.121082
1/1/2009	41.71	40.10041	45.859022
2/1/2009	39.087368	40.961498	46.670834
3/1/2009	47.939091	37.737278	37.86627
4/1/2009	49.646667	46.291824	46.823917
5/1/2009	59.0285	48.155602	46.595553
6/1/2009	69.640909	59.761742	75.266432
7/1/2009	64.152273	67.925598	82.823934
8/1/2009	71.044762	63.907063	74.223524
9/1/2009	69.408095	70.623634	84.34372
10/1/2009	75.715455	68.70919	84.814418
11/1/2009	77.99	75.476807	84.677769
12/1/2009	74.47	76.983093	86.956157
1/1/2010	78.325789	76.095818	83.808142
2/1/2010	76.387368	78.073669	67.878933
3/1/2010	81.203478	78.346596	65.249232
4/1/2010	84.292857	82.587196	70.25078
5/1/2010	73.7435	86.089195	70.697495
6/1/2010	75.335909	72.474373	65.275741

7/1/2010	76.319524	74.798416	65.224048
8/1/2010	76.599091	76.332314	65.140069
9/1/2010	75.241905	76.481583	65.095159
10/1/2010	81.892857	75.274498	78.763571
11/1/2010	84.252857	82.648895	70.484732
12/1/2010	89.145909	87.064743	71.886526
1/1/2011	89.1705	92.931389	71.505077
2/1/2011	88.578421	93.024384	72.141341
3/1/2011	102.856522	96.217186	91.795627
4/1/2011	109.5325	109.67334	96.480255
5/1/2011	100.900476	117.415176	94.557843
6/1/2011	96.264091	104.120392	93.310266
7/1/2011	97.3035	100.45195	92.328527
8/1/2011	86.333043	101.77829	92.556737
9/1/2011	85.515238	87.917213	73.303512
10/1/2011	86.322381	86.41098	73.42235
11/1/2011	97.160476	87.126137	70.533477
12/1/2011	98.562857	99.130135	71.305687
1/1/2012	100.2735	104.797401	72.406314
2/1/2012	102.204	107.040672	72.371661
3/1/2012	106.157727	110.023163	73.506556
4/1/2012	103.321	114.268959	72.081936
9/1/2009	69.408095	70.623634	84.34372
10/1/2009	75.715455	68.70919	84.814418
11/1/2009	77.99	75.476807	84.677769
12/1/2009	74.47	76.983093	86.956157
1/1/2010	78.325789	76.095818	83.808142
2/1/2010	76.387368	78.073669	67.878933
3/1/2010	81.203478	78.346596	65.249232
4/1/2010	84.292857	82.587196	70.25078

Table 3.6. Evaluation Metrics after Financial Global Crisis for Future Months

	evaluation	one month ahead	12 months ahead	four years ahead
RF	MAE	14.05	15.36	13.74
	MSE	197.30	345.67	269.19
LSTM	MAE	3.07	8.17	5.69
	MSE	9.44	107.30	57.62

Figure 3.6. Actual vs Predicted Oil Prices LSTM after Financial Global Crisis for Monthly Dataset

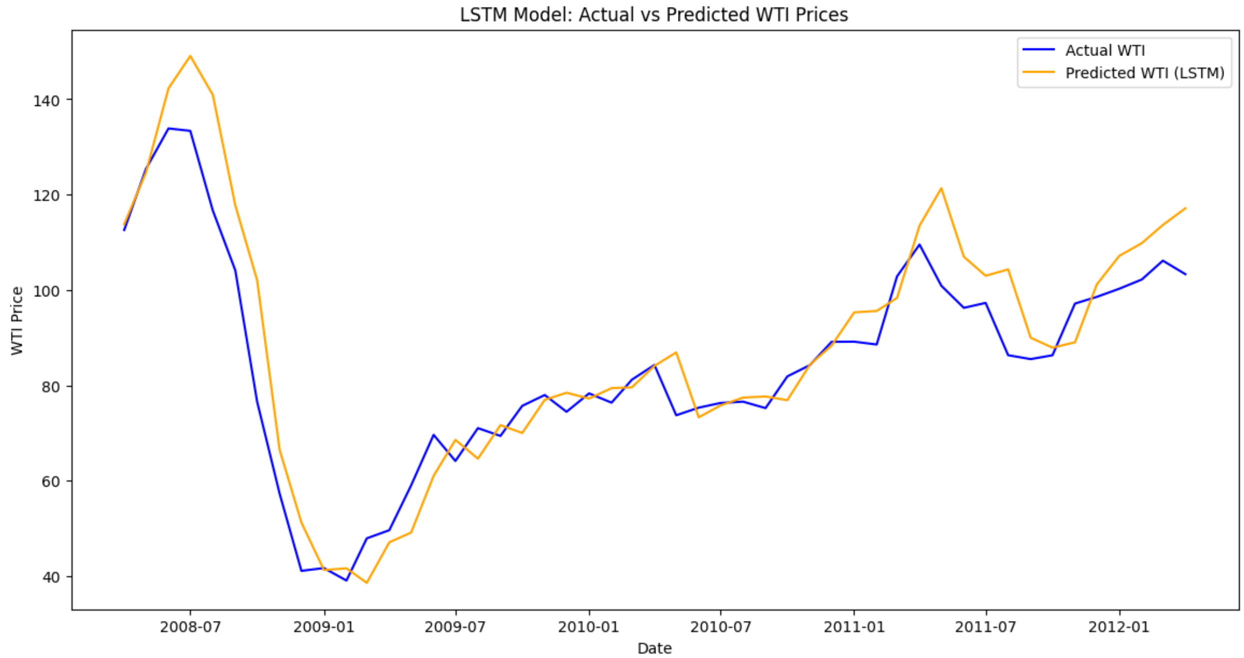
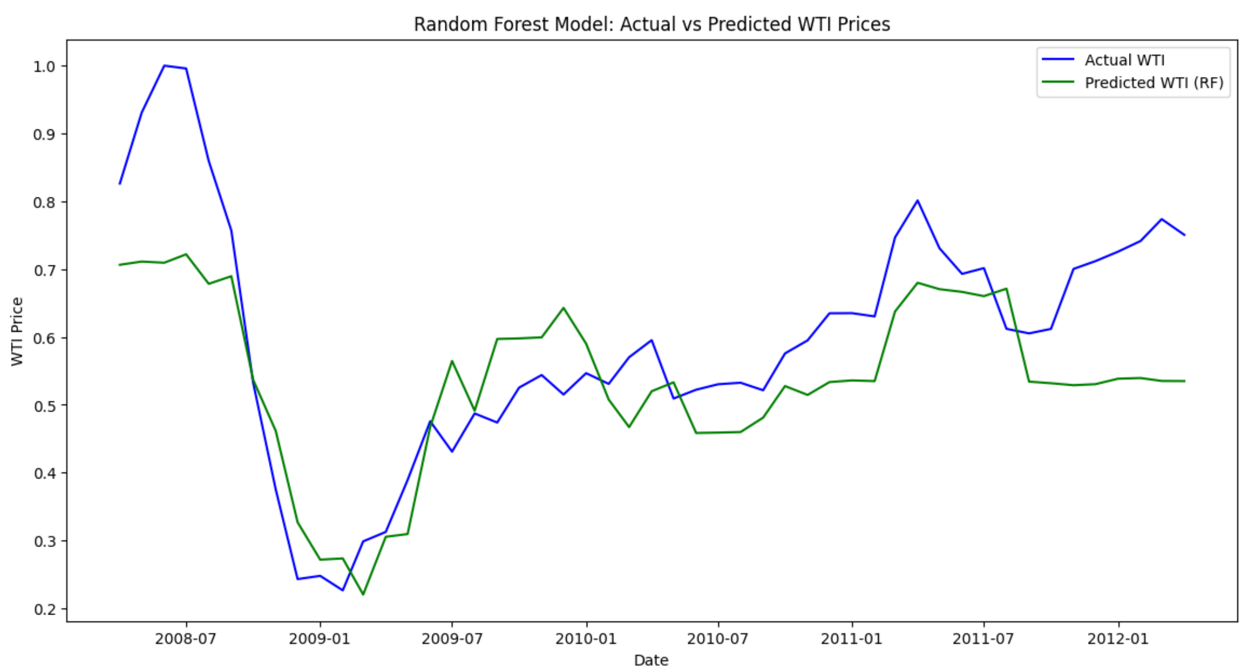


Figure 3.7. Actual vs Predicted Oil Prices RF after Financial Global Crisis for Monthly Dataset



3.4.3.2 COVID-19

Table 3.7 presents the actual and predicted West Texas Intermediate (WTI) crude oil prices using Long Short-Term Memory (LSTM) and Random Forest (RF) models over various dates following the COVID-19 pandemic. It highlights the accuracy of these models in forecasting WTI prices. For instance, on 1/1/20, the actual WTI price was 57.52, while LSTM and RF predicted 64.73 and 59.71, respectively. Similarly, on 1/1/24, the actual WTI price was 74.15, with LSTM predicting 81.29 and RF predicting 71.54. These examples indicate that RF's predictions align more closely with the actual prices compared to LSTM, suggesting that the RF model outperforms LSTM in this context. The consistent performance of RF in capturing the trends and fluctuations in crude oil prices demonstrates its superior accuracy, particularly in the long-term forecast.

Table 3.8 comparing actual vs. predicted oil prices after COVID-19 indicates that both LSTM and RF models have relatively close predictions to actual values, with each excelling in different horizons. For one month ahead, LSTM outperforms RF with a MAE of 1.15 compared to RF's 4.81, and an MSE of 1.32 compared to RF's 23.12. For twelve months ahead, RF shows superior performance with a MAE of 2.17 and MSE of 6.53, while LSTM has a MAE of 5.70 and MSE of 34.86. Over a four-year horizon, RF continues to demonstrate better accuracy with a MAE of 2.73 and MSE of 12.54, compared to LSTM's MAE of 6.72 and MSE of 56.08. These results suggest that RF captures long-term trends and reduces error accumulation more effectively than LSTM, making it more reliable for extended forecasts post-COVID-19. This comparative analysis highlights the strengths of RF in long-term crude oil price forecasting and its robustness in modeling complex market dynamics.

Figures 3.8 and 3.9 provide a visual comparison of actual versus predicted WTI prices for both models. The LSTM model's predictions closely follow the actual price trends, particularly capturing the significant fluctuations seen during the early 2020s. In contrast, the RF model, while still reasonably accurate, shows more deviation from the actual prices, because it emphasizes on mean.

Table 3.7. Actual vs Predicted Oil Prices after COVID-19 for Future Months

Date	Actual WTI	Predicted WTI (LSTM)	Predicted WTI (RF)
12/1/2019	59.816667	58.666592	55.007959
1/1/2020	57.519048	64.726212	59.712859
2/1/2020	50.542632	61.340786	56.857378
3/1/2020	29.207727	52.686272	46.451489
4/1/2020	16.547619	35.588234	30.138072
5/1/2020	28.5625	22.231861	15.807437
6/1/2020	38.307273	33.982265	31.331767
7/1/2020	40.710455	45.237797	41.301564
8/1/2020	42.339048	48.501301	43.627637
9/1/2020	39.634286	49.240364	43.849882
10/1/2020	39.395909	46.101902	43.188777
11/1/2020	40.937368	45.68705	43.027263
12/1/2020	47.025	48.616287	44.096448
1/1/2021	52.008421	56.921322	50.413253
2/1/2021	59.046316	63.851582	54.093544
3/1/2021	62.333043	69.144112	61.919371
4/1/2021	61.716667	74.465073	63.101457
5/1/2021	65.1695	73.642647	62.846239
6/1/2021	71.378182	78.185951	65.340039
7/1/2021	72.485238	80.612961	70.724734
8/1/2021	67.730455	79.659431	73.530463
9/1/2021	71.64619	76.085968	70.166168
10/1/2021	81.476667	83.063889	69.184487
11/1/2021	79.1475	89.892181	88.292971
12/1/2021	71.711818	86.034325	83.331234

1/1/2022	83.222	82.746742	72.444591
2/1/2022	91.641053	93.764771	88.954049
3/1/2022	108.502609	104.386368	95.444103
4/1/2022	101.7775	112.336197	100.129226
5/1/2022	109.552381	109.807709	100.177233
6/1/2022	114.837143	119.167717	103.069268
7/1/2022	101.619	110.701942	102.552285
8/1/2022	93.665217	99.863266	96.99786
9/1/2022	84.258095	93.744659	91.012405
10/1/2022	87.554762	86.488487	86.692918
11/1/2022	84.370476	86.629349	89.801004
12/1/2022	76.437143	83.021477	82.855779
1/1/2023	78.123	83.997337	75.255725
2/1/2023	76.832632	84.474068	75.405269
3/1/2023	73.277826	81.477592	75.218442
4/1/2023	79.446316	79.994766	72.471655
5/1/2023	71.578182	84.289574	81.965011
6/1/2023	70.248095	78.072769	69.312718
7/1/2023	76.0695	76.153717	68.026988
8/1/2023	81.386087	83.414009	74.487059
9/1/2023	89.425	86.538239	86.047659
10/1/2023	85.639524	91.52774	90.181417
11/1/2023	77.685	87.427376	86.675985
12/1/2023	71.9	81.594337	73.281789
1/1/2024	74.152381	81.29232	71.542975

Table 3.8. Evaluation Metrics after COVID-19 for Future Months

	evaluation	one month ahead	12 months ahead	four years ahead
RF	MAE	4.81	2.17	2.73
	MSE	23.12	6.53	12.54
LSTM	MAE	1.15	5.70	6.72
	MSE	1.32	34.86	56.08

Figure 3.8. Actual vs Predicted Oil Prices LSTM after COVID-19 for Monthly Dataset

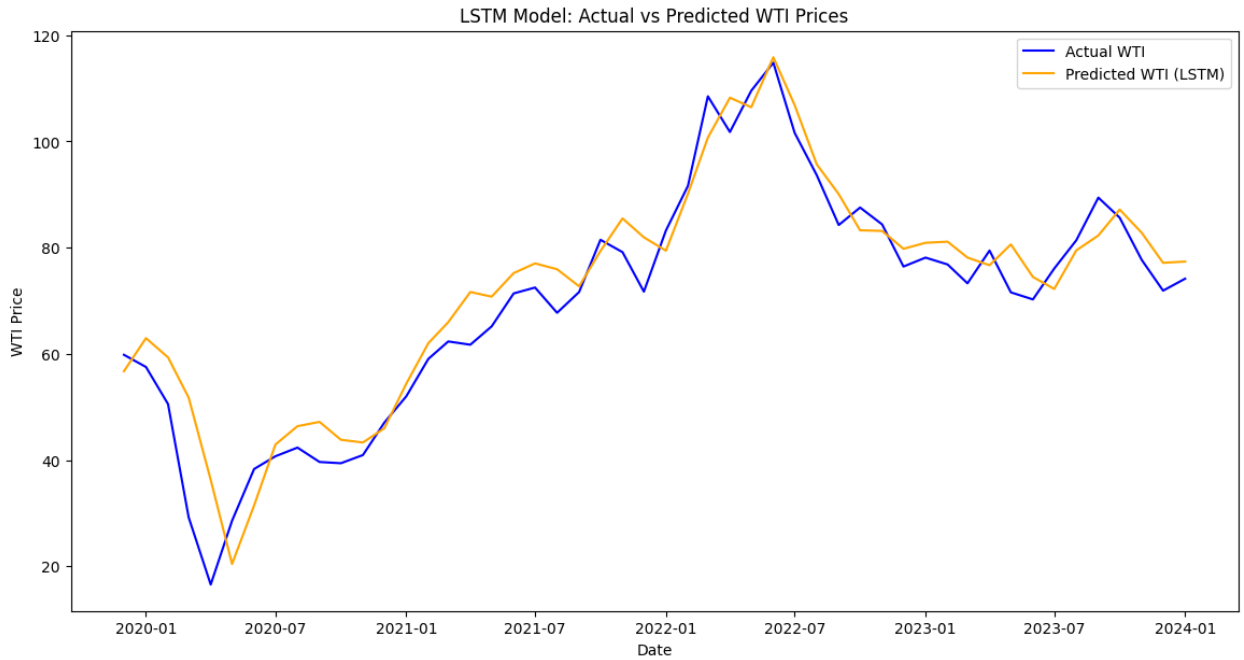
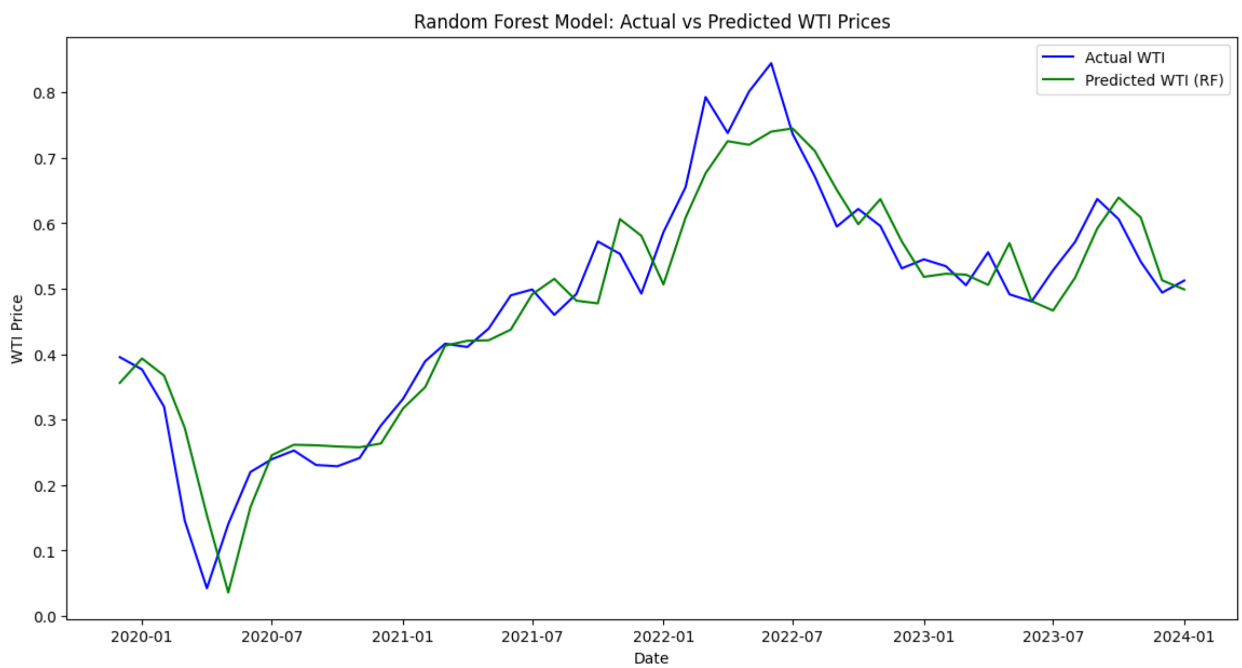


Figure 3.9. Actual vs Predicted Oil Prices RF after COVID-19 for Monthly Dataset

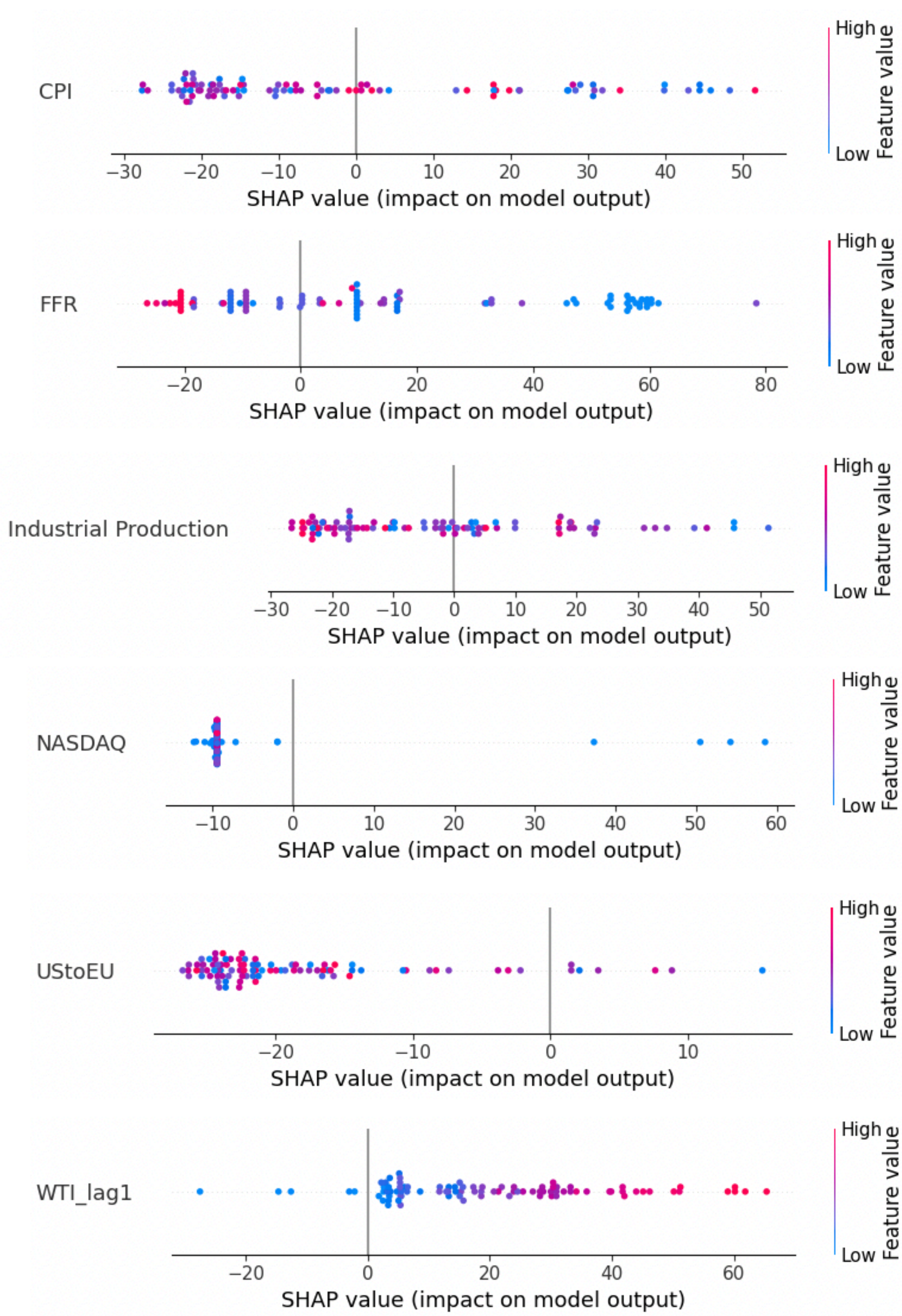


3.4.4 SHAP

The SHAP summary plot (Figure 3.10) illustrates the importance and impact of various economic predictors on the WTI crude oil price forecast. The previous day's WTI price

(WTI_lag1) emerges as the most significant predictor. The exchange rate has a notable negative impact when the dollar is strong, while industrial production and NASDAQ exert moderate influence.

Figure 3.10. SHAP Results



The SHAP analysis reveals that the WTI_lag1 has the most significant impact on WTI price predictions, with higher past prices strongly predicting higher future prices, indicating considerable inertia in the oil market. The USToEU exchange rate shows that lower rates generally decrease WTI prices, while the Federal Funds Rate (FFR) has a varied impact, with higher rates positively influencing WTI prices, possibly due to the expectation of slower economic growth reducing future oil supply. The Consumer Price Index (CPI) indicates that inflationary pressures tend to increase WTI prices, reflecting the cost-push effect. Industrial Production also positively influences WTI prices, though to a lesser extent, and the NASDAQ shows minimal impact, except for a few outliers.

3.5 Conclusion

The analysis of WTI crude oil prices using daily and monthly datasets after significant economic shocks reveals distinct differences between the short-term impact of the COVID-19 pandemic and the long-term effects of the Global Financial Crisis (GFC), demonstrates the effectiveness of machine learning models like Long Short-Term Memory (LSTM) and Random Forest (RF). For the daily dataset, the LSTM model consistently provided more accurate predictions compared to the RF model, particularly in the short-term forecasts. This is evident from the lower Mean Squared Error (MSE) and Mean Absolute Error (MAE) values observed in LSTM's performance over extended periods. Similarly, the monthly dataset revealed that LSTM outperformed RF in predicting crude oil prices post-COVID-19. RF's superior accuracy in long-term forecasts but LSTM is further highlighted by its ability to capture the intricate trends and fluctuations in oil prices, making it a more reliable model for short-term forecasting.

The SHAP analysis provided valuable insights into the predictors influencing WTI price forecasts. It indicates that the lagged WTI price was the most significant predictor, highlighting that past price trends heavily influence future market behavior. Among the macroeconomic variables, the Federal Funds Rate (FFR) demonstrated a notable positive impact on WTI prices, particularly when rates were higher, reflecting the influence of monetary policy on oil markets. The Consumer Price Index (CPI), representing inflationary pressures, also showed a positive impact, consistent with the cost-push effect in the economy. The US to EU exchange rate had a significant negative impact when the dollar was strong, while Industrial Production showed a balanced influence, and the NASDAQ had a minimal overall effect on WTI prices.

References

Alpaydin, E. (2020). Introduction to machine learning. MIT press.

Alper, C. E., & Torul, O. (2008). Effects of Oil Price Changes on Real Economic Activity: The Case of Turkey.

An, J., Mikhaylov, A., & Moiseev, N. (2019). Oil price predictors: Machine learning approach. *International Journal of Energy Economics and Policy*, *9*(5), 1-6.

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Atkins, A., Niranjan, M., & Gerding, E. (2018). Financial news predicts stock market volatility better than close price. *The Journal of Finance and Data Science*, *4*(2), 120-137.

Avramov, D., Cheng, S., & Metzker, L. (2023). Machine learning vs. economic restrictions: Evidence from stock return predictability. *Management Science*, *69*(5), 2587-2619.

Baumeister, C., & Kilian, L. (2016). Forty years of oil price fluctuations: Why the price of oil may still surprise us. *Journal of Economic Perspectives*, *30*(1), 139-160.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). High-dimensional methods and inference on structural and treatment effects. *Journal of Economic Perspectives*, *28*(2), 29-50.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, *24*.

Bernanke, B. S., & Boivin, J. (2003). Monetary policy in a data-rich environment. *Journal of monetary economics*, *50*(3), 525-546.

Bernanke, B. S., Boivin, J., & Eliasz, P. (2005). Measuring the effects of monetary policy: a factor-augmented vector autoregressive (FAVAR) approach. The Quarterly journal of economics, 120(1), 387-422.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of econometrics, 31(3), 307-327.

Bourghelle, D., Jawadi, F., & Rozin, P. (2021). *Oil price volatility in the context of Covid-19, 167, 39–49 [Le prix du pétrole dans le contexte du Covid 19]* (No. hal-04412020).

Cakici, N., Fieberg, C., Metko, D., & Zaremba, A. (2023). Machine learning goes global: Cross-sectional return predictability in international stock markets. *Journal of Economic Dynamics and Control*, *155*, 104725.

Cao, L. J., & Tay, F. E. (2000). Feature selection for support vector machines in financial time series forecasting. In *Intelligent Data Engineering and Automated Learning—IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents: Second International Conference Shatin, NT, Hong Kong, China, December 13–15, 2000 Proceedings 2* (pp. 268-273). Springer Berlin Heidelberg.

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

Chernozhukov, V., Hansen, C., & Spindler, M. (2015). Post-selection and post-regularization inference in linear models with many controls and instruments. *American Economic Review*, *105*(5), 486-490.

Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytics Journal*, *2*, 100015.

Cochrane, J. H. (2011). Presidential address: Discount rates. *The Journal of finance*, *66*(4), 1047-1108.

Corradi, V., & Swanson, N. R. (2013). *A survey of recent advances in forecast accuracy comparison testing, with an extension to stochastic dominance* (pp. 121-143). Springer New York.

Croushore, D. (2011). Frontiers of real-time data analysis. *Journal of economic literature*, *49*(1), 72-100.

Coibion, O., & Gorodnichenko, Y. (2012). Why are target interest rate changes so persistent?. American Economic Journal: Macroeconomics, 4(4), 126-162.

Clarida, R., Gali, J., & Gertler, M. (2000). Monetary policy rules and macroeconomic stability: evidence and some theory. The Quarterly journal of economics, 115(1), 147-180.

Ding, Y. (2018). A novel decompose-ensemble methodology with AIC-ANN approach for crude oil forecasting. *Energy*, *154*, 328-336.

Dong, X., Li, Y., Rapach, D. E., & Zhou, G. (2022). Anomalies and the expected market return. The Journal of Finance, 77(1), 639-681

Doroodian, K., & Boyd, R. (2003). The linkage between oil price shocks and economic growth with inflation in the presence of technological advances: a CGE model. *Energy Policy*, *31*(10), 989-1006.

Fama, E. F., & French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of financial economics*, *25*(1), 23-49.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of financial economics*, *116*(1), 1-22.

Fan, Y., & Xu, J. H. (2011). What has driven oil prices since 2000? A structural change perspective. *Energy Economics*, *33*(6), 1082-1094.

Gao, X., Wang, J., & Yang, L. (2022). An explainable machine learning framework for forecasting crude oil price during the covid-19 pandemic. *Axioms*, *11*(8), 374.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, *63*, 3-42.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223-2273.

Guo, J., & Tuckfield, B. (2020, September). News-based machine learning and deep learning methods for stock prediction. In *Journal of Physics: Conference Series* (Vol. 1642, No. 1, p. 012014). IOP Publishing.

Han, Y., He, A., Rapach, D., & Zhou, G. (2023). Cross-sectional expected returns: New fama-macbeth regressions in the era of machine learning. *Available at SSRN 3185335*.

Harvey, C. R., Liu, Y., & Zhu, H. (2016). … and the cross-section of expected returns. *The Review of Financial Studies*, *29*(1), 5-68.

He, K., Yu, L., & Lai, K. K. (2012). Crude oil price analysis and forecasting using wavelet decomposed ensemble model. *Energy*, *46*(1), 564-574.

Herrera, G. P., Constantino, M., Tabak, B. M., Pistori, H., Su, J. J., & Naranpanawa, A. (2019). Long-term forecast of energy commodities price using machine learning. *Energy*, *179*, 214-221.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67.

Horn, M. (2004). OPEC's optimal crude oil price. *Energy Policy*, *32*(2), 269-280.

Hou, K., Xue, C., & Zhang, L. (2020). Replicating anomalies. *The Review of financial studies*, *33*(5), 2019-2133.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Iglesias, E. M., & Rivera-Alonso, D. (2022). Brent and WTI oil prices volatility during major crises and Covid-19. *Journal of Petroleum Science and Engineering*, *211*, 110182.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

Jena, P. R., & Majhi, R. (2023). Are Twitter sentiments during COVID-19 pandemic a critical determinant to predict stock market movements? A machine learning approach. *Scientific African*, *19*, e01480.

Kamah, M., & Riti, J. S. (2020). Dissecting the Linkages between Variations in Crude Oil Price and Selected Macroeconomic Variables in Nigeria. *Journal of Contemporary Research in Business, Economics and Finance*, *2*(2), 37-46.

Karolyi, G. A., & Van Nieuwerburgh, S. (2020). New methods for the cross-section of returns. *The Review of Financial Studies*, *33*(5), 1879-1890.

Keen, B. D. (2010). The signal extraction problem revisited: a note on its impact on a model of monetary policy. *Macroeconomic Dynamics*, *14*(3), 405-426.

Kilian, L. (2009). Oil price shocks, monetary policy and stagflation.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International journal of computer science*, *1*(2), 111-117.

Landefeld, J. S., Seskin, E. P., & Fraumeni, B. M. (2008). Taking the pulse of the economy: Measuring GDP. *Journal of Economic Perspectives*, *22*(2), 193-216.

Li, X., Shang, W., & Wang, S. (2019). Text-based crude oil price forecasting: A deep learning approach. *International Journal of Forecasting*, *35*(4), 1548-1560.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Lin, W. C., Tsai, C. F., & Chen, H. (2022). Factors affecting text mining based stock prediction: Text feature representations, machine learning models, and news platforms. *Applied Soft Computing*, *130*, 109673.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. *Entropy*, *23*(1), 18.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Riaz, M., Mushtaq, S., & Atta, A. (2020). Oil Price Flux and Macroeconomy of Oil Exporters. *Journal of Accounting and Finance in Emerging Economies*, *6*(2), 651-667.

Malladi, R. K. (2024). Application of supervised machine learning techniques to forecast the COVID-19 US Recession and stock market crash. *Computational Economics*, *63*(3), 1021-1045.

Martin, I. W., & Nagel, S. (2022). Market efficiency in the age of big data. *Journal of financial economics*, *145*(1), 154-177.

McLean, R. D., & Pontiff, J. (2016). Does academic research destroy stock return predictability?. *The Journal of Finance*, *71*(1), 5-32.

Mena, B. (2023, May 25). The US economy grew faster in the first quarter than previously reported. Retrieved July 23, 2023, from https://www.cnn.com/2023/05/25/business/q1-gdp-second-estimate/index.html

Mingming, T., & Jinliang, Z. (2012). A multiple adaptive wavelet recurrent neural network model to analyze crude oil prices. *Journal of Economics and Business*, *64*(4), 275-286.

Narayan, P. K. (2020). Oil price news and COVID-19—Is there any connection?. *Energy Research Letters*, *1*(1).

Orphanides, A. (2001). Monetary policy rules based on real-time data. *American Economic Review*, *91*(4), 964-985.

Panella, M., Barcellona, F., & D'ecclesia, R. L. (2012). Forecasting energy commodity prices using neural networks. *Advances in Decision Sciences*.

Pentreath, N. (2015). *Machine learning with spark* (p. 338). Birmingham: Packt Publishing.

Rapach, D., & Zhou, G. (2022). Asset pricing: Time-series predictability.

Rapach, D. E., & Zhou, G. (2020). Time-series and cross-sectional stock return forecasting: New machine learning methods. *Machine learning for asset management: New developments and financial applications*, 1-33.

Sen, A., & Choudhury, K. D. (2024). Forecasting the Crude Oil prices for last four decades using deep learning approach. *Resources Policy*, *88*, 104438.

Souma, W., Vodenska, I., & Aoyama, H. (2019). Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*, *2*(1), 33-46.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Tian, G., Peng, Y., & Meng, Y. (2023). Forecasting crude oil prices in the COVID-19 era: Can machine learn better?. *Energy Economics*, *125*, 106788.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *58*(1), 267-288.

Wang, J., Athanasopoulos, G., Hyndman, R. J., & Wang, S. (2018). Crude oil price forecasting based on internet concern using an extreme learning machine. *International Journal of Forecasting*, *34*(4), 665-677.

Wheeler, C. M., Baffes, J., Kabundi, A., Kindberg-Hanlon, G., Nagle, P. S., & Ohnsorge, F. (2020). *Adding fuel to the fire: cheap oil during the COVID-19 pandemic*. The World Bank.

White, H. (2000). A reality check for data snooping. *Econometrica*, *68*(5), 1097-1126.

Xie, C., Rajan, D., & Chai, Q. (2021). An interpretable Neural Fuzzy Hammerstein-Wiener network for stock price prediction. *Information Sciences*, *577*, 324-335.

Yun, K. K., Yoon, S. W., & Won, D. (2023). Interpretable stock price forecasting model using genetic algorithm-machine learning regressions and best feature subset selection. *Expert Systems with Applications*, *213*, 118803.

Zhang, K., Hong, M., Zhang, K., & Hong, M. (2022). Forecasting crude oil price using LSTM neural networks. *Data Sci. Financ. Econ*, *2*, 163-180.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301-320.

Appendix A -

- DIV: Dividends
- NCON: Nominal Personal Consumption Expenditures
- NDPI: Nominal Disposable Personal Income
- NOUTPUT: Nominal GNP/GDP
- NPI: Nominal Personal Income
- NPSAV: Nominal Personal Saving
- OLI: Other Labor Income
- P: Price Index for GNP/GDP
- PCON: Price Index for Personal Consumption Expenditures
- PINTPAID: Interest Paid by Consumers
- PROP: Proprietors' Income
- PTAX: Personal Tax & Nontax Payments
- RATESAV: Personal Saving Rate, Constructed
- RCON: Real Personal Consumption Expenditures: Total
- RCOND: Real Personal Consumption Expenditures: Durable Goods
- RCONS: Real Personal Consumption Expenditures: Services
- RENTI: Rental Income of Persons
- REX: Real Exports of Goods and Services
- RG: Real Government Consumption & Gross Investment: Total
- RGF: Real Government Consumption & Gross Investment: Federal
- RGSL: Real Government Consumption & Gross Investment: State and Local

- RIMP: Real Imports of Goods and Services
- rinvbf: Real Gross Private Domestic Investment: Nonresidential
- rinvchi: Change in Private Inventories
- rinvresid: Real Gross Private Domestic Investment: Residential
- RNX: Real Net Exports of Goods and Services
- ROUTPUT: Real GNP/GDP
- SSCONTRIB: Personal Contributions for Social Insurance
- TRANPF: Personal Transfer Payments to Foreigners
- TRANR: Transfer Payments
- WSD: Wage and Salary Disbursements