Quantitative Risk Management: An Application of Machine Learning

Executive Summary: Quantitative Risk Management and Machine Learning by Jørgen Leiros

Jørgen Maurstad Leiros, as part of his MSc in Business Administration and Data Science, conducted an advanced study in **Quantitative Risk Management (QRM)** with a focus on **volatility forecasting using statistical and machine learning models**. His work applies **financial modeling, machine learning, and time-series forecasting** to improve risk estimation strategies, particularly in the context of portfolio management.

Key Focus Areas:

1. **Volatility Forecasting for Financial Risk Management** – The study compared traditional **GARCH models** with **machine learning approaches** such as **Linear Regression, Random Forest, and Long Short-Term Memory (LSTM) networks** to predict stock return volatility.
2. **Application to Norway's Sovereign Wealth Fund (Oljefondet)** – Using stock data from Oljefondet's **top energy sector investments**, Jørgen analyzed **historical stock prices** to evaluate **risk and return dynamics**, highlighting the importance of volatility forecasting for large-scale asset management.
3. **Modeling and Evaluation** – The research implemented different **volatility forecasting techniques**, comparing their effectiveness using **Mean Squared Error (MSE)**. Results showed that:
   a. **GARCH performed well for short-term stability** but struggled with dynamic market fluctuations.
   b. **LSTM and other machine learning models adapted better to long-term trends**, although none fully captured extreme volatility spikes.
4. **Strategic Insights and Future Implications** – The study suggested **enhancing machine learning models with additional financial indicators**, such as **macroeconomic data, sector-specific trends, and market sentiment analysis**, to improve volatility predictions and risk management for institutional investors.

Through this work, Jørgen has demonstrated expertise in **financial modeling, machine learning, and quantitative risk analysis**, providing valuable insights for **portfolio optimization and risk mitigation in financial markets**.

Introduction

Quantitative Risk Management (QRM) applies statistical techniques to systematically identify, assess, and address risks across various domains. Risks take different forms in different industries, and an organization's approach to managing them will vary, but the rational response often involves seeking to minimize or control risk. A more accepting attitude towards risk can be observed in the financial industry, where services such as lending, insurance, investments, trading, and managing derivatives rely heavily on the ability to balance risk and potential reward. This is particularly true given that higher risks often come with the possibility of greater returns.

Norway's sovereign wealth fund, commonly referred to as Oljefondet, manages the surplus revenues of Norway's petroleum reserves. The fund is tasked with deploying this capital to achieve the highest possible return within an acceptable risk, as defined by the Norwegian Ministry of Finance (NBIM, 2024). Accurate risk estimates are crucial for Oljefondet to achieve its dual objectives of maximizing returns while avoiding excessive exposure to potential losses. Moreover, with more precise risk estimates, the fund gains greater flexibility in managing its portfolio, enabling it to explore optimized asset allocation and diversification strategies while staying within defined risk parameters.

Portfolio management for a fund of Oljefondet's scale requires sophisticated tools to balance risk and return effectively. A key aspect of this process is forecasting volatility, which measures the degree of variation in asset returns over time. Accurate volatility forecasts are essential for managing portfolio risk, pricing derivatives, and determining asset allocation strategies. Traditional methods, such as GARCH (Generalized Autoregressive Conditional Heteroskedasticity), have long been used for this purpose, and can provide reasonably reliable short-term predictions. However, recent advancements in machine learning (ML) offer new possibilities for improving volatility forecasting. ML models can identify complex, non-linear patterns in data, potentially outperforming traditional methods in dynamic and volatile markets.

This paper aims to compare the performance of a traditional volatility forecasting method, GARCH, with several machine learning approaches, including Linear Regression, Random Forest, and Long Short-Term Memory (LSTM) models. Through this comparison, the study seeks to evaluate whether machine learning models provide meaningful advantages over statistical methods. While this study focuses on single stocks and applies ML as a novel approach to volatility forecasting, a key motivation is to provide an initial exploration of its potential relevance for portfolio management. This leads to the following problem statement:

"Can machine learning models improve the accuracy of stock volatility forecasts compared to traditional methods, and what implications might this have for risk management in the financial market?"

Related Work

In a 1996 study by Barilsford & Faff, simple traditional models for forecasting volatility are evaluated against what was considered complex models at the time like the ARCH class models (Barilsford & Faff, 1996). Due to the key role of volatility forecasts in economic decision making and analysis, the authors aim to examine which models are superior in forecasting volatility as they argue that existing literature at the time had conflicting evidence. The study compares models including random walk, historical mean, moving average, exponential smoothing, exponentially weighted moving average, simple regression, and several GARCH-based models, including asymmetric GJR-GARCH models. Their findings indicate that while no single model is universally superior, the ARCH class models, particularly the GJR-GARCH (1,1) specification, performed at least as well as, if not better than, simpler models like the historical mean or regression approaches. This highlights the increasing importance of incorporating more sophisticated volatility modeling techniques in financial forecasting.

In their comprehensive survey, Mashrur et al. (2020) explore the application of machine learning techniques in financial risk management, including areas such as volatility forecasting, credit risk evaluation, and fraud detection. The survey presents traditional methods including models such as Exponentially Weighted Moving Average (EWMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and stochastic volatility models like the Heston model. While these approaches are widely used, limitations include their inability to capture asymmetries in volatility and their reliance on structured, univariate data. To address these shortcomings, machine learning techniques, including hybrid models that integrate neural networks with GARCH, standalone methods like Long Short-Term Memory (LSTM) networks, and text-based predictors such as news and social media data, have been developed. Mashrur et al. highlight the potential of advanced approaches using machine learning to outperform traditional models in capturing complex, non-linear patterns and leveraging diverse data sources.

The application of ML for volatility forecasting is examined further by Christensen et al. in a 2023 study comparing various ML models with multiple heterogeneous autoregressive (HAR) models. The authors evaluate a range of HAR models, including extensions such as LevHAR, SHAR, and HARQ, each designed to address specific shortcomings of traditional HAR models and to provide more reliable predictions. These statistical models are evaluated against machine learning models including regularized regression models, tree-based methods, and neural networks, each of which are implemented with minimal hyperparameter tuning. The ML techniques used in this study leverage a broader range of firm-specific and macroeconomic predictors while HAR models primarily rely on lagged realized variance and limited time-series inputs. The findings demonstrate that ML models consistently outperform HAR models, particularly at longer forecasting horizons, due to their ability to capture nonlinearities and extract incremental information from diverse predictors.

Literature

Volatility

Volatility refers to fluctuations in observations over time, but in economics we might describe it as the variability of the unforeseen component of a time series. There are many ways to measure volatility, but the most common measure is the standard deviation of return (Scwert, 1990). Volatility exhibits several notable characteristics, including clustering, where periods of high or low volatility tend to persist, and mean reversion, where volatility eventually stabilizes around an average level (Bose, 2007). Traditional models, such as the GARCH family, have been developed to model these patterns by capturing time-varying volatility, while more recent methods have explored machine learning approaches to address the limitations of traditional models and account for non-linearities in financial data.

ARCH Model Suite

ARCH (Autoregressive Conditional Heteroskedasticity) is a time series model commonly used to forecast changing levels of volatility in financial data (McNeil & Embrechts, 2015, *p. 139*). The model uses past squared returns to predict the current level of volatility (autoregressive), determines volatility based on information available up to the present (conditional), and captures changes in volatility over time (heteroskedastic). This approach allows the model to capture "volatility clusters", where periods of high volatility follow other high-volatility periods, and low-volatility periods follow low ones. ARCH models are useful for understanding the change in risk levels in financial data but can become less reliable when dealing with long-term volatility persistence and may require many parameters for adequate modeling, making them less practical for this task.

GARCH

GARCH (Generalized AutoRegressive Conditional Heteroskedasticity) expands on the ARCH model by incorporating previous volatilities in addition to past squared returns, allowing for a more efficient and flexible way to model volatility (McNeil & Embrechts, 2015, *p. 145*). This inclusion enables GARCH to capture both short-term shocks and long-term volatility persistence, making it particularly effective for financial time series where volatility remains elevated or subdued over extended periods. Like ARCH, GARCH models account for "volatility clustering", however, GARCH achieves this with fewer parameters, reducing the complexity required with ARCH models. As a result, GARCH is widely used in practice, offering a balance between simplicity and the ability to model real-world volatility dynamics.

Machine Learning

Machine Learning (ML) is a subfield of artificial intelligence, and it is the study of computer algorithms which improve at performing nontrivial tasks automatically through experience (or learning from data) (Géron, 2019). ML typically involves feeding a model with data, enabling it to identify patterns, make decisions, or predictions without explicit programming for specific tasks. This makes ML particularly useful for applications where traditional rule-based programming would be infeasible, such as recognizing complex patterns in images, translating languages, or forecasting market trends. Key characteristics of ML include its ability to handle large datasets, adapt to new data through continuous learning, and capture non-linear and high-dimensional relationships, making it a powerful tool for solving a wide range of real-world problems.

There are many different types of ML models with inherent characteristics making them suitable for their own specific groups of tasks, but it is common to separate models based on how they are trained (Géron, 2019). ML models are typically categorized by the level of supervision required during training, with the main groups being supervised and unsupervised learning, each suited to solving specific types of tasks. Supervised learning is when the data used to train the model requires human-labeled target variables, and common tasks include regression and classification. Unsupervised learning, on the other hand, is useful when we want to uncover hidden patterns, groupings, or structures in data without relying on labeled target variables, with common tasks including clustering and dimensionality reduction.

Neural Networks

Neural Networks are machine learning models inspired by the structure of the human brain, with artificial neurons organized in layers (Géron, 2019). The simplest neural network architecture is the perceptron, made up of an input layer with associated weights, a bias term, an activation function, and an output. A single-layer perceptron can be seen as a linear classifier, but when multiple perceptrons are stacked in layers it enables the network to capture non-linear patterns through a variety of activation functions. During training, input data is passed through the network, and the weights and biases of each artificial neuron are adjusted via backpropagation. These are powerful models, capable of capturing complex, non-linear patterns, but increased complexity can also make the model prone to overfitting and the process of backpropagating errors introduces the issue of exploding gradients.

Neural networks can be categorized into three main types, each tailored to specific data types and tasks based on their unique characteristics. Feedforward Neural Networks are the simplest type, where information flows in one direction from input to output, well suited for tasks like regression and classification. Convolutional Neural Networks are designed to process grid-like data such as images, using convolutional layers to detect spatial patterns, making them ideal for image recognition and computer vision. Recurrent Neural Networks (RNNs) are specialized for dealing with sequential data such as text or time series data, as they retain information about previous inputs through feedback loops. RNNs are particularly exposed to the risk of both vanishing and exploding gradient problems because of the structure of feedback loops.

Methodology

Data Collection and Analysis

For this project, I will analyze investments made by Oljefondet, utilizing the closing prices of selected stocks over a five-year period. To identify stocks of interest, I have obtained an updated overview of Oljefondet's stock portfolio, which includes details such as the value of each stake, the company's industry, and the degree of ownership, among other features (NBIM.no, 2024). Since my research focuses on comparing a traditional approach to predicting volatility with advanced machine learning approaches. Sector information is crucial for contextualizing company performance and market volatility. Stock data, including closing prices on the NASDAQ market, will be sourced from Polygon.io.

For this project, I chose to analyze the three highest-valued investments Oljefondet holds in the Energy sector as these stocks significantly impact Oljefondet's overall performance and the companies operate in a volatile industry. To ensure sufficient data for training and validation, I retrieved stock market data for the top three companies from 2019 to the present, resulting in 5,380 observations for each sample. One issue which was encountered at this stage was that Shell Plc had undergone a large restructuring leading to a single line of shares as of January 21st, 2022, with a similar problem arising for the fourth most valuable company (NASDAQ, 2022). For this reason, the fifth most valuable company was chosen for the analysis, Chevron Corp.

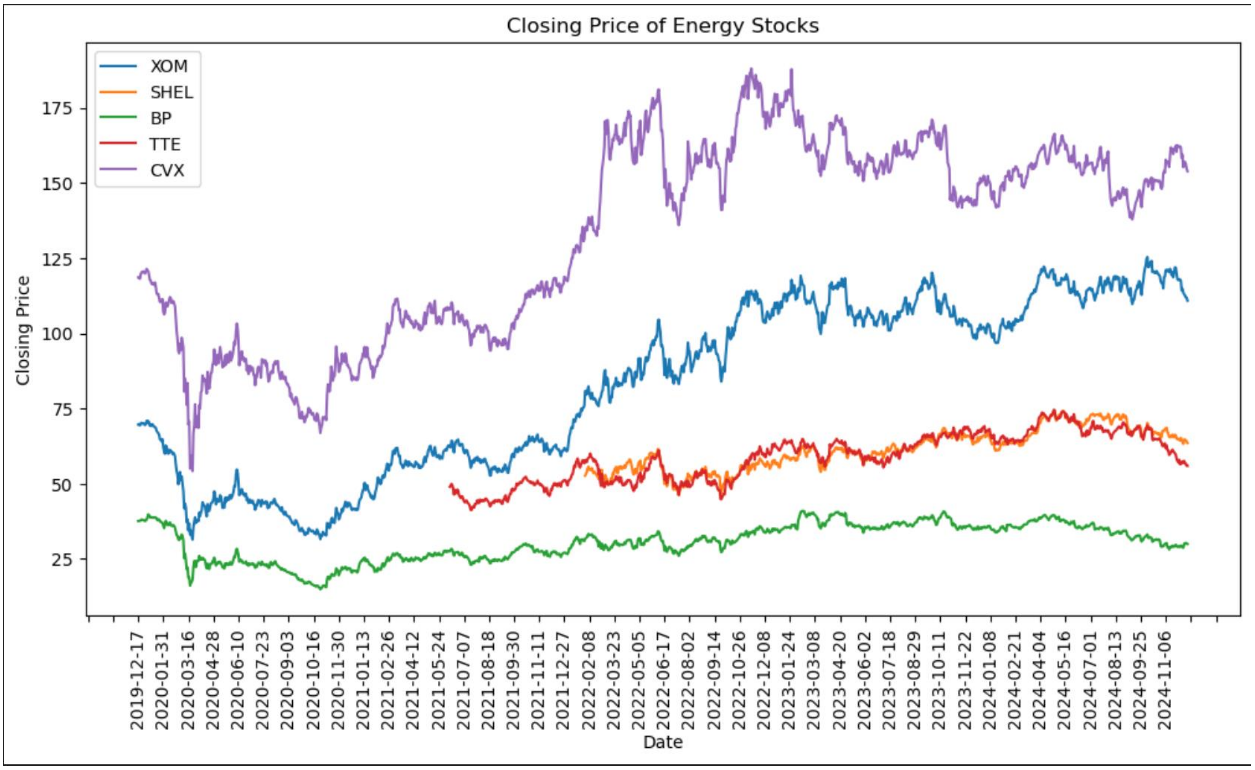

Figure 1: Line graph showing stock price for the top five companies by evaluation of investment by Oljefondet for the previous five years.

Data Preprocessing

Preprocessing of the stock price data involved several steps to ensure its quality and prepare it for analysis. The first step was to drop the observations for the two stocks which did not date back five years, leaving only "XOM", "BP", and "CVX". Once only the relevant data remained, the dataset was checked for missing values and duplicates, but none were found. To identify potential anomalies, the closing price column was plotted, and no significant outliers were detected. Additionally, the data types of each column were checked to ensure they were consistent with their expected formats, including converting the date column to a datetime format to enable proper sorting and time-based operations. dataframe was then sorted chronologically by date to ensure consistency in the temporal structure of the data.

After ensuring the consistency of the data and sorting it on stock ticker and date, the *Return* column was computed as the target variable for this analysis. Daily returns were computed as (Closin Price t / Closing Price t − 1) - 1 to express the percentage change relative to the previous price, capturing how the stock value fluctuated from one day to the next. The return calculation was essential for analyzing stock volatility and served as a basis for modeling and prediction tasks. After computing the return column, the remaining features were dropped except for *Ticker, Date* and *Return*. Finally, the *30_day_lagged* column containing 30 day lagged returns was created to aid the ML models in their predictions and stock as no other features will be added for this comparison.

Modeling

The models used in this project include statistical, basic machine learning, and neural network approaches. We used the ARCH package for implementing the GARCH model, scikit-learn for the linear regression and random forest models, and TensorFlow for the LSTM neural network.

GARCH

GARCH was chosen as the statistical model for predicting volatility due to its suitability for financial time series exhibiting volatility clustering and persistence, as mentioned in the literature section. To select the optimal model configuration, a grid search was conducted over different combinations of lag parameters p and q, which determines the number of past squared returns and past volatilities included in the model, respectively. The Akaike Information Criterion (AIC) was used to identify the best-fitting model, resulting in a GARCH (2,1) specification. This means the model incorporates the two most recent squared returns and the most recent past volatility, striking a balance between model complexity and predictive performance. This approach ensures that the model effectively captures both short-term shocks and the persistence of volatility over time.

Linear Regression

Linear regression (LR) was implemented as a simple machine learning algorithm which models the relationship between a set of features and a target variable by fitting a linear equation to the data (IBM, n.d. a). In this project, an LR model was configured using *30_day_lagged* as feature and the return as the target variable. However, as no specific engineered features beyond lagged returns were provided, the model had limited information to capture the underlying relationships that drive stock returns, making time series forecasting tasks like volatility prediction challenging. Consequently, while linear regression was included for comparison, its suitability for this context was limited due to the absence of meaningful features or time-series-specific transformations.

Random Forest Regressor

Random Forest (RF) was chosen as a machine learning model for its ability to handle regression tasks by leveraging an ensemble of decision trees to produce accurate and robust predictions (IBM, n.d. b). Unlike a single decision tree, which can be prone to overfitting, Random Forest reduces variance and improves prediction accuracy by averaging the outputs of multiple uncorrelated trees, each trained on a random subset of the data and features. Similarly to the LR model, the RF model was configured using *30_day_lagged* as feature and the return as the target variable. However, as no additional engineered features or time-series-specific transformations were included, the model's ability to extract meaningful patterns in stock return data was limited.

Long Short - Term Memory

The Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN), which is implemented to capture temporal dependencies in sequential stock return data (IBM, n.d. c). As mentioned previously, RNNs are specifically designed to learn from sequentially ordered data, making them highly suitable for time-series forecasting tasks. Additionally, LSTMs mitigate the vanishing gradient problem by using specialized "gates" to manage the flow of information through the network. This allows the model to retain relevant information in memory while disregarding less relevant data, making them effective for modeling long-term dependencies in stock returns.

The LSTM model was implemented using TensorFlow and trained solely on the *30_day_lagged* feature. The data was scaled for numerical stability, reshaped to fit the 3D input structure required by LSTMs, and split into training and testing sets. The network architecture consisted of one LSTM layer with 50 units and a dense output layer for single step return prediction. The model was trained for 20 epochs using the Adam optimizer and mean squared error as the loss function. Despite its potential to capture temporal patterns, the lack of additional features beyond lagged returns may have limited the model's predictive performance.

Evaluation

The data for this project has been split by ticker, with training data containing all observations up to the most recent 30 days, which are used as the test set for predictions. The target variable is the percentage change in stock returns, capturing the daily return dynamics for each stock.

The evaluation metric chosen is the Mean Squared Error (MSE), as it effectively penalizes larger errors, making it suitable for assessing the accuracy of return predictions. To provide a more granular analysis, MSE has been computed for both the first 10 days and the full 30-day test period. This approach allows a clearer understanding of short-term versus longer-term prediction performance, particularly important given the volatility and time-sensitive nature of financial data.

Results

Figure 2 illustrates the performance of the GARCH (orange), LR (green), RF (red), and LSTM (purple) models in predicting stock return volatility for three energy sector stocks: XOM, BP, and CVX (actual returns are blue). The GARCH model remains close to the mean volatility and demonstrates stability in its early predictions, particularly for XOM and CVX, but struggles to adapt to fluctuations over time. Both the LR and LSTM models predict return values close to zero across all stocks, though they show slight variability and appear to capture some patterns for XOM, where historical volatility has been lower. In contrast, the RF model produces the most varied predictions, indicating its ability to identify underlying patterns. However, its tendency to deviate significantly when predictions are inaccurate increases its mean squared error (MSE), highlighting a trade-off between sensitivity to patterns and stability.

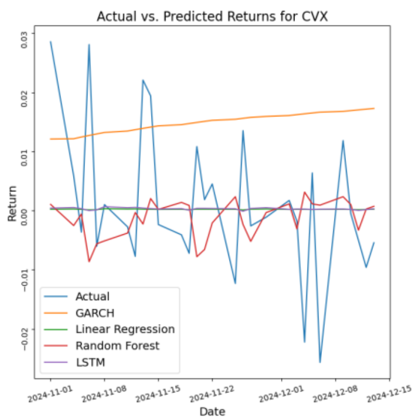

Figure 2: Actual return compared to predictions. Lines represent predictions from different models and each frame corresponds to specific stocks.

The mean squared error (MSE) results found in table 1 further highlight the differences in model performance across the three energy sector stocks. For XOM, the LR, RF, and LSTM models achieve lower MSE values compared to the GARCH model, suggesting that these models were better at capturing short-term variations in returns. Notably, GARCH performs better over the first 10 days, particularly for XOM and CVX, aligning with its stability in predicting mean-level volatility observed in the plots. However, its inability to adapt to a change in trend amongst the fluctuations over longer periods leads to higher overall error. In contrast, the LR, RF, and LSTM models demonstrate more consistent performance across both the first 10 days and the full prediction horizon, with the ML models achieving the same MSE and the lowest error for XOM and CVX.

Stock	**Model**	**MSE (30 day)**	**MSE (10 day)**
XOM	GARCH	0.0004	0.0003
	Linear Regression	0.0001	0.0002
	Random Forest Regressor	0.0001	0.0002
	LSTM	0.0001	0.0002
BP	GARCH	0.0007	0.0008
	Linear Regression	0.0003	0.0004

	Random Forest Regressor	0.0005	0.0006
	LSTM	0.0003	0.0004
CVX	GARCH	0.0004	0.0002
	Linear Regression	0.0002	0.0003
	Random Forest Regressor	0.0002	0.0003
	LSTM	0.0002	0.0003

Table 1: Mean Squared Error (MSE) for GARCH, Linear Regression (LR), Random Forest (RF), and LSTM Models for both 30 and 10 days.

For BP, the GARCH model shows the highest MSE, indicating it struggles the most with the stock's higher volatility. LR and LSTM achieve the lowest errors, while RF slightly underperforms due to its sensitivity to sharp variations, as discussed earlier. This aligns with the observation in the plots that RF's larger deviations, while capturing patterns, are penalized more heavily by the MSE metric. Across all stocks, the LSTM model demonstrates strong performance relative to its complexity, suggesting its ability to balance stability and pattern recognition. Overall, the MSE results reinforce that while GARCH is effective at modeling mean-level volatility, ML models such as LR, RF, and LSTM are better suited to adapt to return fluctuations in more volatile time-series data.

Discussion

Accurate volatility predictions allow fund managers to anticipate market fluctuations, mitigate potential losses, and optimize portfolio performance. However, volatility is difficult to forecast because it is influenced by complex, non-linear patterns, unexpected market events, and economic factors that can be hard to account for. This study employs a simplified approach to evaluate the efficacy of machine learning models for volatility forecasting, using only 30-day lagged returns as features. Despite the minimal input data, the results offer valuable insights into the potential of ML models due to their ability to learn non-linear and complex patterns.

The results of this study highlight differences in the performance of statistical and machine learning models for predicting stock return volatility in the energy sector. While the GARCH model was able to capture mean-level volatility over very short time horizons, it struggled in the longer-term predictions. In contrast, the ML models demonstrated stronger adaptability to long-term fluctuations, with LSTM performing the best overall. However, as seen in Figure 2, even LSTM struggles to identify a consistent and meaningful pattern, underscoring the inherent challenges of this task.

MSE was chosen for the evaluation metric, and it is used when optimizing weights for LR and LSTM, as well as for splits in the RF model, heavily impacting both the construction of these models and the evaluation of their performance. On the one hand, MSE is effective for penalizing large prediction errors, making it suitable for volatility forecasting where significant deviations can have significant impacts on financial decision-making. However, squaring the errors can lead to disproportionately penalized extreme predictions, which can bias the evaluation against models like Random Forest that occasionally make larger errors but seem to better capture underlying patterns. While MSE aligns well with the goal of minimizing high-impact forecasting errors, a balanced approach could be to evaluate the models using multiple metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), to provide a more balanced evaluation of model performance and robustness.

While none of the models successfully captured significant spikes in volatility within the forecasted 30 days, the low MSE scores of the ML models indicate that further optimization could significantly enhance their predictive performance. Given the simplicity of the models used in this project, minor adjustments have the potential to improve performance significantly compared to their statistical counterpart. Incorporating additional data such as macroeconomic indicators, oil price fluctuations, sector-specific news, or financial metrics like trading volume and implied volatility could help the models better capture the drivers of stock return volatility. Additional features engineered from historical data, such as moving averages or volatility indices, might also enhance the ML models' ability to detect patterns and predict more significant shifts in volatility.

The ML models in this project have been implemented with minimal hyperparameter tuning and while they are functional, simple adjustments could improve predictions. The LSTM model, for example, could benefit from a more robust architecture, such as increasing the number of units or adding recurrent dropout to prevent overfitting. Additionally, the Random Forest model might perform better with refined hyperparameter tuning, such as increasing the number of estimators or experimenting with the maximum depth of trees to balance bias and variance. Similarly, the Linear Regression model could be enhanced by applying regularization techniques like Ridge or Lasso to improve its stability and generalization. These relatively simple adjustments could provide substantial performance gains, making the models more effective at capturing the inherent complexities of stock return volatility.

This study's focus on individual stock volatility provides a foundation for understanding how machine learning models might be applied to broader portfolio management. While predicting single-stock volatility is valuable, the true potential of these methods emerges when considering groups of stocks or entire portfolios. Portfolio volatility is not only influenced by individual stock behavior but also by the correlations between assets, which play a crucial role in diversification and risk mitigation. Extending the use of ML models to analyze subsections of the market or full portfolios could unlock insights into these interdependencies, offering a more holistic view of risk. This relevance underscores the importance of studies like this one, which test ML models in a controlled setting, as they lay the groundwork for applying these techniques to more complex and impactful portfolio-level problems.

Conclusion

This study explored the performance of GARCH, Linear Regression, Random Forest, and LSTM models in forecasting stock return volatility for three energy sector stocks. The findings revealed that while GARCH was effective at capturing mean-level volatility and performed best for very short-term horizons, such as next-day predictions, it struggled with more dynamic fluctuations over longer periods. Machine learning models, particularly LSTM, demonstrated greater adaptability to these fluctuations, suggesting their potential for improved volatility forecasting, despite limited features and hyperparameter tuning. However, none of the models fully captured significant volatility spikes, emphasizing the inherent challenges of the task. Despite these limitations, the study highlights the promise of ML approaches, particularly when applied to portfolio-level analysis, where the ability to handle complex interactions between multiple assets could lead to significant advancements in risk management strategies.

References

Bose, S. (2007). Understanding the volatility characteristics and transmission effects in the Indian stock index and index futures market. ICRA Bulletin, Money & Finance.

Brailsford, T. J., & Faff, R. W. (1996). An evaluation of volatility forecasting techniques. *Journal of Banking & Finance*, *20*(3), 419-438.

Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly Media.

IBM (n.d. a) What is Linear Regression. Retrieved from https://www.ibm.com/topics/linear-regression

IBM (n.d. b) What is a recurrent neural network (RNN)? Retrieved from https://www.ibm.com/think/topics/recurrent-neural-networks

IBM (n.d. c) What is random forest? Retrieved from https://www.ibm.com/topics/random-forest

McNeil, A. J., Frey, R., & Embrechts, P. (2015). Quantitative risk management: concepts, techniques and tools-revised edition. Princeton university press.

NASDAQ. (2022). Royal Dutch Shell Plc Changes Name to Shell Plc. Retrieved from https://www.nasdaq.com/articles/royal-dutch-shell-plc-changes-name-to-shell-plc

NBIM (2024), Oljefondets investeringer. Retrieved from https://www.nbim.no/no/oljefondet/investeringene/#/2024/investments/equities

NBIM (2024). Risikostyring i Oljefondet. Retrieved from https://www.nbim.no/no/oljefondet/slik-er-fondet-investert/risikostyring/

Schwert, G. W. (1990). Stock market volatility. Financial analysts journal, 46(3), 23-34.