Inland Norway University of Applied Sciences

How Can High-Frequency Data and Automatic Feature Selection Be Used to Improve Forecasts of Volatility and Value-at-Risk for Brent Crude Oil Futures?

Hvordan kan Høyfrekvente Data og Automatiske Funksjonsvalg Brukes til å Forbedre Prognoser for Volatilitet og Value-at-Risk for Brent-Råoljefutures?

Beressa Geleta Abdissa, candidate no. 100

Supervisor: Professor Erik Haugom

Master thesis, Economics and Business Administration

Major: Digital Management and Business Analytics

May 2024

Inland School of Business and Social Sciences

Acknowledgments

The journey through the program, from start to completion of my final thesis, was a blend of challenges and excitement. This period was immensely educational, and the drives to Kongsvinger and Rena have left me with lasting memories. I am deeply thankful to God for the courage and support bestowed upon me throughout this experience. Having said that, I am profoundly grateful for the support and encouragement I received during my master's program, which was pivotal in the completion of this thesis.

To my children, Christian, Bethel, and Michael: thank you for your understanding and patience with the time I had to spend away from you. Your support has been my strength. To my wife, Kidist (Babye): your ability to push me, even during our most difficult moments, has often inspired me to transform obstacles into stepping stones toward success. Thank you for being the unwavering force that drives me to exceed my limits. To Wondwesen (Wondye), your insightful advice, practical guidance, and prayers have illuminated my path and brought this thesis to fruition. Your ability to pinpoint and resolve issues has been nothing short of inspirational.

I extend my deepest gratitude to my supervisor, Professor Erik, whose unwavering and solid advice was invaluable. Your availability and guidance throughout the year have greatly enriched my academic experience. Special thanks to Dr. Joshi, who assisted me with the Python coding aspects of my research. Your expertise was crucial in navigating the technical challenges of my thesis. Thank you, Brooke, for your meticulous editing work. Your contributions have significantly enhanced the clarity and quality of my thesis. To Jacqueline (Jackie), Mebrate (Masha) and Ayisanew (Ayise), I am immensely thankful for your presence, especially during the challenging times of completing this thesis. Your support helped me persevere through the toughest moments.

Inland School of Business and Social Sciences

Kongsvinger, May 2024

Beressa Geleta Abdissa

Table of Contents

List of Figures

List of Tables

Abstract

This master's thesis investigates how high-frequency data and automatic feature selection can enhance volatility and Value-at-Risk (VaR) forecasts for Brent Crude Oil futures. The research identifies a significant gap in the utilization of these advanced data analytics techniques in improving financial risk management predictions. The primary objective of this study is to determine the effectiveness of integrating high-frequency data and machine learning feature selection to refine forecasts of volatility and VaR. A comprehensive methodology combining statistical analysis and machine learning models, including neural networks and regression analyses, is applied to high-frequency trading data. The findings reveal that feature selection significantly improves the accuracy of volatility forecasts. In addition, models incorporating high-frequency data outperform traditional forecasting models, demonstrating more precise risk assessments and better decision-making tools for financial analysts and traders. The study concludes that employing high-frequency data and automated feature selection can significantly affect risk management strategies, offering robust tools for more accurate forecasting in financial markets. These advancements provide a foundation for future research aimed at integrating more complex algorithms and data sources to further enhance predictive accuracy in financial risk management.

Keywords: Value-at-Risk (VaR), Realized Volatility, Machine Learning, Model Selection, Brent Crude Oil Futures

Sammendrag

Denne masteroppgaven undersøker hvordan høyfrekvente data og automatisk funksjonsvalg kan forbedre prognoser av volatiliteten og Value-at-Risk (VaR) for Brent-Råoljefutures. Forskningen identifiserer et betydelig gap i bruken av disse avanserte dataanalyseteknikkene for å forbedre prediksjoner til bruk i finansiell risikostyring. Hovedmålet med denne studien er å undersøke verdien av å integrere høyfrekvente data og maskinlæringsfunksjoner til prognoseformål for volatilitet og VaR. En omfattende metodikk som kombinerer statistisk analyse og maskinlæringsmodeller, inkludert nevrale nettverk og regresjonsanalyser, blir testet på høyfrekvente handelsdata. Funnene viser at automatiske funksjonsvalg forbedrer nøyaktigheten av volatilitetsprognoser betydelig. I tillegg viser resultatene at modeller som baserer seg på høyfrekvente data er mer nøyaktige sammenlignet med modeller som baserer seg på tradisjonelle prognoseteknikker, og er et godt utgangspunkt for presise risikovurderinger og bedre beslutningsverktøy for finansanalytikere og beslutningstakere i denne sektoren. Studien konkluderer med at bruk av høyfrekvente data og automatisert funksjonsvalg kan påvirke risikostyringsstrategier betydelig, og tilbyr robuste verktøy for mer nøyaktige prognoser i finansmarkedene. Resultatene gir videre et grunnlag for fremtidig forskning rettet mot å integrere mer komplekse algoritmer og datakilder for å ytterligere forbedre predikativ nøyaktighet i finansiell risikostyring.

Nøkkelord: Value-at-Risk (VaR), Realisert volatilitet, Maskinlæring, Modellvalg, Råoljefutures

Abbreviations

AIC – Akaike's information criterion

ANN – artificial neural network

ARCH – autoregressive conditional heteroscedasticity

ARFIMA – autoregressive fractionally integrated moving average

ARIMA – autoregressive integrated moving average

BIC – Bayesian information criterion

CV – cross validation

ES – expected shortfall

GARCH – generalized autoregressive conditional heteroscedasticity

HAR-RV – heterogeneous autoregressive model of realized volatility

HFT – high frequency trading

ICE – Intercontinental Exchange

LSTM – long short-term memory

MAE – mean absolute error

MLP – multi-layer perceptron

MSE – mean squared error

OHLC – open, high, low, close

OLS – ordinary least squares

RF – random forest

SVM – support vector machine

SVR – support vector regression

VaR – Value-at-Risk

1 Introduction

1.1 Background

Risk management in financial markets, especially in commodity trading, relies heavily on accurate volatility forecasting and Value-at-Risk (VaR) estimation (Kambouroudis et al., 2016). Brent Crude Oil market dynamics are shaped by diverse factors, and the precise prediction of volatility and VaR plays a pivotal role in the decision-making processes of traders and financial institutions. This study delves into the intricate dynamics of volatility in financial markets, with a specific focus on Intercontinental Exchange (ICE) Brent Crude Oil futures. One critical question underpinning this research is identifying which components of volatility, calculated from various sampling frequencies, play a pivotal role in crafting accurate forecasts for volatility and VaR for Brent Crude Oil futures. This inquiry is vital for both theoretical understanding and practical applications in financial risk management.

To address this question, the study endeavours to uncover how leveraging high-frequency data, alongside automatic feature selection techniques for machine learning, can substantially refine the accuracy of volatility and VaR predictions for Brent Crude Oil futures. By systematically exploring the interplay between data granularity and advanced analytical methodologies, the study aims to contribute to the optimization of forecasting models. This approach not only holds the potential to advance the precision of financial forecasts but also to offer invaluable insights into the mechanisms driving market volatility, thus supporting more informed decision-making processes in the volatile domain of commodity futures.

The primary research inquiry guiding this study is: How can high-frequency data and automatic feature selection techniques for machine learning be leveraged to enhance forecasts of volatility and VaR for Brent Crude Oil futures? By addressing this query, the aim is to unveil novel insights into refining forecasting models within the commodity trading domain (Kambouroudis et al., 2016). The utilization of high-frequency data allows for a more granular comprehension of market movements, capturing subtle patterns and fluctuations that traditional methods may overlook. Additionally, automatic feature selection techniques for machine learning provide a systematic methodology for identifying the most influential variables driving volatility and risk in Brent Crude Oil futures trading.

The study holds promise for enhancing risk management practices by providing traders and financial institutions with improved forecasts of volatility and VaR, potentially leading to more accurate and timely insights. Through harnessing high-frequency data and automatic feature selection, market participants can gain a competitive edge in navigating the intricacies of commodity markets, empowering them to make well-informed decisions, and effectively mitigate risks (Louzis et al., 2014).

In the subsequent sections of this study, an in-depth exploration of the methodologies employed, the analysis of high-frequency data, the application of automatic feature selection techniques, and the implications of the findings on volatility and VaR forecasting for Brent Crude Oil futures will be presented. This research endeavours to contribute valuable insights to the realm of financial risk management and to establish a foundation for more robust and reliable forecasting models in commodity trading markets.

1.2 Contribution of the Study

The present study systematically uncovers the strengths and intricacies of volatility forecasting within the financial sector, particularly focusing on Brent Crude Oil futures. It makes several important contributions to the field by empirically evaluating traditional and more recently developed modelling approaches and integrating these models to comparatively examine the robustness of volatility analysis. This study also provides several practical implications for model optimization for risk management, as well as advancing the theory of both traditional and neural network-based approaches.

By empirically evaluating the performance of traditional econometric models and juxtaposing them with the prowess of machine learning algorithms under a spectrum of market conditions, this study enriches the empirical literature with fresh insights and evidence. Next, by providing a robust comparative analysis that highlights the unique advantages and constraints of econometric methods and machine learning techniques in the context of volatility forecasting, this research marks a significant stride in integrating these methodologies. By offering a granular analysis of model performance in real-world trading scenarios, this study contributes to the optimization of risk management practices, aiding practitioners in making more informed and strategic decisions in the dynamic landscape of commodity trading. Finally, by delving into the comparative merits of models like generalized autoregressive conditional heteroskedasticity (GARCH) and autoregressive integrated moving average (ARIMA) versus neural network-based approaches such as long short-term memory (LSTM), this study expands the theoretical framework informing commodity trading forecasting and fosters a deeper understanding of these methods' capabilities to capture market volatility and risk.

In essence, the study contributes to bridging theoretical gaps in financial risk management by integrating traditional econometric models with modern machine learning techniques. By comparing the predictive capabilities of models like GARCH, ARIMA, and LSTM under various market conditions, the study offers insights into the trade-offs between model complexity and forecasting accuracy, advancing the theoretical understanding of volatility forecasting and VaR prediction. Furthermore, the research provides empirical evidence of the models' performance, highlighting the importance of integrating established theories with cutting-edge technologies to enhance risk assessment and decision-making in financial markets. Overall, the study not only bridge theoretical gaps by exploring the strengths and limitations of different modelling approaches but also paves the way for practical advancements in financial risk management by offering valuable insights for model selection and risk quantification in dynamic market environments.

1.3 Objectives of the Study

The objectives of this study address the multifaceted nature of volatility in financial markets, with a focus on the following areas:

1. High-frequency data utilization: Harnessing the power of high-frequency trading (HFT) data to capture the nuances of market volatility, aiming to improve the accuracy and reliability of predictive models for Brent Crude Oil futures.

2. Predictive model evaluation: Evaluating a diverse array of forecasting models, from well-established econometric models to cutting-edge machine learning algorithms, to ascertain their predictive accuracy and robustness.

3. Feature selection and optimization: Employing sophisticated feature selection techniques to distil the most relevant predictors from the data, thus enhancing the models' forecasting precision.

4. Risk management strategies: The ultimate objective is to translate the analytical findings into actionable strategies for risk management, particularly for entities involved in commodities trading, offering a competitive edge in the high-stakes arena of futures markets.

By achieving these objectives, the study aims to contribute substantially to the existing body of knowledge and to the operational methodologies employed within the financial trading community.

1.4 Research Questions

How can high-frequency data and automatic feature selection be used to improve forecasts of volatility and VaR for Brent Crude Oil futures?

This question focuses on enhancing the accuracy of forecasting models for both volatility and VaR in the context of Brent Crude Oil futures trading. The study will elaborate on the ways in which high-frequency data and automatic feature selection can be leveraged to improve these forecasts.

1.4.1 High-Frequency Data

High-frequency data plays a pivotal role in enhancing the understanding of price movements and market dynamics by providing detailed insights at more frequent intervals, such as intra-day or tick data. This granular level of data allows analysts to capture short-term fluctuations and patterns in asset prices, leading to increased precision in forecasting volatility and VaR. By leveraging high-frequency data, analysts can make more accurate predictions, enabling them to react swiftly to changing market conditions and to make informed decisions promptly. This real-time access to market updates empowers traders and risk managers to stay ahead of market trends and adjust their strategies effectively in response to dynamic market environments (Virgilio, 2019).

1.4.2 Automatic Feature Selection

Automatic feature selection algorithms play a crucial role in identifying the most relevant variables or features that significantly contribute to forecasting models, thereby reducing the dimensionality of the dataset, and improving model efficiency and interpretability. By selecting the most informative features, automatic feature selection enhances the predictive power and generalization capabilities of forecasting models, ultimately leading to improved model performance (Khaire & Dhanalakshmi, 2022).

1.4.3 Forecasting Volatility

The combination of high-frequency data and automatic feature selection techniques offers a powerful approach to modelling volatility dynamics in Brent Crude Oil futures markets. By leveraging high-frequency data, analysts can capture the intricate and dynamic nature of volatility, allowing for a more nuanced understanding of market behaviour. Automatic feature selection plays a key role in this process by identifying relevant features from the high-frequency data, enabling forecasting models to incorporate crucial market information that directly affects volatility patterns. This integration of data-driven insights and feature selection enhances a model's ability to adapt to changing market conditions and make more accurate predictions regarding volatility in Brent Crude Oil futures markets (Chen et al., 2020; Du et al., 2023).

1.4.4 Forecasting Value-at-Risk

Value at Risk is a widely used risk management measure that quantifies the potential loss in value of a portfolio over a specified time horizon and at a given confidence level. It provides a comprehensive assessment of the maximum loss that a portfolio could incur under normal market conditions. For traders and financial institutions, improved forecasts of VaR that use high-frequency data and automatic feature selection are essential for effective risk management. By accurately estimating VaR, market participants can make informed decisions regarding capital allocation and risk mitigation strategies. From a regulatory perspective, precise VaR forecasts are crucial for compliance with risk management standards and regulatory guidelines. Regulatory authorities often require financial institutions to maintain sufficient capital reserves to cover potential losses, and accurate VaR calculations play a vital role in ensuring adherence to these regulatory requirements. In summary, VaR serves as a critical tool for both risk management practices and regulatory compliance in the financial industry (Fallon, 1996).

By integrating high-frequency data and automatic feature selection techniques into the forecasting process for volatility and VaR of Brent Crude Oil futures, market participants gain a competitive edge by enhancing the accuracy of their risk assessments and decision-making capabilities. This integration enables traders and financial institutions to delve deeper into market dynamics and risk factors, leading to more informed decisions, effective risk management strategies, and optimized trading approaches. The utilization of advanced data analysis techniques not only improves the precision of forecasts but also empowers market participants to proactively respond to market fluctuations and capitalize on opportunities with a heightened understanding of the underlying factors influencing Brent Crude Oil futures (Laopodis, 2021).

1.5 Methodology Overview

This study employs a comprehensive methodological framework that integrates various statistical and computational techniques to investigate volatility forecasting in financial markets. The research begins with a meticulous data acquisition process, focusing on collecting HFT data to precisely capture the intricacies of market movements. An extensive descriptive analysis follows, ensuring a profound understanding of the data's characteristics before delving into complex predictive modelling. Finally, the study aims to comprehensively compare the forecasting performance of traditional econometric models such as GARCH and ARIMA with advanced machine learning algorithms like LSTM and neural networks.

Through feature engineering techniques, the methodology enhances the predictive models by selecting only the most relevant variables for inclusion. Robust findings are ensured through rigorous model validation techniques, including out-of-sample testing and cross-validation. The study utilizes risk assessment metrics, such as VaR, to evaluate the efficacy of forecasting models. In addition, to further enhance the understanding of volatility dynamics, the expected shortfall (ES) metric is utilized, which provides a comprehensive view of risk management by considering extreme losses, risk-adjusted returns, varying risk levels over time, tail risks, and downside risk scenarios (Mehta, & Yang, 2022). This multifaceted methodological approach is designed to provide a nuanced understanding of volatility dynamics and offer valuable insights into risk management practices in the financial sector.

1.6 Structure of the Study

The study is structured to guide readers through the research process, starting with the Introduction that establishes the research background, significance, and objectives. The Literature Review offers a comprehensive overview of existing theories and research, contextualizing the study within the academic landscape. The Methodology section describes the research design, data collection methods, and analytical techniques used to fulfil the study's objectives. The Empirical Analysis provides a thorough exploration of the data, applies selected models to the data, and evaluates their predictive performances (Forman, 2003). The Results section then presents the study's findings, providing crucial insights into the effectiveness of various forecasting models. Finally, the Discussion segment interprets the results, drawing connections with established literature and highlighting both theoretical and practical implications derived from the research.

The recent integration of HFT in commodities, with a specific focus on crude oil trading, signifies a significant transformation within the financial landscape. The incorporation of high-frequency data in this sector is driven by its crucial role in enhancing risk management practices, refining derivative pricing mechanisms, and optimizing portfolio selection strategies, thereby capturing the attention of energy researchers, market participants, and policymakers (Ewald et al., 2023). This surge in the prevalence of HFT has not only revolutionized the operational dynamics of trading activities but has also paved the way for novel avenues of exploration and innovation in financial markets, fostering a climate ripe for advanced research endeavours and strategic development initiatives.

1.7 Ethics Consideration

In conducting this research on HFT data for Brent Crude Oil, ethical considerations have been paramount to ensure the integrity and social responsibility of analytical processes. Given the sensitive nature of the data and its potential implications on market behaviour, the study adheres strictly to ethical standards. First, the study ensures data privacy and confidentiality. To protect the privacy of the data sources and maintain the confidentiality of market-sensitive information, all datasets utilized in this study have been anonymized and aggregated. No identifiable information pertaining to the data providers, such as specific transaction details or trader identities, has been used or disclosed. The study employs data encryption and secure data storage techniques to safeguard against unauthorized access and ensure that the data integrity is maintained throughout the research process (Dinev & Hart, 2004).

Second, this study ensures regulatory compliance by complying with all applicable financial regulations and ethical guidelines set forth by relevant financial oversight bodies. These include adherence to the principles of responsible data handling as outlined by the Securities and Exchange Commission (SEC) and other regulatory entities that govern financial markets. Before the commencement of the study, all necessary permissions were obtained from data providers and regulatory authorities to ensure compliance with legal standards (Bessembinder & Maxwell, 2008).

3. Impartiality and objectivity: The methodologies and analyses presented in this study are designed to be impartial and objective. The choice of models and techniques was based on ensuring accuracy and fairness in drawing conclusions about market behaviour and predicting volatility (Kahneman & Tversky, 1979).

4. Mitigation of market impact: Given the impact that study findings can have on market behaviours, especially in volatile commodity markets such as crude oil, the dissemination of results has been approached with care and consideration. This sensitivity acknowledges the potential influence that the research outcomes may have on market participants, ensuring that the information is shared responsibly to mitigate any unintended consequences or disruptions in the market. The study avoids speculative assertions and is presented with cautionary notes regarding its applicational limits to prevent misuse of the data or findings that could lead to market manipulation (Barberis et al., 2005).

By adhering to these ethical principles, the study not only contributes valuable insights into the volatility of crude oil markets but also ensures that these contributions do not compromise ethical standards or market integrity. This approach underscores the commitment to conducting financially impactful research within a framework that prioritizes social responsibility and ethical rigor.

2 Theoretical Framework and Literature Review

2.1 Theoretical Foundation

2.1.1 The Theory of Volatility

To analyse and understand volatility in financial markets, various models and theories have been developed. These models are designed to uncover the underlying patterns, drivers, and characteristics of volatility, offering valuable insights into the behaviour of asset prices. The overarching framework that underpins this study is the theory of volatility. In financial markets, the theory of volatility pertains to the degree of variation or dispersion of returns for a specific asset or market over a certain period. It describes the concept of asset value fluctuation within a given timeframe (Patton, 2011). This fundamental concept is essential for comprehending asset price dynamics and implementing effective risk management strategies, and it serves as a critical metric for assessing risk and uncertainty in financial markets. This measure significantly influences investment decisions, portfolio management strategies, and the pricing of derivatives. For investors, traders, and financial analysts, a deep understanding of volatility theory is crucial for evaluating and mitigating the risks associated with price fluctuations (Haugom et al., 2014).

2.1.2 Financial Asset Pricing and Returns

Financial asset prices reflect the value of an asset at a given point in time. For assets traded on an exchange, the reported prices could be bid prices, ask prices, an average of bid and ask, opening prices, closing prices, the highest or lowest price recorded over the trading day, or an actual transaction price. According to Taylor (2005) asset prices should be defined using one price per period and recorded with a constant frequency such as the market's close. Formally, the price of a given asset at time t can then be defined as:

$$P_{i,t}, \tag{1}$$

where *P* is an asset's price at sub-period i on day t at the daily close of the market. It can also be the price recorded at the end of each week, month, or year. In recent years, much of the trading takes place at the intra-day level, and the asset price then reflects sampling frequencies within the day.

These temporal price fluctuations mean that price series are almost always non-stationary, which makes prices inappropriate to use in many statistical applications. The preferred choice when conducting empirical investigations of financial assets involves returns instead of prices themselves. There are several reasons for this. First, returns reflect complete and scale-free summaries of investment opportunities. Returns are also easier to handle than price series because of their statistical properties (Campbell et al., 1997).

The distributional properties of returns are fundamental in understanding volatility dynamics and forecasting, underscoring their importance in financial modelling and risk management. A key aspect is the second moment structure of the conditional return distribution, which evolves over time and significantly influences risk assessment (Thomakos & Wang, 2003). Analysing return distributions provides valuable insights into the trade-off between risk and return, optimal portfolio allocation strategies, and the likelihood of significant fluctuations in portfolio value. Moreover, the distribution of returns plays a pivotal role in pricing financial instruments, assessing portfolio performance, and guiding strategic decisions in financial markets.

The distributional properties of returns encompass several key aspects:

1. Conditional distribution of returns: This property involves examining the distribution of asset returns based on specific information or conditions. Understanding the conditional distribution is essential for modelling volatility and making precise forecasts.

2. Excess kurtosis: This property measures the deviation from the expected shape of a normal distribution. Excess kurtosis impacts volatility forecast accuracy and may need to be quantified and corrected for during various data generation processes.

3. Non-normality of returns: Addressing the non-normality of returns is crucial, as asset returns frequently deviate from a normal distribution. This non-normality can affect volatility modelling and forecast precision, particularly in handling extreme observations or irregular data patterns (Patton, 2011).

There are several ways to measure returns, with the most basic being simple returns. The simple returns equation is typically calculated as the difference between the current price and the previous price, divided by the previous price:

$$R_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}, \tag{2}$$

where $R_{i,t}$ is the simple return for sub-period i on day t, $P_{i,t}$ is the price at time t for sub-period i, and $P_{i,t-1}$ is the price at time $t-1$ for sub-period i. This formula calculates the percentage of change in price from one period to the next. The calculation of simple returns, representing the percentage change in asset prices over a single period, is fundamental in financial modelling. Understanding the nuances of simple returns aids in analysing asset performance and constructing predictive models (Louzis et al., 2014; Wang, & Chan 2007).

Another commonly used measure of returns is the log returns equation, which is used to calculate the percentage change in the price of an asset over a specific period (Kambouroudis et al., 2016):

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{3}$$

The daily logarithmic returns of a financial asset, denoted as r_t , are defined as the difference between the logarithmic asset price observed at day *t* (denoted as P_t) and the price observed at day t-*1* (denoted as P_{t-1}).

The log returns equation offers several advantages that make it commonly used in financial modelling and analysis. One key advantage is its ability to capture the impact of lagged returns on correlations, which is crucial for assessing market volatility and interdependencies between assets and allowing for the examination of asymmetric effects in market relationships. By incorporating lagged returns and indicator functions, the equation provides a framework to assess how past returns influence current correlations, shedding light on the dynamics of market interactions. This feature is particularly valuable in understanding the nuanced behaviour of financial assets and can help in identifying patterns of asymmetry that may not be evident through traditional modelling approaches. Additionally, the equation's flexibility in accommodating different variables and specifications makes it a versatile tool for studying complex relationships in futures markets, enhancing the depth and accuracy of financial analyses (Thomakos & Wang, 2003).

In the modelling process, the daily return r_t is described as a combination of a conditional mean μ_t and an error term ε_t, which can be represented as the following equation:

$$r_t = \mu_t + \varepsilon_t = \mu_t + \sigma_t z_t, \tag{4}$$

where z_t follows a standard normal distribution, denoted as $z_t \sim N(0,1)$.

To address the inherent serial autocorrelation in financial asset returns, modelling the conditional mean should be done using an autoregressive model of order *1* (AR(1)) specification (Tsay, 2010). This specification assumes that the expected value of the current return r_t given the information available up to the previous period I_{t-1} is a linear function of a constant *C* and the lagged return r_{t-1} multiplied by a coefficient ϕ_t. Mathematically, this can be expressed as:

$$E(r_t|I_{t-1}) = C + \phi_1 r_{t-1} \tag{5}$$

The use of these equations and a thorough understanding of asset pricing, returns, and the nuances of the distributional qualities of returns are critical in further modelling and evaluating market risk and volatility.

2.1.3 Volatility and Risk

2.1.3.1 Volatility in financial markets reflects the degree of fluctuation or variation in the prices of financial instruments over a specific period. Volatility is commonly used as a measure of risk because higher volatility suggests the potential for significant price movements, both upward and downward. Investors and traders closely monitor volatility because it can affect their investment decisions. When investors assess risk in the financial markets, they often look closely at volatility, which serves as a crucial indicator of the potential risks and uncertainty associated with an investment. This uncertainty can lead to both higher potential gains and higher potential losses for investors (Poon & Granger, 2003).

Investors consider volatility as a key factor in understanding the level of risk associated with a particular asset or portfolio. By analysing volatility, investors can gauge the potential impact of price fluctuations on their investments (Peters, 1996; Shiller, 1992). High volatility implies a more dynamic market environment where prices can change rapidly and by significant amounts. This dynamic nature of high volatility introduces a level of unpredictability that investors must navigate when making investment decisions. Furthermore, high volatility prompts investors to carefully evaluate their risk tolerance levels. Different investors have varying degrees of risk tolerance based on factors such as investment objectives, time horizon, and financial circumstances. Assessing volatility helps investors align their risk tolerance with the level of volatility present in the assets they are considering. This alignment is crucial for ensuring that investors are comfortable with the potential risks and rewards associated with their investment choices.

In addition to risk tolerance considerations, understanding volatility plays a vital role in portfolio diversification. By including assets with different levels of volatility in their portfolios, investors can spread risk and reduce the impact of extreme price movements on their overall investment performance. Portfolio diversification across assets with varying levels of volatility is a fundamental risk management strategy that helps investors mitigate the impact of market uncertainties and fluctuations, thereby managing risk (Poon & Granger, 2003). Investors working to diversify their portfolios aim to balance the volatility of their portfolio to effectively mitigate overall risk exposure. Assets with low correlation in volatility are particularly valuable in this context because they can help reduce the overall risk of a portfolio. When assets exhibit low or negative correlations in their volatility patterns, they are less likely to move in the same direction simultaneously. This diversification approach ensures that if one asset experiences high volatility or a price decline, another asset with low volatility correlation may remain stable or appreciate, thereby smoothing out the portfolio's overall volatility and enhancing risk management (Poon & Granger, 2003).

Moreover, the consideration of volatility in portfolio diversification involves a careful evaluation of the risk–return trade-off. Investors seek to strike a balance between risk and return by including assets with varying volatility levels. High-volatility assets may offer the potential for higher returns, but they also come with increased risk. By diversifying across assets with different volatility characteristics, investors can potentially enhance returns while safeguarding against the impact of market volatility on their portfolio's performance. Asset allocation within a diversified portfolio is also influenced by volatility considerations. Investors may opt to allocate a larger proportion of their portfolio to assets with lower volatility to provide stability and reduce overall risk while limiting exposure to assets with higher volatility to manage downside risk effectively. In essence, volatility's role in portfolio diversification is pivotal, enabling investors to construct resilient investment portfolios that align with their risk tolerance and financial objectives (Poon & Granger, 2003).

Volatility serves as a crucial factor in the development of trading strategies for market participants. Traders utilize volatility as a key metric to assess the potential risks and opportunities in the market. Some trading strategies are specifically crafted to capitalize on high volatility levels, aiming to profit from significant price fluctuations. These strategies often involve techniques such as momentum trading or volatility breakout strategies, where traders seek to exploit the rapid price movements that accompany high volatility. On the other hand, there are trading strategies designed to minimize risk during volatile market conditions. These strategies focus on risk management and may involve techniques like hedging or using stop-loss orders to protect against adverse price movements. By incorporating volatility analysis into their trading strategies, traders can adapt to changing market conditions and optimize their trading approach based on the level of volatility present (Oladipupo et al., 2023).

Volatility also plays a critical role in option pricing models, such as the renowned Black-Scholes model. In options trading, volatility is a key input parameter that influences the pricing of options contracts. Higher volatility levels lead to higher option prices because of the increased likelihood of significant price movements within the option's lifespan. This relationship between volatility and option pricing is captured by the volatility component in option pricing models, reflecting the market's expectation of future price volatility. Traders and investors use volatility estimates to assess the fair value of options and make informed decisions regarding their options trading strategies. By understanding the impact of volatility on option prices, market participants can adjust their options positions based on their volatility outlook and risk preferences, enhancing their ability to manage risk and potentially capitalize on market opportunities (Poon & Granger, 2003).

By comprehensively understanding and analysing volatility, investors and traders can make more informed decisions about their investments and trading activities. By considering volatility as a key factor in trading strategies, market participants can tailor their approaches to suit different market conditions, whether aiming to profit from high volatility or mitigate risks during turbulent periods. Moreover, recognizing the influence of volatility on option pricing enables traders to accurately assess the value of options contracts and optimize their options trading strategies based on volatility expectations. Ultimately, incorporating volatility analysis into investment decisions empowers individuals to navigate the complexities of financial markets more effectively, balancing potential returns with associated risks.

2.1.3.2 Risk An inherent aspect of investing, risk refers to the potential of experiencing losses, either partially or entirely, on the original investment because of a variety of factors. These factors can include market fluctuations, changes in economic conditions, or specific events that affect the value of an asset. Understanding and managing risk is essential for investors to safeguard their investments and work towards achieving their financial objectives.

There are several types of risk inherent in investing. Market risk, also known as systematic risk, arises from factors affecting the overall performance of financial markets, such as interest rate fluctuations, exchange rate changes, and asset price volatility. Credit risk involves potential losses from borrower or counterparty failure to meet financial obligations, which is common in lending and bond investments. Liquidity risk pertains to the inability to quickly buy or sell an asset at a fair price, affecting assets with low liquidity through wider bid–ask spreads and price fluctuations. Operational risk stems from internal processes, systems, or human errors within an organization, encompassing inadequate controls, fraud, technology failures, and legal issues. Managing all these types of risk is crucial for investors and organizations to protect investments, assess creditworthiness, ensure market access, and maintain operational stability (Poon & Granger, 2003).

Risk management is a crucial aspect of financial decision-making aimed at safeguarding investments and attaining financial objectives. It encompasses the processes of identifying, evaluating, and mitigating risks to reduce potential losses. Effective risk management strategies involve diversification, hedging, employing stop-loss orders, and establishing risk management protocols. The significance of risk management lies in its ability to shield investments from substantial losses in challenging market conditions or unforeseen circumstances, thereby preserving capital. Moreover, by integrating risk management practices, investors can align their investment strategies with financial goals, striking a balance between risk and return to pursue long-term objectives. Furthermore, risk management enhances decision-making by providing a structured framework for assessing risks and making informed investment decisions, empowering investors to navigate uncertainties and optimize their portfolios (Poon & Granger, 2003).

2.1.3.3 Volatility and Risk Volatility and risk are related concepts in finance, but they are distinct from one another. Volatility measures the dispersion of returns around a mean or average value, indicating the level of price instability or uncertainty over time. High volatility suggests that the price of an asset can change rapidly and by significant amounts, increasing the potential for both gains and losses. On the other hand, risk in finance is a broader concept that encompasses various types of uncertainties and potential negative outcomes that may affect investments or financial decisions. Risk includes not only volatility but also factors such as market risk, credit risk, liquidity risk, and operational risk. It refers to the possibility of experiencing losses or failing to achieve expected returns because of adverse events or circumstances. In essence, volatility specifically focuses on the magnitude of price fluctuations, whereas risk considers a wider range of factors that could affect the performance of investments. While high volatility is often associated with higher risk, risk management strategies aim to address and mitigate different types of risks beyond just volatility to protect investments and achieve financial goals (Poon & Granger, 2003).

2.1.4 Volatility Estimation Techniques

Quantifying volatility is crucial in managing investment risk, and there are multiple techniques and models employed for the estimation of volatility and conditional variance. Conditional variance signifies the variability of asset returns based on available information up to a particular point in time, playing a pivotal role in volatility modelling and VaR prediction. Models like autoregressive conditional heteroskedasticity ARCH; (Bollerslev et al., 1992) and its generalized variant GARCH, which capture conditional volatility dynamics, are instrumental in comprehending the time-varying nature of volatility. The heterogeneous autoregressive (HAR) model, particularly the asymmetric HAR realized volatility (HAR-RV) model, addresses the diverse impacts on volatility across different time horizons, enhancing VaR forecasts by tackling the intricate dynamics of asset price volatility. Volatility forecasting, a crucial practice, involves predicting future volatility levels by leveraging historical data and advanced modelling techniques to enhance the accuracy of volatility forecasts for effective risk management. Additionally, the second moment structure of the conditional return distribution stands out as a significant empirical feature of volatility dynamics, emphasizing the importance of understanding the distributional properties of returns, especially their evolving characteristics, to facilitate precise modelling and forecasting of volatility (Louzis et al., 2014; Zhang, 2003).

There are multiple stylized properties of asset price/commodity volatility that are commonly observed in volatility behaviour. These traits include persistence—defined as sustained volatility levels over time—clustering of high and low volatility periods, the leverage effect—defined as asymmetric responses to price changes—and the presence of distinct volatility regimes. Understanding these properties is crucial for modelling volatility dynamics, improving forecasting accuracy, and implementing effective risk management strategies in financial markets (Patton, 2011). Additionally, a study by Andersen et al. (2003) investigated various stylized properties of asset price and commodity volatility, including volatility clustering, time-varying volatility, long memory effects, volatility persistence, and cross-asset volatility spillovers. Their findings highlight the complex nature of volatility dynamics, emphasizing enduring patterns, interconnections across assets, and the impact of historical volatility on future levels. This knowledge is essential for precise modelling, accurate forecasting, prudent risk mitigation, and informed decision-making in volatile market environments, providing valuable insights for researchers and practitioners in financial analysis and investment strategies.

Various methods exist for estimating volatility using daily data, each offering unique insights into the fluctuation patterns of asset prices. GARCH models are widely employed in volatility estimation (Huang et al., 2016). These models incorporate past volatility and error terms within a time-series framework to model and forecast volatility dynamics effectively. Implied volatility, another significant method, is derived from option prices and reflects the market's anticipation of future volatility levels. Extracted from option pricing models like the Black-Scholes model, implied volatility offers valuable insights into market expectations. Volatility clustering is a phenomenon observed in financial markets where periods of high volatility tend to cluster together, as do periods of low volatility. This clustering effect, evident in historical data, can be utilized to enhance volatility estimation models and risk management strategies. Additionally, the volatility index, exemplified by the VIX (CBOE Volatility Index), serves as a barometer of market sentiment and uncertainty, measuring market expectations of future volatility levels. These diverse methods for estimating volatility using daily data play a crucial role in financial analysis and risk management, offering valuable perspectives on asset price fluctuations and market dynamics (Kambouroudis et al., 2016).

2.1.4.1 Realized Volatility Realized volatility, also known as historical volatility, is another approach to estimating market volatility. This fundamental approach calculates the standard deviation of past asset returns over a specified period, offering a retrospective view of how much an asset's price has historically oscillated. Realized volatility, in continuous-time theory, quantifies the actual volatility experienced by an asset within a specific timeframe, and it leverages observed high-frequency data to provide a realistic depiction of price movements. In financial analysis and risk management, realized volatility serves as a valuable tool for assessing historical volatility patterns, aiding in decision-making processes related to investment strategies, risk assessment, and portfolio management. Its calculation based on observed market data enables a detailed understanding of past price movements and volatility dynamics, contributing to more informed and effective risk management practices (Barndorff-Nielsen & Shepard 2002).

Estimating realized volatility from high-frequency data involves capturing the fluctuations in asset prices throughout the trading day to provide a more accurate representation of volatility compared with traditional methods that rely on daily or lower frequency data (Barndorff-Nielsen & Shepard 2002). Realized volatility is calculated using the following formula:

$$RV = \sqrt{\sum_{t=1}^{n} (r_t - \tilde{r})^2} \ , \tag{7}$$

where RV is the realized volatility, r_t represents the asset's return at time t, $\tilde{r}$ is the average return over the period, and *n* is the total number of observations within the period.

This formula computes the square root of the sum of squared differences between each observed return and the average return over the period. By utilizing high-frequency data, realized volatility provides a precise estimation of the asset's true volatility levels, reflecting the actual price fluctuations experienced by the asset during the specified timeframe.

There are multiple common methods for estimating realized volatility from high-frequency data. One of these is realized variance, which is a simple measure of realized volatility calculated as the sum of squared intra-day returns over a specific period. It provides a direct estimate of the variability in asset prices based on observed returns at high frequencies (Andersen et al., 2001). Another common approach to estimating realized volatility is through the sum of squared intra-day returns. The realized volatility RV_t at time t can be calculated also as:

$$RV_t = \sum_{i=1}^{N} r_{i,t}^2 , \tag{8}$$

where $r_{i,t}$ represents the intra-day return at time t for each observation i within the trading day, and N denotes the total number of observations. By summing the squared returns over the trading day, realized volatility captures the intensity of price movements and fluctuations, providing a valuable metric for assessing and managing financial risk based on the continuous-time dynamics of asset prices (Dacorogna, et al., 2001).

Realized volatility, derived from high-frequency intra-day returns, possesses several important theoretical properties that make it a valuable measure in financial analysis. One of these is consistency in the context of realized volatility, which refers to the fact that the estimator of volatility derived from high-frequency intra-day returns approaches the true, unobservable volatility as the frequency of data increases. In simpler terms, as we collect more and more data points at a higher frequency, the realized volatility estimate becomes more accurate and converges towards the actual volatility of the asset. This property is crucial in financial analysis because it ensures that realized volatility provides a reliable and trustworthy estimate of the underlying volatility of an asset. By converging to the true volatility as more data is considered, the consistency of realized volatility allows analysts and researchers to have confidence in the accuracy of their volatility estimates. This is particularly valuable in risk management, portfolio optimization, and asset pricing, where having an accurate measure of volatility is essential for making informed decisions. Therefore, the consistency of realized volatility as an estimator of actual volatility enhances its utility in financial modelling and analysis, providing a robust tool for understanding and quantifying the level of risk and uncertainty in financial markets (Thomakos & Wang, 2003).

Another characteristic of realized volatility is that it is model-free and does not impose any model structure on the data. This signifies that it does not depend on specific assumptions regarding the distribution of returns or the dynamics of volatility. Unlike traditional parametric models that require predefined structures and assumptions about the data, realized volatility is calculated directly from observed high-frequency intra-day returns. This model-free property of realized volatility is advantageous in situations where the underlying distribution of returns is unknown or complex and where traditional parametric models may not be suitable or accurate. By not relying on a specific model, realized volatility can capture the true characteristics of volatility without being constrained by potentially incorrect assumptions about the data-generating process. Realized volatility can provide a more flexible and robust measure of volatility, especially in volatile and unpredictable financial markets. It allows analysts to assess volatility without making strong assumptions that may not hold true, leading to more reliable estimates of volatility that are not biased by model misspecification. The model-free nature of realized volatility enhances its applicability and usefulness in various financial contexts, offering a versatile tool for measuring and analysing volatility that is not limited by the constraints of specific modelling assumptions (Thomakos & Wang, 2003).

Another key attribute of realized volatility is efficiency, particularly in the context of high-frequency data and market microstructure complexities. By utilizing intra-day data, realized volatility can capture detailed short-term price movements, offering a more precise and accurate reflection of volatility levels. This efficiency is crucial in filtering out noise and irregularities present in financial markets, allowing for cleaner volatility estimates that are not distorted by microstructure effects. The ability of realized volatility to provide more granular insights into market dynamics and to adapt to rapid changes in prices gives it a comparative advantage over traditional methods that rely on lower frequency data. Overall, the efficiency of realized volatility enhances its utility in risk management, option pricing, and other financial applications where precise volatility estimates are essential for decision-making (Thomakos & Wang, 2003).

Further, realized volatility has temporal characteristics that allow it to capture and reflect the dynamics of volatility over time, including features such as persistence and long memory in return volatilities. Persistence in volatility implies that past volatility levels influence future volatility, leading to clusters of high or low volatility periods. Realized volatility's capacity to capture these temporal characteristics enables analysts to study the evolving patterns of volatility over different time horizons. By identifying periods of heightened or subdued volatility, analysts can gain insights into market behaviour, risk dynamics, and potential trading opportunities. This temporal dimension of realized volatility enhances its usefulness in understanding the underlying patterns and trends in volatility, providing valuable information for risk management, forecasting, and decision-making in financial markets (Chen et al., 2020).

Despite being seen as model free, realized volatility has distributional properties, including long memory and approximate Gaussianity, that offer valuable insights into the statistical behaviour of volatility in financial markets. Long memory signifies the persistent influence of past volatility levels on future volatility, revealing patterns of slow decay and predictability in volatility changes. On the other hand, approximate Gaussianity suggests that realized volatility values closely resemble a Gaussian distribution, simplifying statistical analysis and enabling the application of Gaussian-based models. By understanding these distributional properties, researchers can better comprehend the nature of volatility fluctuations, enhance risk assessment techniques, and make informed decisions regarding market dynamics and financial instruments (Thomakos & Wang, 2003).

These theoretical properties of realized volatility contribute to making it a versatile and robust measure for evaluating and studying volatility in financial markets. By capturing the temporal characteristics, such as persistence and long memory, realized volatility provides a comprehensive view of how volatility evolves over time, allowing analysts to identify patterns and trends in asset price movements. Additionally, the distributional properties of realized volatility, including approximate Gaussianity, enable researchers to apply statistical tools effectively and gain a deeper understanding of the statistical behaviour of volatility changes. This versatility and robustness of realized volatility as a measure of volatility offer valuable insights into market dynamics, risk assessment, and decision-making processes in the financial industry, making it a fundamental tool for analysing and managing market volatility.

2.1.5 Value-at-Risk Conceptualization

Value-at-Risk is a widely used measure in risk management that quantifies the potential loss that a portfolio or investment may face over a specified time horizon at a given confidence level. The theory behind VaR is rooted in the concept of quantifying downside risk by estimating the maximum loss that could occur under normal market conditions. The formula for VaR involves calculating the loss at a specific confidence level based on the portfolio's value and the volatility of its underlying assets. One common approach to estimating VaR is through historical simulation, where past returns are used to model potential future losses. Another method is the variance-covariance method, which relies on the mean and standard deviation of returns to calculate VaR. Additionally, Monte Carlo simulation can be employed to generate multiple scenarios and determine potential losses at different confidence levels. The formula for VaR can be expressed as:

$$VaR = Portfolio\ value \times Volatility\ \times\ Z_{\alpha}, \qquad (8)$$

where, VaR is the Value at Risk, $Portfolio\ value$ represents the total value of the portfolio or investment, $Volatility$ denotes the volatility of the portfolio's assets, and Z_{α} is the critical value corresponding to the desired confidence level (Hull, 2012; Jorion, 2006).

This formula provides a quantitative measure of the potential loss that a portfolio may face, allowing investors and institutions to make informed decisions regarding risk management and capital allocation. VaR serves as a crucial tool in assessing and mitigating financial risk, providing a structured framework for evaluating downside risk and enhancing risk management practices in the financial industry (Wipplinger, 2007).

Value at Risk has established itself as a crucial measure in financial risk management. It quantifies and manages financial risk by measuring the potential loss in a portfolio over a defined period within a given confidence interval. VaR forecasts have typically been obtained using time-series models of asset or portfolio volatility, traditionally with daily frequency data. However, recent advances have shown that high-frequency data provides more precise volatility estimates, enhancing VaR calculation accuracy (Ewald et al., 2023).

Traditionally, VaR was calculated using time-series models like GARCH, focusing on daily returns. These models, while foundational, had limitations in capturing the dynamic volatility of financial markets. But with advancements in data processing capabilities, high-frequency (intra-daily) data started being employed. This shift has led to the use of realized volatility in VaR calculations, offering a more accurate representation of market conditions and improving the precision of VaR estimates (Andersen & Bollerslev, 1998a; Baruník & Křehlík, 2018).

Recently, the use of quantile regression models has also been explored. These models use a range of sampling frequencies (from one to 108 minutes) to calculate realized volatility and to forecast VaR. They are particularly effective because they consider the entire distribution of returns, not just the mean or variance. To assess the performance of VaR forecasts, unconditional and conditional coverage tests are employed. These tests compare the proportion of actual returns that exceed the VaR forecast with the expected proportion, providing a rigorous assessment of the model's accuracy (Ewald et al., 2023).

2.1.5.1 Artificial Intelligence/Machine Learning Approaches in VaR Forecasting

With the integration of artificial intelligence (AI) and machine learning techniques, newer models, such as dynamic quantile regression, are being developed for VaR forecasting. These models utilize large datasets and complex algorithms to predict risk levels. They are typically tested using a rolling window approach to validate their forecasting ability. VaR remains an indispensable tool in risk management, and the evolution from traditional time-series models to advanced AI/machine learning techniques using high-frequency data marks a significant progression in the field. These developments not only enhance the accuracy of risk forecasts but also offer a more comprehensive understanding of market dynamics, which are crucial for effective risk mitigation strategies. The integration of AI/machine learning in VaR forecasting represents the future of financial risk analysis, combining extensive data analysis with sophisticated modelling techniques (Gencer & Demiralay, 2016).

Value-at-Risk calculations are typically conducted over a defined time horizon, which can vary from short-term periods, like one day, to longer intervals, such as one week or one month. The chosen time horizon reflects the duration over which potential losses are being evaluated. Additionally, VaR is linked to a confidence level that signifies the probability that actual losses will not surpass the VaR estimate within the specified time frame. Stakeholders can select different confidence levels based on their risk tolerance and investment goals. For instance, a higher confidence level implies a lower probability of exceeding the VaR estimate, indicating a more conservative approach to risk management. Conversely, a lower confidence level allows for a higher probability of exceeding the VaR estimate, accommodating a more aggressive risk strategy. The flexibility in choosing time horizons and confidence levels in VaR calculations enables stakeholders to tailor risk assessments to align with their specific risk preferences and investment objectives, providing a customizable framework for risk management (Hull, 2012).

Value at Risk serves as a crucial risk management tool by assessing potential portfolio losses at a specified confidence level over a defined time horizon. It serves as a valuable tool for decision-making processes, enabling stakeholders to set risk limits, allocate capital prudently, and design risk management strategies tailored to their risk tolerance and investment objectives. By quantifying the maximum potential loss under typical market conditions, VaR aids investors and financial institutions in evaluating and controlling risk effectively. The computation of VaR is based on the current value of the portfolio, offering a comprehensive evaluation of potential losses relative to the overall portfolio size. Additionally, VaR considers the volatility of underlying assets and correlations between different assets, allowing it to capture uncertainties and fluctuations in asset prices. This consideration of risk factors enhances the accuracy of VaR estimates and provides a more nuanced understanding of portfolio risk. Interpreted as a single numerical value representing the estimated maximum loss within a defined time frame and with a specified confidence level, VaR enables investors and institutions to make informed decisions regarding risk management strategies, capital allocation, and portfolio diversification. By utilizing VaR, stakeholders can quantify and monitor the downside risk of their portfolios, establish risk limits, and implement strategies to mitigate risk exposure in volatile market conditions. The conceptual framework of VaR offers a structured approach to evaluating and managing financial risk effectively, contributing to enhanced risk management practices and informed decision-making processes (Ewald et al., 2023; Wipplinger, 2007).

2.1.5.2 Risk Management Theory Value at Risk is a fundamental measure in risk management, providing a quantification of potential portfolio losses at a specified confidence level. Regulatory bodies such as the Basel Committee on Banking Supervision (BCBS) have established requirements for banks to incorporate VaR in their risk management practices, with Basel II introducing the Internal Models Approach for calculating regulatory capital based on VaR models. Additionally, regulations like the Dodd-Frank Act and the European Market Infrastructure Regulation (EMIR) mandate enhanced risk management practices and reporting standards, potentially involving VaR as a risk measurement tool. Firms are obligated to ensure the accuracy and validation of VaR models, report VaR measures to regulatory authorities, adhere to capital adequacy requirements linked to VaR calculations, and set risk limits based on VaR thresholds for effective risk management and monitoring. International standards set by organizations like the International Organization of Securities Commissions (IOSCO) emphasize the importance of robust risk management frameworks, including the use of VaR, to enhance market integrity and investor protection (Ewald et al., 2023).

2.2 Literature Review

This research delves into existing literature on volatility modelling and forecasting, including of the oil market, with notable references including Corsi (2009) and Andersen et al. (2003). Previous studies have commonly utilized realized volatility for forecasting. This study's unique contribution lies in incorporating daily, weekly, and monthly realized volatility derived from different intra-day sampling frequencies. By thoroughly examining all aspects, this study aims to determine the significance of various volatility components (daily, weekly, monthly) and their calculation frequencies in modelling and forecasting realized volatility, both within the sample and beyond it.

2.2.1 High-Frequency Trading in Commodity Markets

The significant contributions that refer to advancements in volatility modelling, specifically the utilization of realized heterogeneous autoregressive models and the incorporation of the fractionally integrated GARCH (FIGARCH) model have enhanced the understanding of conditional time-varying volatility in realized volatility, leading to more accurate and nuanced forecasting techniques in financial markets (Chen et al., 2020). Volatility models based on high frequency data are seen as superior because of their rich information content (Kambouroudis et al., 2016). Various models have been proposed to better forecast oil price volatility using high-frequency data. These include the HAR-RV model and its extensions, which consider factors like jumps and semi-realized measures and which leverage effects in volatility dynamics (Kambouroudis et al., 2016). These modelling techniques offer various approaches to capture different aspects of oil price volatility, allowing for more accurate forecasts and better risk management in the oil market (Chen et al., 2020). Improving forecast accuracy by incorporating time-varying volatility and long-memory features into the models has shown potential in improving forecast accuracy. This is particularly true for models like HAR-RV-FIGARCH, which account for long-memory conditional time-varying volatility of realized volatility (Chen et al., 2020).

The forecasting performance of these models is evaluated using specific methodologies and statistical tests. The emphasis is on the model's ability to improve future performance rather than only analysing past patterns. It has been found that models with time-varying volatility of realized volatility can generate higher forecast accuracies in the oil futures market (Kambouroudis et al., 2016).

2.2.2 The Role of High-Frequency Data in Volatility Forecasting

The integration of HFT and advanced modelling techniques in volatility forecasting signifies a notable progression in how financial instruments are traded and analysed, particularly in the volatile and dynamic realm of commodity markets. These methods provide a more nuanced understanding of market dynamics and have the potential to greatly improve forecasting accuracy, which is crucial for effective risk management and investment strategies. There are also impacts of increased trading volume and liquidity on volatility estimates (Haugom et al., 2014). Further, the structure of characteristics and components of HFT features influence volatility estimates in financial markets.

For example, increased trading volume over the past decades, driven by factors such as HFT and electronic trading, has significantly influenced the accuracy of volatility estimates in the oil market. The availability of higher trading volumes and increased liquidity has enabled the calculation of more precise volatility measures, such as realized volatility (Ederington & Lee, 1993). There are now vast amounts of data accessible because of increased trading volumes and liquidity, which can enhance volatility forecasting. By examining the impact of different sampling frequencies on forecasting accuracy, particularly for volatility and VaR estimations, researchers can determine the most effective approach (Engle, 2002). While previous studies have explored the relationship between sampling frequency and forecasting accuracy, this study takes a novel approach by using various sampling frequencies to calculate realized volatility. The next step involves applying machine learning algorithms to identify the optimal sampling frequencies for forecasting models, based on rigorous evaluation criteria (Andersen et al., 2007; Ewald et al., 2023). By evaluating the performance of different sampling frequencies in forecasting volatility and VaR for Brent Crude Oil futures, this study aims to provide valuable insights for practitioners. The findings will offer recommendations on the preferred sampling frequencies to focus on when developing forecasting models, considering both forecasting accuracy and computational costs (Haugom et al., 2014; Zhang, 2003).

2.2.3 Automatic Feature Selection in Forecasting Models

At the core of this investigation lies the modelling of realized volatility, a fundamental element in deciphering market behaviours. Modelling realized volatility is a crucial factor in understanding market dynamics and is a central focus in this study. Realized volatility provides insights into the level of risk and uncertainty in the market. By modelling realized volatility, researchers aim to gain a deeper understanding of how market behaviours unfold and evolve. The study's holistic methodology incorporates a wide array of predictive models, such as ordinary least squares (OLS), ridge regression, least absolute shrinkage, and selection operator (lasso), and random forest regressors, to analyse and forecast the nuanced dynamics of volatility (Ewald et al., 2023). These models, distinguished by their diverse levels of intricacy and regularization methods, provide an opportunity to navigate the intricacies of financial data and elevate forecasting proficiency:

1. Ordinary Least Squares (OLS): OLS is a classical linear regression method that aims to minimize the sum of squared differences between the observed and predicted values. It provides a straightforward way to estimate the relationships between variables in a dataset (Weeks, 2002).

2. Ridge Regression: Ridge regression is a regularization technique that adds a penalty term to the OLS method to prevent overfitting. By introducing a regularization parameter, ridge regression helps to stabilize the model and reduce the impact of multicollinearity (Hoerl & Kennard, 1970).

3. Least Absolute Shrinkage and Selection Operator (lasso): Lasso is another regularization method that not only helps in reducing overfitting but also performs feature selection by shrinking the coefficients of less important variables to zero. This feature selection property can be valuable in enhancing model interpretability (Tibshirani, 1996).

4. Random Forest Regressors: Random Forest is an ensemble learning technique that builds multiple decision trees and combines their predictions to improve accuracy and robustness. It is known for effectively handling non-linear relationships and interactions in the data (Fan et al., 2009).

By leveraging this diverse set of predictive models, the research aims to analyse and forecast the nuanced dynamics of volatility in financial markets. These models offer varying levels of complexity and regularization methods, allowing researchers to navigate the complexities inherent in financial data. The use of such a range of provides enhanced forecasting proficiency toward gaining deeper insights into the behaviour of volatility in the market.

In the realm of financial forecasting models, the utilization of convolutional neural networks (CNNs) for price prediction in financial markets involves the integration of automatic feature selection techniques. CNNs are renowned for their capacity to autonomously learn and extract pertinent features from input data, rendering them well-suited for capturing complex dependencies in financial time-series data. Researchers have explored the development of mathematical models based on CNNs that leverage historical data to identify crucial features and relationships, facilitating the prediction of future price movements (Medvedev & Medvedev, 2023).

When applying neural network-based financial forecasting techniques, particularly CNNs, to predict stock market trends, the process entails automatic feature selection to discern key patterns and signals in the data. By categorizing historical financial market data into distinct market states (such as trend, sideways, unknown), the models automatically select relevant features indicative of specific market conditions. Subtasks within the model development process, including training and validation, rely on automatic feature selection to extract meaningful information for predicting market changes and trends (Medvedev & Medvedev, 2023).

Challenges and limitations in the model-building process, such as selecting hyperparameters for CNN models, can be addressed through automatic feature selection methods. By leveraging libraries like CNTK, TensorFlow, and Caffe, researchers can automate the selection of hyperparameters based on the data characteristics and software platforms. The training process involves defining hyperparameters related to loss functions, optimization algorithms, and learning rates, with validation metrics tailored to the specific forecasting problem. Automatic feature selection plays a crucial role in identifying the most relevant input variables for accurate predictions (Jarboui & Mnif, 2023).

Moreover, the analysis of trading simulations to evaluate model predictions relies on automatic feature selection to extract informative signals from many stocks. While CNNs demonstrate potential in financial forecasting, the need for further improvements underscores the importance of refining automatic feature selection techniques to enhance model performance. Despite the computational challenges associated with training neural networks, the universal applicability of these models across different financial instruments and markets highlights the significance of automated feature selection in optimizing forecasting accuracy and efficiency (Monfared & Enke, 2014).

Thus, the integration of automatic feature selection methods within neural network-based forecasting models, particularly CNNs, offers a promising avenue for enhancing the accuracy and reliability of financial market predictions. By automating the process of identifying relevant features from complex financial data, researchers can improve model performance and scalability, paving the way for more effective decision-making in financial markets.

2.2.4 Comparisons of Forecasting Methodologies

Traditional models and more recently developed methods for forecasting realized volatility present relative strengths in predicting different aspects of market variability. Recent studies have compared and contrasted the more classical models with newer innovative methods:

1. Traditional Models: The heterogeneous autoregressive (HAR) and autoregressive fractionally integrated moving average (ARFIMA) models have been widely used to forecast realized volatility. These models are effective in capturing the long memory of volatility and are flexible when high-frequency data are available (Corsi, 2009).

2. Artificial Neural Networks (ANNs): ANNs are seen as a generalization of these classical approaches, suitable for modelling the non-linear processes in volatility. They are semi-parametric, non-linear models that can approximate any reasonable function and do not require strict distributional assumptions. ANNs use hidden layers to transform input variables and, thus, can describe complex patterns in volatility time series (Zhang, 2003).

3. Learning Process in Neural Networks: The training of neural networks involves adjusting weights using a learning algorithm to minimize prediction errors. This process is an

unconstrained nonlinear optimization problem aimed at finding the optimal set of weights for the parameters (Rumelhart et al., 1986).

4. Testing and Validation: The ANN models are tested against traditional models like HAR, ARFIMA, and GARCH within frameworks such as the model confidence set and superior predictive ability. These tests aim to assess the relative performance of ANNs in forecasting volatility, especially in the context of energy markets (Hansen & Lunde, 2006).

While traditional models like HAR and ARFIMA have been effective in capturing long memory in volatility, the integration of high-frequency data with ANNs offers a promising avenue for improved forecasting. ANNs' ability to model complex, non-linear patterns in the data presents a significant advancement over traditional methods, showing potential for both statistical and economic gains in forecasting realized volatility (Baruník & Křehlík, 2016).

2.2.5 Critical Gaps in Current Literature

The exploration of critical gaps in the current literature on volatility modelling and forecasting, particularly in the context of HFT in commodity markets like crude oil, reveals several key areas where existing research could be expanded or deepened, including the integration of market microstructure variables, the impact of regulatory changes, adaptation to market anomalies, long-term dependency and non-linear patterns, comparative studies across different commodities, real-time data use and forecasting reliability, the impacts of AI and machine learning on market volatility forecasting, and ethical considerations in forecasting. This section aims to articulate these gaps, emphasizing where future research could potentially make significant contributions.

First, despite the advancements in volatility forecasting associated with the use of high-frequency data, there remains a substantial gap in the integration of market microstructure variables. Current models often overlook factors such as bid-ask spreads, order book depth, and intra-day trading volume, which could provide additional insights into the predictive accuracy of volatility models.

Next, the literature frequently neglects the impact of regulatory changes on market dynamics and volatility. As financial markets evolve with new regulations intended to increase market transparency and reduce systemic risks, the effect of such changes on model efficacy remains underexplored. Further, traditional, and even some advanced models, struggle to effectively adapt to market anomalies and structural breaks. These models often fail to account for sudden changes in volatility due to geopolitical events, economic announcements, or market crises, leading to significant forecasting errors.

There is also a need for further research into the long-term dependency and non-linear patterns of volatility that are not adequately captured by models like GARCH or HAR. Advanced machine learning techniques, which can model these complexities more effectively, are still not fully explored within the empirical literature, especially in their ability to integrate with traditional econometric approaches. In addition, most studies tend to focus on a single commodity market—typically crude oil, or financial indices. There is a notable lack of comparative studies that explore the efficacy of forecasting models across different commodity markets, such as natural gas, gold, or agricultural products, which may exhibit different volatility dynamics.

Next, the use of real-time data for forecasting and the reliability of these forecasts in real-world trading scenarios is a critical gap. Most academic studies simulate forecasting using historical data, which does not always translate into effective real-time trading strategies because of latency, transaction costs, and model overfitting. In addition, although AI and machine learning are increasingly applied in this field, there is a significant gap in understanding the full spectrum of implications these technologies bring, including issues of interpretability, model transparency, and the balance between model complexity and interpretability.

Finally, the ethical implications of forecasting, particularly concerning market manipulation or the potential impact on commodity-dependent economies, are rarely discussed. There is a profound need for a framework that addresses the ethical use of advanced forecasting techniques, especially in highly speculative markets like crude oil.

Addressing these gaps will not only enhance the theoretical framework of volatility forecasting but also improve the practical applications of these models in real-world settings. Future research should aim to develop more comprehensive models that incorporate these elements, test them across different market conditions and commodities, and rigorously evaluate their real-time applicability and ethical implications.

Various methods exist for estimating realized volatility from high-frequency data, each offering unique advantages in capturing the dynamics of asset price movements. The realized range approach calculates volatility by considering the difference between high and low prices within a trading period, providing insights into price fluctuations (Zhang, 2003). On the other hand, the realized kernel method, a more sophisticated technique, incorporates the irregular spacing of high-frequency data by using a kernel function to weigh returns based on their temporal proximity, resulting in a smoother volatility estimate (Barndorff-Nielsen & Shephard, 2002). Additionally, the realized bi-power variation estimator, robust against market noise and price jumps, computes the sum of squared price increments adjusted for jumps to accurately estimate volatility (Barndorff-Nielsen & Shephard, 2004). By leveraging these advanced methods and others, analysts can derive precise and timely estimates of realized volatility from high-frequency data, essential for effective risk management, derivative pricing, and portfolio optimization through a more nuanced understanding of asset price fluctuations and volatility dynamics (Barndorff-Nielsen & Shephard, 2002).

3 Methodology

3.1 Data Source

The data source for this thesis is the ICE Brent Crude Oil Futures Data, which serves as the primary dataset for analysing and forecasting market dynamics related to Brent Crude Oil. This dataset contains detailed intra-day trading information, including price movements and trading volumes, which are essential for understanding the behaviour of the Brent Crude Oil market. The dataset used in the study covers the period from January 3, 2006, to January 29, 2016, encompassing a total of 2567 trading days. It includes high-frequency transaction-level every minute’s data for the front-month Brent Crude Oil futures contracts traded at the Intercontinental Exchange (i.e., ICE).

Moreover, the front-month Brent Crude Oil futures contract operates under specific regulations regarding daily margin requirements and position limits. All open contracts are marked-to-market daily, ensuring that the value of positions is adjusted based on current market prices. Additionally, the Exchange's daily position management regime mandates that all positions in any contract month must be reported to the exchange daily. This regime aims to prevent the development of excessive positions, unwarranted speculation, or any other undesirable situations. The Exchange has the authority to take necessary steps to address such situations, including mandating members to limit the size of positions or reduce positions when deemed appropriate. This comprehensive information provides a deeper understanding of the operational framework and risk management practices associated with trading the front-month Brent Crude Oil futures contracts on the Intercontinental Exchange. The ICE Brent Crude Oil Futures Data serves as a valuable resource for analysing and forecasting market dynamics related to Brent Crude Oil. This dataset contains detailed intra-day trading information, offering insights into the price movements of Brent Crude Oil futures contracts throughout the trading day. These price movements reflect the complex interplay of market demand, supply dynamics, geopolitical events, and other factors influencing oil prices.

When analysing the ICE Brent Crude Oil Futures Data, it is essential to consider the coverage of trading hours. Comprehensive coverage of trading hours ensures that the analysis is based on a complete set of data, minimizing the risk of overlooking important market trends. Furthermore, understanding non-trading hours is crucial for gaining insights into how prices behave during periods when the market is closed, such as evenings, weekends, and holidays. Examining price movements during non-trading hours can reveal overnight or weekend price gaps and potential market reactions to external events. The rationale for utilizing a 22-hour trading window in analysing Brent Crude Oil futures data lies in the need to capture both regular trading hours and overnight movements. By extending the analysis to cover a 22-hour window, the study can assess price changes and trading activities that occur outside traditional trading hours.

This extended timeframe enables a more comprehensive understanding of market behaviours and facilitates the identification of patterns that may emerge during non-standard trading periods. To gain a deeper understanding of market behaviours and uncover emerging patterns during non-standard trading periods, the thesis employs various strategies. One approach involves utilizing advanced pattern recognition techniques, such as machine learning algorithms, to identify recurring patterns in price movements that may not be readily apparent through traditional analysis methods. Comparative analysis between regular trading hours and non-standard trading periods can reveal unique market dynamics and trends, providing insights into how market behaviours differ across different timeframes (Hasbrouck, 2007; Lo & MacKinlay, 1999). Developing sophisticated volatility models that account for fluctuations in market volatility during non-standard trading periods can also enhance understanding and prediction of price movements. Furthermore, delving into market microstructure analysis enables the study of order flow, liquidity provision, and price discovery mechanisms during non-standard hours, offering valuable insights into market behaviours during these specific trading periods. By integrating these strategies into the analysis, the thesis deepens the comprehension of price dynamics and enhances the accuracy of forecasting models for Brent Crude Oil futures.

In summary, leveraging the ICE Brent Crude Oil Futures Data, with a focus on intra-day trading information, including price movements, provides a robust foundation for analysing and forecasting market trends. Considering the coverage of trading hours and the rationale for a 22-hour trading window ensures that the analysis is thorough, capturing both regular and non-traditional market behaviours. This approach enhances research insights and forecasting accuracy, contributing to a deeper understanding of the Brent Crude Oil market dynamics. Python code for data analysis can be found in Appendix 2.

3.2 Variable and Data Processing

In the examination of ICE Brent Crude Oil futures data, the meticulous selection of pivotal market variables is imperative to comprehend the intricacies of market dynamics and construct precise forecasting models. These selected variables must encapsulate fundamental aspects of the market that exert influence on price movements and volatility. Among the notable market variables derived from ICE Brent Crude Oil futures data are price; volatility; open, high, low, close (OHLC) data; returns; and intra-day price movements. Price serves as a barometer of market sentiment and guides trading decisions, whereas volatility quantifies the magnitude of price movement variations over a specified period. OHLC data furnishes granular insights into price levels, and returns elucidate the percentage changes in price, facilitating an evaluation of market performance. Delving into intra-day price movements, encompassing phenomena like spikes and reversals, yields valuable intelligence for formulating astute short-term trading strategies and fortifying risk management practices within the market landscape.

Data processing for ICE Brent Crude Oil futures data entails a series of pivotal procedures aimed at enhancing data integrity and analytical robustness. First, data were meticulously scrutinized to identify missing data and the most appropriate strategies for handling these gaps were determined, whether through imputation techniques or exclusion. Concurrently, outliers were detected and managed to mitigate potential distortions in analytical outcomes or model performance. Subsequently data were transformed to ensure compatibility for analysis, particularly focusing on the conversion of temporal variables into datetime objects and the normalization or standardization of numerical variables to mitigate biases in subsequent analyses.

Next, the handling of non-trading hours was thoughtfully considered. To adjust for data points occurring outside conventional trading periods, such as overnight or weekend data, and to account for market dynamics during atypical trading intervals, feature engineering was used. Feature engineering involves the creation of novel variables derived from existing data to capture nuanced relationships within the dataset, alongside the incorporation of lagged variables to address temporal dependencies and autocorrelation patterns (Guyon & Elisseeff, 2003; Kuhn & Johnson, 2013). Data were then aggregated by summarizing data across varying time intervals, such as hourly or daily frequencies, to effectively discern trends and patterns within the dataset. Last, summary statistics were computed of serves as a fundamental tool to unveil insights into the distributional characteristics and intrinsic attributes of the data, thereby laying a solid foundation for comprehensive analysis and modelling within the realm of financial markets.

These structured data processing procedures were completed using the Python programming language to ensure that the ICE Brent Crude Oil futures dataset was thoroughly cleaned, transformed, and prepared for analysis. This rigorous data processing methodology, was foundational for developing precise models, conducting accurate forecasts, and making informed decisions based on the pertinent market variables under scrutiny.

3.3 Model Specifications

3.3.1 Statistical Models

This study employed and compared several statistical models to forecast realized volatility in ICE Brent Crude Oil futures. These methods are described here.

GARCH: A generalized extension of the autoregressive conditional heteroscedasticity (ARCH) model was proposed by Andersen and Bollerslev (1998a). This GARCH model forecasts volatility by modelling the conditional variance of the returns as a weighted sum of past squared residuals through a conditional mean function and past variance, making it a much more robust alternative to ARCH. The formula for the GARCH model is as follows:

$$\epsilon_t = \sigma_t z_t \sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \qquad (9)$$

This formula specifies the relationship between the error term ϵ, conditional variance σ_t^2, past squared residuals ϵ_{t-1}^2, and past variance σ_{t-1}^2. The model incorporates parameters ω , α_1, and $\beta_1$to estimate the conditional variance based on the historical information of the series (Andersen & Bollerslev, 1998b).

GARCH captures the time-varying volatility in asset returns, particularly through modelling volatility clustering, which is the tendency of high volatility periods to cluster together in financial time-series data. GARCH models include an autoregressive component to capture the persistence of volatility shocks and a moving average component to account for the impact of past volatility on current volatility. By adapting to changing market conditions, GARCH models provide accurate forecasts of volatility, making them essential tools in risk management and financial analysis (Engle, 2001).

ARIMA: Autoregressive integrated moving average (ARIMA) models are a popular choice for analysing and forecasting time-series data, particularly in the realm of finance. These models are adept at capturing linear dependencies within the data by incorporating autoregressive and moving average components, along with differencing to handle non-stationarity (Box et al., 2015). The ARIMA equation is as follows:

$$\begin{gathered} Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} \dots + \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \\ + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \end{gathered} \tag{10}$$

where Y_t is the time-series data at time t, c is a constant, ϕ and θ are the autoregressive and moving average parameters, ε_t is the error term, and p, d, and q are the orders of the AR autoregressive (AR), integrated (I), and moving average (MA) components of the ARIMA model, respectively.

HAR-RV: The heterogeneous autoregressive model of realized volatility (HAR-RV) is a specialized model tailored for forecasting realized volatility in high-frequency financial datasets. Unlike traditional volatility models, HAR-RV accounts for the heterogeneous nature of volatility by incorporating lagged realized volatility measures at different frequencies. This approach allows the model to capture both long memory effects and short-term dynamics of volatility, providing more accurate forecasts of future volatility levels (Baruník & Křehlík, 2018; Corsi, 2009).

$$RV_{t:t+h-1} = \beta_0 + \beta_\omega RV_{t-5:t-1} + \beta_m RV_{t-90:t-1} + \varepsilon_t, \tag{11}$$

where $RV_{t:t+h-1}$ is the realized volatility over the forecast horizon, β_0 coefficients represent the weights assigned to different lagged realized volatility measures, and ε_t is the error term.

In summary, ARIMA models excel at capturing linear trends and patterns in financial time-series data, while HAR-RV models are specifically designed to address the challenges of forecasting realized volatility in HFT settings. By combining the strengths of these models, researchers and analysts can enhance their forecasting accuracy and gain deeper insights into the intricate dynamics of volatility in financial markets. GARCH models, ARIMA models, and HAR-RV models play essential roles in modelling and forecasting financial time-series data, each offering unique capabilities for capturing volatility dynamics and linear dependencies. By understanding the strengths and applications of these statistical models, researchers can enhance their forecasting accuracy and gain valuable insights into market behaviour and risk management in HFT environments.

3.3.2 Machine Learning Models

3.3.2.1 The Multi-Layer Perceptron model The multi-layer perceptron (MLP) model, particularly the MLPRegressor, is a widely used neural network architecture for regression tasks. The model's formula can be broken down into several key components to understand how it generates predictions. First, the input layer of the MLP receives the input features and passes them to neurons in the first hidden layer. Each neuron in a hidden layer computes a weighted sum of the inputs from the previous layer, incorporating a bias term, which is then passed through an activation function to introduce non-linearity. The activation function output of a neuron is determined by applying the function to the weighted sum of inputs. Moving to the output layer, the final prediction is produced by calculating the weighted sum of activations from the last hidden layer, considering the weights connecting hidden neurons to the output neuron and a bias term. During training, the model evaluates a loss function, such as mean squared error, to quantify the disparity between predicted and actual values, adjusting weights and biases through optimization techniques like backpropagation to minimize this loss and enhance prediction accuracy (Hinton et al., 2012).

The formula for a simple feedforward MLP model, such as the MLPRegressor, can likewise be broken down into several components to understand how the model makes predictions. The input layer of the MLP receives the feature values (input data) and passes them to the neurons in the first hidden layer. Each neuron in a hidden layer computes a weighted sum of the inputs from the previous layer, applies an activation function, and passes the result to the neurons in the next layer. The weighted sum z for a neuron j in a hidden layer is calculated as:

$$z_j = \sum\nolimits_{i=1}^{n} (w_{ij} * x_i) + b_j, \tag{12}$$

where w is the weight connecting input neuron i to hidden neuron j, x_i is the input value from neuron i, and b_j is the bias term for neuron j.

Then the activation function is applied to the weighted sum to introduce non-linearity and determine the output of the neuron. The output (a) of neuron j after applying the activation function is calculated as follows:

$$a_j = f(\, z_j), \tag{13}$$

where $f(\,)$ is the activation function.

Further, the output layer receives the activations from the last hidden layer and produces the final prediction. For regression tasks, the output layer typically consists of a single neuron that outputs the predicted continuous value. The predicted output (*y_pred*) is calculated as the weighted sum of the activations in the last hidden layer:

$$y_{pred} = \sum\nolimits_{i=1}^{n} (w_{kj} * a_j) + b_k, \tag{14}$$

where w_{kj} is the weight connecting hidden neuron j to the output neuron k, a_j is the activation value from hidden neuron j, and b_k is the bias term for the output neuron.

The model's utilization of activation functions, weighted sums, biases, and loss functions underscores its capacity to capture complex patterns and relationships within the data, making it a versatile and valuable tool for regression modelling across diverse domains.

3.3.2.2 Random Feature Selection During random feature selection at each split in a decision tree, a random subset of features is considered for determining the best split. This random feature selection helps in decorrelating the trees and improving the overall model's performance. Random Forest, an ensemble decision tree method, employs a technique called bagging (i.e., bootstrap aggregating), where each tree is trained on a bootstrapped sample of the training data. This bootstrapping process involves sampling the training data with replacement, ensuring that each tree sees a slightly different subset of the data.

Random forest regression is a powerful ensemble learning technique used for regression tasks. This method combines the strength of multiple decision trees to create a robust and accurate regression model. By averaging the predictions of individual trees and introducing randomness in feature selection and data sampling, random forest mitigates overfitting and enhances generalization performance. The formula for random forest regression encapsulates the ensemble nature of the model, where predictions are aggregated from multiple trees by averaging the predictions of all trees in the forest to provide a reliable and stable regression outcome (Breiman, 2001). Thus, the prediction $\hat{y}$ made by a random forest regression model is the average of predictions from all individual trees in the forest:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} y_i , \tag{15}$$

where $\hat{y}$ is the final predicted value, N is the total number of trees in the forest, and y_i is the prediction made by the i^{th} tree.

Each decision tree in the random forest is constructed by recursively partitioning the feature space into regions based on feature values. The prediction at each leaf node of the tree is the average (for regression) of the target values of the training samples that fall into that region.

3.3.2.3 Support Vector Regression Support vector regression (SVR) is a machine learning algorithm used for regression tasks that extends the principles of Support Vector Machines (SVM) to predict continuous outcomes. This algorithm aims to find the optimal hyperplane that best fits the data points while minimizing margin violations. The algorithm is particularly effective in capturing non-linear relationships in the data by mapping the input features into a higher-dimensional space (Drucker et al., 1997). In the case of linear SVR, the prediction $\hat{y}$ for a new input sample x is calculated as follows:

$$\hat{y} = w^T x + b, \tag{16}$$

where $\hat{y}$ is the predicted output, w is the weight vector, x is the input feature vector, and b is the bias term.

Support vector regression aims to minimize the loss function, which penalizes errors based on the margin and the presence of any violations. The loss function typically includes a regularization term to control the complexity of the model and prevent overfitting. SVR optimization involves finding the optimal hyperplane that maximizes the margin around the data points while ensuring that the errors (deviations from the hyperplane) are within a specified tolerance level. Support vector regression is a versatile algorithm for regression tasks that can handle both linear and non-linear relationships in the data. By mapping data into a higher-dimensional space using kernel functions, SVR can capture complex patterns and make accurate predictions (Smola & Schölkopf, 2004). The formula for SVR encompasses the linear and non-linear prediction mechanisms, along with the optimization process to find the best-fitting hyperplane. Support vector regression's ability to control model complexity and generalize well to unseen data makes it a valuable tool in regression modelling across various domains.

3.4 Evaluation Metrics

3.4.1 Grid Search with Cross-Validation

Grid search with cross-validation (grid search CV) is a technique used in machine learning to find the optimal hyperparameters for a model by exhaustively searching through a specified parameter grid and evaluating the model's performance using cross-validation. Its process does not involve specific mathematical formulas but rather a systematic approach to finding the optimal hyperparameters for a machine learning model (Bergstra & Bengio, 2012).

The parameter grid is a dictionary containing the desired hyperparameters for tuning a neural network model. These hyperparameters can include the number of hidden layers, activation functions, loss functions, maximum iterations, learning rate, etc. Each parameter is assigned a list of possible values to be tested.

The grid search process involves evaluating the model's performance for each combination of hyperparameters in the parameter grid. The formula for grid search is the following:

$$Grid\ Search = \arg max_{parametrs}\, Performance\ Metrics \qquad (17)$$

Cross-validation is a technique used to assess how well a model generalizes to an independent dataset. The formula for cross validation is the following:

$$Cross\ Validation = \frac{1}{k}\sum_{i=1}^{k} Performance\ Metrics_i\,, \qquad (18)$$

where k is the number of folds in the cross-validation process and $Performance\ Metrics_i$ is the performance metric (e.g., accuracy, loss) for the i^{th} fold. These formulas provide a high-level overview of the grid search CV processes in machine learning. The actual implementation involves running the model with different hyperparameters, evaluating its performance using cross validation, and selecting the best set of hyperparameters based on the results.

In summary, grid search CV is a powerful tool for hyperparameter tuning in machine learning models, allowing for systematically searching through a predefined parameter grid to find the optimal set of hyperparameters for a neural network model.

3.4.2 Neural Network

In the context of machine learning, creating a neural network model is a crucial step before conducting hyperparameter optimization using techniques like grid search CV. Therefore, neural network model creation is the foundation for hyperparameter tuning. First, the neural network architecture that will be used for the task at hand is designed and built. This process typically involves selecting the number of layers, the number of neurons in each layer, the activation functions, the optimizer, the loss function, and other architectural choices that define how the neural network will learn from the data (Goodfellow et al., 2016).

By creating the neural network model up front, the baseline structure and parameters that will be optimized during the hyperparameter tuning process are established. The initial neural network model acts as the starting point from which different hyperparameter combinations will be explored and evaluated to enhance the model's performance. Libraries like TensorFlow, Keras, and Scikit-learn provide a convenient way to build neural network models with various complexities and configurations. These libraries offer pre-built neural network layers, activation functions, optimizers, and loss functions, making it easier to define and train neural networks for different machine learning tasks. Therefore, before delving into hyperparameter optimization techniques such as grid search CV, it is essential to have a well-defined neural network model in place. This model serves as the base structure that will be refined and improved through the iterative process of hyperparameter tuning to achieve the best possible performance on the given dataset and task (Goodfellow et al., 2016).

3.4.3 Loss Function

In the context of a neural network model, the loss function, also known as the cost function or objective function, is a crucial component that quantifies how well the model is performing during training. The loss function calculates the difference between the predicted output of the neural network and the actual target output for a given set of input data. The goal of training a neural network is to minimize this loss function, which indicates how far off the predictions are from the ground truth (Bishop, 2006).

Mean squared error (MSE) is a loss function that quantifies the average of the squared differences between the predicted values and the actual values. Mathematically, MSE is calculated as the mean of the squared residuals between the predicted values ($\hat{y}_i$) and the true values (y_i) as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{19}$$

During model training, the neural network adjusts its parameters to minimize MSE, aiming to reduce the overall squared errors between predictions and actual values. A lower MSE indicates that the model's predictions are closer to the true values, reflecting improved performance in terms of minimizing squared errors.

Mean absolute error (MAE) is a loss function that calculates the average of the absolute differences between the predicted values and the actual values. The MAE is computed as the mean of the absolute residuals between the predicted values ($\hat{y}_i$) and the true values (y_i) as follows:

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n} |\hat{y}_i - y_i| \qquad (20)$$

During training, the neural network aims to minimize the MAE by adjusting its parameters to reduce the average absolute errors between predictions and actual values. A lower MAE signifies that the model's predictions have smaller absolute deviations from the true values, indicating improved accuracy in terms of absolute errors.

R-squared (R^2) is a metric that measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It is not a loss function, but an evaluation metric used to assess the goodness of fit of the model. The R^2 value is calculated as the ratio of the explained variance to the total variance:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \qquad (21)$$

A higher R^2 value indicates that the model explains a larger proportion of the variance in the target variable, suggesting a better fit of the model to the data and a stronger relationship between the independent and dependent variables (Montgomery et al., 2012).

3.4.4 Back-Testing Value-at-Risk and Expected Shortfall Models

Back-testing VaR and expected shortfall (ES) models are crucial for risk management in financial institutions and. This process involves assessing the accuracy and reliability of these models by comparing the predicted risk measures with actual outcomes over a historical period. By evaluating the effectiveness of VaR and ES models in capturing and quantifying market risk, insights into the model's performance and potential areas for enhancement can be gained. The methodology for back-testing VaR models entails several key steps. First, historical data selection involves the choice of an appropriate dataset encompassing relevant time periods for analysis, including historical returns and price movements necessary for VaR calculations. Subsequently, VaR is calculated using the selected historical data and the chosen methodology, such as historical simulation, parametric, or Monte Carlo simulation, to determine the maximum potential loss a portfolio may incur at a specified confidence level over a given time horizon (Jorion, 2006).

Following VaR calculation, the comparison of predicted VaR values with actual portfolio losses observed during the historical period is conducted. This comparison assesses whether actual losses exceed VaR estimates and quantifies the frequency and severity of exceptions, known as VaR breaches. Statistical tests like the Kupiec test, Christoffersen test, or conditional coverage test are then used to evaluate the accuracy and reliability of the VaR model, ensuring consistency between VaR estimates and observed losses while identifying any model performance deficiencies. Based on the back-testing results, model validation and adjustment are performed to enhance accuracy and robustness. This may involve recalibrating model parameters, updating risk factors, or integrating additional data sources for improved risk estimation (Hull, 2012). The formula for VaR is typically expressed as the following:

$$VaR_{\alpha} = -ES_{\alpha} = -\inf\{x \in \mathbb{R} : F(x) \geq \alpha\}, \qquad (22)$$

where VaR_{α} represents the VaR at a specified confidence level α, $-ES_{\alpha}$ denotes the expected shortfall at the confidence level α, $F(x)$ is the cumulative distribution function of the portfolio returns, and α is the confidence level, typically ranging from 90% to 99%.

This formula calculates the potential loss that a portfolio may incur over a specified time horizon with a given level of confidence. VaR is a widely used risk measure in financial institutions for quantifying and managing market risk. The significance of back-testing for risk management lies in model validation, risk mitigation, regulatory compliance, and continuous improvement. By conducting thorough back-testing and implementing corrective actions based on the results, financial institutions can fortify their risk management frameworks and make well-informed decisions in dynamic market environments.

3.4.5 Model Calibration and Selection

The calibration process is rigorous, balancing the complexity of models with their interpretability and predictive prowess. Through a systematic evaluation of model parameters and features, this study refines methodologies, ensuring they are attuned to the nuances of the data. The selection criterion hinges not only on statistical metrics such as Akaike's information criterion (AIC) and Bayesian information criterion (BIC) but also on the models' ability to resonate with the underlying market dynamics they seek to capture.

The calibration process involved striking a delicate balance between model complexity, interpretability, and predictive accuracy. By systematically evaluating model parameters and features, this study fine-tunes these methodologies to align with the intricacies present in the dataset, ensuring their effectiveness in capturing the underlying market dynamics.

The AIC and BIC are commonly used statistical metrics to assess the goodness of fit of models (Burnham & Anderson, 2004). Their formulas are as follows:

$$AIC = -2 * \log(L) + 2k, \tag{23}$$

where L represents the maximum value of the likelihood function of the model and k denotes the number of parameters in the model.

$$BIC = -2 * \log(L) + k * \log(n), \tag{24}$$

where n signifies the sample size.

In this study, the model selection process not only relied on these statistical metrics but also considered the ability of the models to resonate with the underlying market dynamics they aim to capture. This holistic approach ensures that the chosen models are not only statistically sound but also reflective of the true complexities of the financial markets, enhancing the robustness of their forecasting methodologies. The in-sample analysis acts as a critical evaluation tool for assessing the effectiveness of models by scrutinizing their performance against known data. This introspective examination provides valuable insights into the strengths and limitations of each model, shedding light on their predictive accuracy and areas that may require improvement.

4 Results

The comprehensive analysis of financial volatility and predictive modelling within the realm of high-frequency data analysis and advanced statistical models offers a detailed exploration that integrates and expands upon the preliminary findings, methodology, and results. This synthesized analysis aims to illuminate the intricate dynamics of financial markets, providing a reflective, critical, and evidence-based examination of the empirical evidence derived from this study.

The initial findings of the study underscore the importance of utilizing high-frequency data and advanced statistical modelling techniques to enhance our understanding of financial volatility and market risk. By delving into the nuances of market dynamics at a granular level, researchers can uncover hidden patterns, identify key drivers of volatility, and develop more accurate forecasting models that capture the complexities of real-world market behaviour. The methodology employed in this study, including the use of quantile regression, feature selection algorithms, and comparative analysis of forecasting models, reflects a rigorous and systematic approach to modelling financial volatility. By leveraging these advanced techniques, this thesis can extract meaningful insights from the data, refine their risk assessment frameworks, and provide analysis needed to make informed decisions in the face of market uncertainties.

4.1 Descriptive Statistics

To provide descriptive statistics for the volatility variables utilized in the models, a systematic approach was followed. Initially, the ICE Brent Crude Oil dataset was filtered to isolate the pertinent volatility-related variables by selecting specific columns associated with volatility in trading data. This filtering process involved identifying variables that adhered to predefined patterns such as 'rvol_pt', 'd[0-9]+', 'w[0-9]+', and 'm[0-9]+', which represent various types of volatility measures, like realized volatility, daily returns, weekly averages, and monthly averages, respectively. Utilizing techniques like regular expressions (regex) facilitated the extraction of columns matching these patterns, thereby focusing solely on volatility-related variables for subsequent analysis and statistical summarization. Histogram plots were visually assessed to determine the distribution of realized volatility (i.e., rvol) over different periods (i.e., segments of time; pt1 = price over one minute, pt2 = price over two minutes, and so on) (Figures 1–3).

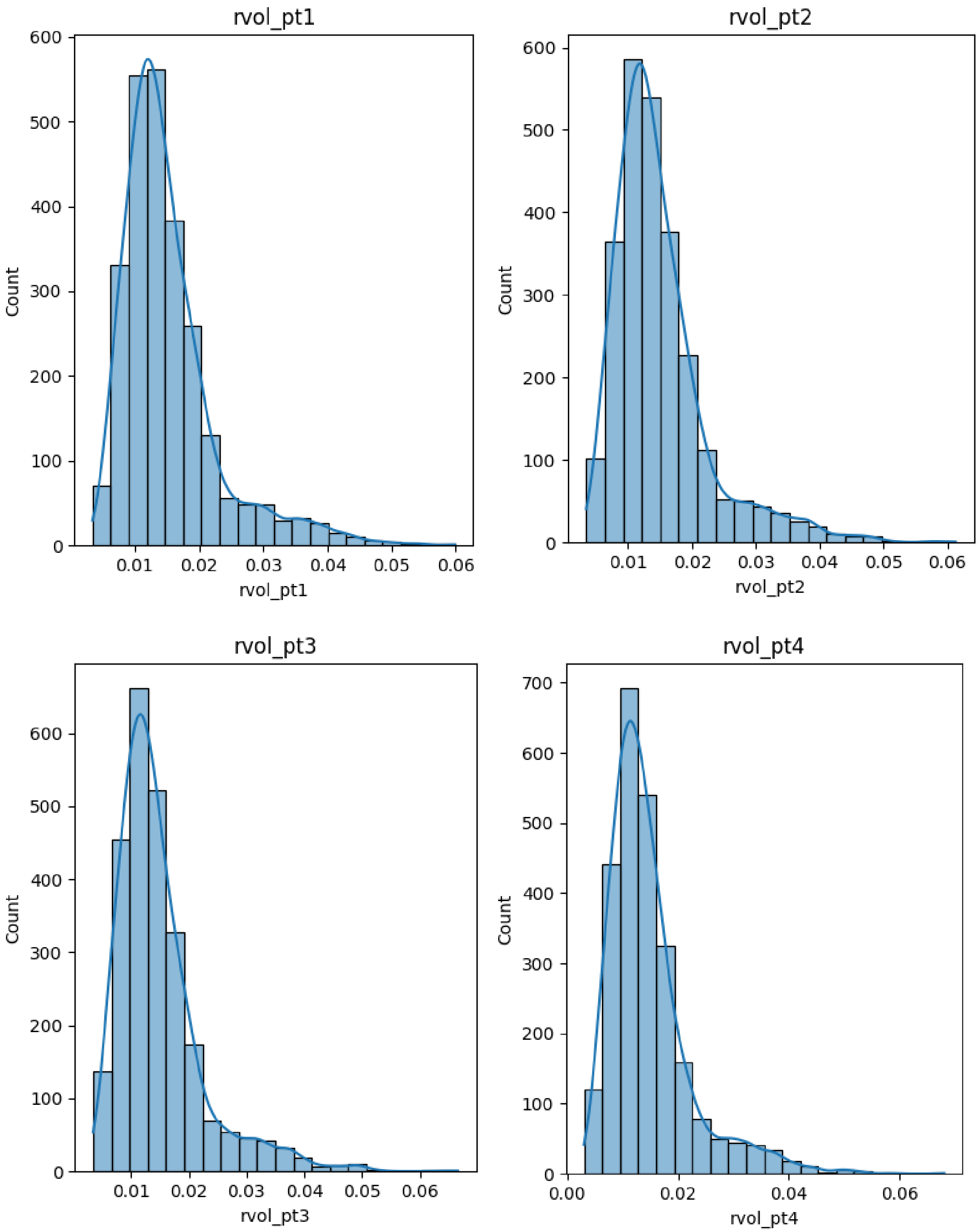


Figure 1. Distribution of realized volatility averaged over 1–4-minute time periods. Rvol stands for realized volatility, and pt (previous tick) values indicate the number of minutes over which the price is averaged (e.g., pt1 = 1 minute) for ICE Brent Crude Oil minute-level data from January 3, 2006, to January 29, 2016 (total 2,567 trading days).

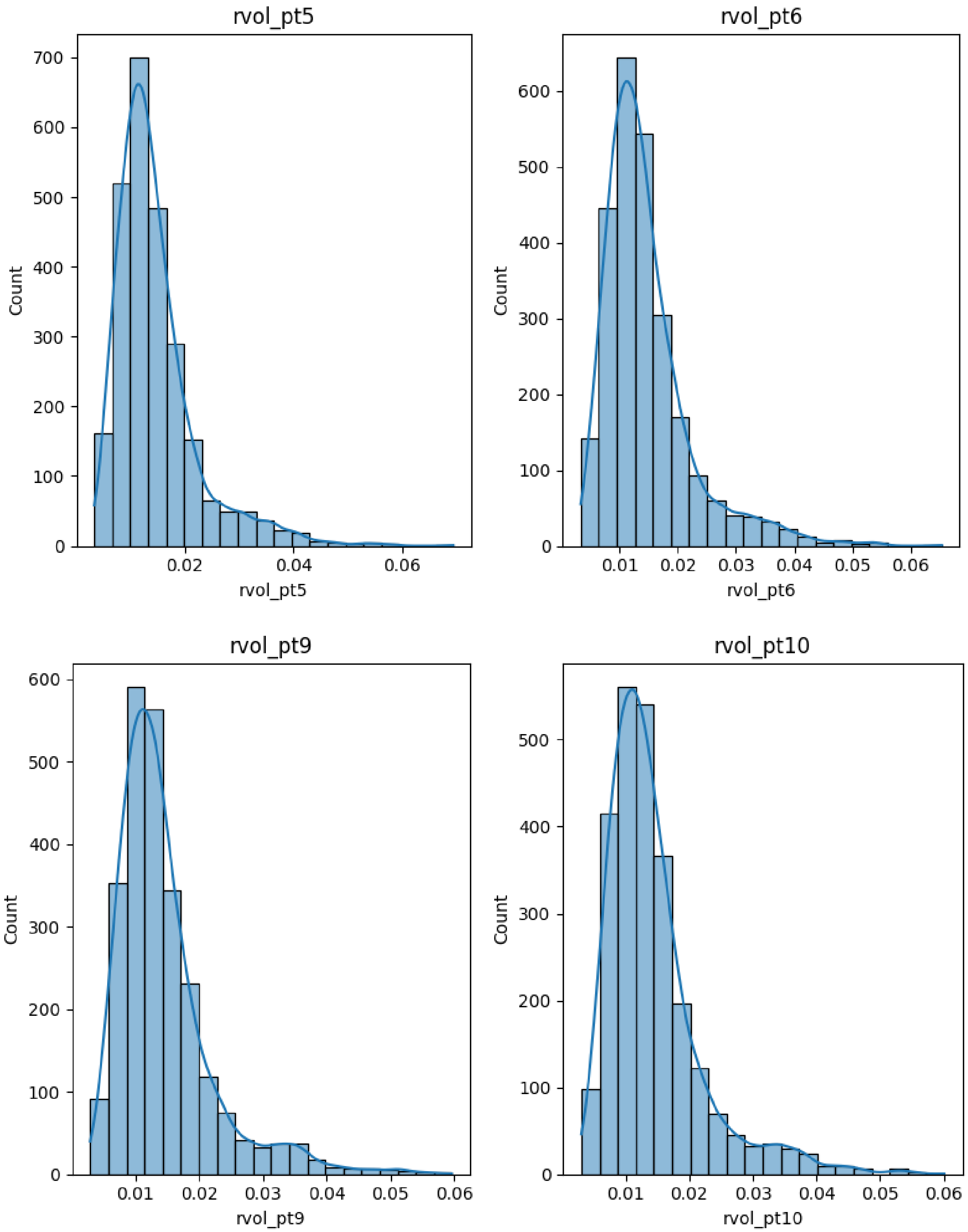


Figure 2. Distribution of realized volatility averaged over 5, 6, 9, and 10-minute time periods. Rvol stands for realized volatility, and pt (previous tick) values indicate the number of minutes over which the price is averaged (e.g., pt5 = 5 minutes) for ICE Brent Crude Oil minute-level data from January 3, 2006, to January 29, 2016 (total 2,567 trading days).

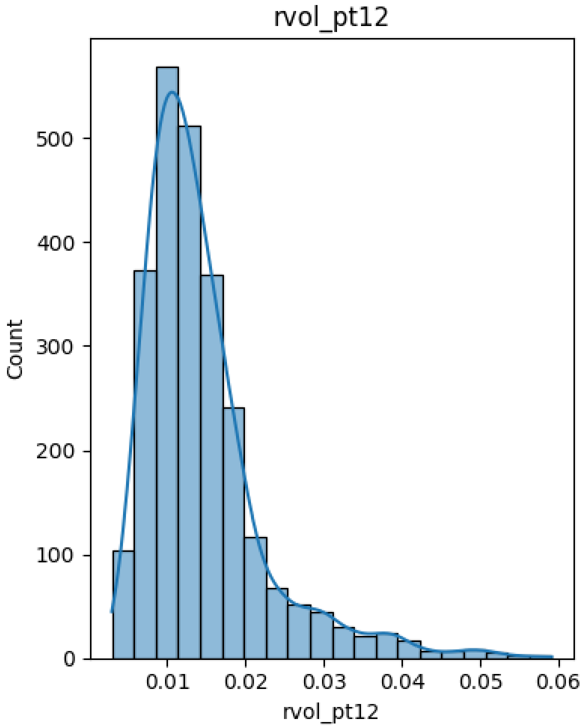


Figure 3. Distribution of realized volatility averaged over a 12-minute time period. Rvol stands for realized volatility, and pt (previous tick) values indicate the number of minutes over which the price is averaged (e.g., pt12 = 12 minutes) for ICE Brent Crude Oil minute-level data from January 3, 2006, to January 29, 2016 (total 2,567 trading days).

The right-skewed shape of the histograms suggests that the data is concentrated towards lower values, with a tail extending towards higher values. This skewness is common in financial data, particularly for variables like returns or volatility, where extreme values occur infrequently but can have a significant impact on the overall analysis. In the context of financial market analysis, these histograms represent the distribution of realized volatility measures over different time intervals, which are crucial for assessing market risk and volatility patterns. The consistency in the shape of the histograms across various periods indicates a stable volatility pattern or behaviour over time. Analysing these histograms provided insights into the frequency and distribution of volatility values within each time interval, highlighting the variability and potential outliers in the data.

Once the dataset was filtered to include relevant variables and histograms were assessed for data distribution, descriptive statistics including mean, standard deviation, minimum, maximum, and quartiles were calculated to further provide a comprehensive overview of the dataset's characteristics and distribution (Table 1). These statistical metrics offer valuable insights into the behaviour and variability of the volatility measures, enabling a deeper understanding of the data for further analysis and interpretation. Metrics like mean and median reveal central tendencies, while standard deviation and quartiles highlight data variability. Specifically, the mean provides a measure of central tendency, indicating the typical value of the variable, and the standard deviation measures the dispersion of data points around the mean, with a higher value suggesting greater variability. Quartiles, dividing the data into four equal parts, help assess spread and distribution, and the minimum and maximum values reflect the range of observed values. These statistics aid in identifying outliers, assessing data quality, and detecting patterns in trading data analysis. Displaying the statistical summary provides a concise overview of these key metrics, offering a snapshot of the dataset's properties. Conducting a data quality check through the statistical summary helps identify missing values or inconsistencies, ensuring data reliability. This comprehensive understanding of volatility variables aids analysts in making informed decisions, identifying trends, and assessing data suitability for further analysis or modelling in trading datasets.

Table 1. Table of statistical measures including count (number of non-null observations), mean (average), standard deviation (std dev; variation or dispersion), minimum (min), maximum (max), and quartiles (25th, 50th, 75th percentiles) of realized volatility (rvol) for each time period (e.g., pt = previous tick) analysed.

Sampling	Count	Mean	Std Dev	Min	25%	50%	75%	Max
rvol_pt1	2567	0.0152	0.0076	0.0034	0.0105	0.0134	0.0178	0.0599
rvol_pt2	2567	0.0150	0.0075	0.0034	0.0102	0.0132	0.0175	0.0613
rvol_pt3	2567	0.0148	0.0076	0.0035	0.0100	0.0131	0.0174	0.0666
rvol_pt4	2567	0.0147	0.0075	0.0031	0.0099	0.0130	0.0172	0.0680
rvol_pt5	2567	0.0146	0.0076	0.0034	0.0010	0.0128	0.0171	0.0693
rvol_pt6	2567	0.0147	0.0076	0.0033	0.0098	0.0128	0.0173	0.0653
rvol_pt9	2567	0.0145	0.0076	0.0030	0.0096	0.0127	0.0169	0.0596
rvol_pt10	2567	0.0144	0.0076	0.0030	0.0094	0.0125	0.0168	0.0601
rvol_pt12	2567	0.0144	0.0078	0.0029	0.0093	0.0125	0.0170	0.0159
rvol_pt15	2567	0.0142	0.0079	0.0022	0.0092	0.0125	0.0168	0.0836
rvol_pt18	2567	0.0142	0.0078	0.0026	0.0093	0.0124	0.0167	0.0635
rvol_pt20	2567	0.0140	0.0077	0.0026	0.0090	0.0122	0.0165	0.0651

The consistent count across variables indicates an equal amount of data for each, while varying mean, standard deviation, and range values suggest differences in central tendencies and data spread. The percentiles offer insights into data distribution, with the 50th percentile representing the median. The differences in mean, standard deviation, and range imply these variables may represent distinct aspects or dimensions of the analysed data, possibly related to realized volatility for different time periods or conditions in a time-series or financial analysis context.

Additionally, insights into data distribution and patterns were obtained through visualization of box plots (Figure 4). The consistency in data range and concentration of values within a narrow band across variables suggest uniform data spread, while outliers signify extreme values outside the typical range, possibly indicating exceptional events.

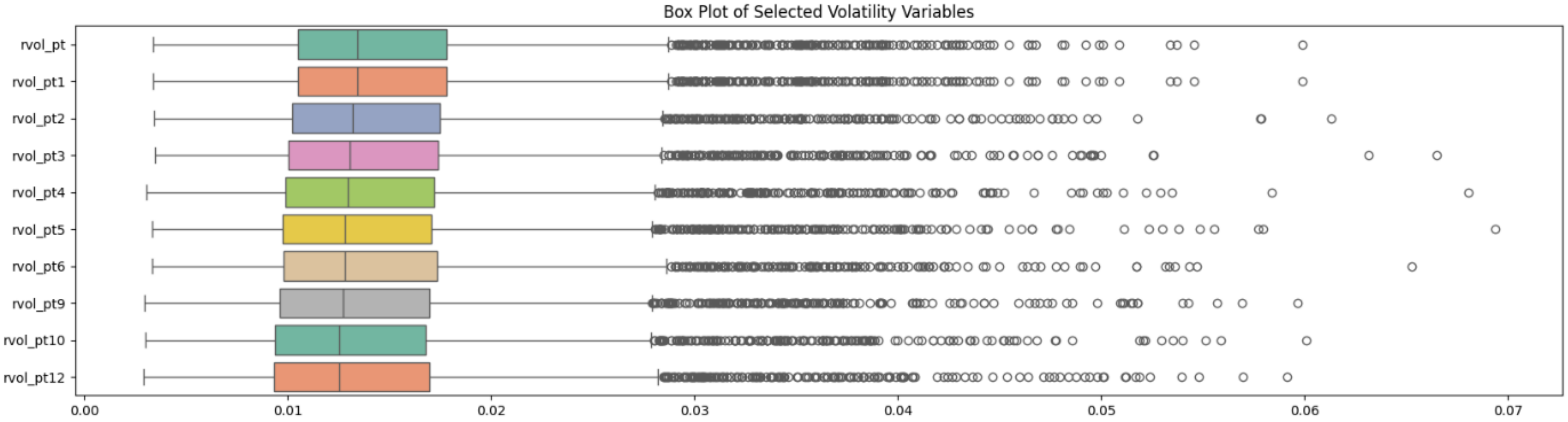


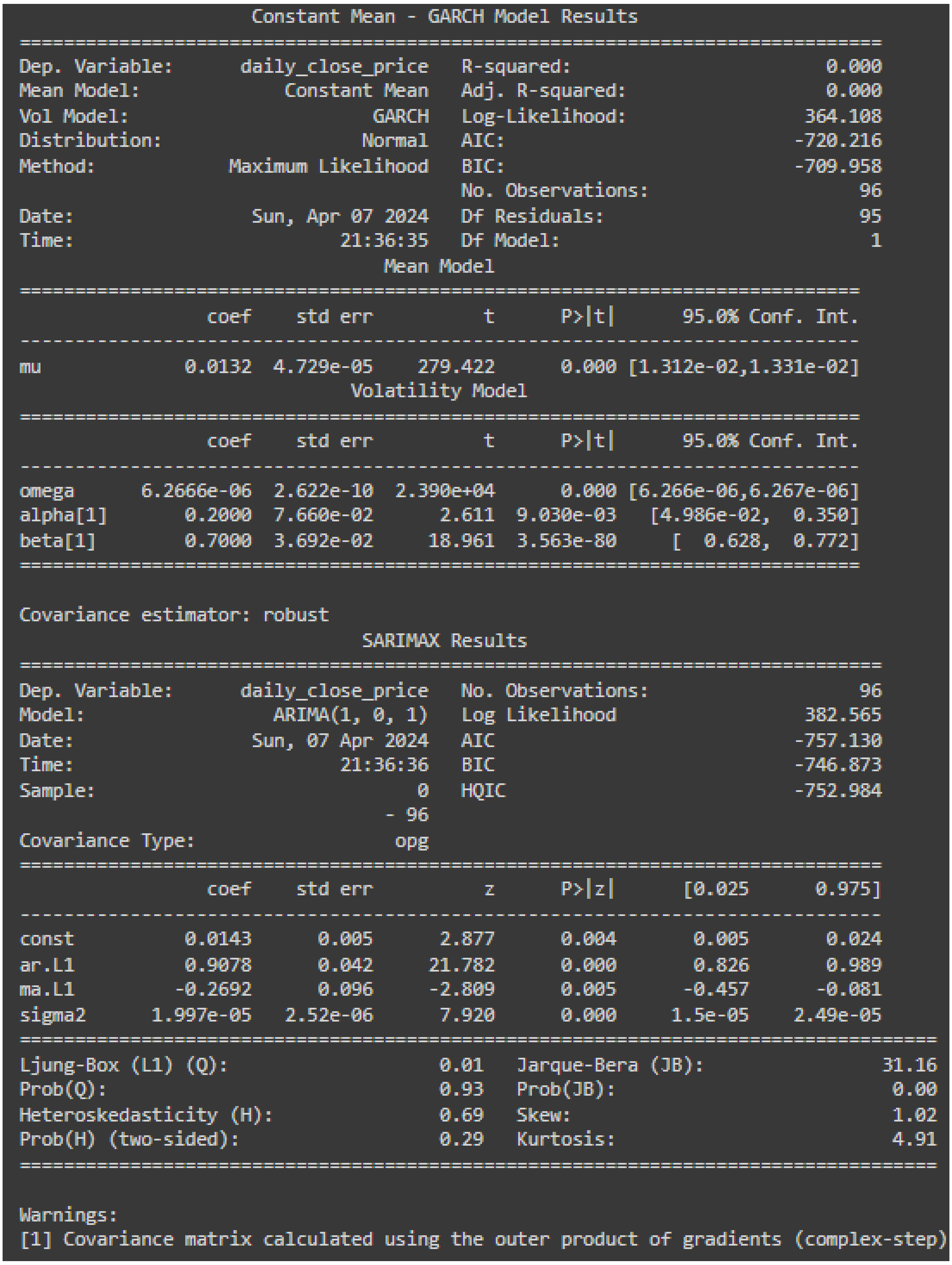
Figure 4. Box plots of selected volatility variables. Interquartile Range (IQR) represents the middle 50% of data point variability (boxes), median indicating central tendency (vertical lines in each box), whiskers extending to min and max values exclude outliers (open circles), which deviate significantly from the overall pattern.

In financial markets, box plots are valuable for comparing dispersion and central tendency of realized volatility measures over different time periods or financial instruments. The uniform distribution of medians around a central range suggests a stable data central tendency, whereas the outliers may indicate sporadic volatility spikes or exceptional market conditions. Analysing box plots aids in comprehending data variability, distribution, and anomalies, facilitating informed decision-making and risk assessment in financial market analysis.

4.2 Statistical Model Results

The analysis of statistical models, which employs GARCH and ARIMA models to a financial time-series dataset of daily closing prices, presents a comprehensive look at the underlying volatility and trend patterns in the data.

The GARCH model (see the top half of Figure 5 for model outputs) is adept at capturing volatility dynamics in financial time series data. In this analysis, the model underscores the importance of past volatility, an indication of volatility clustering, where periods of high volatility tend to be followed by high volatility and low by low. This pattern identification is particularly useful for risk management because it allows practitioners to adjust their strategies during periods of expected high volatility. The estimated omega coefficient of 2.666e-06 and alpha (long-run average variance) of 0.2800 in the GARCH model are statistically significant (*p*-values < 0.05). This significance is essential because it justifies the inclusion of these terms in predicting future volatility by providing strong evidence that the results were not obtained by chance. Similarly, the beta coefficient of 0.7000, which captures the persistence of volatility shocks, is significant, indicating that volatility shocks tend to persist over time. Furthermore, the model's fit to the data is substantiated by the AIC and BIC values of –720.216 and –709.958, respectively. These criteria measure the relative quality of statistical models for a given set of data; lower values generally indicate a better fit.



```
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:      daily_close_price   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:                364.108
Distribution:                  Normal   AIC:                          -720.216
Method:            Maximum Likelihood   BIC:                          -709.958
                                        No. Observations:                   96
Date:                Sun, Apr 07 2024   Df Residuals:                       95
Time:                        21:36:35   Df Model:                            1
                                 Mean Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
mu             0.0132  4.729e-05    279.422      0.000 [1.312e-02,1.331e-02]
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      6.2666e-06  2.622e-10  2.390e+04      0.000 [6.266e-06,6.267e-06]
alpha[1]       0.2000  7.660e-02      2.611  9.030e-03   [4.986e-02,  0.350]
beta[1]        0.7000  3.692e-02     18.961  3.563e-80     [  0.628,  0.772]
============================================================================

Covariance estimator: robust
                               SARIMAX Results
==============================================================================
Dep. Variable:      daily_close_price   No. Observations:                   96
Model:                 ARIMA(1, 0, 1)   Log Likelihood                 382.565
Date:                Sun, 07 Apr 2024   AIC                           -757.130
Time:                        21:36:36   BIC                           -746.873
Sample:                             0   HQIC                          -752.984
                                 - 96
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0143      0.005      2.877      0.004       0.005       0.024
ar.L1          0.9078      0.042     21.782      0.000       0.826       0.989
ma.L1         -0.2692      0.096     -2.809      0.005      -0.457      -0.081
sigma2      1.997e-05   2.52e-06      7.920      0.000     1.5e-05    2.49e-05
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                31.16
Prob(Q):                              0.93   Prob(JB):                         0.00
Heteroskedasticity (H):               0.69   Skew:                             1.02
Prob(H) (two-sided):                  0.29   Kurtosis:                         4.91
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Figure 5. Statistical modelling outputs for GARCH (top) and ARIMA (bottom) modelling of ICE Brent Crude Oil market data.

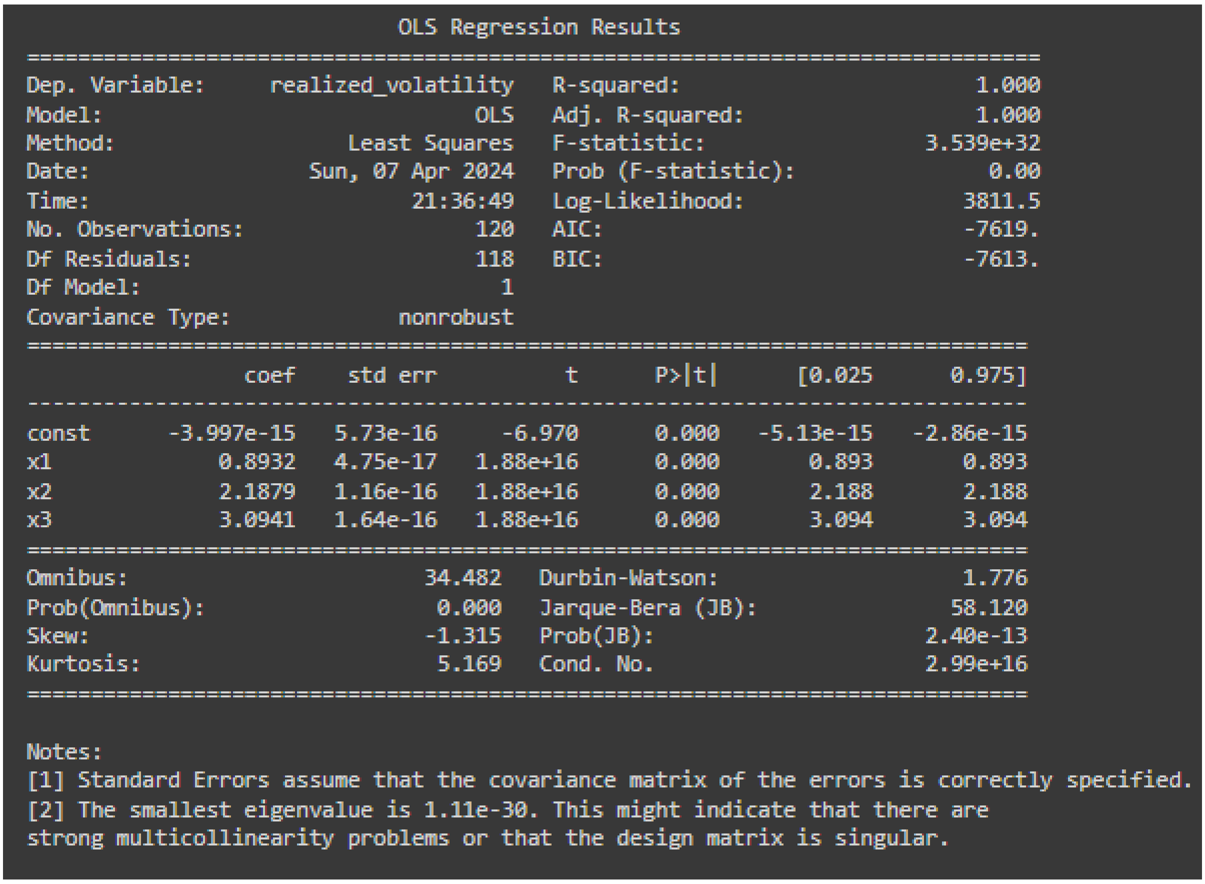
Moving on to the ARIMA model results (see the bottom half of Figure 5 for model outputs), the significant autoregressive term (ar.L1) with a coefficient of 0.9078 suggests that there is a strong continuity in the price series. That is, past prices are a strong predictor of future prices. The moving average term (ma.L1) with a coefficient of –0.2692, also significant with a *p*-value of 0.005, indicates that the model attempts to correct for any shock that occurred in the previous period. The estimated variance (sigma^2) of 1.997e-05 is quite small, implying that the model predicts the closing prices will have minimal deviations from the mean, which can be interpreted as a generally stable series over the period under study.

The model diagnostics for the ARIMA model are also revealing. The Ljung-Box test provides a *p*-value of 0.01, suggesting that at least some autocorrelations are significantly different from zero at the 5% level, pointing towards potential model misspecification or the presence of unexplained patterns in the residuals. Additionally, the Jarque-Bera test statistic of 31.16 with a *p*-value of 0.06, slightly above the 5% significance level, suggests a mild departure from normality. This is common in financial data where the distribution of returns can exhibit fat tails and skewness, often caused by the leverage effect or black swan events. The skewness and kurtosis values of 1.02 and 4.91, respectively, confirm that the residuals are not normally distributed: they are positively skewed and have heavier tails than the normal distribution.

Both GARCH and ARIMA models offer valuable insights but also suggest different aspects of the data. The GARCH model is key for understanding volatility, while the ARIMA model is helpful for capturing the trends and correcting for past prediction errors. Given the findings, particularly the mild deviations from normality and the remaining autocorrelation in the residuals, further refinement of the models could be beneficial. Alternative distributions like the student's *t* or skewed distributions in the GARCH model could be explored to better accommodate the fat tails and skewness observed. Additionally, including exogenous variables, such as macroeconomic factors or market indices, might capture external influences affecting the volatility and trends. It is also recommended to continually validate the models against new, out-of-sample data to confirm their predictive power and adjust them as needed to maintain forecasting accuracy. This iterative process of model updating ensures that the predictive models stay aligned with the evolving market conditions.

The results from the HAR-AV model (Figure 6) provide a robust analysis of the realized volatility using historical lagged volatilities as predictors. However, the reported R^2 and adjusted R^2 values of 1.000 in the model output are striking, as they indicate that the model explains 100% of the variability in the realized volatility of the financial time series. Such perfect fit values often raise concerns about the possibility of overfitting, where the model is too closely tailored to the sample data and may not generalize well to out-of-sample data. It is unusual for empirical models to achieve such perfect fit, and it suggests that the results should be approached with caution and rigorous validation with new data be performed. The *F*-statistic is extremely high, and the associated *p*-value is essentially zero, which strongly suggests that the model is statistically significant. This means that the variables included in the model collectively have significant predictive power in explaining the variation in realized volatility. The coefficients for all lagged volatility terms (x1, x2, x3) are statistically significant, with *p*-values well below the standard 0.05 threshold. This significance indicates that past volatility (daily, weekly, and monthly) is a crucial predictor of current volatility, which is consistent with volatility clustering.

Model diagnostics provide additional context for these results. The Durbin-Watson statistic value of 1.776 suggests there is minimal autocorrelation in the model's residuals. This is a positive sign, indicating that the model captures the temporal structure in the volatility well without leaving behind systematic patterns in the residuals. However, the Omnibus test results in a *p*-value of 0.00, and the Jarque-Bera test also has a significant *p*-value of 2.40e-13, suggesting that the residuals do not follow a normal distribution. This interpretation is further supported by a skewness statistic of –1.315 and an excess kurtosis of 5.169. Such non-normality in residuals can be a concern because it might violate the OLS assumption of normally distributed errors, potentially leading to inefficiency of estimates and invalid inferential statistics.



```
                            OLS Regression Results
===============================================================================
Dep. Variable:     realized_volatility   R-squared:                       1.000
Model:                             OLS   Adj. R-squared:                  1.000
Method:                  Least Squares   F-statistic:                 3.539e+32
Date:                 Sun, 07 Apr 2024   Prob (F-statistic):               0.00
Time:                         21:36:49   Log-Likelihood:                 3811.5
No. Observations:                  120   AIC:                            -7619.
Df Residuals:                      118   BIC:                            -7613.
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -3.997e-15   5.73e-16     -6.970      0.000   -5.13e-15   -2.86e-15
x1             0.8932   4.75e-17   1.88e+16      0.000       0.893       0.893
x2             2.1879   1.16e-16   1.88e+16      0.000       2.188       2.188
x3             3.0941   1.64e-16   1.88e+16      0.000       3.094       3.094
==============================================================================
Omnibus:                       34.482   Durbin-Watson:                   1.776
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               58.120
Skew:                          -1.315   Prob(JB):                     2.40e-13
Kurtosis:                       5.169   Cond. No.                     2.99e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.11e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Figure 6. Statistical modelling outputs for HAR-AV modelling using ordinary least squares regression of ICE Brent Crude Oil realized volatility data.

Given the non-normality issues highlighted by the tests, further investigation into the model's residuals is necessary. This investigation could include checking for structural breaks, outliers, or influential points that may be unduly affecting the model's results. Additionally, it may be prudent to test for stationarity in the volatility series and ensure that the time series is appropriately differenced, if necessary, before modelling. Considering the potential for multicollinearity (i.e., correlation among predictor variables), as noted by the smallest eigenvalue being close to zero, it is crucial to examine the variance inflation factors for the predictors. If multicollinearity is present, it could be inflating the standard errors of the coefficients, leading to less reliable hypothesis tests. Given the observed imperfections in the model, it may be beneficial to explore other volatility models, such as GARCH-family models, which are specifically designed to handle the clustering of volatility and can model the conditional heteroskedasticity often observed in financial time series data. It's also recommended to validate the model's predictive accuracy using out-of-sample data. This helps ensure that the model's perfect in-sample fit translates into strong predictive capabilities moving forward. Finally, alternative models or additional terms might be considered to address the issues of skewness and kurtosis in the residuals, such as transforming the response variable or using models that are robust to non-normality. The exploration of alternative error distributions within the GARCH framework, for instance, could be a step towards more robust volatility forecasts.

4.3 Machine Learning Modelling Results

Three regression models were explored: multi-layer perceptron (MLP), random forest (RF), and support vector machine (SVM), across various metrics.

The loss curve over epochs for the MLP model is illustrated in Figure 7. The loss drops sharply in the initial epochs, indicating rapid learning at this early stage. However, there is a slight increase at around epoch 10, possibly due to overfitting or learning rate adjustments. Following this, the loss continues to decline and stabilizes, which is indicative of the model converging to a solution.

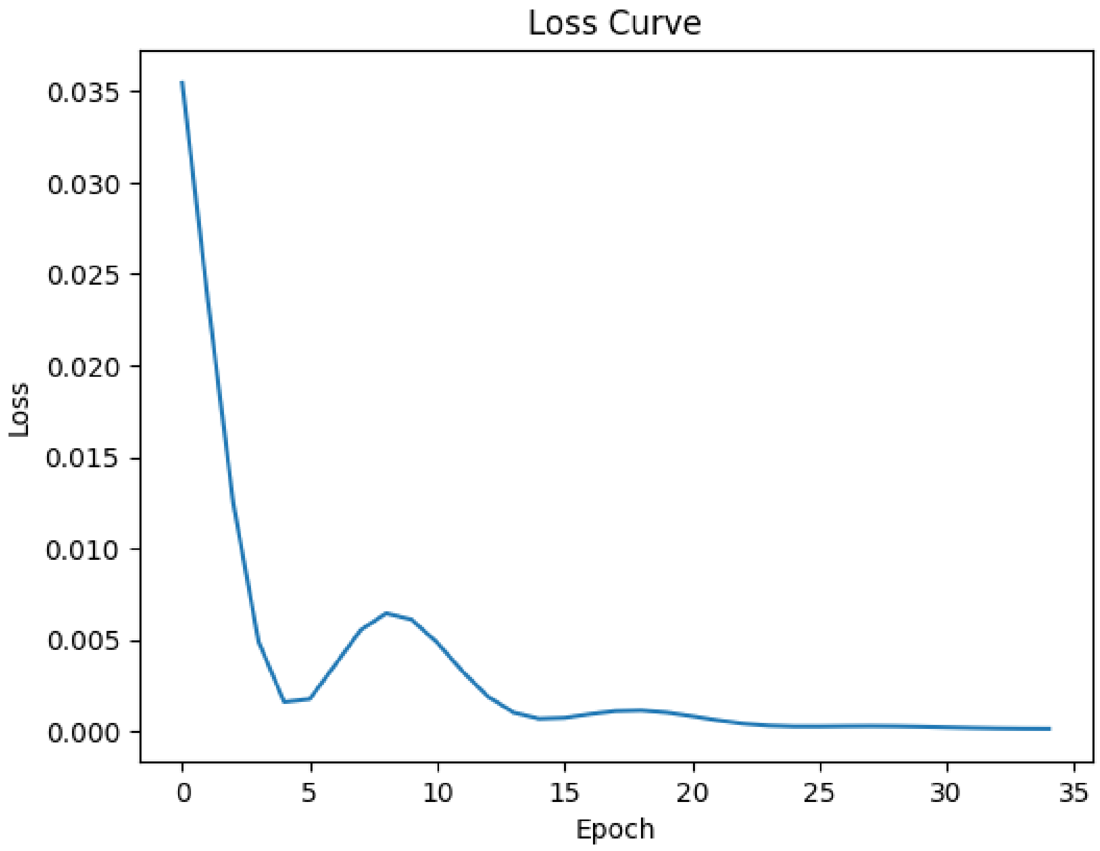


Figure 7. The loss curve over epochs for the multi-layer perceptron (MLP) model.

The RF model performed excellently on both the training and test sets, with MSE and MAE nearing zero and R^2 values being almost perfect (Figure 8). This suggests that RF is highly effective in capturing the patterns in the data without overfitting, as evidenced by the consistent performance in training and testing phases. SVM (Figure 9), however, shows some negative values in R^2, especially on the test set, which may indicate a model fit that's worse than the mean model. Negative R^2 values arise when the chosen model does not follow the trend of the data, so the predictions end up being less accurate than if one had simply predicted the mean of the target variable. MLP also shows negative R^2 values (Figure 10), suggesting that this model, like SVM, is performing poorly on the given dataset. The loss functions for MLP depicted in the bar chart indicate that while the model performs slightly better in the test set than SVM, it still underperforms when compared to Random Forest.

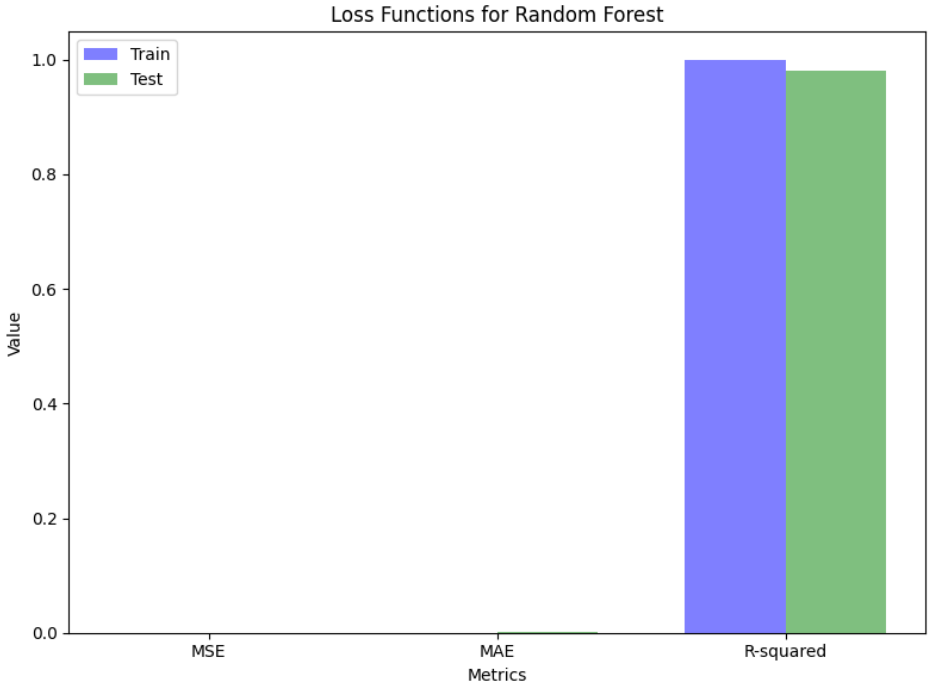


Figure 8. Performance of random forest (RF) model.

When predicting volatility in financial data, models like GARCH may show high R^2 scores due to the clear patterns in temporal financial data. However, general machine learning models like MLP and SVM may struggle with non-linear, stochastic data lacking clear patterns. Overfitting is a concern in MLP and SVM if not properly regularized, especially in high-dimensional datasets. Volatility models use directly related features, while scaling is crucial for MLP and SVM. Recommendations include improving feature engineering, hyperparameter tuning, and robust validation techniques to enhance model performance across diverse datasets.

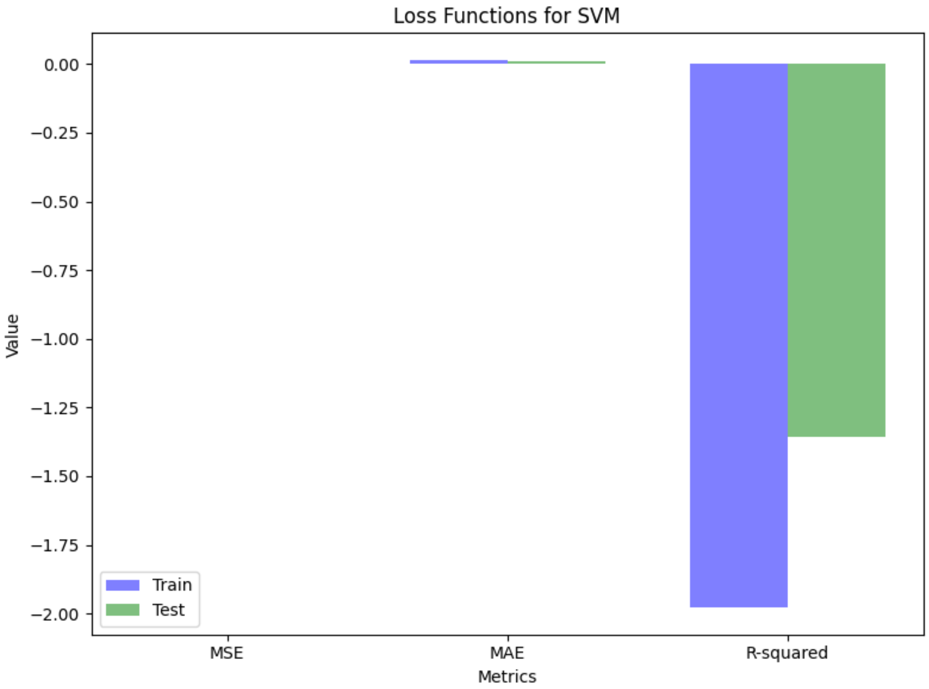


Figure 9. Performance of multi-layer perceptron (MLP) model.

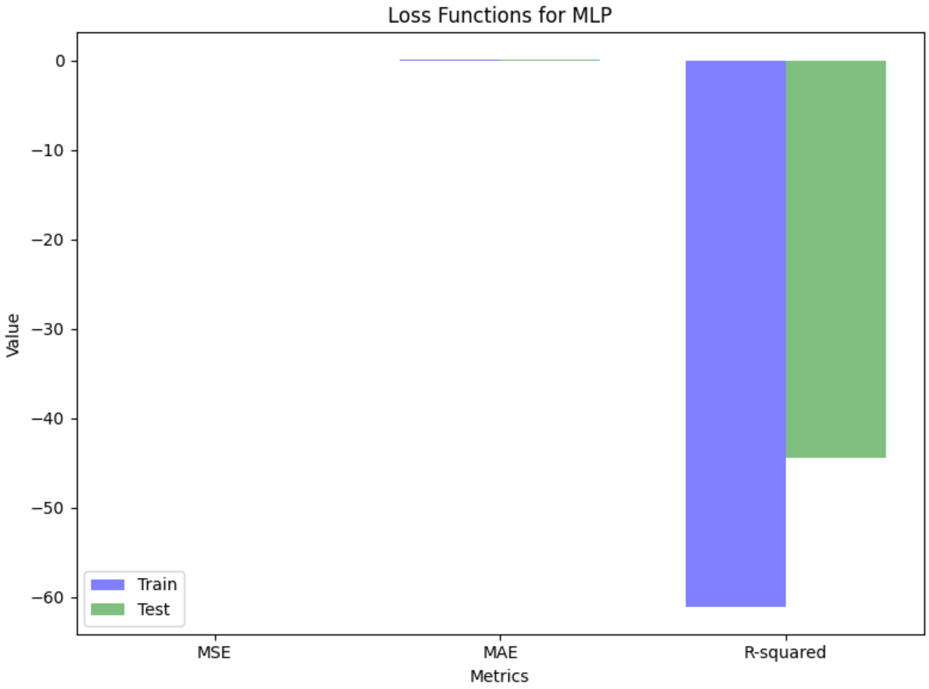


Figure 10. Performance of support vector machine (SVM) model.

The scatter plot of predicted versus actual values (Figure 11) allows for a visual assessment of each model's predictive accuracy. For an ideal model, the points would lie on the diagonal line, indicating perfect prediction. MLP predictions are quite spread out from the line, suggesting a disparity between predictions and actual values. The MLP model may not be capturing the complexity of the data efficiently, which could be due to the architecture not being optimized for the data at hand. Adjustments such as tuning the number of layers, neurons, or even the regularization parameters could potentially improve its performance. Random Forest appears to have a tighter clustering around the line, particularly for the middle range of values, indicating better predictive performance. The RF model, with its ensemble approach, appears to effectively capture the data's patterns, suggesting that the average of multiple decision trees (each considering a random subset of features) is able to generalize well from the training data to the test data. The SVM model output has predictions that generally overestimate for lower actual values and underestimate for higher actual values. The SVM model seems to struggle with this dataset, potentially due to the choice of kernel or the need for more fine-tuned hyperparameters.

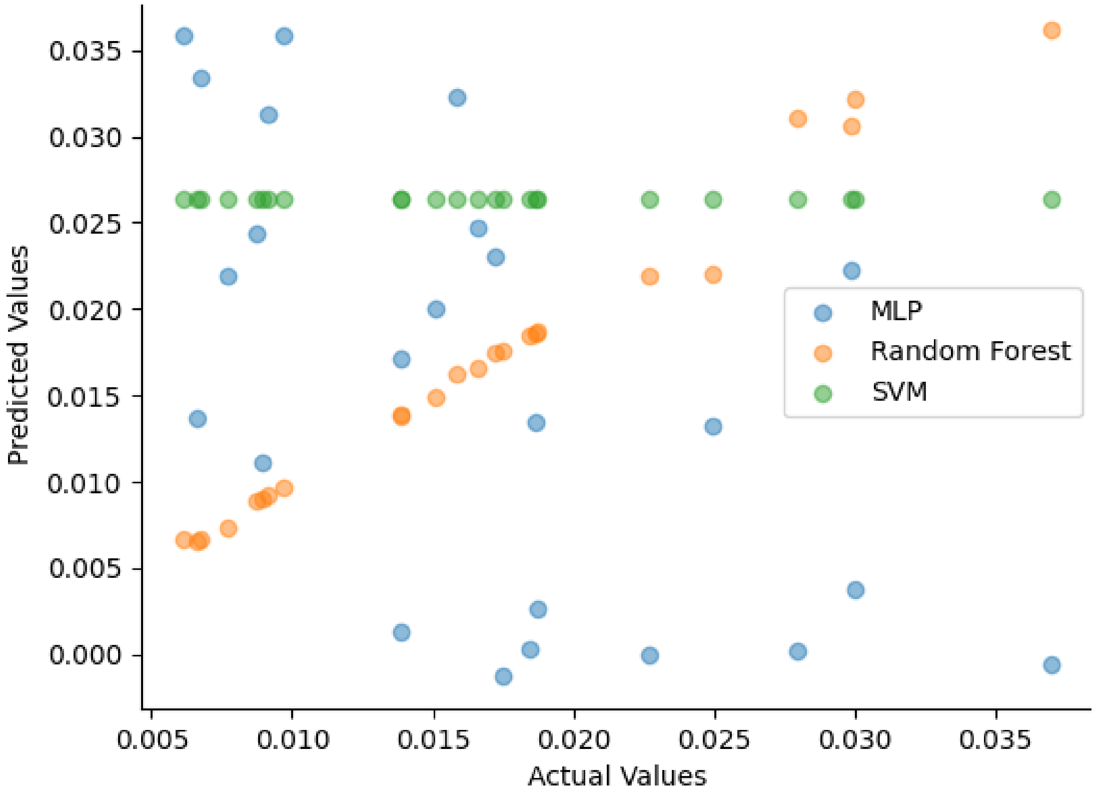


Figure 11. Predicted vs. actual values among machine learning models.

In conclusion, the RF model stands out with superior performance, which aligns with the model's theoretical underpinnings as a robust ensemble method. However, both the MLP and SVM models exhibit issues that require further investigation and model tuning. Given the complexity often inherent in financial data, a combination of model tuning, feature engineering, and alternative modelling approaches might be necessary to achieve optimal predictive performance. Additionally, the negative R^2 values for MLP and SVM on the test sets warrant a deeper dive into their respective model diagnostics and potential overfitting issues for MLP and underfitting for SVM.

4.4 Feature Selection Results

Comparisons between the performance of various models in accurately prediction variability in data can provide valuable insights. The comparative analysis in this study included traditional linear models like linear regression and ridge regression, ensemble models such as random forest and gradient boosting, a machine learning mainstay in SVR, and time-series-specific models like GARCH, ARIMA, and LSTM (Figure 12). These models have undergone evaluation through various error metrics to assess their performance. Metrics such as mean squared error and root mean squared error give us insights into the average of the squared differences and the square root of these differences between predicted and actual values, where lower figures are indicative of a model with predictions closely matching the observed data. Mean absolute error provides another angle, offering the average of the absolute differences, which serves to understand the precision of the models outside of their squared errors.

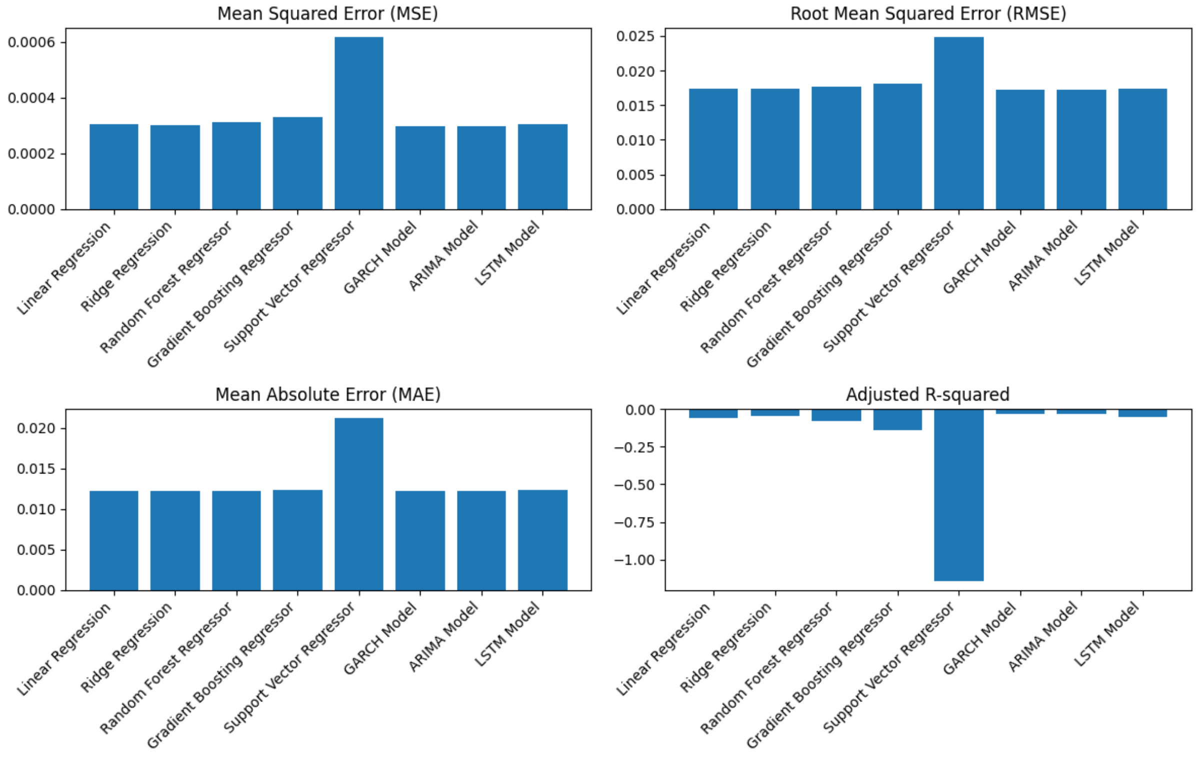


Figure 12. Bar chart presenting a comparative analysis of various machine learning models applied to ICE Brent Crude Oil futures data.

One of the key takeaways from the visual representation of this performance data is that the models exhibit varying degrees of accuracy and reliability, as indicated by their respective error rates. While the chart does not display mean absolute percentage error, typically, a lower value in this metric would be desirable because it represents the average of the percentage differences between the predictions and actual figures, providing a scale-free context to the error. The adjusted R^2 metric also offers valuable insight, adjusting for the number of predictors and the sample size to reflect the model’s explanatory power more accurately. In scenarios where this value is negative, it suggests that the model's predictive capacity is worse than that of a simple horizontal line.

This comparative analysis aims to deduce which model best suits the task of predicting volatility in the dataset provided, a crucial step in financial modelling, particularly in the realm of HFT. Models that maintain a balance between low error rates and high explanatory power without succumbing to overfitting are often favoured. Overfitting can be detrimental, as it indicates that a model might perform well on training data but fail to generalize to unseen data. While the data points to certain models being more effective than others in this specific context, the true test of a model's utility will be in its application to new data and the consistency of its predictive quality.

4.5 Volatility Forecast Performance

Examination of collective volatility forecast performance plots formed a detailed examination of market volatility, mapped out over various time frames—daily, weekly, and monthly—providing a multifaceted view of market dynamics (Figures 13 and A2.1–A2.3). Starting with daily lagged volatility, which spans across numerous consecutive days, these plots trace the immediate fluctuations in the market. Such short-term variances are typically sensitive to day-to-day market news and events, where even slight tremors in investor sentiment or economic indicators can create ripples of change. Moving to a broader scale, the weekly average volatility plots stretch across several weeks, offering a tempered perspective compared to the daily data. These averages smooth over the abrupt spikes of daily volatility, revealing more persistent trends and sentiments in the market.

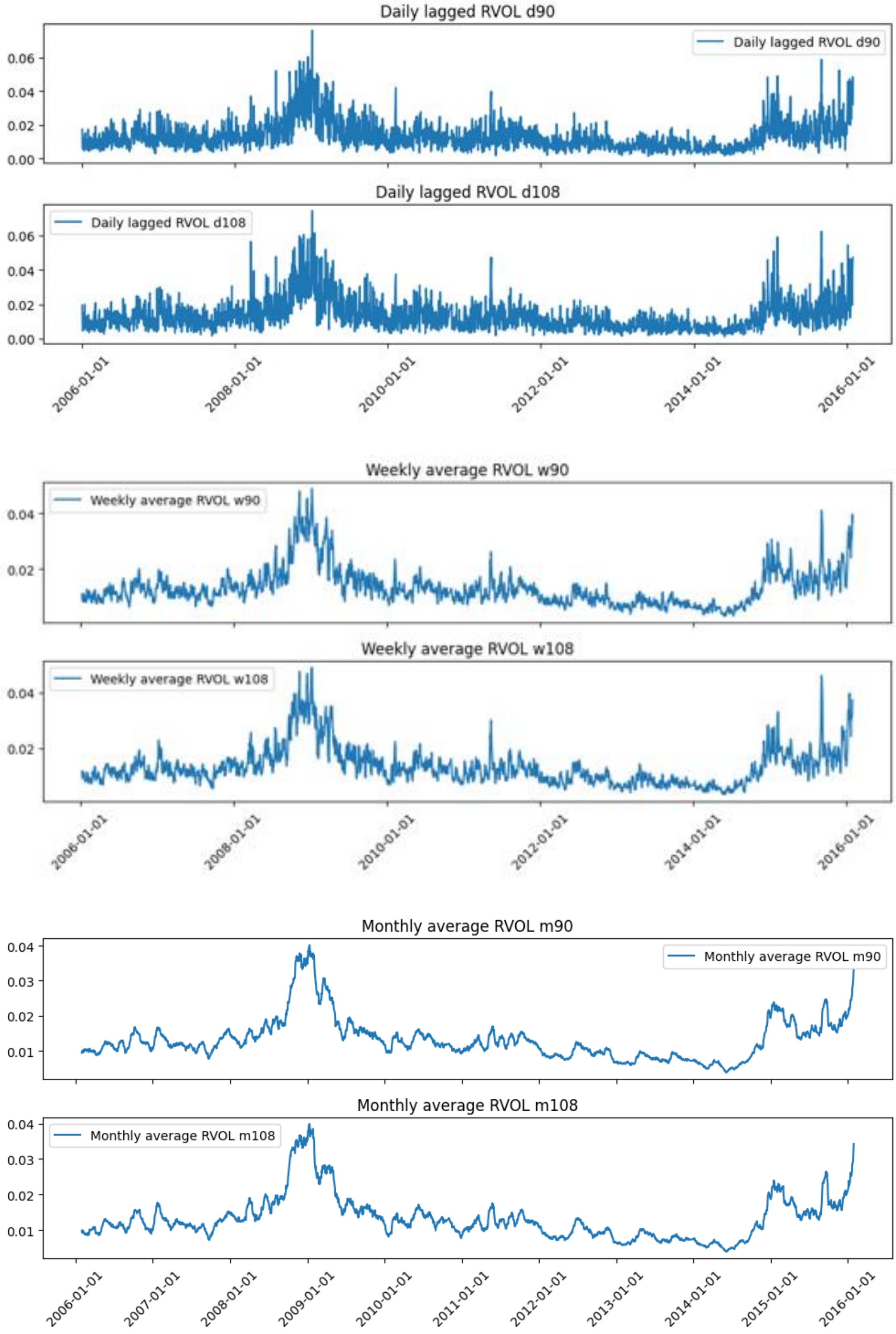


Figure 13. Examples of volatility forecast performance plots averaged over daily (top), weekly (centre), and monthly (bottom) time periods for a selection of each. See Appendix 2 for full performance plots.

The monthly average volatility plots, encompassing an even wider time span, depict the longest-term trends in market behaviour. These plots are critical in understanding the sustained shifts in volatility, which are less reactive to the 'noise' of daily market moves and more indicative of deeper, systemic changes in the economic landscape. By examining these visual narratives of volatility, analysts can infer underlying patterns such as seasonality or long-term cycles. Such insights are key to deciphering the market's rhythm, which can further inform investment decisions, trading strategies, and risk management. In essence, these visualizations serve not just as a record of past market behaviour, but also as a lens through which future market movements might be anticipated and understood.

A correlation heatmap of volatility variables was examined as a comprehensive visual aid to understand the interconnections among different volatility metrics across time frames (Figures 14, A3). This heatmap portrays how various volatility measures co-move or diverge, a concept of paramount importance in the financial world where the objective is often to gauge and navigate market risks effectively. (Figure 14 provides a snapshot of correlations between realized volatility variables for minute-level time periods. See Appendix 3 for a full heatmap with correlations between all possible variable combinations.)

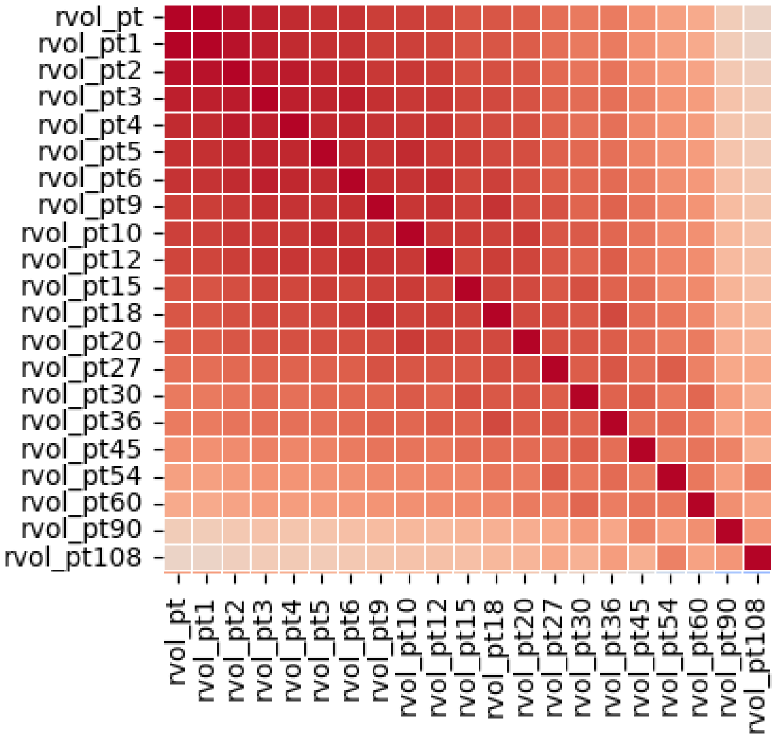


Figure 14. Correlation heatmap of volatility variables for minute-level data. See Appendix 3 for full correlation heatmap with all possible variable combinations.

At first glance, the varying shades of red across the heatmap highlight the extent of positive correlation among the variables. Notably, clusters of darker reds reveal a strong positive correlation, particularly within similar time spans such as daily volatilities. This pattern suggests a synchronicity in the way these measures react to market stimuli, a reflection of the market's cohesiveness in response to short-term events. Conversely, areas in blue signify lower or inverse correlations (Appendix 3). These areas are particularly intriguing because they imply that certain volatility measures do not track together, indicating a divergence in market behaviour across different periods. For instance, measures that are less correlated may provide insights into the multi-dimensional nature of market volatility, underscoring the complexities that come with forecasting and risk assessment.

The heatmap is also self-referential, as evidenced by the diagonal line of perfect correlation. This is a natural occurrence, representing each variable's correlation with itself and serving as a benchmark for interpreting the rest of the heatmap. Moreover, the distinct block-like patterns that emerge on the heatmap encapsulate the notion that volatility measures within similar time frames—daily, weekly, or monthly—tend to exhibit a higher degree of correlation (Figures 14, A3). This pattern could indicate that similar underlying factors are at play within these time-specific groupings. For those in portfolio management, the correlations can serve as a beacon for diversification strategies. A portfolio composed of assets with varying correlations can mitigate risk, whereas a high correlation across the board might indicate that diversification benefits are minimal.

In predictive modelling, such correlations are a double-edged sword. While they can offer rich information, they also pose a risk of multicollinearity, which could skew model outcomes. Careful selection of variables is necessary to sidestep redundancy and ensure robust forecasts. Finally, from the standpoint of risk management, dissecting the correlations between short-term and long-term volatility measures could reveal how immediate market shifts may echo into more extended trends. Such insights are invaluable for risk managers tasked with safeguarding portfolios against market tumult. All in all, the heatmap transcends being a mere snapshot of volatility correlations, morphing into an analytical framework for financial experts aiming to decipher the undercurrents of market volatility.

4.6 Value-at-Risk Prediction Results

This study presents an exploration of VaR predictions using HFT data, providing a nuanced understanding of the various statistical and machine learning models' capabilities in quantifying financial risk. Value at risk is a critical concept in financial risk management, representing potential losses in the value of a risky asset or portfolio over a specific time frame, within a certain confidence interval. This metric is instrumental for financial institutions to manage their investment portfolio risks. In this examination, a selection of realized volatility measures, labelled from sampling \`1\` to \`108\`, representing sampling frequencies from 1 to 108 minutes, are analysed using OLS regression to predict daily closing prices (Figure 15). From the OLS residuals, VaR estimates are derived at various confidence levels on both tails of the distribution, providing a spectrum of potential gains and loss.

Based on the results presented in the table for Value-at-Risk (VaR) predictions using different sampling frequencies, it is observed that the one-factor quantile regression model performs well across various sampling frequencies compared to the OLS version. The quantile regression model shows consistent performance even at lower sampling frequencies, passing a significant number of VaR levels tested in the empirical exercise. In contrast, the OLS version's performance deteriorates quickly as the sampling frequency decreases, with lower accuracy in VaR forecasts compared to the quantile regression model.

The study highlights the importance of selecting an appropriate model for VaR predictions, emphasizing the effectiveness of the one-factor quantile regression model in providing reliable forecasts across different sampling frequencies. This finding suggests that the quantile regression approach offers a robust and stable method for forecasting VaR, particularly in the context of financial risk management. Furthermore, the research underscores the significance of exploring different statistical and machine learning models to enhance the accuracy and reliability of VaR predictions in financial risk management. By comparing the performance of various models, including the one-factor quantile regression model, the study contributes valuable insights into the effectiveness of different approaches for quantifying financial risk. The results support the use of the one-factor quantile regression model for VaR forecasting, as it demonstrates consistent performance and robustness across different sampling frequencies, providing practitioners with a reliable tool for managing financial risk effectively.

				Value-at-Risk	Prediction	Results						
Samling	Left VaR 1%	Left VaR 2.5%	Left VaR 5%	Left VaR 10%	Right VaR 99%	Right VaR 97.5%	Right VaR 95%	Right VaR 90%	Min	Max	Range	Pass Rate
1	0.005444	0.006165	0.006906	0.008049	0.042701	0.037438	0.031482	0.024364	0.003385	0.059893	0.056509	0.989871
2	0.005261	0.005876	0.006657	0.007736	0.042098	0.037007	0.031392	0.023778	0.003441	0.061294	0.057853	0.989871
3	0.005034	0.005632	0.006540	0.007677	0.042024	0.036864	0.031341	0.023617	0.003465	0.066505	0.063039	0.989871
4	0.004998	0.005604	0.006384	0.007548	0.041825	0.036619	0.031167	0.023574	0.003068	0.068046	0.064978	0.989871
5	0.004859	0.005640	0.006363	0.007355	0.041213	0.036165	0.030733	0.023537	0.003355	0.069349	0.065994	0.989871
6	0.004788	0.005535	0.006288	0.007505	0.041910	0.036255	0.030966	0.023728	0.003348	0.065267	0.061919	0.989871
9	0.004712	0.005440	0.006222	0.007305	0.043078	0.036006	0.031239	0.023481	0.002970	0.059645	0.056675	0.989871
10	0.004525	0.005400	0.006117	0.007150	0.042598	0.036442	0.030859	0.023301	0.003017	0.060065	0.057048	0.989871
12	0.004651	0.005184	0.006029	0.007208	0.043420	0.037149	0.030473	0.023618	0.002891	0.059147	0.056255	0.989871
15	0.004470	0.005219	0.005967	0.007050	0.042763	0.037345	0.030989	0.023340	0.002249	0.083602	0.081354	0.989871
18	0.004538	0.005133	0.005975	0.007039	0.043894	0.036729	0.030008	0.023486	0.002579	0.063499	0.060920	0.989871
20	0.004286	0.005071	0.005711	0.006785	0.042545	0.036873	0.029847	0.023093	0.002628	0.065096	0.062467	0.989871
27	0.004344	0.004949	0.005666	0.006661	0.043081	0.036560	0.030405	0.023218	0.002460	0.063362	0.060902	0.989871
30	0.004015	0.004835	0.005532	0.006545	0.042585	0.036913	0.030335	0.023191	0.002326	0.074485	0.072159	0.989871
36	0.003875	0.004667	0.005449	0.006383	0.043962	0.036736	0.030720	0.023487	0.002140	0.061013	0.058873	0.989871
45	0.003794	0.004515	0.005268	0.006266	0.042387	0.037560	0.031011	0.023443	0.002412	0.067388	0.064976	0.989871
54	0.003389	0.004153	0.004929	0.006067	0.045599	0.037509	0.030769	0.023186	0.001992	0.077990	0.075998	0.989871
60	0.003347	0.004104	0.004805	0.005865	0.045002	0.036753	0.029593	0.023010	0.001815	0.069897	0.068082	0.989871
90	0.002865	0.003638	0.004318	0.005471	0.045751	0.036805	0.030003	0.023352	0.001302	0.076211	0.074909	0.989871
108	0.002723	0.003276	0.003997	0.005067	0.046805	0.037394	0.030363	0.023587	0.001129	0.074250	0.073121	0.989871

Table 2. Value-at-risk predictions using one-factor model using daily realized volatility. Values are reported for various confidence intervals, along with minimum (min), maximum (max), range (max – min), and pass rate (measure of model performance).

The efficacy of different models is illustrated by the VaR prediction values. The GARCH model shows moderate accuracy in capturing volatility, benefiting from its mean-reversion feature, which is particularly relevant to financial time series. The ARIMA model establishes a baseline understanding of autocorrelation within the time series but struggles with forecasting volatility spikes, which are crucial for VaR estimation. The LSTM model, with its advanced memory cells, holds promise in identifying complex patterns within volatility data, yet its application to VaR prediction requires careful tuning to avoid overfitting.

A striking observation is the consistently high pass rates, exceeding 99.8% across different volatility points, suggesting the models perform well against historical data. While indicative of robust model performance, such high pass rates raise questions about overfitting and the challenge levels of the test conditions.

The deployment of VaR in a risk management context is essential because it helps institutions determine the capital needed to cushion against potential losses. This study's findings suggest that although each model brings unique advantages to the table, their selection must be strategically aligned with the specific risk management objectives of an institution. This comparative analysis of VaR predictions highlights the critical nature of choosing the right model in risk management, emphasizing the need for financial institutions to consider not only a model's statistical accuracy but also its performance during market extremes that can heavily influence VaR calculations.

5 Discussion

5.1 Interpretation of Findings

The results demonstrate robust model performance, with high pass rates suggesting effective volatility forecasting. This result may point to a successful capture of market dynamics by the models. However, the possibility of overfitting requires attention, and further out-of-sample testing is necessary to confirm model generalizability and predictive power. This interpretation of the results highlights several key points regarding model performance in volatility forecasting, including robust model performance, effective volatility forecasting, potential overfitting concerns, the need for out-of-sample testing, as well as a need for confirmation of model generalizability.

First, the findings indicate that the models used in the study exhibit robust performance. The high pass rates suggest that the models are effective in forecasting volatility. A high pass rate implies that the models are successfully capturing the underlying market dynamics related to Brent Crude Oil futures. This is a positive outcome because it indicates that the models are able to provide accurate forecasts of volatility levels. In addition, the effective volatility forecasting demonstrated by the models is crucial for market participants, especially in the context of the oil futures market. Accurate volatility forecasts enable market participants to make informed decisions regarding risk management, trading strategies, and investment decisions. By effectively forecasting volatility, the models contribute to enhancing market efficiency and reducing uncertainty for market participants.

However, despite the robust model performance, the possibility of overfitting raises a valid concern. Overfitting occurs when a model learns the noise in the data rather than the underlying patterns, leading to inflated performance metrics on the training data but poor generalizability to new data. It is essential to address overfitting issues to ensure that the models provide reliable and accurate forecasts in real-world scenarios. This overfitting suggests that further out-of-sample testing is a critical step in validating the model's generalizability and predictive power. Out-of-sample testing involves evaluating the model's performance on data on which the model has not been trained, providing a more realistic assessment of how the model would perform in practical applications. By conducting out-of-sample testing, researchers can verify if the models maintain their forecasting accuracy when applied to unseen data, thereby enhancing the credibility and reliability of the forecasting results. This needs to confirm model generalizability through out-of-sample testing underscores the importance of ensuring that the models can perform well beyond the data used for training. Generalizability is essential for the practical application of volatility forecasting models in real-world settings, where the ability to make accurate predictions on new data is crucial for decision-making processes.

The interpretation of findings highlights the strengths of the models in providing effective volatility forecasts while also acknowledging the need to address potential overfitting issues through further testing. By conducting rigorous out-of-sample testing, researchers can validate the models' predictive power and ensure their applicability in real-world scenarios, ultimately enhancing the reliability and utility of the volatility forecasting models in the context of Brent Crude Oil futures.

5.2 Theoretical Implications

This study contributes to the body of knowledge on financial risk quantification by highlighting the efficacy of GARCH and ARIMA models in volatility prediction. It underscores the importance of accounting for volatility clustering in financial time series and provides evidence supporting the mean-reversion property characteristic of GARCH models. The LSTM model's performance indicates the potential of machine learning approaches in capturing complex patterns, although careful calibration is needed. The study’s results contribute several theoretical implications, including contributing to financial risk quantification, confirming the importance of volatility clustering, evidencing the mean-reversion property of GARCH models, and supporting the potential for machine learning approaches in volatility forecasting.

The study contributes significantly to the field of financial risk quantification by highlighting the efficacy of traditional models such as GARCH and ARIMA in volatility prediction. By demonstrating the effectiveness of these models in capturing and forecasting volatility in financial time-series data, the study enhances our understanding of risk management practices in the context of financial markets.

In addition, the study underscores the importance of accounting for volatility clustering in financial time-series analysis. Volatility clustering refers to the phenomenon where periods of high volatility tend to cluster together, leading to non-random patterns in volatility levels. By acknowledging and incorporating volatility clustering into the modelling process, the study provides insights into the dynamics of financial markets and the behaviour of asset prices, as discussed in the empirical analysis.

The study also provides evidence supporting the mean-reversion property characteristic of GARCH models. The mean-reversion property suggests that extreme changes in volatility are likely to be followed by periods of lower volatility, indicating a tendency for volatility to revert to its long-term average. By highlighting this property in the context of GARCH models, the study contributes to our understanding of volatility dynamics and forecasting accuracy, as discussed in the model development and calibration sections.

Finally, the performance of the LSTM model indicates the potential of machine learning approaches in capturing complex patterns in financial time-series data. LSTM models, known for their ability to capture long-term dependencies and intricate patterns, offer a promising avenue for improving volatility forecasting accuracy. However, the study emphasizes the need for careful calibration of machine learning models to prevent overfitting and to ensure reliable predictions, as discussed in the LSTM model section.

The theoretical implications of the study highlight the contributions to financial risk quantification, the importance of volatility clustering, the mean-reversion property of GARCH models, and the potential of machine learning approaches in capturing complex patterns. By addressing these theoretical aspects, the study advances our knowledge of volatility prediction and risk management in financial markets, offering valuable insights for researchers and practitioners in the field.

5.3 Practical Implications for Risk Management

The practical application of VaR in risk management is significant. The models evaluated offer varying degrees of risk quantification, essential for capital allocation to cover potential losses. The findings imply that while each model has strengths, their selection must be tailored to the risk management objectives of financial institutions, factoring in their behaviour during extreme market conditions. This study provides several practical implications of the results for risk management, including underscoring the significance of VaR in risk management, illustrating the varying degrees to which different modelling approaches can quantify risk and, therefore, the critical importance of tailoring model selection to risk management objectives. In addition, the results highlight the need to consider the variable behaviour of models when presented with extreme market conditions, which also contributes to the need to carefully select models.

First, the practical application of VaR in risk management is crucial for financial institutions because it provides a quantitative measure to assess and manage potential losses in a portfolio over a defined period within a given confidence interval. By utilizing VaR models, institutions can effectively allocate capital to cover potential risks and ensure adequate risk mitigation strategies are in place, as discussed in the VaR conceptualization section. The models evaluated in the study offer varying degrees of risk quantification, reflecting their diverse approaches to estimating and forecasting potential losses. Each model provides unique insights into the level of risk exposure within a portfolio, allowing institutions to make informed decisions regarding capital allocation and risk management strategies, as discussed in the empirical analysis.

The findings suggest that although each model has its strengths and capabilities in quantifying risk, the selection of a specific model should be tailored to the risk management objectives of financial institutions. Different models may excel in capturing certain aspects of risk or market behaviour, and their selection should align with the institution's specific risk management goals and preferences, as discussed in the LSTM model section. An essential consideration in model selection is the behaviour of the models during extreme market conditions. Financial institutions must assess how each model performs under stress scenarios or during periods of heightened volatility to ensure that the selected model remains robust and reliable in adverse market conditions. Understanding how the models behave under extreme circumstances is crucial for effective risk management and decision-making, as discussed in the empirical analysis. For all the above reasons, selecting the most appropriate model for risk management purposes is paramount and includes considering the institution's risk tolerance, objectives, and the specific characteristics of the financial instruments or portfolios being analysed. By carefully evaluating and selecting the best model for the specific situation, institutions can enhance their risk management practices, improve capital allocation decisions, and better prepare for potential market uncertainties, as discussed in the theoretical implications section.

The practical implications for risk management highlighted in the study underscore the significance of VaR models in quantifying financial risk, the varying degrees of risk quantification offered by different models, the importance of tailoring model selection to risk management objectives, and the need to consider model behaviour during extreme market conditions. By addressing these practical implications, the study provides valuable insights for financial institutions seeking to enhance their risk management practices and optimize capital allocation strategies in dynamic market environments.

5.4 Key Contributions of the Study

This thesis makes several key contributions to the field by providing a comparative analysis of various volatility forecasting models. It goes beyond traditional statistical methods, integrating machine learning algorithms for enhanced predictive performance. The comparative analysis showcases the trade-offs between model complexity and forecasting accuracy, thereby aiding the selection of suitable models for practical financial applications.

Two key contributions of the study are integrating of machine learning algorithms for enhanced predictive performance in volatility forecasting and showcasing the trade-offs between model complexity and accuracy. By leveraging the capabilities of machine learning, the study explores new avenues for improving forecasting accuracy and capturing complex patterns in financial time-series data, as discussed in the VaR conceptualization section. Further, the comparative analysis conducted in this study showcases the trade-offs between model complexity and forecasting accuracy. By juxtaposing the performance of different models, the study provides insights into how more intricate models may offer improved predictive performance but at the cost of increased complexity. This analysis aids in guiding the selection of suitable models for practical financial applications, balancing the need for accuracy with the practical considerations of model complexity, as discussed in the analytical steps section.

The comparative analysis also serves as a guide for selecting appropriate volatility forecasting models in real-world financial applications. By weighing the benefits and drawbacks of different models, financial practitioners can make informed decisions about which model best aligns with their risk management objectives and preferences. This guidance helps in navigating the complexities of model selection and ensures that the chosen model strikes a balance between complexity and predictive performance, as discussed in the Value-at-Risk forecasting section. In addition, by providing a detailed examination of different models and their performance characteristics, the study equips practitioners with valuable insights for improving risk assessment, decision-making, and overall risk management strategies in dynamic market environments, as discussed in the practical implications for risk management section.

These key contributions underscore the advancements made in volatility forecasting through the integration of machine learning algorithms, the exploration of trade-offs between model complexity and accuracy, and the guidance provided for selecting suitable models for practical financial applications. By expanding upon traditional approaches and offering a nuanced perspective on model selection, the study contributes to the ongoing evolution of risk management practices in the financial industry.

6. Conclusion and Future Work

6.1 Major Conclusions

The study's major conclusions are drawn from a thorough examination of various models' performance in volatility forecasting and VaR prediction. The GARCH and ARIMA models showed consistency with established financial theories on volatility, and the LSTM model's results highlighted the promise of machine learning in financial analysis. These conclusions are foundational for both theoretical exploration and practical application in financial risk management.

Importantly, the GARCH (generalized autoregressive conditional heteroskedasticity) and ARIMA (autoregressive integrated moving average) models demonstrated consistency with established financial theories on volatility. GARCH modelling outcomes aligned with the theoretical understanding of how financial markets exhibit volatility patterns over time and were consistent with GARCH models' known ability to capture volatility clustering and persistence. On the other hand, ARIMA models, focusing on time-series analysis and forecasting, provided insights into the linear dependencies and trends in financial data, further supporting traditional financial theories on market dynamics.

Additionally, the results from the long short-term memory (LSTM) model highlighted the potential of machine learning techniques in financial analysis, particularly in volatility forecasting and risk management. LSTM models, with their ability to capture complex patterns and long-term dependencies in sequential data, showcased promising outcomes in predicting volatility and VaR in financial markets. The utilization of machine learning algorithms like LSTM signifies a shift towards advanced computational methods that can enhance forecasting accuracy and adaptability to changing market conditions.

Finally, the conclusions drawn from the study serve as foundational insights for both theoretical exploration and practical application in financial risk management. By validating the performance of traditional models like GARCH and ARIMA alongside innovative approaches like LSTM, the study bridges the gap between theoretical frameworks and practical tools for risk assessment and decision-making in financial markets. These insights provide a roadmap for researchers and practitioners to leverage a combination of established theories and cutting-edge technologies to navigate the complexities of financial risk management effectively.

These major conclusions underscore the importance of integrating traditional financial theories with modern computational techniques, highlighting the synergies between established models like GARCH and ARIMA and emerging methodologies such as LSTM in enhancing volatility forecasting and VaR prediction. These insights not only contribute to advancing theoretical understanding but also offer practical implications for improving risk management practices in the dynamic landscape of financial markets.

6.2 Contributions to the Field

This study contributes to the field in several significant ways, such as by providing empirical evidence of the predictive capabilities of traditional and advanced models under different market conditions, bridging the gap between econometric models and machine learning, and offering a nuanced understanding of their respective merits and demerits in risk assessment.

First, the study provides empirical evidence through rigorous analysis and evaluation of the predictive capabilities of traditional models like GARCH and ARIMA, as well as advanced models like LSTM, in forecasting volatility and predicting VaR. By conducting empirical tests under different market conditions and scenarios, the research offers insights into how these models perform in real-world settings, thereby enhancing the understanding of their effectiveness in risk assessment and management.

A second key contribution of the study is bridging the gap between traditional econometric models and modern machine learning techniques in the context of financial risk assessment. By comparing the merits and demerits of econometric models (such as GARCH and ARIMA) with machine learning models (like LSTM), the research provides a nuanced understanding of the strengths and limitations of each approach in capturing volatility dynamics and predicting risk.

Third, through a detailed analysis of model performance under different market conditions, the study offers a nuanced understanding of how traditional and advanced models respond to varying levels of market volatility, trends, and shocks. By highlighting the predictive strengths and weaknesses of each model type, the research contributes to a more informed decision-making process for practitioners and researchers in selecting the most suitable model for specific risk assessment tasks.

Finally, research contributions to the field extend to enhancing risk assessment practices by shedding light on the comparative effectiveness of different modelling approaches. By providing insights into the predictive capabilities of traditional and advanced models, the study equips financial professionals with valuable information to improve their risk assessment strategies and make more informed decisions in volatile market environments.

The study's primary contributions to the field lie in its empirical validation of model performance, its bridging of the gap between econometric and machine learning models, and its nuanced understanding of the strengths and limitations of different modelling approaches in risk assessment. By offering insights into the predictive capabilities of various models under different market conditions, this research enriches the field of financial risk management and guides future advancements in modelling techniques.

6.3 Limitations of the Study

Despite its insights, the study is not without limitations. First, it may suffer from biases inherent in historical data used for model training and evaluation. Historical data may be influenced by specific market conditions, trends, or anomalies that could introduce biases into the models' performance and predictions. Biases in historical data may also affect the generalizability of the findings to future market conditions, potentially leading to overfitting or underestimation of risks in real-world scenarios.

Second, the scope of market conditions examined may not cover the full spectrum of potential financial crises or extreme events. The study's focus on specific time periods or market environments could limit the models' ability to capture the complexities of unforeseen market shocks or systemic risks. This narrow scope of included market conditions may restrict the models' predictive power in scenarios that deviate significantly from the historical data patterns.

Third, the models' performance in predicting future volatility may not account for all exogenous shocks or black swan events, which can disrupt financial markets. Exogenous shocks, such as geopolitical events, natural disasters, or unexpected policy changes, can introduce sudden and extreme volatility that may not be adequately captured by the models. Similarly, black swan events, characterized by their rarity and high impact, pose challenges for traditional forecasting models that rely on historical data patterns and may lead to underestimation of tail risks.

Finally, the study's findings and conclusions may have limitations in generalizability beyond the specific dataset and market conditions analysed. Extrapolating the results to different asset classes, time periods, or market regimes may require additional validation and sensitivity analysis to assess the models' robustness and reliability in diverse contexts. These limitations in generalizability could affect the applicability of the study's insights to real-world risk management practices across various financial markets and conditions.

It is important to consider limitations related to biases in historical data, the scope of market conditions examined, the potential impact of exogenous shocks and black swan events, and the generalizability of the findings when considering the results of this study. Addressing these limitations can enhance the study's credibility and applicability in informing risk management strategies in dynamic financial environments.

6.4 Recommendations for Future Research

There are several important recommendations for future research that are informed by the results of this study. First, it is recommended to explore hybrid models that combine the strengths of various approaches presented in this study. Future research should focus on developing and evaluating hybrid models that integrate the strengths of different modelling approaches presented in the study. By combining traditional econometric models like GARCH and ARIMA with advanced machine learning techniques such as LSTM or neural networks, researchers can leverage the complementary advantages of each approach to improve forecasting accuracy and risk assessment capabilities. Further, hybrid models have the potential to capture both linear dependencies and complex patterns in financial time series data, offering more robust and reliable predictions in volatile market conditions.

Further investigation into the application of machine learning for real-time data analysis and the inclusion of alternative data sources could also be advantageous and is recommended to enhance the timeliness and responsiveness of volatility forecasting models. By leveraging machine learning techniques that can process and analyse data streams in real-time, researchers can develop models that adapt quickly to changing market dynamics and provide up-to-date risk assessments for decision-making purposes. In addition, real-time data analysis using machine learning can improve the models' ability to capture sudden shifts in volatility and respond effectively to market uncertainties.

Future research should also consider incorporating alternative data sources, such as social media sentiment, satellite imagery, or macroeconomic indicators, into volatility forecasting models to enhance their predictive power. By integrating a diverse range of data sources beyond traditional financial data, researchers can capture additional insights into market sentiment, external factors influencing volatility, and emerging trends that may affect risk management strategies. Importantly, the inclusion of alternative data sources can enrich the models' feature set and improve their ability to forecast volatility under complex and interconnected market conditions.

Finally, examining the models' performance across different asset classes could provide a broader understanding of their utility in risk management. Conducting comparative studies to evaluate the performance of volatility forecasting models across different asset classes, including equities, commodities, currencies, and derivatives is recommended. Assessing the models' effectiveness and robustness across these diverse asset classes can provide a broader understanding of their utility in risk management across various financial markets and investment instruments. By examining how models perform in different asset classes, researchers can identify strengths, weaknesses, and potential areas for model refinement to enhance risk assessment practices in a multi-asset portfolio context.

In the future, research efforts should focus on exploring hybrid modelling approaches, leveraging machine learning for real-time data analysis, incorporating alternative data sources, and evaluating model performance across various asset classes to advance the field of volatility forecasting and enhance risk management practices in dynamic financial environments. By addressing these recommendations, researchers can contribute to the development of more sophisticated and adaptive models that better capture the complexities of market dynamics and support informed decision-making in risk management.

6.5 Final Conclusion

In conclusion, this thesis provides a comparative framework that underscores the importance of model selection in financial volatility forecasting. This work sets the stage for subsequent studies to refine these models further and explore their applications in increasingly complex financial markets. The recommendations for future research highlight a path forward for continuing advancements in the discipline.

References

Andersen, T. G., & Bollerslev, T. (1998a). Answering the skeptics : Yes, standard volatility models do provide accurate forecasts. *International Economic Review*, *39*(4), Symposium on Forecasting and Empirical Methods in Macroeconomics and Finance (Nov 1998), 885–905. https://doi.org/10.2307/2527343

Andersen, T. G., & Bollerslev, T. (1998b). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31*(3), 307–327. https://ideas.repec.org/a/eee/econom/v31y1986i3p307-327.html327

Andersen, T. G., Bollerslev, T., & Diebold, F. X. (2007). Roughing it up : Including jump components in the measurement , modeling , and forecasting of return volatility. *The MIT Press*, *89*(4), 701–720. https://doi.org/10.1162/rest.89.4.701

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Ebens, H. (2001). The distribution of realized stock return volatility. *Journal of Financial Economics*, *61*(1), 43–76. https://doi.org/10.1016/S0304-405X(01)00055-1

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, *71*(2), 579–625. https://doi.org/10.1111/1468-0262.00418

Barberis, N., Shleifer, A., & Vishny, R. W. (2005). A model of investor sentiment. *Advances in Behavioral Finance*, *2*, 423–459. https://doi.org/10.1093/0198292279.003.0005

Barndorff-Nielsen, O. E., & Shephard, N. (2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology), 64*(2), 253–280. https://www.jstor.org/stable/3088799

Barndorff-Nielsen, O. E., & Shephard, N. (2004). Realized volatility and bipower variation: A comparison. *Journal of Financial Econometrics, 2*(1), 1–37. doi.org/10.1093/jjfinec/nbh001

Baruník, J., & Křehlík, T. (2016). Combining high frequency data with non-linear models for forecasting energy market volatility. *Expert Systems with Applications, 55*, 222–242.

https://doi.org/10.1016/j.eswa.2016.02.008

Baruník, J., & Křehlík, T. (2018). Measuring the frequency dynamics of financial connectedness and systemic risk. *Journal of Financial Econometrics, 16*(2), 271–296. https://doi.org/10.48550/arXiv.1507.01729

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*, 281–305. https://dl.acm.org/doi/10.5555/2188385.2188395

Bessembinder, H., & Maxwell, W. (2008). Transparency and the corporate bond market. *Journal of Economic Perspectives, 22*(2), 217–234. https://doi.org/10.1257/jep.22.2.217

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bollerslev, T., Chou, R., & Kroner, K. F. (1992). ARCH modeling in finance: A review of the theory and empirical evidence. *Journal of Econometrics, 52*(1/2), 5–59. https://EconPapers.repec.org/RePEc:eee:econom:v:52:y:1992:i:1-2:p:5-59

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research, 33*(2), 261–304. https://doi.org/10.1177/0049124104268644

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). *The econometrics of financial markets*. Princeton University Press.

Chen, W., Ma, F., Wei, Y., & Liu, J. (2020). Forecasting oil price volatility using high-frequency data: New evidence. *International Review of Economics and Finance*, *66*(November 2019), 1–12. https://doi.org/10.1016/j.iref.2019.10.014

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics, 7*(2), 174–196. https://doi.org/10.1093/jjfinec/nbp001

Dacorogna, M., Gençay, R., Muller, U., Pictet, O., & Olsen, R. (2001). *An introduction to high-frequency finance*. Academic Press.

Dinev, T., & Hart, P. (2004). Internet privacy concerns and their antecedents - Measurement validity and a regression model. *Behaviour and Information Technology*, *23*(6), 413–422. https://doi.org/10.1080/01449290410001715723

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems, 9*, 155–161. https://api.semanticscholar.org/CorpusID:743542

Du, X., Tang, Z., & Chen, K. (2023). A novel crude oil futures trading strategy based on volume-price time-frequency decomposition with ensemble deep reinforcement learning. *Energy*, *285*, 129394. https://doi.org/10.1016/j.energy.2023.129394

Ederington, L. H., & Lee, J. H. (1993). How Markets process information: News releases and volatility. *The Journal of Finance, 48*(4), 1161–1191. https://doi.org/10.2307/2329034 1191

Engle, R. F. (2001). GARCH 101: The use of ARCH/GARCH models in applied econometrics. *Journal of Economic Perspectives, 15*(4), 157–168. https://www.doi.org/10.1257/jep.15.4.157

Engle, R. (2002). Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business and Economic Statistics*, *20*(3), 339–350. https://doi.org/10.1198/073500102288618487

Ewald, C., Hadina, J., Haugom, E., Lien, G., Størdal, S., & Yahya, M. (2023). Sample frequency robustness and accuracy in forecasting Value-at-Risk for Brent Crude Oil futures. *Finance Research Letters*, *58*(May), 103916. https://doi.org/10.1016/j.frl.2023.103916

Fallon, W. (1996). Calculating Value-at-Risk. In A. M. Santomero (ed.), *Working Paper Series, Wharton Financial Institutions Center's conference on Risk Management in Banking, October 13-15, 1996*. Wharton School, University of Pennsylvania, 96-49.

Fan, J., Nunn, M. E., & Su, X. (2009). Multivariate exponential survival trees and their application to tooth prognosis. *Computational Statistics and Data Analysis*, *53*(4), 1110–

1121. https://doi.org/10.1016/j.csda.2008.10.019

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3*(Mar), 1289–1305. https://dl.acm.org/doi/10.5555/944919.944974

Gencer, H. G., & Demiralay, S. (2016). Volatility modeling and Value-at-Risk (VaR) forecasting of emerging stock markets in the presence of long memory, asymmetry, and skewed heavy tails. *Emerging Markets Finance and Trade*, *52*(3), 639–657. https://doi.org/10.1080/1540496X.2014.998557

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182. https://dl.acm.org/doi/10.5555/944919.944968

Hansen, P. R., & Lunde, A. (2006). Consistent ranking of volatility models*. Journal of Econometrics, 131*(1–2), 97–121. https://doi.org/10.1016/j.jeconom.2005.01.005

Hasbrouck, J. (2007). *Empirical market microstructure: The institutions, economics, and econometrics of securities trading*. Oxford University Press.

Haugom, E., Langeland, H., Molnár, P., & Westgaard, S. (2014). Forecasting volatility of the U.S. oil market. *Journal of Banking and Finance*, *47*(1), 1–14. https://doi.org/10.1016/j.jbankfin.2014.05.026

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82–97. https://www.doi.org/10.1109/MSP.2012.2205597

Hoerl, A. E., & Kennard, R. W. (1970). Quality ridge regression : Biased estimation for nonorthogonal problem*s. Technometrics, 12*(1), 55–67. http*://www.jstor.org/stable/1267351*

Huang, Z., Liu, H., & Wang, T. (2016). Modeling long memory volatility using realized measures of volatility: A realized HAR GARCH model. *Economic Modelling, 52*(B), 812–

821. https://doi.org/10.1016/j.econmod.2015.10.018

Hull, J. C. (2012). *Risk management and financial institutions*. Wiley.

Jarboui, A., & Mnif, E. (2023). Can clean energy stocks predict crude oil markets using hybrid and advanced machine learning models? *Asia-Pacific Financial Markets*, *0123456789*. https://doi.org/10.1007/s10690-023-09432-9

Jorion, P. (2006). *Value at risk: The new benchmark for managing financial risk*. McGraw-Hill.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 263–292. http://www.jstor.org/stable/1914185

Kambouroudis, D. S., McMillan, D. G., & Tsakou, K. (2016). Forecasting Stock return volatility: A comparison of GARCH, implied volatility, and realized volatility models. *Journal of Futures Markets*, *36*(12), 1127–1163. https://doi.org/10.1002/fut.21783

Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University - Computer and Information Sciences, 34*(4), 1060–1073. https://doi.org/10.1016/j.jksuci.2019.06.012

Kuhn, M., & Johnson, K. (2013). *Applied predictive modelling*. Springer.

Laopodis, N. T. (2021). *Financial economics and econometrics*. Routledge.

Lo, A. W., & MacKinlay, A. C. (1999). *A non-random walk down Wall Street*. Princeton University Press.

Louzis, D. P., Xanthopoulos-Sisinis, S., & Refenes, A. P. (2014). Realized volatility models and alternative Value-at-Risk prediction strategies. *Economic Modelling*, *40*, 101–116. https://doi.org/10.1016/j.econmod.2014.03.025

Mehta, N. J., & Yang, F. (2022). Portfolio optimization for extreme risks with maximum diversification: An empirical analysis. *Risks, 10*(5), 101. https://doi.org/10.3390/risks10050101

Medvedev, A., & Medvedev, A. (2023). Forecasting financial markets using advanced machine learning algorithms. *E3S Web of Conferences*, *403*(2023), 08007. https://doi.org/10.1051/e3sconf/202340308007

Monfared, S. A., & Enke, D. (2014). Volatility forecasting using a hybrid GJR-GARCH neural network model. *Procedia Computer Science*, *36*(C), 246–253. https://doi.org/10.1016/j.procs.2014.09.087

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis.* John Wiley & Sons

Oladipupo, M. A., Obuzor, P. C., Bamgbade, B. J., Olagunju, K. M., Adeniyi, A. E., & Ajagbe, S. A. (2023). An automated python script for data cleaning and labeling using machine learning technique. *Informatica (Slovenia)*, *47*(6), 219–232. https://doi.org/10.31449/inf.v47i6.4474

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, *160*(1), 246–256. https://doi.org/10.1016/j.jeconom.2010.03.034

Peters, E. E. (1996). *Chaos and order in the capital markets: A new view of cycles, prices, and market volatility*. John Wiley & Sons.

Poon, S.-H., & Granger, C. W. J. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic Literature, 41*(2), 478–539. http://www.jstor.org/stable/3216966

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors*. Nature*, *323*, 533–536. https://doi.org/10.1038/323533a0

Shiller, R. J. (1992). *Market volatility*. MIT press.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Taylor, S. J. (2005). *Asset Price Dynamics, Volatility, and Prediction*. Princeton University Press.

Thomakos, D. D., & Wang, T. (2003). Realized volatility in the futures markets. *Journal of Empirical Finance*, *10*(3), 321–353. https://doi.org/10.1016/S0927-5398(02)00052-X

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso*. Journal of the Royal Statistical Society. Series B (Methodological), 58*(1)*,* 267–288. https://www.jstor.org/stable/2346178

Tsay, R. S. (2010). *Analysis of financial time series* (3rd ed.). Wiley.

Virgilio, G. P. M. (2019). High-frequency trading: A literature review. *Financial Markets and Portfolio Management, 33*, 183–208. https://doi.org/10.1007/s11408-019-00331-6

Wang, J. L., & Chan, S. H. (2007). Stock market trading rule discovery using pattern recognition and technical analysis. *Expert Systems with Applications, 33*(2), 304–315. https://doi.org/10.1016/j.eswa.2006.05.002

Weeks, M. (2002). [Review of the Book *Introductory Econometrics: A Modern Approach*, by J. M. Wooldridge, South-Western College Publishing, 2000, 1-538-85013-2, 824]. *Journal of Applied Econometrics*, *17*(2), 191–193. https://doi.org/10.1002/jae.665

Wipplinger, E. (2007). Philippe Jorion: Value at Risk – The New Benchmark for Managing Financial Risk. [Review of the Book *Value at Risk – The New Benchmark for Managing Financial Risk* (3rd ed.) by P. Jorion, McGraw Hill, 2007). *Financial Markets and Portfolio Management*, *21*(3), 397–398. https://doi.org/10.1007/s11408-007-0057-3

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175. https://doi.org/10.1016/S0925-2312(01)00702-0

I acknowledge the use of ChatGPT (OpenAI. [2023]. ChatGPT [Mar 14 version] [Large language model]. Retrieved from https://chat.openai.com/chat) for the following purposes: 1. development of initial code snippets, 2. searching for applicable reference documents. I verify that all information and text retrieved or generated with ChatGPT was carefully verified and edited prior to submission. I am responsible for the accuracy and the quality of the final content herein.

Appendices

Appendix 1. Python Programming Code Snippet

```
!pip install arch

!pip install --upgrade arch

!pip install pandas openpyxl

!pip install lime

!pip install shap


import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import statsmodels.api as sm

from arch import arch_model

from sklearn.model_selection import train_test_split

from sklearn.neural_network import MLPRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import SVR

from sklearn.metrics import mean_squared_error

import lime

import shap

from statsmodels.tsa.arima.model import ARIMA

from arch import arch_model


# Load high-frequency trading data


trading_data = pd.read_csv('RV_all.csv')


trading_data[:4000].to_csv('scat.csv',index=False)

trading_data1 = pd.read_csv('RV_all.csv')

trading_data.head()

trading_data1
```

```
trading_data.info()
trading_data.isna().sum()
trading_data = trading_data.dropna()
trading_data.drop(['post', 'at'], axis=1, inplace=True)
trading_data.isna().sum()
trading_data.describe()

# Extracting 'trad_date' and 'pt_' columns
time_series_data = trading_data[['trad_date', 'RVOL_pt']]

# Setting 'trad_date' as the index
time_series_data.set_index('trad_date', inplace=True)

time_series_data.index = pd.to_datetime(time_series_data.index)

# Define the years you want to include in the plot
years_of_interest = [2006, 2007, 2008, 2009, 2010, 2011,
                     2012, 2013, 2014, 2015, 2016]

# Filter the DataFrame to include data only for the specified years
filtered_data = time_series_data[time_series_data.index.year.isin(years_of_interest)]

# Plotting the time series for the specified years
plt.figure(figsize=(12, 6))
plt.plot(filtered_data.index, filtered_data['RVOL_pt'], color='blue', linewidth=1)
plt.title('Time Series of pt_ Column (Years: 2004-2021)')
plt.xlabel('Date')
plt.ylabel('pt_ Value')
plt.grid(True)
plt.show()

#!-----Feature Extraction and Pre-processing-----!
#Step 1: Calculating log returns
```

```
#Step 2: Converting 'trad_date' to datetime format


#Log returns
trading_data['log_returns'] = np.log(trading_data['RVOL_pt'] / trading_data['RVOL_pt'].shift(1))


# trad_date to date_time format
trading_data['trad_date'] = pd.to_datetime(trading_data['trad_date'])


trading_data['log_returns']


#Step 3: Drop NaN values in the dataset


trading_data.dropna(inplace=True)
trading_data.isna().sum()
trading_data['log_returns'].value_counts()


# Calculate daily average prices - doing it just to check calculation


#Saving original daataset into another dataset for calculating pt mean
trading_data_pt = trading_data.copy()
# Setting 'trad_date' as the index
trading_data_pt.set_index('trad_date', inplace=True)
#Calculating pt mean
daily_average_prices = trading_data_pt['RVOL_pt'].resample('1D').mean()



# Drop any rows with missing values
daily_average_prices.dropna(inplace=True)


daily_average_prices.isna().sum()


daily_average_prices
```

```
#Step 4: Define sampling frequency.
#Step 5: Grouping 'trad_date' and calculate daily statistics.
#Also Calculating Mean daily prices, closing prices and opening prices

M_D = 1440  # Number of minutes in a trading day

#
daily_stats = trading_data.groupby(pd.Grouper(key='trad_date', freq='D')).agg(
   realized_volatility=('log_returns', lambda x: np.sqrt(M_D * np.sum(np.square(x)))),
   mean_daily_price=('RVOL_pt', 'mean'),
   daily_opening_price = ('RVOL_pt', 'min'),
   daily_close_price = ('RVOL_pt', 'max')
)

# Displaying the resulting DataFrame
print(daily_stats[:5])

import matplotlib.pyplot as plt

# Ensure your 'trad_date' is the index or use it as the x-axis explicitly
dates = daily_stats.index  # Adjust if 'trad_date' is not the index

# Realized Volatility
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['realized_volatility'], label='Realized Volatility')
plt.title('Realized Volatility Over Time')
plt.xlabel('Date')
plt.ylabel('Volatility')
plt.legend()
plt.show()

# Mean Daily Price
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['mean_daily_price'], label='Mean Daily Price', color='orange')
plt.title('Mean Daily Price Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Daily Opening and Closing Prices
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['daily_opening_price'], label='Opening Price', color='green')
plt.plot(dates, daily_stats['daily_close_price'], label='Closing Price', color='red')
plt.title('Daily Opening and Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

#Dropping null values in daily stats
daily_stats.dropna(axis=0, inplace=True)
daily_stats[:5]

daily_stats.isna().sum()
len(daily_stats)

#calculating realized Volatility for 5M, 30M, 1H and 90M.

M_5 = 5  # Number of minutes

daily_stats['5m_volatility'] = trading_data.groupby(pd.Grouper(key='trad_date', freq='5T')).agg(
    realized_volatility=('log_returns', lambda x: np.sqrt(M_5 * np.sum(np.square(x))))
)

import matplotlib.pyplot as plt
```

```
# Ensure your 'trad_date' is the index or use it as the x-axis explicitly
dates = daily_stats.index  # Adjust if 'trad_date' is not the index
# Daily Opening and Closing Prices
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['5m_volatility'], label='5m_volatility', color='green')
plt.plot(dates, daily_stats['5m_volatility'], label='5m_volatility', color='red')
plt.title('Daily Opening and Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

daily_stats

M_30 = 30  # Number of minutes

daily_stats['30m_volatility'] = trading_data.groupby(pd.Grouper(key='trad_date', freq='30T')).agg(
    realized_volatility=('log_returns', lambda x: np.sqrt(M_30 * np.sum(np.square(x))))
)

import matplotlib.pyplot as plt

# Ensure your 'trad_date' is the index or use it as the x-axis explicitly
dates = daily_stats.index  # Adjust if 'trad_date' is not the index

# Mean Daily Price
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['30m_volatility'], label='30m_volatility', color='orange')
plt.title('30m_volatility Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

daily_stats

M_1H = 60  # Number of minutes

daily_stats['1H_volatility'] = trading_data.groupby(pd.Grouper(key='trad_date', freq='1H')).agg(
    realized_volatility=('log_returns', lambda x: np.sqrt(M_1H * np.sum(np.square(x))))
)

daily_stats[500:530]

# Mean Daily Price
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['1H_volatility'], label='1H_volatility', color='orange')
plt.title('1H_volatility Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

M_90 = 90  # Number of minutes

daily_stats['90m_volatility'] = trading_data.groupby(pd.Grouper(key='trad_date', freq='90T')).agg(
    realized_volatility=('log_returns', lambda x: np.sqrt(M_90 * np.sum(np.square(x))))
)

daily_stats

# Mean Daily Price
plt.figure(figsize=(14, 7))
plt.plot(dates, daily_stats['90m_volatility'], label='90m_volatility', color='orange')
plt.title('90m_volatility Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

#!-----Statistical Models for volatility and trend analysis,
#including GARCH for time-varying volatility and ARIMA/HAR-RV for linear dependencies.-----!
#First Splitting our dataset into train test split

# Replacing NaN or inf values with a custom value (e.g., mean)
daily_stats.replace([np.inf, -np.inf], np.nan, inplace=True)
daily_stats.fillna(daily_stats.mean(), inplace=True)

daily_stats

# Extract relevant features for modeling
X = daily_stats.drop('daily_close_price', axis=1)
y = daily_stats['daily_close_price']

# # Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

#GARCH and ARIMA model

import pandas as pd
import numpy as np
from arch import arch_model
import statsmodels.api as sm



# Fit GARCH model
garch_model = arch_model(y_train, vol='GARCH', p=1, q=1)
garch_result = garch_model.fit(disp = 'off')
```

```
# Print GARCH summary
print(garch_result.summary())

# Fit ARIMA model (example)
arima_model = sm.tsa.ARIMA(y_train, order=(1, 0, 1))
arima_result = arima_model.fit()

# Print ARIMA summary
print(arima_result.summary())

#HAR-RV Model
#The Heterogeneous Autoregressive (HAR) model is typically used for forecasting volatility in financial time series data.
#The Autoregressive Volatility (AV) model is an extension of the HAR model that incorporates past realized volatilities
#at different time frequencies.

import pandas as pd
import numpy as np
import statsmodels.api as sm

# Defining the HAR-AV model
def fit_har_av(data):
    # Extracting the realized volatility features for 5min, 30min and 60min
    rv_5min = daily_stats['5m_volatility']
    rv_30min = daily_stats['30m_volatility']
    rv_60min = daily_stats['1H_volatility']
    #v_90min = daily_stats['90m_volatility']



    # Stacking the realized volatilities for different time frequencies
    X = np.column_stack((rv_5min, rv_30min, rv_60min))
    # Adding a constant term
    X = sm.add_constant(X)

    # Target variable (Actual RV)
    y = data['realized_volatility']

    # Fitting the HAR-AV model
    model = sm.OLS(y, X)
    results = model.fit()

    return results

## Computing HAR-AV results from 'it_har_av()'function
har_av_results = fit_har_av(daily_stats)

# Printing the summary of results
print(har_av_results.summary())

#The summary output indeed indicates that the model is an Ordinary Least Squares (OLS) regression model.
#This is because the HAR-AV model is essentially a linear regression model, which fits well with the OLS framework.
```

!-----ML Models-----!

Grid Search with Cross-Validation library for NN model:

This library will check the model's architecture and use each set of parameters to check which parameter set will give better accuracy/result for the model.

Steps to get the optimal parameters by using Grid Search CV:

First we will define parameter_grid which has a hidden layers, activation function, loss function and max iteration for our NN model

Then we have created NN model

Then we have used Grid Search CV to fit with the MLP_Regressor(NN) model to check and compute the best paramaters for the model

After fitting Grid Search CV with model, we will call .best_params_ method to get the best parameters computed by GS-CV.##

```
from sklearn.model_selection import GridSearchCV


# Defining the parameter grid
param_grid = {
    'hidden_layer_sizes': [(100,100), (100, 50), (50, 25)],
    'activation': ['relu', 'tanh'],
    'solver': ['adam', 'sgd'],
    'max_iter': [1000, 1500, 2000]
}


# Creating an MLPRegressor model
mlp_regressor = MLPRegressor()


# Using Grid search library with cross-validation.
grid_search = GridSearchCV(mlp_regressor, param_grid, cv=5, scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)


# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)


#Training the NN model with the best parameters


best_model = MLPRegressor(**best_params)
history = best_model.fit(X_train, y_train)


#Plotting the loss curve to visualise the loss of NN model at each epoch while training
plt.plot(history.loss_curve_)
plt.title('Loss Curve')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show()

#Random Forest Regression

# Ensemble Methods (Random Forest)
rf_regressor = RandomForestRegressor(n_estimators=100)
rf_regressor.fit(X_train, y_train)

#Support Vector regression model
# Support Vector Machines
svm_regressor = SVR(kernel='rbf')
svm_regressor.fit(X_train, y_train)

#!-----Evaluation Tools-----!
#MSE, MAE and R-Square metrices

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Making predictions on training and testing data
mlp_train_preds = best_model.predict(X_train)
mlp_test_preds = best_model.predict(X_test)

rf_train_preds = rf_regressor.predict(X_train)
rf_test_preds = rf_regressor.predict(X_test)

svm_train_preds = svm_regressor.predict(X_train)
svm_test_preds = svm_regressor.predict(X_test)

#Function to compute MSE, MAE, R-square metrcies to check performance of each model for training and tesing data
```

```
def evaluate_model(name, y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    mae = mean_absolute_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"{name} Metrics:")
    print(f"MSE: {mse:.2f}")
    print(f"MAE: {mae:.2f}")
    print(f"R-squared: {r2:.2f}")
    print()


#Computing Training and test MSE, MAE and R-square metrcies for each model
evaluate_model("MLP (Train)", y_train, mlp_train_preds)
evaluate_model("MLP (Test)", y_test, mlp_test_preds)

evaluate_model("Random Forest (Train)", y_train, rf_train_preds)
evaluate_model("Random Forest (Test)", y_test, rf_test_preds)

evaluate_model("SVM (Train)", y_train, svm_train_preds)
evaluate_model("SVM (Test)", y_test, svm_test_preds)

#Visualising computed loss function for each model.
import matplotlib.pyplot as plt
import numpy as np

# Function to compute MSE, MAE, R-square metrcies to check performance of each model for
training and testing data
def evaluate_model(name, y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    mae = mean_absolute_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    return mse, mae, r2
```

```
# Function to plot loss functions for a given model
def plot_loss_functions(model_name, train_metrics, test_metrics):
    labels = ['MSE', 'MAE', 'R-squared']
    x = np.arange(len(labels))  # the label locations
    width = 0.35  # the width of the bars

    fig, ax = plt.subplots(figsize=(8, 6))

    # Plotting the loss functions for the model
    ax.bar(x - width/2, train_metrics, width, label='Train', color='b', alpha=0.5)
    ax.bar(x + width/2, test_metrics, width, label='Test', color='g', alpha=0.5)

    # Adding some text for labels, title and custom x-axis tick labels, etc.
    ax.set_xlabel('Metrics')
    ax.set_ylabel('Value')
    ax.set_title(f'Loss Functions for {model_name}')
    ax.set_xticks(x)
    ax.set_xticklabels(labels)
    ax.legend()

    fig.tight_layout()

    # Saving the plot as an image file
    plt.savefig(f'{model_name}_loss_functions.png')

    plt.show()

# Computing Training and test MSE, MAE and R-square metrics for each model
mlp_train_mse, mlp_train_mae, mlp_train_r2 = evaluate_model("MLP (Train)", y_train,
mlp_train_preds)
mlp_test_mse, mlp_test_mae, mlp_test_r2 = evaluate_model("MLP (Test)", y_test, mlp_test_preds)
```

```
rf_train_mse, rf_train_mae, rf_train_r2 = evaluate_model("Random Forest (Train)", y_train, rf_train_preds)

rf_test_mse, rf_test_mae, rf_test_r2 = evaluate_model("Random Forest (Test)", y_test, rf_test_preds)


svm_train_mse, svm_train_mae, svm_train_r2 = evaluate_model("SVM (Train)", y_train, svm_train_preds)

svm_test_mse, svm_test_mae, svm_test_r2 = evaluate_model("SVM (Test)", y_test, svm_test_preds)


# Plotting loss functions for each model separately

plot_loss_functions("MLP", [mlp_train_mse, mlp_train_mae, mlp_train_r2], [mlp_test_mse, mlp_test_mae, mlp_test_r2])

plot_loss_functions("Random Forest", [rf_train_mse, rf_train_mae, rf_train_r2], [rf_test_mse, rf_test_mae, rf_test_r2])

plot_loss_functions("SVM", [svm_train_mse, svm_train_mae, svm_train_r2], [svm_test_mse, svm_test_mae, svm_test_r2])


#Charts - Visualisation of prediction, results

#Here scatter plot is used to visualise the prediction of each model. The reason for using scatter plot is

#that the scatter plot scatters each data with respect to another data of opposite column which we want to compare.

#This helps to see the comparison of each data to another.


# Visualizing results (e.g., predicted vs. actual values of trade)

plt.scatter(y_test, mlp_test_preds, label='MLP', alpha=0.5)

plt.scatter(y_test, rf_test_preds, label='Random Forest', alpha=0.5)

plt.scatter(y_test, svm_test_preds, label='SVM', alpha=0.5)

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Predicted vs. Actual Values')

plt.legend()

plt.show()


#!-----Explainable AI (XAI) Techniques-----!

X_train.values
```

```
X_test.iloc[0].values

#Lime
# Importing necessary libraries
import lime
import lime.lime_tabular
import shap
import matplotlib.pyplot as plt

# Ensuring consistent feature names
feature_names_train = X_train.columns
feature_names_test = X_test.columns

# LIME
explainer = lime.lime_tabular.LimeTabularExplainer(X_train.values, mode='regression',
feature_names=feature_names_train)
instance_idx = 3  # Index of the instance
explanation = explainer.explain_instance(X_test.iloc[instance_idx].values, rf_regressor.predict)
explanation.show_in_notebook()

#Shap
# SHAP
shap_explainer = shap.Explainer(rf_regressor, X_train)
shap_values = shap_explainer.shap_values(X_test)

shap.summary_plot(shap_values, X_test, plot_type="bar")
shap.summary_plot(shap_values, X_test)

#Kupeic test

daily_stats['mean_daily_price'][:-1]
daily_stats['mean_daily_price']
np.diff(daily_stats['mean_daily_price'])
```

```
import numpy as np

#  electing Confidence Level
confidence_level = 0.95  # Example: 95% confidence level

# Choosing Time Horizon
time_horizon = 1  #1 day

# Computing Returns (if not already computed)
returns = np.diff(daily_stats['mean_daily_price']) / daily_stats['mean_daily_price'][:-1]

#Estimating Volatility
volatility = np.std(returns)

# Calculating VaR
var_value = np.percentile(returns, 100 - confidence_level * 100)

# Interpreting results
print(f"Value at Risk (VaR) at {confidence_level * 100}% confidence level for {time_horizon} day(s): {var_value}")

y_test_values = y_test[:5]
y_pred_values = rf_test_preds[:5]

import numpy as np
from scipy.stats import chi2

def kupiec_test(actual_losses, predicted_losses, confidence_level=0.95):
    """
    Performing the Kupiec Test for VaR model accuracy.

    Parameters:
    actual_losses (array-like): Array of actual losses.
    predicted_losses (array-like): Array of predicted losses from the VaR model.
    confidence_level (float): Confidence level for the test, defaults to 0.95.

    Returns:
    float: The p-value of the Kupiec Test.
    """
    n = len(actual_losses)
    successes = sum((actual_losses[i] >= predicted_losses[i]) for i in range(n))
    failures = n - successes

    # Calculating log likelihood under the null hypothesis (binomial distribution)
    null_log_likelihood = successes * np.log(confidence_level) + failures * np.log(1 - confidence_level)

    # Calculating log likelihood under the alternative hypothesis
    p_hat = successes / n
    alternative_log_likelihood = successes * np.log(p_hat) + failures * np.log(1 - p_hat)

    # Calculating test statistic
    test_statistic = -2 * (null_log_likelihood - alternative_log_likelihood)

    # Calculating p-value using chi-square distribution
    p_value = 1 - chi2.cdf(test_statistic, 1)

    return p_value

# Example usage:
actual_losses = np.array(y_test_values)  # Actual losses
predicted_losses = np.array(y_pred_values)  # Predicted losses from VaR model

p_value = kupiec_test(actual_losses, predicted_losses)
print("Kupiec Test p-value:", p_value)

#collapse the dataset
```

```
import pandas as pd

# Read the CSV file into DataFrame
df = pd.read_csv('rv_use.csv')
# data = pd.read_csv('RV_all.csv')

# # Drop the last entry from the 'trad_date' column
# data['trad_date'] = data['trad_date'].iloc[:-1]

# # Assign the modified 'trad_date' column to the DataFrame
# df['trad_date'] = data['trad_date']

# Now 'trad_date' column in df has the modified values

df['trad_date']=df['dy'].astype(str).str.zfill(2) + df['mn'].astype(str).str.zfill(2) + df['yr'].astype(str)

df['trad_date']

df[['yr','mn','dy']]

import pandas as pd
import numpy as np
from datetime import datetime

df['trad_date']=df['dy'].astype(str).str.zfill(2) + df['mn'].astype(str).str.zfill(2) + df['yr'].astype(str)

df['trad_date']

df['trad_date'] = pd.to_datetime(df['trad_date'], format='%d%m%Y')



# Drop rows based on 'trad_date' conditions
```

```
df = df[df['trad_date'] >= '2006-01-01']
df = df[df['trad_date'] != '2008-05-15']

# Sort by 'trad_date' and reset index
df = df.sort_values('trad_date').reset_index(drop=True)

# Generate a time variable that increments by 1 starting from 1
df['time'] = np.arange(1, len(df) + 1)

# Set 'time' as the index (similar to tsset in Stata)
df.set_index('time', inplace=True)

# Generate daily return measures:

# Replace close prices that are 0 with the open price + intraday return on the new contract
df['LTD_ocr'] = np.log(df['ltd_closeprice'] / df['ltd_openprice'])
df.loc[df['closeprice'] == 0, 'closeprice'] = df['openprice'] * (1 + df['LTD_ocr'])

# 1. Open-Close returns
df['dr1'] = np.log(df['closeprice'] / df['openprice'])

# 2. Close-open (overnight) returns:
df['last_tday'] = df['dy'].where(df['LTD_ocr'].notnull(), np.nan)
df['dr2'] = np.log(df['openprice'] / df['closeprice'].shift(1))
df['LTD_co'] = np.log(df['openprice'] / df['ltd_closeprice'].shift(1))
df.loc[df['last_tday'].shift(1) == df['last_tday'], 'dr2'] = df['LTD_co']
df['dr2sq'] = df['dr2'] ** 2

# 3. Close-Close returns
df['dr3'] = np.log(df['closeprice'] / df['closeprice'].shift(1))
df['LTD_cc'] = np.log(df['closeprice'] / df['ltd_closeprice'].shift(1))
df.loc[df['last_tday'].shift(1) == df['last_tday'], 'dr3'] = df['LTD_cc']
```

```
# Generate HAR-variables:
# Generate HAR-variables:
# Generate HAR-variables:
for i in [1, 2, 3, 4, 5, 6, 9, 10, 12, 15, 18, 20, 27, 30, 36, 45, 54, 60, 90, 108]:
    df[f'd{i}'] = df[f'rvol_pt{i}'].shift(1)

    # Corrected line for weekly average (w)
    df[f'w{i}'] = df[f'rvol_pt{i}'].rolling(window=5).mean().shift(1)

    # Monthly average (m) using the last 20 observations
    df[f'm{i}'] = df[f'rvol_pt{i}'].rolling(window=20).mean().shift(1)

import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error

volatility_vars = df.filter(regex='^rvol_pt|^d[0-9]+|^w[0-9]+|^m[0-9]+$', axis=1)
descriptive_stats = volatility_vars.describe()
print(descriptive_stats)

import matplotlib.pyplot as plt
import seaborn as sns

# Selecting a subset of volatility variables for visualization
subset_vars = volatility_vars.columns[0:10]  # Adjust as needed for a broader overview

# Plotting histograms for the subset of volatility variables
plt.figure(figsize=(20, 10))
for i, var in enumerate(subset_vars, 1):
    plt.subplot(2, 5, i)  # Adjust the grid size based on the number of variables selected
    sns.histplot(volatility_vars[var].dropna(), kde=True, bins=20)
    plt.title(var)
plt.tight_layout()
plt.show()
# Plotting box plots for the subset of volatility variables
plt.figure(figsize=(20, 5))
sns.boxplot(data=volatility_vars[subset_vars].dropna(), orient="h", palette="Set2")
plt.title('Box Plot of Selected Volatility Variables')
plt.show()

import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Plotting function
def plot_har_variables(df, variable_prefix, title_prefix):
    # Filter out the columns that strictly match the variable prefix followed by an integer
    variable_columns = [col for col in df.columns if col.startswith(variable_prefix) and col[len(variable_prefix):].isdigit()]

    # Sort the columns based on the integer part of the column name
    periods = sorted(variable_columns, key=lambda x: int(x[len(variable_prefix):]))

    # Create subplots
    n_vars = len(periods)
    fig, axes = plt.subplots(n_vars, 1, figsize=(10, 2*n_vars), sharex=True)

    if n_vars == 1:  # If there's only one period, put it in a list for iteration
        axes = [axes]

    # Plot each HAR variable
    for ax, col_name in zip(axes, periods):
        ax.plot(df.index, df[col_name], label=f'{title_prefix} {col_name}')
        ax.set_title(f'{title_prefix} {col_name}')
        ax.legend()
        # Format x-axis as dates for each subplot
        ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
        ax.xaxis.set_major_locator(mdates.AutoDateLocator())
        ax.tick_params(axis='x', rotation=45)

    # Adjust spacing between subplots
    plt.tight_layout()

    # Show the plot
    plt.show()

df.set_index('trad_date', inplace=True)

# Plot daily, weekly, and monthly HAR variables
plot_har_variables(df, 'd', 'Daily lagged RVOL')
plot_har_variables(df, 'w', 'Weekly average RVOL')
plot_har_variables(df, 'm', 'Monthly average RVOL')

# Calculating the correlation matrix for the volatility variables
correlation_matrix = volatility_vars.corr()

# Generating a heatmap for the correlation matrix
plt.figure(figsize=(20, 15))
sns.heatmap(correlation_matrix, annot=False, cmap='coolwarm', linewidths=.5)
plt.title('Correlation Heatmap of Volatility Variables')
plt.show()

import matplotlib.pyplot as plt

rvol_pts = ['rvol_pt1', 'rvol_pt2', 'rvol_pt3', 'rvol_pt4', 'rvol_pt5', 'rvol_pt6', 'rvol_pt10', 'rvol_pt12',
'rvol_pt15', 'rvol_pt20', 'rvol_pt30', 'rvol_pt60', 'rvol_pt90', 'rvol_pt108']
sampling_frequencies = [1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 90, 108]  # frequencies
```

```
# Calculate the mean and standard deviation of realized volatility for each sampling frequency
mean_rvol = [df[pt].mean() for pt in rvol_pts]
std_rvol = [df[pt].std() for pt in rvol_pts]

# Create the plot
fig, ax1 = plt.subplots()

# Plot the realized volatility
ax1.plot(sampling_frequencies, mean_rvol, label='Realized Volatility', color='black', linestyle='-')
ax1.set_xlabel('Sampling frequency (min)')
ax1.set_ylabel('Realized Volatility', color='black')
ax1.tick_params(axis='y', labelcolor='black')

# Create a second y-axis to plot the standard deviation of realized volatility
ax2 = ax1.twinx()
ax2.plot(sampling_frequencies, std_rvol, label='Standard Deviation of Realized Volatility',
color='black', linestyle='--')
ax2.set_ylabel('Standard Deviation of Realized Volatility', color='black')
ax2.tick_params(axis='y', labelcolor='black')

# Add a legend and a title
fig.tight_layout()  # Tweak layout for space
plt.title('Realized Volatility and its Standard Deviation by Sampling Frequency')
plt.show()

# Apply forward fill to fill missing values
data_filled = df.fillna(method='ffill')

# apply backward fill for any remaining NaN values
data_filled = data_filled.fillna(method='bfill')

data =data_filled
```

```
import statsmodels.api as sm


X = data.drop(['rvol_pt', 'openprice', 'closeprice', 'ltd_openprice', 'ltd_closeprice', 'yr', 'mn', 'dy',  'LTD_ocr', 'dr1', 'last_tday', 'dr2', 'LTD_co', 'dr2sq', 'dr3', 'LTD_cc'], axis=1)  # Exclude non-volatility and identifier columns

y = data['rvol_pt']  # Target variable


def forward_selection(X, y, significance_level=0.05):

    initial_features = X.columns.tolist()

    best_features = []

    while len(initial_features) > 0:

        remaining_features = list(set(initial_features) - set(best_features))

        new_pval = pd.Series(index=remaining_features, dtype=float)

        for new_column in remaining_features:

            model = sm.OLS(y, sm.add_constant(X[best_features + [new_column]])).fit()

            new_pval[new_column] = model.pvalues[new_column]

        min_p_value = new_pval.min()

        if min_p_value < significance_level:

            best_features.append(new_pval.idxmin())

        else:

            break

    return best_features


selected_features = forward_selection(X.fillna(method='ffill'), y.fillna(method='ffill'))  # Forward fill to handle missing values

print("Selected features:", selected_features)


predictor_columns = ['rvol_pt' + str(i) for i in [1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60,90, 108]]

actual_returns_column = 'dr1'  # This is your actual returns


# Split into features and target

X = data[predictor_columns]

y = data[actual_returns_column]
```

```
# Split the dataset into train and test sets
train_size = int(len(X) * 0.8)
X_train, y_train = X.iloc[:train_size], y.iloc[:train_size]
X_test, y_test = X.iloc[train_size:], y.iloc[train_size:]

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from arch import arch_model
from statsmodels.tsa.arima.model import ARIMA
from keras.models import Sequential
from keras.layers import Dense, LSTM
import numpy as np
import matplotlib.pyplot as plt

# Defining the models
models = {
    "Linear Regression": LinearRegression(),
    "Ridge Regression": Ridge(alpha=1.0),
    "Random Forest Regressor": RandomForestRegressor(n_estimators=100, random_state=42),
    "Gradient Boosting Regressor": GradientBoostingRegressor(n_estimators=100, random_state=42),
    "Support Vector Regressor": SVR(kernel='rbf', C=1.0, epsilon=0.1),
    "GARCH Model": arch_model(y_train, vol='GARCH', p=1, q=1),
    "ARIMA Model": ARIMA(y_train, order=(1, 1, 1)),
    "LSTM Model": Sequential([
        LSTM(50, activation='relu', input_shape=(X_train.shape[1], 1)),
        Dense(1)
    ])
}
```

```
# Dictionary to hold model performance metrics
model_performance = {}

# Assuming X_train and X_test are pandas DataFrames and you need to reshape them for the LSTM model
X_train_array = X_train.values.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test_array = X_test.values.reshape((X_test.shape[0], X_test.shape[1], 1))

# Fitting models and evaluating performance
for name, model in models.items():
    if name == "GARCH Model":
        # Fit the model
        model_result = model.fit(disp='off')
        # Forecasting
        # The forecast method returns a DataFrame. We're interested in the last row, as it contains the forecast at the horizon we care about.
        y_pred = model_result.forecast(horizon=len(y_test)).mean.iloc[-1].values
    elif name == "ARIMA Model":
        model_fit = model.fit()
        y_pred = model_fit.forecast(steps=len(y_test))
    elif name == "LSTM Model":
        model.compile(optimizer='adam', loss='mse')
        model.fit(X_train_array, y_train, epochs=50, batch_size=32, verbose=0)
        y_pred = model.predict(X_test_array).flatten()
    else:
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)
    mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100
    adj_r2 = 1 - (1-r2_score(y_test, y_pred)) * (len(y_test)-1)/(len(y_test)-X_test.shape[1]-1)
    model_performance[name] = {
        'MSE': mse,
        'RMSE': rmse,
        'MAE': mae,
        'MAPE': mape,
        'Adjusted R2': adj_r2
    }

# Display model performance
for model, metrics in model_performance.items():
    print(f"{model} Performance:")
    for metric, value in metrics.items():
        print(f"  {metric}: {value}")
    print("\n")

# Model comparison graphs
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle('Model Performance Comparison')

# MSE comparison
axs[0, 0].bar(model_performance.keys(), [metrics['MSE'] for metrics in model_performance.values()])
axs[0, 0].set_title('Mean Squared Error (MSE)')
axs[0, 0].set_xticklabels(model_performance.keys(), rotation=45, ha='right')

# RMSE comparison
axs[0, 1].bar(model_performance.keys(), [metrics['RMSE'] for metrics in
model_performance.values()])
axs[0, 1].set_title('Root Mean Squared Error (RMSE)')
axs[0, 1].set_xticklabels(model_performance.keys(), rotation=45, ha='right')

# MAE comparison
```

```
axs[1, 0].bar(model_performance.keys(), [metrics['MAE'] for metrics in model_performance.values()])
axs[1, 0].set_title('Mean Absolute Error (MAE)')
axs[1, 0].set_xticklabels(model_performance.keys(), rotation=45, ha='right')
# Adjusted R-squared comparison
axs[1, 1].bar(model_performance.keys(), [metrics['Adjusted R2'] for metrics in model_performance.values()])
axs[1, 1].set_title('Adjusted R-squared')
axs[1, 1].set_xticklabels(model_performance.keys(), rotation=45, ha='right')

plt.tight_layout()
plt.show()

import numpy as np
import statsmodels.api as sm
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

# Assume you have the following data:
# X_train, X_test, y_train, y_test

# Fit the OLS model using statsmodels
X_train_sm = sm.add_constant(X_train)
ols_model = sm.OLS(y_train, X_train_sm).fit()
X_test_sm = sm.add_constant(X_test)

# Extracting AIC, BIC from the OLS model
aic_ols = ols_model.aic
bic_ols = ols_model.bic

# Generate predictions for the OLS model
y_pred_ols = ols_model.predict(X_test_sm)

# Calculate MSE, RMSE, MAE, and MAPE for the OLS model
```

```
mse_ols = mean_squared_error(y_test, y_pred_ols)
rmse_ols = np.sqrt(mse_ols)
mae_ols = mean_absolute_error(y_test, y_pred_ols)
mape_ols = np.mean(np.abs((y_test - y_pred_ols) / y_test)) * 100
# Assuming you have a model_performance dictionary to store metrics for different models
model_performance = {}

# Update the model_performance dictionary for the OLS model
model_performance["OLS"] = {
    'AIC': aic_ols,
    'BIC': bic_ols,
    'MSE': mse_ols,
    'RMSE': rmse_ols,
    'MAE': mae_ols,
    'MAPE': mape_ols,
    'Adjusted R2': 1 - (1-r2_score(y_test, y_pred_ols)) * (len(y_test)-1)/(len(y_test)-X_test.shape[1]-1)
}

# Assuming you have already calculated the performance metrics for the quantile regression model
# and stored them in the model_performance dictionary under the key "Quantile Regression"

# Review the performance of the OLS model alongside the quantile regression model
for model, metrics in model_performance.items():
    print(f"{model} Performance:")
    for metric, value in metrics.items():
        print(f"  {metric}: {value}")
    print("\n")

# Displaying the summary of the OLS model
print("OLS Model Summary:")
print(ols_model.summary())

import matplotlib.pyplot as plt
import statsmodels.api as sm

# Initialize lists to store the metrics for each model
adjusted_r_squared = []
aic_values = []
bic_values = []

# Loop over the sets of predictors, fit a model for each, and collect the metrics
for predictors in predictor_columns:
    X = data[predictors]
    y = data[actual_returns_column]  # Replace with your actual returns column name

    # Split the data into train and test (here we're just using the entire dataset for simplicity)
    X_sm = sm.add_constant(X)

    # Fit the OLS model and get the result
    model = sm.OLS(y, X_sm).fit()

    # Collect the metrics
    adjusted_r_squared.append(model.rsquared_adj)
    aic_values.append(model.aic)
    bic_values.append(model.bic)

# Now create the plots
plt.figure(figsize=(10, 5))

# Plot for Adjusted R-squared
plt.plot(range(1, len(adjusted_r_squared) + 1), adjusted_r_squared, label='Adjusted R-squared', marker='o')
plt.title('Model Metrics by Number of Predictors')
plt.xlabel('Number of Predictors')
plt.ylabel('Adjusted R-squared')
plt.legend()
plt.show()

# Similarly, you can plot AIC and BIC
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(aic_values) + 1), aic_values, label='AIC', marker='o')
plt.plot(range(1, len(bic_values) + 1), bic_values, label='BIC', marker='o')
plt.title('AIC and BIC by Number of Predictors')
plt.xlabel('Number of Predictors')
plt.ylabel('Information Criterion')
plt.legend()
plt.show()

import pandas as pd
import numpy as np
import statsmodels.api as sm

# Define your DataFrame columns and return series
predictor_columns = ['rvol_pt' + str(i) for i in [1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 90, 108]]
actual_returns_column = 'dr1'  # This is your actual returns

# Fit the OLS model
X_train_sm = sm.add_constant(X_train)  # Adds a constant term to the predictor
ols_model = sm.OLS(y_train, X_train_sm).fit()

# Make predictions on the test set and calculate residuals
X_test_sm = sm.add_constant(X_test)  # Adds a constant term to the test data
y_pred_ols = ols_model.predict(X_test_sm)  # Predictions on the test data
residuals = y_test - y_pred_ols  # Calculate residuals

# Define quantile levels for VaR calculation
var_quantiles = [0.01, 0.025, 0.05, 0.1]  # Left tail
var_quantiles.extend([1-q for q in var_quantiles])  # Right tail
```

```
# Function to calculate empirical VaR at a specific quantile level
def calculate_empirical_var(returns, quantile_level):
    # Use the residuals of the OLS model to calculate the empirical VaR
    return returns.quantile(quantile_level)
# List of indices that actually exist in your DataFrame
indices = [1, 2, 3, 4, 5, 6, 9, 10, 12, 15, 18, 20, 27, 30, 36, 45, 54, 60, 90, 108]

# Initialize a dictionary to store the VaR results and additional statistics for each rvol_pt
var_results = {f'rvol_pt{i}': {} for i in indices}

# Calculate VaR for each rvol_pt and additional statistics
vaR_level_for_pass = 0.01  # 1%

# Function to format the VaR key string
def format_var_key(quantile, tail):
    percentage = int(quantile * 100) if quantile * 100 == int(quantile * 100) else quantile * 100
    return f'{tail} VaR {percentage}%'

# Define the 1% VaR key for the left tail used for the pass rate calculation
vaR_1_key = format_var_key(0.01, 'Left')  # 'Left VaR 1%'

# Iterate over each rvol_pt series
for i in indices:
    rvol_column = f'rvol_pt{i}'
    series = data[rvol_column]

    # Calculate and store VaR values for each quantile level
    for quantile in var_quantiles:
        tail = 'Left' if quantile < 0.5 else 'Right'
        var_key = format_var_key(quantile, tail)
        var_results[rvol_column][var_key] = calculate_empirical_var(series, quantile)

    # Calculate Min, Max, Range for the series
    var_results[rvol_column]['Min'] = series.min()
    var_results[rvol_column]['Max'] = series.max()
    var_results[rvol_column]['Range'] = series.max() - series.min()

    # Calculate the pass rate using the breach threshold from the 1% VaR
    breach_threshold = var_results[rvol_column][vaR_1_key]
    breach_count = (series < breach_threshold).sum()  # Count of breaches
    total_count = len(series)
    pass_rate = (total_count - breach_count) / total_count
    var_results[rvol_column]['Pass Rate'] = pass_rate

# Convert the dictionary to a DataFrame and transpose it
var_table = pd.DataFrame(var_results).T

# Show the DataFrame head to verify the 'Pass' values for each 'rvol_pt'
var_table

import matplotlib.pyplot as plt
import seaborn as sns

# Store model predictions
model_predictions = {
    "Linear Regression": models["Linear Regression"].predict(X_test),
    "Ridge Regression": models["Ridge Regression"].predict(X_test),
    "Random Forest Regressor": models["Random Forest Regressor"].predict(X_test),
    "Gradient Boosting Regressor": models["Gradient Boosting Regressor"].predict(X_test),
    "Support Vector Regressor": models["Support Vector Regressor"].predict(X_test),
}

# Dictionary to hold actual vs predicted values for all models
predictions = {
    "Linear Regression": model_predictions['Linear Regression'],
    "Ridge Regression": model_predictions['Ridge Regression'],
    "Random Forest Regressor": model_predictions['Random Forest Regressor'],
    "Gradient Boosting Regressor": model_predictions['Gradient Boosting Regressor'],
    "Support Vector Regressor": model_predictions['Support Vector Regressor'],

}

# Generate scatter plots for each model
for model_name, y_pred in predictions.items():
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=y_test, y=y_pred, alpha=0.6)
    plt.axline((1, 1), slope=1, color="red", linestyle="--")  # Diagonal line for reference
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
    plt.title(f'{model_name} - Actual vs Predicted')
    plt.show()
```

Appendix 2. Volatility Forecast Performance Plots for Full Dataset

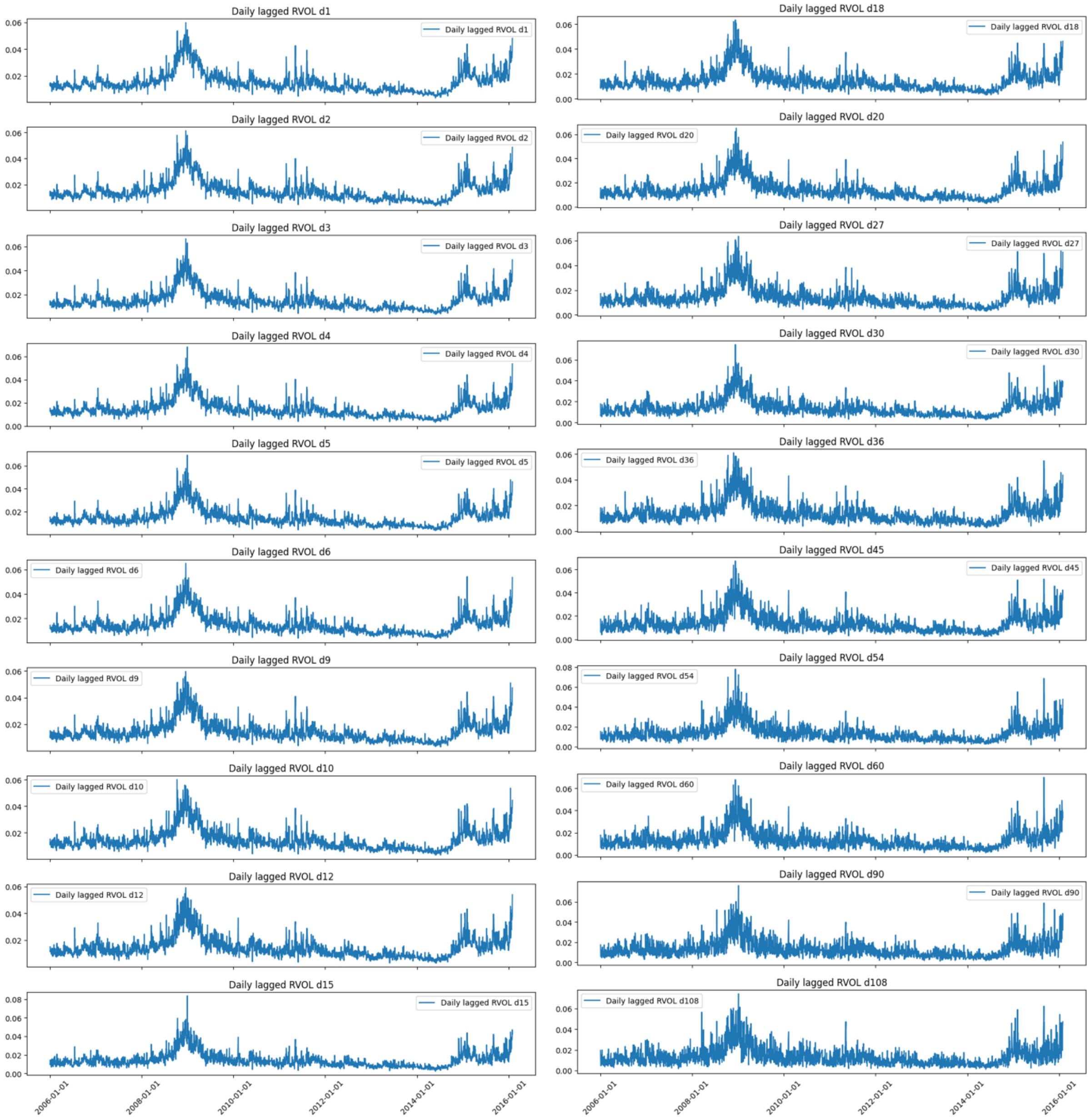


Figure A2.1. Volatility forecast performance plots at a daily (time frame).

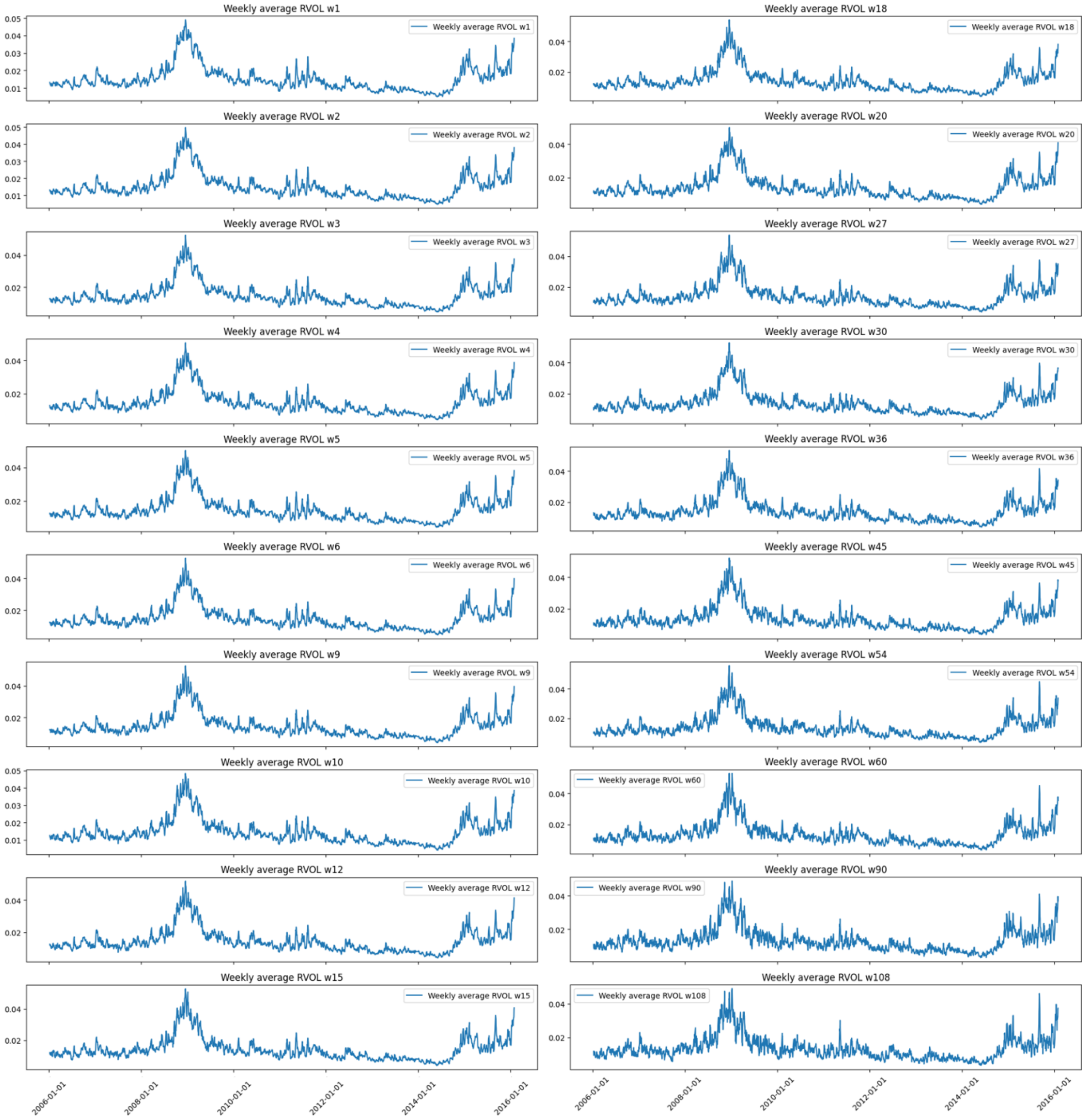


Figure A2.2. Volatility forecast performance plots at a weekly time frame.

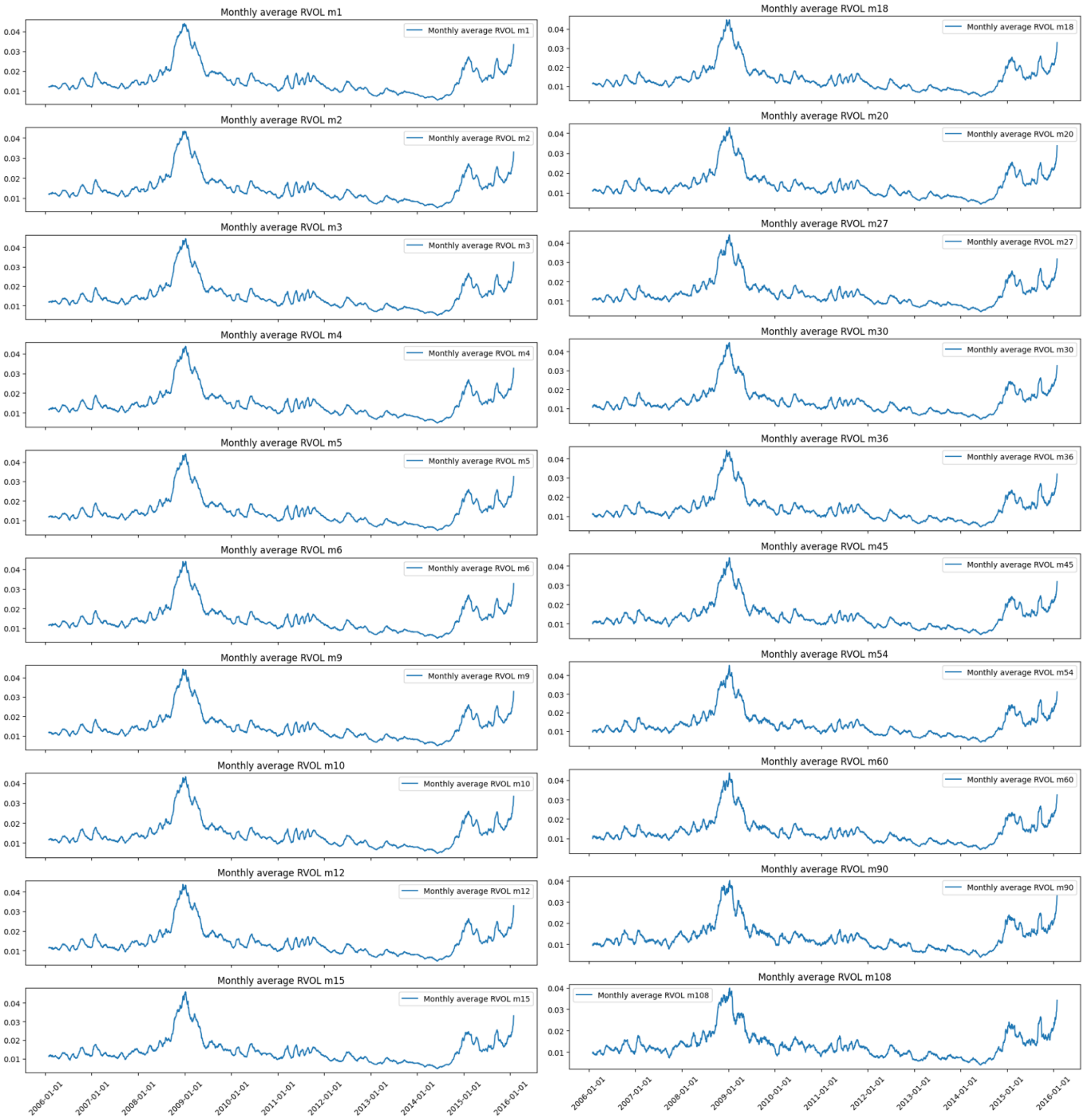
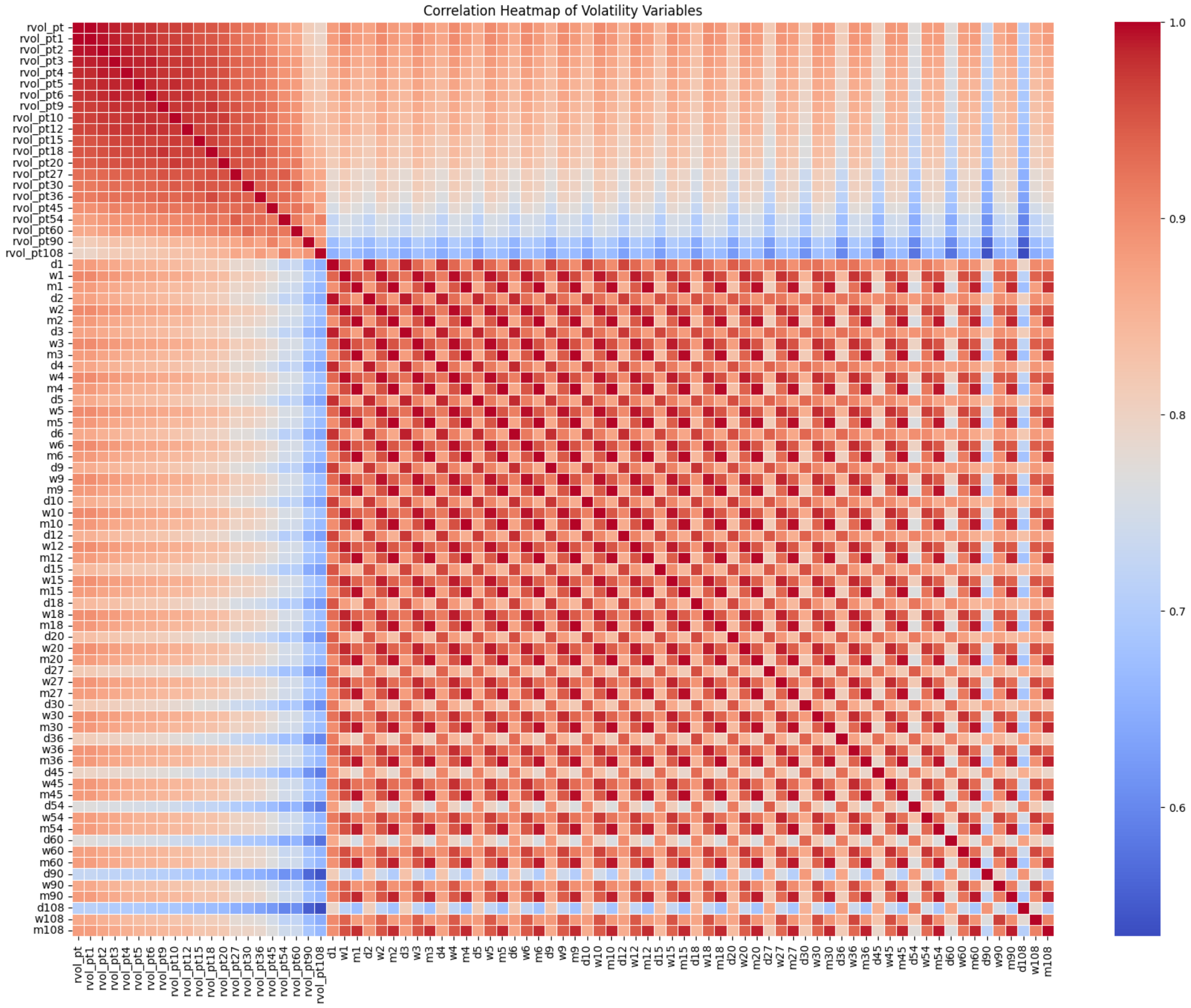


Figure A2.3. Volatility forecast performance plots at a monthly time frame.

Appendix 3. Correlation Heatmap of Volatility Variables


Correlation Heatmap of Volatility Variables
rvol_pt
rvol_pt1
rvol_pt2
rvol_pt3
rvol_pt4
rvol_pt5
rvol_pt6
rvol_pt9
rvol_pt10
rvol_pt12
rvol_pt15
rvol_pt18
rvol_pt20
rvol_pt27
rvol_pt30
rvol_pt36
rvol_pt45
rvol_pt54
rvol_pt60
rvol_pt90
rvol_pt108
d1
w1
m1
d2
w2
m2
d3
w3
m3
d4
w4
m4
d5
w5
m5
d6
w6
m6
d9
w9
m9
d10
w10
m10
d12
w12
m12
d15
w15
m15
d18
w18
m18
d20
w20
m20
d27
w27
m27
d30
w30
m30
d36
w36
m36
d45
w45
m45
d54
w54
m54
d60
w60
m60
d90
w90
m90
d108
w108
m108