COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

XAI FOR VOLATILITY FORECASTING

Ignacio Bayón Jiménez-Ugarte

Madrid, December 2025

UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería ICAI
Master's Degree in Artificial Intelligence

Abstract

This project presents an end-to-end Explainable AI (XAI) case study focused on forecasting financial volatility. Using historical market data for **Tesla (TSLA)**, we model the *difference in realised volatility* using Machine Learning techniques, including Random Forest and LSTM. The core of this study applies XAI techniques to unbox these models, deriving actionable insights for trading risk management and critically evaluating the faithfulness of the generated explanations.

Contents

1 Introduction

Problem & Motivation
Financial markets are defined by volatility. This project addresses the challenge of forecasting the **Difference in Realised Volatility** (RV_{diff}) for **Tesla (TSLA)**. Unlike standard price prediction, forecasting the *change* in volatility allows for better detection of regime shifts.

A central conflict in quantitative finance is the trade-off between accuracy and interpretability. While Deep Learning models (like LSTMs) theoretically capture complex patterns, their "black box" nature makes them risky for deployment. This study employs Explainable AI (XAI) to bridge this gap, using methods like SHAP and Integrated Gradients to validate whether models (Random Forest vs. LSTM) are learning genuine market dynamics or merely overfitting to noise.

Stakeholders
The transparency provided by this framework serves:

- **Risk Managers:** To validate predictions against market fundamentals before adjusting VaR limits.
- **Quantitative Traders:** To identify specific technical drivers (e.g., RSI) behind signals for hedging.
- **Regulators:** To ensure algorithmic accountability and prevent reliance on spurious correlations.

2 Data and Methods

2.1 Dataset and Feature Engineering

We utilize daily OHLC data for **Tesla Inc. (TSLA)** (Jan 2015–Present). To prevent data leakage, we applied a strict temporal split: **Train (70%)**, **Validation (15%)**, and **Test (15%)**.

2.1.1 Target: Differential Volatility

The forecasting target is the **Difference in Realised Volatility** ($\Delta RV = RV_t - RV_{t-1}$). Initial experiments forecasting absolute RV_t failed to beat naive baselines due to high autocorrelation. By targeting the *difference*, we stationarize the series, forcing the model to learn the *drivers of change* (market shocks) rather than simply memorizing the previous day's value.

2.1.2 Input Features

We constructed 8 features to capture volatility and momentum dynamics with a $T = 30$ day window:

- **Volatility:** Lagged RV and Yang-Zhang Volatility.[1]
- **Momentum:** RSI (`RSI_14`, Relative Strength Index) and MACD (Moving Average Convergence/Divergence) for overbought/oversold detection.
- **Dynamics:** Bollinger Band Width (`BB_Width`) for compression/expansion.
- **Price:** Log Returns of Close, Close/Open (`Log_CO`), and High/Low (`Log_HL`).

2.2 Modelling Strategy

2.2.1 Random Forest

We implemented a Random Forest Regressor where the 30-day window is flattened into a vector ($30 \times 8 = 240$ inputs). Hyperparameters were optimized using **Time Series Cross-Validation** to respect temporal order. The final configuration uses: `n_estimators=300`, `max_depth=20`, and `min_samples_leaf=5` (to prevent overfitting).

2.2.2 Deep Learning: LSTM Network

To capture temporal dependencies without flattening, we implemented an **LSTM** (Long Short-Term Memory) network in PyTorch.

- **Architecture:** Input (30×8) $\rightarrow$ LSTM Layer (Hidden=32) $\rightarrow$ Dropout ($p = 0.3$) $\rightarrow$ Linear Output Head.
- **Training:** Adam Optimizer, MSE Loss, and Early Stopping (Patience=10) on validation loss.

2.3 Explainability Framework (XAI)

We employ two XAI techniques to validate model behavior:

1. **Global Interpretability: SHAP**

We use **SHAP (SHapley Additive exPlanations)** to generate summary plots for the

[1] Formula based on Yang, D. and Zhang, Q. (2000) "Drift-Independent Volatility Estimation", *Journal of Business*. Implementation adapted from `https://www.pyquantnews.com/the-pyquant-newsletter/how-to-compute-volatility-6-ways`.

Random Forest. This game-theoretic approach attributes prediction output fairly among features, allowing us to verify if the model relies on financial fundamentals or spurious correlations.

2. Sequential Interpretability: Integrated Gradients (IG)
For the LSTM, we use **Integrated Gradients**, a gradient-based method satisfying the axiom of *Completeness*. Unlike perturbation methods, IG provides a rigorous "white-box" view of how the network's memory cells react to specific sequential patterns over the 30-day window.

3 Results and Analysis

3.1 Model Performance Evaluation

We evaluated the predictive performance of the Random Forest and LSTM models against a naive baseline, which assumes zero change in volatility ($\Delta RV_t = 0$).

Table 1: Performance Comparison on Test Set (Target: ΔRV)

Model	MAE	RMSE	Improvement vs. Baseline
Naive Baseline	0.0990	0.2078	—
Random Forest	0.0896	0.1897	+9.43%
LSTM	0.0982	0.2053	+0.77%

The **Random Forest** achieved a robust **9.43% improvement**. In contrast, the standard **LSTM** performed marginally better than the baseline (+0.77%).

Hypothesis: Vanishing Gradients
Given the Random Forest's success with the same data, the LSTM's failure appears architectural. We hypothesize that the model suffers from *Vanishing Gradients*, causing it to bias heavily towards recent time steps ($t-1$) while losing the long-term structural patterns ($t-21$) required for this forecast.

To validate this, we introduced an **Attention LSTM** to allow direct access to past states without gradient decay.

Table 2: Performance Comparison between LSTMs on Test Set (Target: ΔRV)

Model	MAE	RMSE	Improvement vs. Baseline
LSTM	0.0982	0.2053	+0.77%
LSTM with Attention	0.0986	0.2069	+0.36%

Contrary to expectations, the Attention mechanism did not improve performance. The increased complexity likely caused the model to "attend" to stochastic noise rather than robust signals, slightly degrading the error metrics.

3.2 Visual XAI Analysis

3.2.1 Random Forest: Reverse-Engineering the Formula

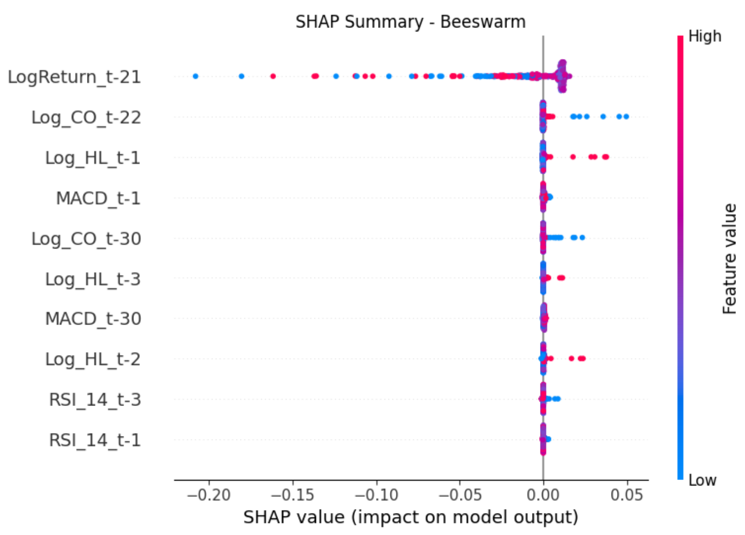

Figure 1: **SHAP Summary (RF)**

The SHAP plot reveals that the Random Forest relies heavily on `LogReturn_t-21`. This is not arbitrary; it reflects the mathematical definition of Realised Volatility (RV), calculated over a 21-day rolling window. The model successfully "discovered" that the change in volatility is mathematically determined by the difference between the new observation (t) and the old one leaving the window ($t - 21$).

3.2.2 Standard LSTM: Evidence of Vanishing Gradients

The attribution map confirms our hypothesis. Despite theoretical access to the past, the model places near-zero weight on the critical $t - 21$ lag identified by the RF. The gradients decay rapidly, restricting the model's focus to at most $t - 10$. Consequently, it fails because it attempts to predict a 21-day cycle using only 10 days of context.

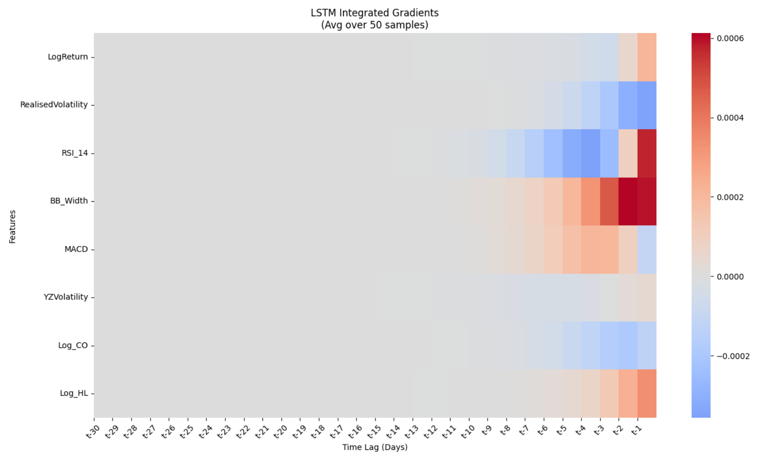

Figure 2: **Integrated Gradients (Standard LSTM).**

3.2.3 Attention LSTM: Signal Dilution

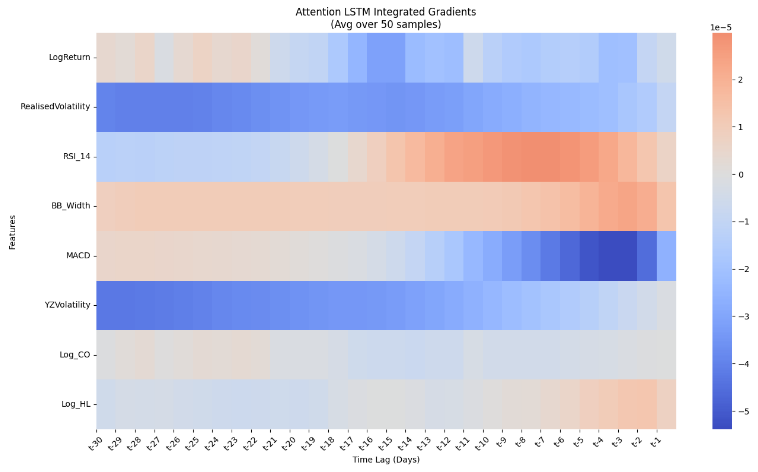

Figure 3: **Integrated Gradients (Attention LSTM)**

The Attention mechanism solved the gradient flow issue, as evidenced by the attribution spread across the window. However, the model fails to *focus*. Instead of pinpointing the deterministic $t - 21$ signal, it distributes importance across the entire month, effectively averaging the volatility regime. While this captures general market dynamics, it dilutes the specific mathematical signal required for accurate forecasting.

Feature Selection Strategy: The analysis highlights redundancy in features like `YZVolatility` (similar to `RealisedVolatility`) and `Log_HL` (similar to `LogReturn`). As Attention LSTM is the only architecture capable of assessing relevance across the entire sequence without the bias of gradient decay (Standard LSTM) or static structural constraints (Random Forest), we rely on its explanations to guide feature selection.

4 Actions and Insights

4.1 Performance of Optimized Models

Following the insights from the previous section, we performed **Explanation-Guided Feature Selection**. We identified the top 5 predictive features (Log Returns, MACD, BB_Width, RSI, and Realised Volatility) based on the Attention LSTM's attribution and retrained all architectures.

Table 3: Impact of Feature Selection on Test Performance

	Full Feature Set		**Reduced Feature Set**		
Model	**MAE**	**RMSE**	**MAE**	**RMSE**	**Change (MAE)**
Naive Baseline	0.0990	0.2078	—	—	—
Random Forest	**0.0896**	**0.1897**	0.0899	0.1900	+0.25%
Standard LSTM	0.0981	0.2051	0.0980	0.2054	-0.08%
Attention LSTM	0.0988	0.2070	0.0985	0.2065	-0.32%

- The **Random Forest** remained remarkably stable, confirming its internal ability to filter noise without manual intervention.
- The **LSTMs** showed negligible improvement. The fact that removing noisy inputs did not significantly lower the error reinforces our conclusion that the limitation is architectural (gradient flow) rather than data-related.

However, it is also important to notice that achieving comparable performance with **37.5% fewer features** (5 vs. 8) is a positive outcome. It demonstrates that the removed variables were indeed redundant, and we have successfully reduced model complexity and training cost without sacrificing predictive power.

4.2 Visual XAI: Why Feature Selection Failed

4.2.1 Random Forest: The "Sniper" Strategy Persists

As SHAP gave almost all of the importance to `LogReturn_t-21`, it was expected that removing non-important features would barely affect performance.

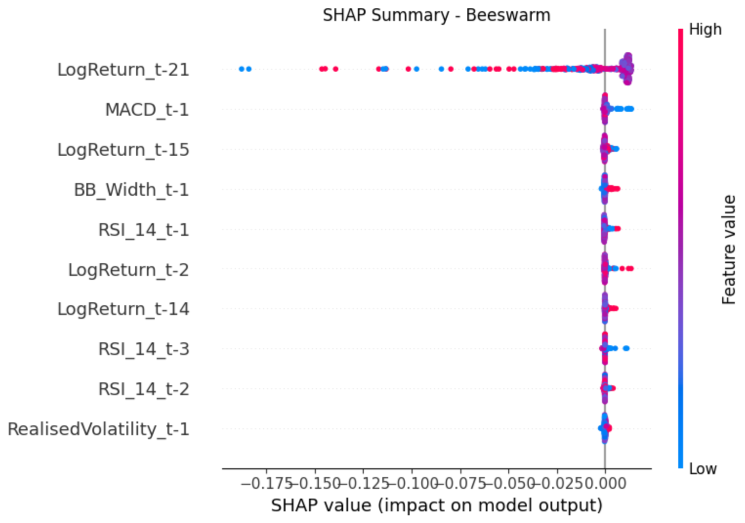

Figure 4: **SHAP Summary (RF with Feature Selection).**

4.2.2 Standard LSTM: The "Recency Trap" Continues

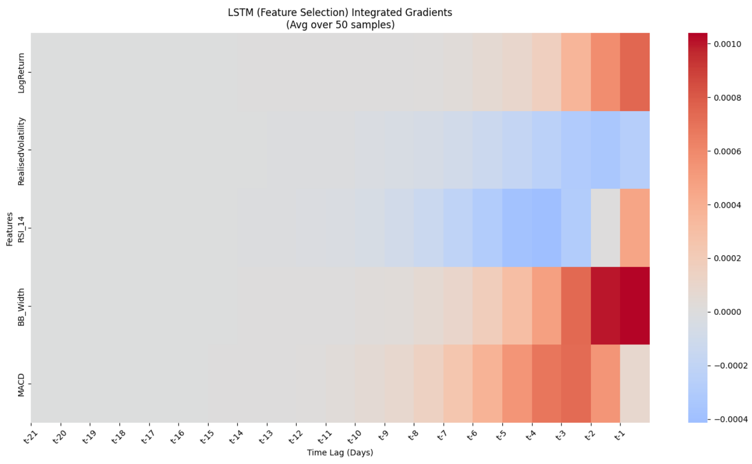

Figure 5: **Integrated Gradients (Standard LSTM).**

Despite removing noisy features, the Standard LSTM remains structurally incapable of reaching the $t{-}21$ signal. The heatmap is entirely grey from $t{-}21$ to $t{-}10$, proving that **Vanishing Gradients** prevent the model from learning the optimal strategy. It is forced to over-rely on `BB_Width` at $t - 1$, explaining why its performance remained stagnant—it was effectively ignoring the extra features regardless of whether they were present or not.

4.2.3 Attention LSTM: The "Wrong Focus"

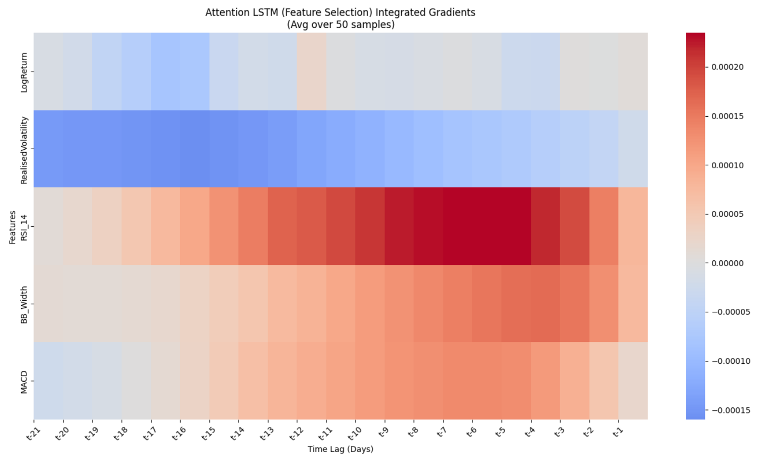

Figure 6: **Integrated Gradients (Attention LSTM).**

The Attention mechanism used the cleaner feature set to find a *new* pattern: a block of intense `RSI` activity between $t-4$ and $t-9$ (Red Block). While this "Weekly Momentum" is a valid signal that allowed the model to maintain parity with the full version, the metrics prove it is **inferior** to the "Monthly Mean Reversion" ($t-21$) found by the Random Forest.

4.3 Domain Recommendations

1. **Simplicity Wins:** For daily stock data ($\approx 2,500$ samples), Random Forests outperform LSTMs because they can capture sparse, long-term dependencies ($t-21$) without the vanishing gradient bottleneck.
2. **Engineer the Cycle:** Since $t-21$ is the dominant predictor, future models should explicitly include a `Return_Lag_21` feature rather than relying on neural networks to learn this subtraction from raw sequences.

5 Discussion and Future Work

5.1 Limitation: The "Arithmetic Artifact"

While the Random Forest achieved the highest accuracy, the XAI analysis reveals a critical limitation. The model effectively "reverse-engineered" the arithmetic definition of the target variable rather than learning genuine market dynamics.

Since Realised Volatility (RV) is calculated over a rolling 21-day window, the change (ΔRV) is mathematically dominated by the difference between the new return (t) and the old return leaving the window ($t-21$):

$$\Delta RV \approx |r_t|^2 - |r_{t-21}|^2$$

The Random Forest simply learned to perform a delayed subtraction: if the return at $t-21$ was small, volatility is likely to rise. While this minimizes RMSE, it renders the solution trivial. The model is acting as a calculator, not a forecaster, which explains why the LSTMs failed—they attempted to learn sequence patterns where there was only a simple arithmetic identity.

5.2 Insights from Neural Architectures

Despite lower performance, the LSTMs provided deeper financial insights when analyzed through Integrated Gradients:

1. Attention LSTM (The "Trader"):
Instead of exploiting the $t-21$ artifact, this model identified a distinct block of high importance on **RSI** between $t-4$ and $t-9$. This indicates the network attempted to learn a "Weekly Momentum" strategy. While this signal was weaker than the arithmetic trick, it validates the Attention mechanism's ability to isolate complex behavioral patterns that simple decision trees miss.

2. Standard LSTM (The "Efficient Skeptic"):
Despite suffering from Vanishing Gradients and being blind to $t-21$, the Standard LSTM matched the Attention model's performance (MAE : 0.0981 vs 0.0988). By relying solely on yesterday's **Bollinger Band Width** ($t-1$), it proved that in high-noise environments, immediate context often contains as much predictive power as the full history. This challenges the assumption that "Longer Memory" is always superior.

5.3 Future Work: Transitioning to Intraday Data

To eliminate the "Rolling Window" artifact, future research must shift to **Intraday Data** (e.g., 5-minute bars).

Currently, the target depends heavily on data leaving the 21-day window. By using high-frequency data, we can calculate volatility *within* a single trading day:

$$RV_{today} = \sum_{i=1}^{N} (r_{5min}^2)$$

This calculation resets daily, removing the dependency on the past. Predicting this target would force Neural Networks to learn genuine microstructure dynamics—such as liquidity drying up or panic selling—rather than simply performing a delayed subtraction.