Comparative Sensitivity Analysis of LSTM and XGBoost in Financial Forecasting

Baheux Samuel, Fritz Axel, Pariente-Launay Samuel

February 29, 2024

Abstract

This paper presents a sensitivity analysis of two advanced Machine Learning models - Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost) - specifically applied to financial forecasting. The primary objective is to assess how these models perform in predicting closing prices and volatility, with a focus on two critical factors: the forecast horizon and the data split between training and testing sets. Utilizing a dataset of five stocks from the S&P 500, we construct a representative equiweighted portfolio to analyze these aspects. Our approach systematically explores the impact of varying forecast horizons and data splits on the predictive accuracy and reliability of the LSTM and XGBoost models in forecasting closing prices and estimating market volatility. The findings of this study provide insightful conclusions on the adaptability and effectiveness of these models in financial data analysis, emphasizing the significance of model tuning and data structuring in achieving optimal forecasting performance. The research aims to offer a practical guide for practitioners in selecting and deploying these models for more accurate financial market predictions.

Keywords : Machine Learning, LSTM (Long Short-Term Memory), XGBoost (Extreme Gradient Boosting), Financial Forecasting, Sensitivity Analysis, Predictive Models

Directed by : Rania Hentati Kaffel

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Contents

1 Methodology

1.1 Data

To conduct this study, we retrieved the daily data of 5 of the largest market capitalizations from the S&P 500: Amazon, Berkshire, Google, Microsoft, Exxon. We have the following information for each asset: Date, opening, closing and adjusted closing price, highest and lowest price of the day, volume of stock traded. These data were retrieved from Yahoo Finance, covering a period of 10 years, from January 2, 2013, to December 29, 2024.

1.2 Creation of our variables of interest

Our paper focuses on two main variables: closing price and volatility. To represent the closing price predictions, we calculate returns from actual and predicted prices for each asset. From these returns, we will create a portfolio valued at 1000 euros, composed of 20% of each asset. Regarding volatility, we use a 10-day rolling window from our asset's price, and we retrieve its standard deviation.

1.3 Modelization

1.3.1 XG-Boost

To model our variables of interest, we first used an XGBoost model. We chose to use a rolling window approach, which was as follows:

1. We define the hyperparameters of our rolling window: the size of the training window and the forecast horizon.
2. We retrieve the training data corresponding to the number of days of the window size as well as the test data, which is an equal number of days to the forecast horizon after the training window.
3. We train the model and then make predictions on the test data.
4. We shift the window by one day and then start again from step 2.

For each learning period, we apply an XGBoost model with the following estimators: n_estimators = 400, max_depths = 30, learning_rate = 0.01. These were found through a grid search.

1.3.2 LSTM

To predict our variables of interest using LSTM, we employed two distinct model architectures:

- **For Closing Price** : A single LSTM layer with 200 units followed by a dense layer sized according to the forecast horizon.
- **For Volatility** : Two LSTM layers, each with 50 units, concluding with a dense layer matching the forecast horizon in size.

Both models were trained utilizing Mean Squared Error as the loss function and Adam as the optimizer.

2 Impact of varying the forecast horizon

In this section, we undertake a detailed analysis of the impact that varying forecast horizons have on the performance of LSTM and XGBoost predictive models. We specifically focus on assessing the prediction errors of these two models using four different time horizons: 1, 7, 14, and 28 days. This approach aims to determine how the length of the forecast horizon affects the accuracy and reliability of the predictions provided by these models.

2.1 XGBoost

2.1.1 Closing Prices

As discussed in the first part, we use a sliding window to predict closing prices. Thus, we ran our models for 4 different forecast horizons with the following hyperparameters: a window size of 30 days and a train/test split of 80%/20%. For each model, we calculated 3 error metrics: MSE, RMSE, and MAE. The results obtained are as follows:

Errors on Portfolio Value	**MSE**	**RMSE**	**MAE**
1 day forecast horizon	2398.60	48.98	42.3
7 days forecast horizon	13636.12	116.77	103.83
14 days forecast horizon	19594.74	139.98	126.13
28 days forecast horizon	7801.43	88.33	75.51

Table 1: Metrics table on Portfolio Value Forecast using XGBoost

The analysis of the results presented in the table for the forecast of portfolio closing prices using XGBoost reveals two significant trends regarding the model's performance sensitivity across different forecast horizons:

1. **Increase in Error with Forecast Horizon**: The error metrics (MSE, RMSE, MAE) generally exhibit an increase as the forecast horizon extends. This trend is particularly noticeable when transitioning from a 1-day to a 14-day forecast horizon, indicating a decline in model accuracy for longer-term forecasts. This observation aligns with the inherent challenge in predicting price movements over extended periods due to increased uncertainty and the influence of unknown market variables.

2. **Anomaly in Error Reduction for the 28-day Horizon**: Intriguingly, there is a decrease in error metrics for the 28-day forecast horizon compared to the 14-day horizon. This suggests that the model may capture certain patterns or trends more effectively over monthly periods, possibly due to recurring cycles or events influencing closing prices. However, this improvement is relative, and the model demonstrates significantly higher precision for shorter horizons.

These trends can also be seen on the graphical representations :

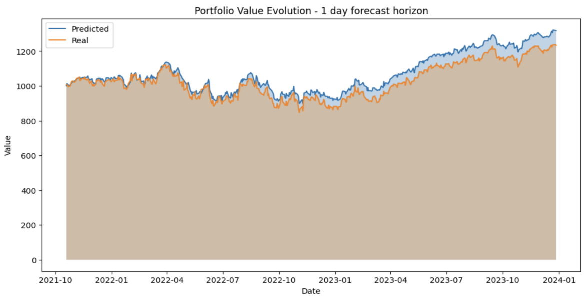

Figure 1: Portfolio Value Evolution - 1 day forecast horizon

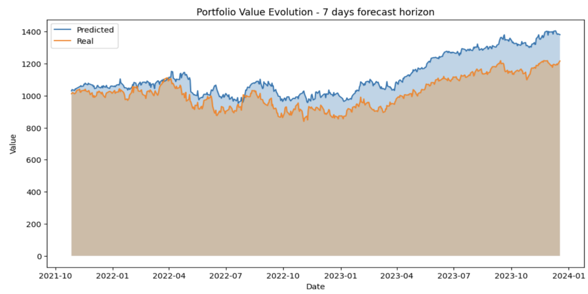

Figure 2: Portfolio Value Evolution - 7 days forecast

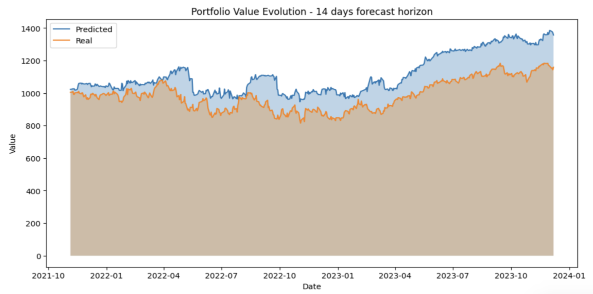

Figure 3: Portfolio Value Evolution - 14 days forecast horizon

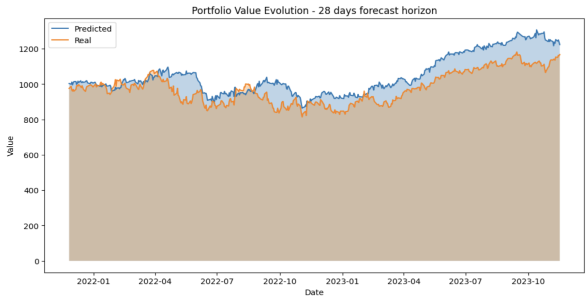

Figure 4: Portfolio Value Evolution - 28 days forecast

These graphical representations provide a visual corroboration of the errors table. For the 1-day forecast horizon, the graph shows a close tracking between the predicted and real values, confirming the model's high precision for short-term forecasts. Moving to the 7-day forecast horizon, a wider divergence between predicted and actual values begins to emerge, especially in periods of high volatility. The 14-day forecast horizon graph exhibits further separation between prediction and reality, particularly where the market shows significant movement. This visual separation is an empirical reflection of the increased forecast errors detailed earlier and highlights the compounded difficulty of predicting further into the future.

Interestingly, for the 28-day forecast horizon, while the previous table indicated a reduction in error compared to the 14-day forecast, the graph does not show a marked improvement in tracking the actual values. The decrease in error metrics may reflect the model's ability to capture broader trends despite missing short-term fluctuations.

2.1.2 Volatility

We used here the same hyperparameters that has been used in the closing prices prediction section. For the volatility, the results are the following :

Errors on Portfolio Volatility	**MSE**	**RMSE**	**MAE**
1 day forecast horizon	0.000003	0.001707	0.001148
7 days forecast horizon	0.000015	0.003894	0.002980
14 days forecast horizon	0.000022	0.004671	0.003554
28 days forecast horizon	0.000016	0.004059	0.002924

Table 2: Metrics table on Portfolio Volatility Forecast using XGBoost

We observe in this table the same phenomenon observed for the closing prices prediction : the accuracy diminishes as the forecast horizon extends from 1 to 14 days. This is typical due to increasing market unpredictability over time. However, there is a slight decrease in error at the 28-day forecast horizon, suggesting the model may be capturing monthly volatility patterns more effectively than bi-weekly fluctuations.

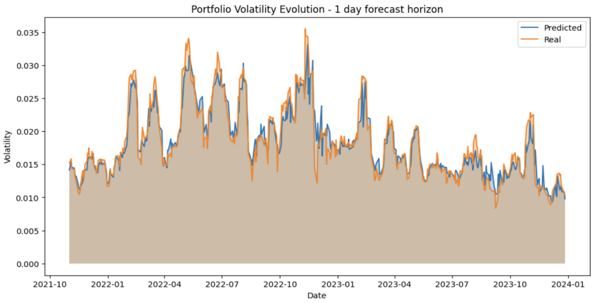

Figure 5: Portfolio Value Evolution - 1 day forecast horizon

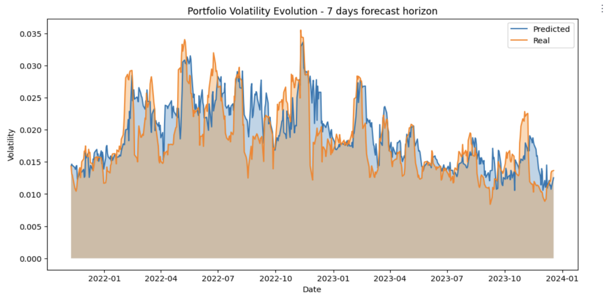

Figure 6: Portfolio Value Evolution - 7 days forecast

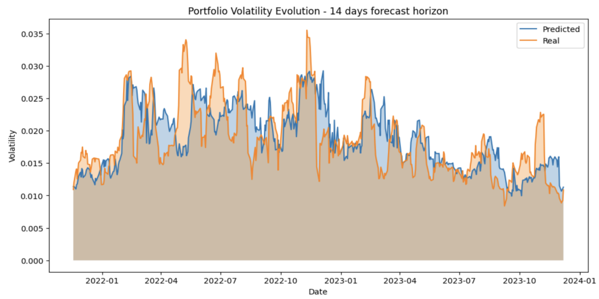

Figure 7: Portfolio Value Evolution - 14 days forecast horizon

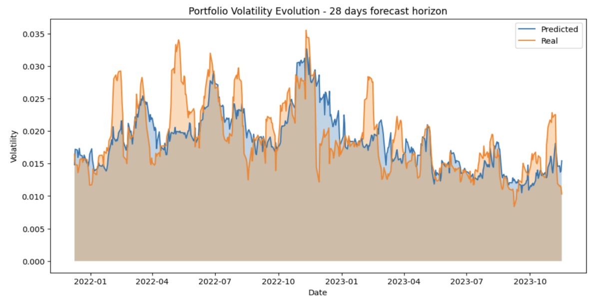

Figure 8: Portfolio Value Evolution - 28 days forecast

These graphs offer a more visual observation of the results obtained on volatility forecasting. This confirms the trend observed in the metrics: the farther out the forecast, the less accurate it becomes. The slight decrease in error for the 28-day forecast is not very noticeable graphically. What is particularly evident is a smoothing of the predicted volatility.

2.2 LSTM

We have just observed that the performance of XGBoost diminishes as the forecast horizon extends, both for volatility and closing prices. What about the LSTM model now?

Long Short-Term Memory (LSTM) networks, a special kind of Recurrent Neural Networks (RNNs), are widely used for time series forecasting. Unlike traditional RNNs, LSTMs are designed to remember long-term dependencies, making them highly effective for sequential data analysis. The key to their functionality lies in a unique structure that includes memory cells and three types of gates: input, forget, and output. These gates control the flow of information, allowing the network to retain or discard data over time. This architecture enables LSTMs to overcome the vanishing gradient problem common in standard RNNs, where the network struggles to learn from earlier inputs in a long sequence. As a result, LSTMs maintain stability in learning and are particularly adept at capturing the intricacies of time series data, such as trends and seasonality, making them ideal for complex forecasting tasks.

2.2.1 Closing Prices

After running our models with the following hyperparameters: 50 epochs, 30 days training sequence length and 80%/20% train-test split, we obtain the following results:

Errors on Portfolio Value	MSE	RMSE	MAE
1 day forecast horizon	600.52	24.5	21.32
7 days forecast horizon	1420.75	37.69	29.53
14 days forecast horizon	3131.11	55.96	47.40
28 days forecast horizon	7993.87	89.41	69.56

Table 3: Metrics table on Portfolio Value Forecast using LSTM

We can clearly observe the same trend as before: the further the forecast horizon extends, the less accurate the model becomes and the more its performance decreases. The accuracy of the predictions can also be visually observed :

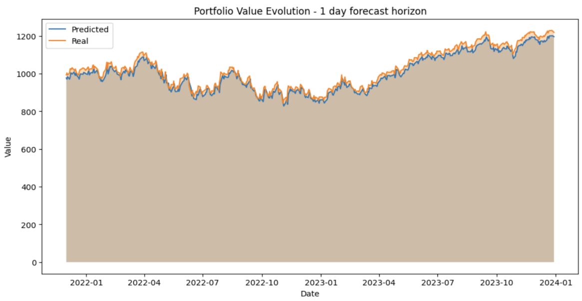

Figure 9: Portfolio Value Evolution - 1 day forecast horizon

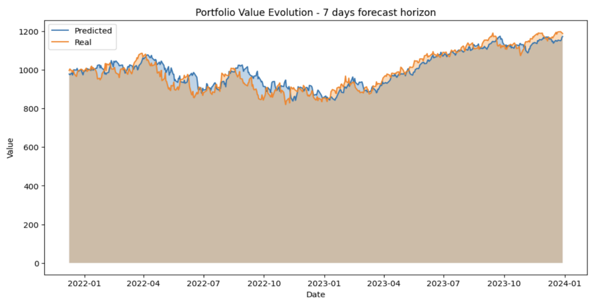

Figure 10: Portfolio Value Evolution - 7 days forecast

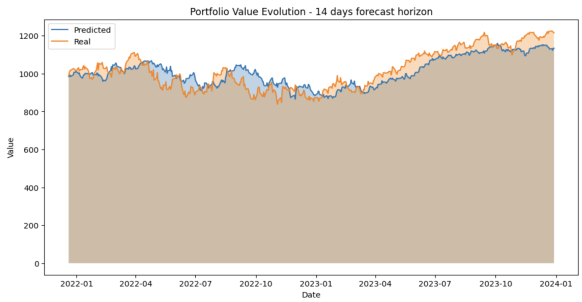

Figure 11: Portfolio Value Evolution - 14 days forecast horizon

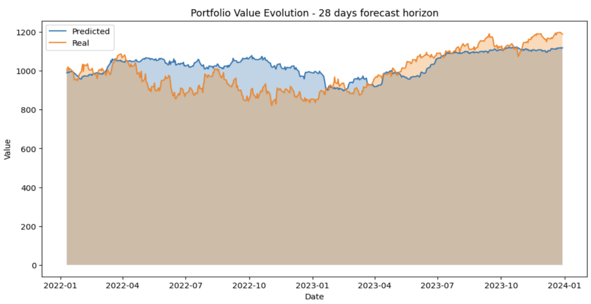

Figure 12: Portfolio Value Evolution - 28 days forecast

The graphical analysis of our LSTM model's predictive performance reveals a clear pattern across various forecast horizons. For the 1-day forecast horizon, the predictions are impressively close to the actual portfolio values, underscoring LSTM's strength in short-term forecasting accuracy. This confirms the widely recognized capability of LSTM models for near-term market predictions.

As we extend the forecast to a 7-day horizon, the predictions remain closely aligned with the actual values, albeit with a noticeable lag. This slight delay in capturing market movements could be attributed to the model's intrinsic reaction time to recent trends.

However, when we examine the 14-day and 28-day forecast horizons, the precision of the predictions substantially decreases. While the LSTM model manages to capture the general direction of the portfolio's trend, there is a significant divergence from the actual values. This drop in accuracy with longer forecast horizons is consistent with expectations, as predicting further into the future inherently comes with increased uncertainty and challenge.

Overall, these observations emphasize the LSTM model's proficiency in capturing market behavior over short periods while highlighting the complexities involved in longer-term financial forecasting.

2.2.2 Volatility

Regarding the prediction of volatility, we ran our models using the same hyperparameters as for the closing price forecasts: 50 epochs, a training sequence length of 30 days, and an 80%/20% train-test split. We obtained the following results:

Errors on Portfolio Volatility	**MSE**	**RMSE**	**MAE**
1 day forecast horizon	0.000003	0.001638	0.001072
7 days forecast horizon	0.000022	0.004676	0.003488
14 days forecast horizon	0.000035	0.005920	0.004459
28 days forecast horizon	0.000043	0.006588	0.004997

Table 4: Metrics table on Portfolio Volatility Forecast using LSTM

Our analysis of the LSTM model's predictive accuracy on portfolio volatility across varying forecast horizons yields insightful trends. For the 1-day forecast, the model demonstrates remarkable precision with minuscule error rates, evidenced by an MSE of 0.000003, RMSE of 0.001638, and MAE of 0.001072, suggesting that LSTMs are highly adept at short-term market volatility predictions. Moving to a 7-day horizon, there is an uptick in errors, yet they remain modest, implying the model's continued applicability in a weekly forecast scenario. However, as we extend the prediction window to 14 and 28 days, the errors notably increase, with the largest observed at the 28-day horizon (MSE of 0.000043, RMSE of 0.006588, MAE of 0.004997). This incremental growth in prediction errors as the forecast horizon extends indicates the challenges inherent in modeling financial market volatility over longer periods. Despite the increase in errors, the LSTM model's performance in predicting longer-term volatility shows potential, albeit with diminished accuracy, capturing the broader trends if not the precise fluctuations. These findings underscore the strength of LSTM models in short-term forecasting while also marking the boundaries of their effectiveness as the prediction horizon expands.

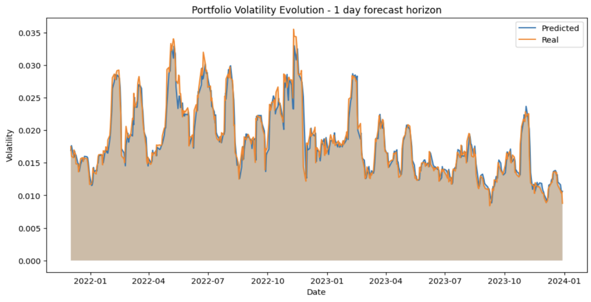

Figure 13: Portfolio Volatility Evolution - 1 day forecast horizon

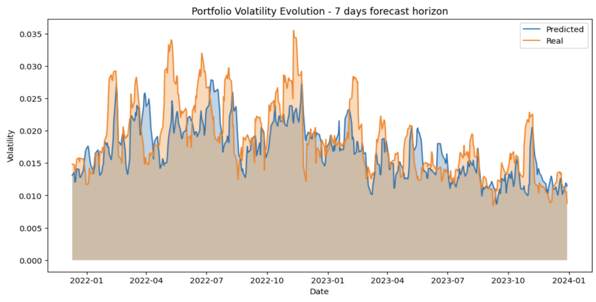

Figure 14: Portfolio Volatility Evolution - 7 days forecast horizon

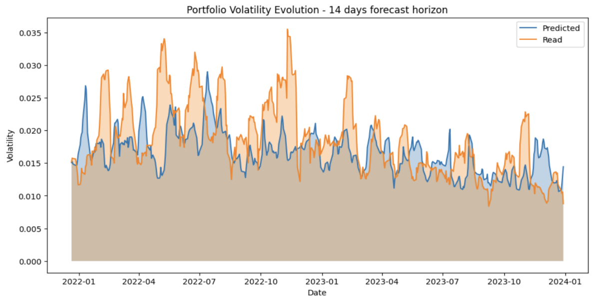

Figure 15: Portfolio Volatility Evolution - 14 days forecast horizon

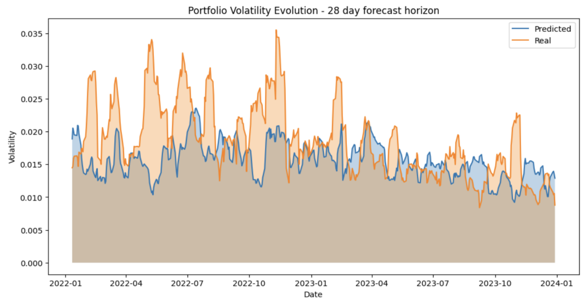

Figure 16: Portfolio Volatility Evolution - 28 days forecast horizon

The graphical analysis of these predictions aligns with the previously discussed error metrics, elucidating the relationship between increased forecast horizons and diminished accuracy. For the 1-day horizon, the model's predictions tightly follow the actual market volatility, as reflected in the low error rates—MSE, RMSE, and MAE. This demonstrates the LSTM's strength in short-term forecasting. Progressing to a 7-day horizon, although the predictions still closely track real volatility, the errors begin to widen, showing slight delays in response to market changes. The 14-day predictions further accentuate this trend, with the model missing more nuanced volatility shifts, despite capturing the overall market direction. The 28-day horizon exhibits the most considerable discrepancy, with the LSTM model's predictions smoothing over the actual volatility and missing the finer details, highlighted by the highest error values.

The visual and numerical evidence together suggest that while LSTMs are adept at short-term volatility forecasting, their effectiveness wanes as the prediction interval lengthens, struggling to navigate the complex and unpredictable long-term market behaviors.

3 Impact of varying the period split for training

In this section, we will discuss the impact of period splits on the performance of our models. Specifically, we have a 10-year history (from January 2, 2013, to December 29, 2023) for all our stocks. For training the LSTM model, we will examine three different cases: an 80% training, 20% testing split; a second split of 50% training, 50% testing; and finally, training on data from 2022 for testing on 2023. The rationale behind these varied splits is to explore the effects on model performance when the training data is altered. For XGBoost, given that its rolling window mechanism does not imply a train/test split analogous to that of the LSTM, we will vary the training window size between 30, 60, and 90 days.

3.1 XGBoost

3.1.1 Closing Prices

We thus ran three different models using the same hyperparameters as described in Section 2.1.1, with the only variation being the size of the rolling training windows. The forecast horizon was set to 7 days. We calculated the prediction errors for each model, which we have summarized in the following table:

Errors on Portfolio Value	MSE	RMSE	MAE
30 days	13636.12	116.77	103.83
60 days	24481.43	156.47	134.68
90 days	51453.15	226.83	194.92

Table 5: Metrics table on Portfolio Value Forecast using XGBoost

With a 30-day rolling window, the model shows relatively moderate error values, suggesting a reasonable balance between capturing short-term market trends and responding to recent price movements. This window size appears to be effective for the XGBoost algorithm in navigating the complexities of financial data, allowing for a nuanced understanding of market dynamics over a one-month period.

However, as the window size increases to 60 days, all error metrics rise significantly. It may be due to the model incorporating older data points, which could dilute the impact of more recent and relevant market trends on the forecast, leading to less accurate predictions. The errors becomes more pronounced with a 90-day window. As the training window expands, the model's ability to accurately predict portfolio values diminishes. This could be attributed to the inclusion of too much historical data, potentially making the model less responsive to short-term market fluctuations and more susceptible to overfitting to past trends that may not be indicative of future movements.

These findings underscore the importance of optimizing the size of the rolling training window when using machine learning models like XGBoost for financial forecasting. They reveal a trade-off between the breadth of historical data incorporated into the model and the precision of its predictions.

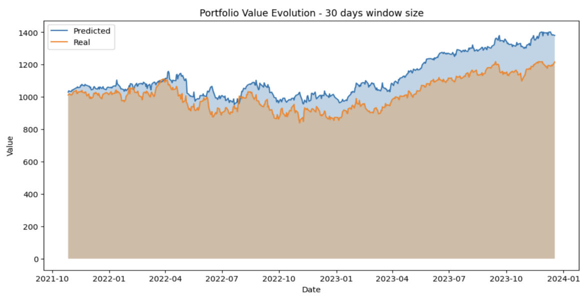

Figure 17: Portfolio Value Evolution - 30 days window size

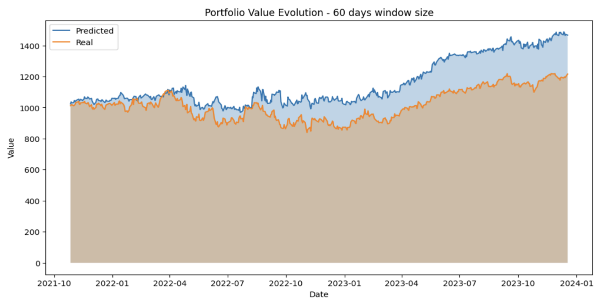

Figure 18: Portfolio Value Evolution - 60 days window size

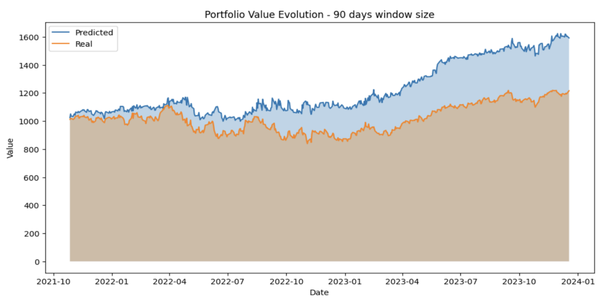

Figure 19: Portfolio Value Evolution - 90 days window size

3.1.2 Volatility

We used the same approach for the portfolio volatility forecast, the errors are the following :

Errors on Portfolio Volatility	MSE	RMSE	MAE
30 days	0.000015	0.003894	0.002980
60 days	0.000016	0.004045	0.003157
90 days	0.000017	0.004086	0.003251

Table 6: Metrics table on Portfolio Volatility Forecast using XGBoost

Initially, for the 30-day window, the model demonstrates commendable accuracy, as indicated by the lowest error metrics (MSE, RMSE, MAE), suggesting its efficacy in capturing short-term volatility patterns within the market. This performance is critical for short-term investment strategies where understanding monthly volatility trends can significantly influence decision-making processes.

Nevertheless, when the window size is expanded to 60 days or 90 days, it is noteworthy that the increase in error metrics is not as pronounced as observed in the portfolio value predictions. This subtlety suggests a relatively stable model performance despite the extended forecasting period. What about it graphically?

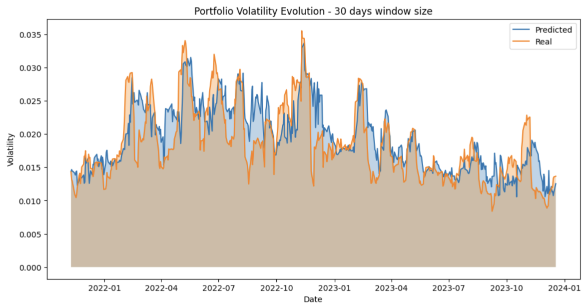

Figure 20: Portfolio Volatility Evolution - 30 days window size

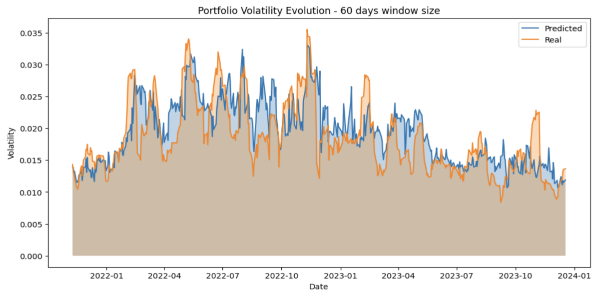

Figure 21: Portfolio Volatility Evolution - 60 days window size

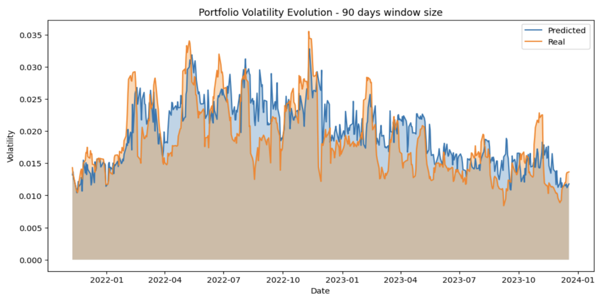

Figure 22: Portfolio Volatility Evolution - 90 days window size

In the 30-day window graph, there's a noticeable alignment between the predicted and actual volatility, with the model capturing the peaks and troughs reasonably well. This close tracking is consistent with lower error metrics and suggests that the model is quite reactive to changes in volatility over a short-term period.

Moving to the 60-day window graph, the model still follows the general trend of actual volatility, but discrepancies begin to appear. These are moments where the predicted values either overestimate or underestimate the actual volatility, but the overall pattern remains in sync.

The 90-day window graph shows a continuation of this trend, with predicted volatility often lagging or leading the actual figures. The peaks and troughs are less pronounced in the predicted data, suggesting that the longer window size may be diluting the model's sensitivity to short-term volatility spikes and dips.

Across all three graphs, it is evident that as the window size increases, the model's ability to capture rapid changes in volatility diminishes.

3.2 LSTM

3.2.1 Closing Prices

Thus, we ran 3 models with the same hyperparameters : 7 days of forecast horizon, 50 epochs and 30 days training sequence. The 3 splits can be visualized trough a stock, let's take Exxon for example :

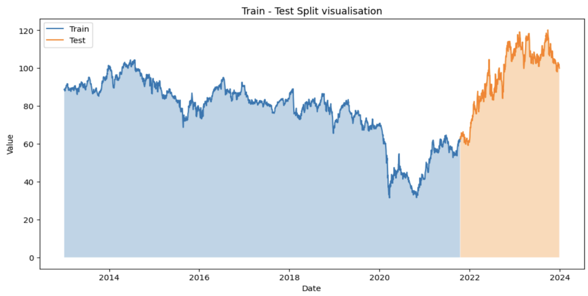

Figure 23: Exxon's Closing Prices - 80%/20% split

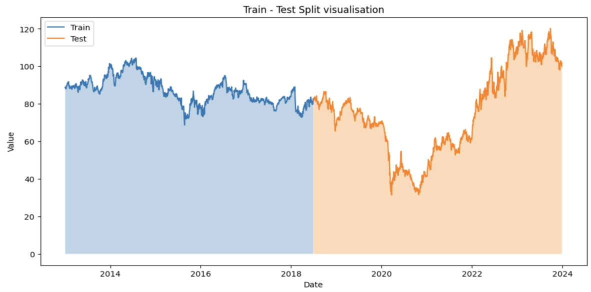

Figure 24: Exxon's Closing Prices - 50%/50% split

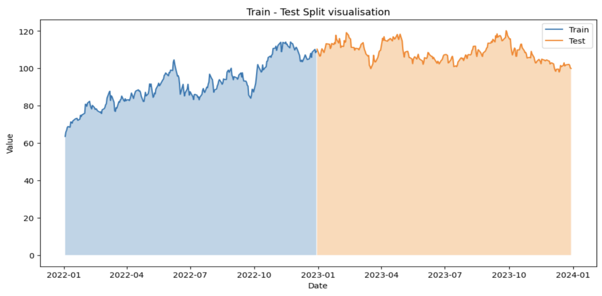

Figure 25: Exxon's Closing Prices - 2022/2023 split

It's crucial to acknowledge the inclusion or exclusion of anomalous events such as the Covid-19 crisis in the training splits, which can significantly affect model performance. In the 80/20 training/testing split, this period of heightened volatility and market disruption will be part of the training dataset. However, in the 50/50 split, the model will not be trained on data from the Covid-19 crisis, potentially providing insights into the model's performance in more stable times.

The strategic decision to train on 2022 data and test on 2023 data for the LSTM model is a deliberate attempt to avoid the pandemic period, allowing us to assess the model's predictive power in a post-crisis market environment. This approach aims to understand the model's effectiveness in a presumably more predictable and stable market, free from the unprecedented shocks experienced during the pandemic. It also offers an opportunity to evaluate the robustness of the model when such significant external variables are absent from the training data, which could be indicative of the model's adaptability to new market conditions.

After running our 3 models, the results are the following :

Errors on Portfolio Value	MSE	RMSE	MAE
80% / 20% Split	1420.75	37.69	29.53
50% / 50% Split	114155.87	337.87	263.47
2022/2023 Split	3436.22	58.62	49.29

Table 7: Metrics table on Portfolio Value Forecast using LSTM

The comparative analysis of model performance across varying training/testing splits reveals distinct outcomes in error metrics, underscoring the importance of dataset composition in predictive accuracy. An 80%/20% split yields the lowest errors (in terms of MSE, RMSE or MAE), signifying a robust predictive capability when the model is allowed to learn from a substantial historical data set. Conversely, the 50%/50% split incurs a drastic escalation in error values, with MSE and RMSE increasing markedly. This degradation in performance might result from two main reasons : a reduced training dataset and the exclusion of pivotal market events such as the Covid-19 crisis from the training phase.

The 2022/2023 split, designed to train on post-pandemic data, results in intermediate error levels. Although errors are higher than in the 80%/20% scenario, they remain substantially lower than in the 50%/50% split, indicating a moderate predictive proficiency. This particular split's performance could be attributed to the exclusion of the pandemic's anomalous market behavior, which, while stabilizing training data, also removes a potentially valuable context for the model's learning process.

The findings suggest that the incorporation of extensive and event-rich historical data facilitates a more accurate forecast, implying that the scope and quality of training data are critical for model efficacy. The analysis also hints at the potential challenges in model generalizability to new data regimes, particularly when significant market events are omitted during training.

These results can also be analyzed graphically :

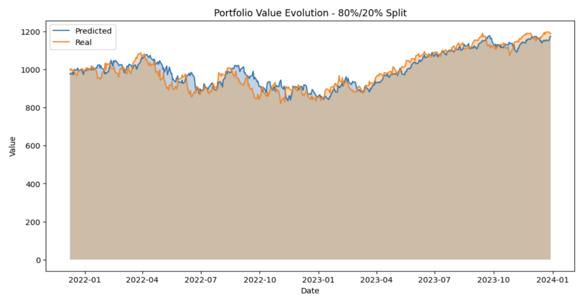

Figure 26: Portfolio Value Evolution - 80%/20% split

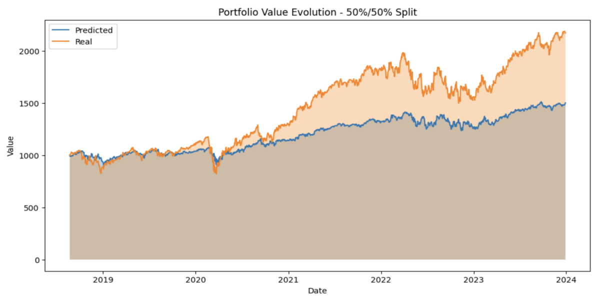

Figure 27: Portfolio Value Evolution - 50%/50% split

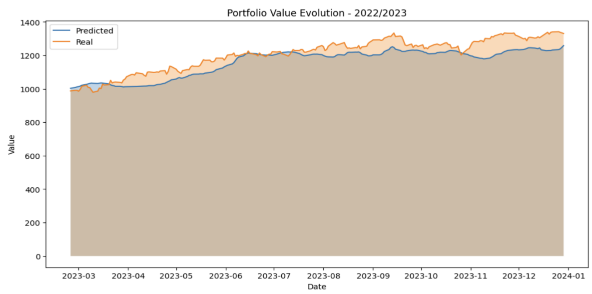

Figure 28: Portfolio Value Evolution - 2022/2023 split

3.2.2 Volatility

We used the same hyperparameters used in the precedent subsection. We can visualize the split on the Exxon stock volatility :

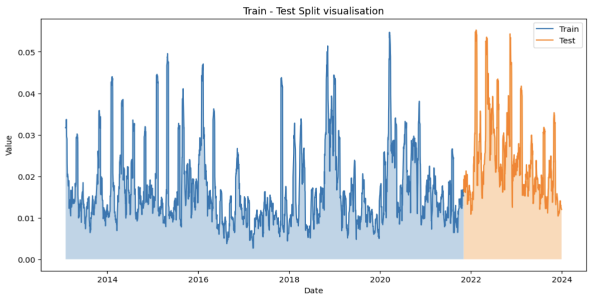

Figure 29: Exxon's Volatility - 80%/20% split

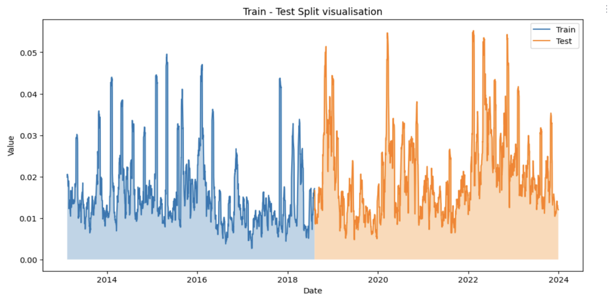

Figure 30: Exxon's Volatility - 50%/50% split

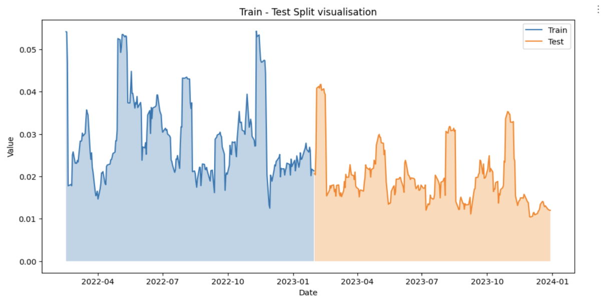

Figure 31: Exxon's Volatility - 2022/2023 split

The Covid crisis is less visually apparent in the volatility graphs than it was in the closing price trends. However, a clear shift in volatility regime between 2022 and 2023 can be observed, which could lead to poorer results. The results are the following :

Errors on Portfolio Volatility	**MSE**	**RMSE**	**MAE**
80% / 20% Split	0.000020	0.004488	0.003382
50% / 50% Split	0.000050	0.007090	0.004551
2022/2023 Split	0.000071	0.008441	0.007413

Table 8: Metrics table on Portfolio Volatility Forecast using LSTM

The 80%/20% Split shows the lowest error values among the configurations, indicating a strong model performance when a substantial portion of the data is allocated for training. This split likely allows the model to capture the underlying volatility patterns more effectively, leading to more accurate forecasts as in the portfolio value prediction subsection.

Conversely, the 50%/50% Split results in a noticeable increase in all error metrics. This suggests that reducing the training dataset's size, thereby limiting the model's exposure to the range of historical volatility, adversely affects its forecasting accuracy.

The 2022/2023 Split presents the highest error values, implying the poorest model performance among the three configurations. This split, designed to assess the model's ability to predict post-pandemic market conditions by training exclusively on data from 2022 and testing on 2023, highlights significant challenges. The elevated errors can be attributed to the model's struggle to adapt to the abrupt shift in market dynamics between the two years, underscoring the impact of recent historical events on forecast accuracy.

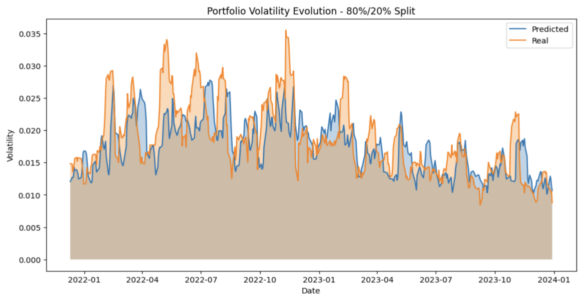

Figure 32: Portfolio Volatility Evolution - 80%/20% split

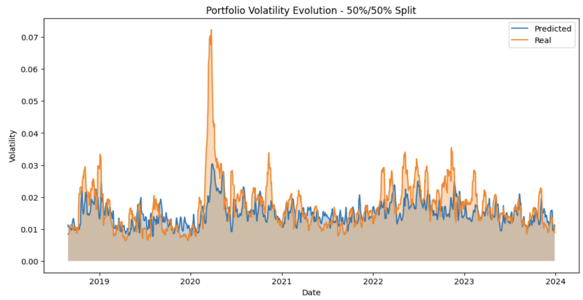

Figure 33: Portfolio Volatility Evolution - 50%/50% split

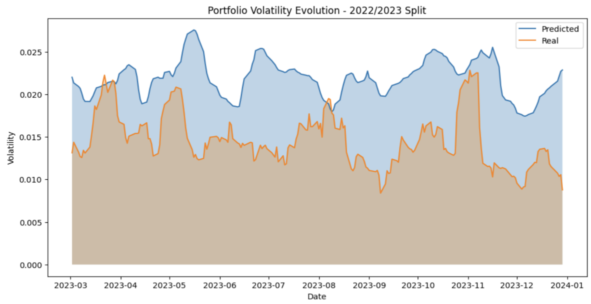

Figure 34: Portfolio Volatility Evolution - 2022/2023 split

Therefore, it is immediately evident that the predictions closely align with actual outcomes in the 80%/20% split. Regarding the 50%/50% split, it is immediately apparent that the model entirely failed to predict the surge in volatility in 2020 associated with the Covid-19 crisis. Aside from this error, it is noticeable that the model still captures the overall trend of our portfolio's volatility well. Finally, as we discussed during the split, the model failed to accurately predict the year 2023 when trained on 2022 data, due to the regime change that is clearly visible in the graphical representation.

This analysis underscores the critical role of training data selection in predictive model performance, especially in the context of financial markets known for their complexity and susceptibility to external shocks. The results suggest that while LSTM models are capable of capturing market volatility to a degree, their performance is heavily influenced by the data split strategy, with more extensive training datasets generally yielding better results.

4 Conclusion

In conclusion, this study provides a comprehensive sensitivity analysis of LSTM and XGBoost models in the context of financial forecasting. Through rigorous testing across various forecasting horizons and data splits, we have delineated the strengths and limitations of these advanced machine learning techniques. The research has clearly demonstrated that while both models exhibit proficiency in short-term predictions, their performance invariably declines as the prediction horizon widens. This effect is attributed to the increased uncertainty inherent in longer-term financial projections.

Our findings indicate that the LSTM model, renowned for capturing temporal dependencies, shows exceptional promise in predicting closing prices and volatility over shorter forecast horizons. However, its efficacy diminishes with the extension of the forecast period, which is reflected in the gradual increase in prediction errors.

XGBoost, with its robust and scalable nature, also performs well for short-term forecasts. Yet, similar to LSTM, it faces challenges in maintaining accuracy over longer horizons. Interestingly, the decrease in error rates at certain longer forecast horizons suggests the models' potential ability to adapt to market dynamics over specific temporal windows.

The variation in model performance across different training and testing splits underscores the critical importance of dataset structuring in predictive modeling. The inclusion of significant market events, such as the Covid-19 crisis, within the training data, has proven to be essential for model robustness. Conversely, the exclusion of such events in the 50%/50% split led to a marked deterioration in model performance, highlighting the necessity for comprehensive data that encapsulates a full spectrum of market conditions.

Ultimately, the study affirms that while machine learning models are powerful tools in financial forecasting, their deployment must be carefully considered. The selection of appropriate model hyperparameters, training/testing splits, and the understanding of each model's unique characteristics are paramount for optimizing predictive performance.

References

[1] Jian Cao, Zhi Li, and Jian Li. "Financial time series forecasting model based on CEEMDAN and LSTM". In: *Physica A: Statistical Mechanics and its Applications* 519 (Apr. 2019), pp. 127–139. ISSN: 0378-4371. DOI: 10.1016/j.physa.2018.11.061. URL: https://www.sciencedirect.com/science/article/pii/S0378437118314985 (visited on 02/02/2024).

[2] Atsushi Inoue, Lu Jin, and Barbara Rossi. "Rolling window selection for out-of-sample forecasting with time-varying parameters". In: *Journal of Econometrics* 196.1 (Jan. 2017), pp. 55–67. ISSN: 0304-4076. DOI: 10.1016/j.jeconom.2016.03.006. URL: https://www.sciencedirect.com/science/article/pii/S0304407616301713 (visited on 02/02/2024).

[3] Aistis Raudys and Edvinas Goldstein. "Forecasting Detrended Volatility Risk and Financial Price Series Using LSTM Neural Networks and XGBoost Regressor". en. In: *Journal of Risk and Financial Management* 15.12 (Dec. 2022). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, p. 602. ISSN: 1911-8074. DOI: 10.3390/jrfm15120602. URL: https://www.mdpi.com/1911-8074/15/12/602 (visited on 02/02/2024).

[4] Jujie Wang, Qian Cheng, and Ying Dong. "An XGBoost-based multivariate deep learning framework for stock index futures price forecasting". In: *Kybernetes* 52.10 (Jan. 2022). Publisher: Emerald Publishing Limited, pp. 4158–4177. ISSN: 0368-492X. DOI: 10.1108/K-12-2021-1289. URL: https://doi.org/10.1108/K-12-2021-1289 (visited on 02/02/2024).

[5] Anita Yadav, C K Jha, and Aditi Sharan. "Optimizing LSTM for time series prediction in Indian stock market". In: *Procedia Computer Science*. International Conference on Computational Intelligence and Data Science 167 (Jan. 2020), pp. 2091–2100. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.03.257. URL: https://www.sciencedirect.com/science/article/pii/S1877050920307237 (visited on 02/02/2024).

[6] Xuan Zhang et al. "AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction". en. In: *IOP Conference Series: Materials Science and Engineering* 569.5 (July 2019). Publisher: IOP Publishing, p. 052037. ISSN: 1757-899X. DOI: 10.1088/1757-899X/569/5/052037. URL: https://dx.doi.org/10.1088/1757-899X/569/5/052037 (visited on 02/02/2024).