Modelling Equity Index Dynamics Using a Lightweight Attention-Based Neural Network

A Time-Series Study of the S&P 500

Abstract

This project explores the use of a lightweight attention-inspired neural network to model S&P 500 index dynamics using daily market data from 2018 to 2026. Implemented from scratch in PyTorch, the model is evaluated against linear baselines under two distinct prediction tasks: next-day price-level forecasting and next-day return forecasting.

A central finding of this study is that these two tasks exhibit fundamentally different levels of difficulty and interpretability. While the neural network achieves moderate performance when predicting price levels (test $R^2 \approx 0.62$), this result is largely driven by trend persistence and does not necessarily imply strong predictive insight. In contrast, return prediction proves substantially more challenging, reflecting the weak autocorrelation and high noise characteristic of daily equity returns.

Comparisons with Ridge regression indicate that increased model complexity does not automatically yield superior performance in this setting, with neural and linear models exhibiting broadly similar behavior on return-based targets. These results highlight the importance of careful target definition, robust baseline comparisons, and realistic expectations when applying neural networks to financial time series. This project is intended as an exploratory and educational exercise, emphasizing methodological rigor and transparent evaluation over trading performance claims.

1. Introduction and Motivation

Financial time series present a challenging modeling environment due to pronounced temporal dependence, structural nonstationarity, and sensitivity to evolving macroeconomic conditions. Classical econometric approaches such as ARIMA models and linear regression offer interpretability and well-understood statistical properties, but often struggle to accommodate nonlinear interactions and regime-dependent behavior observed in real-world markets. In contrast, modern machine learning methods—including neural networks and attention-based architectures—are capable of modeling complex relationships, albeit at the cost of reduced transparency if not carefully constrained and evaluated.

The objective of this project is to examine whether a simple, interpretable neural network incorporating an attention-like gating mechanism can extract meaningful structure from daily S&P 500 data when paired with economically motivated covariates. Rather than targeting trading strategy optimization or portfolio construction—tasks that introduce additional considerations such as transaction costs, execution constraints, and risk management—this study focuses on understanding model behavior, identifying predictive limits, and benchmarking neural approaches against transparent linear alternatives.

The project is inspired by recent practitioner work that applied attention-based neural networks to equity index forecasting using PyTorch. Building on that foundation, this study extends the analysis in several key directions. First, it applies a comparable modeling framework to the S&P 500, the most widely referenced U.S. equity index. Second, it explicitly distinguishes between price-level prediction and return prediction, highlighting the conceptual and statistical differences between these objectives. Third, it incorporates diagnostic analyses of autocorrelation, volatility dynamics, and regime-specific behavior to contextualize

model performance. Finally, it conducts formal comparisons with Ridge regression to assess whether increased model flexibility provides tangible benefits in a noisy financial environment.

This project is deliberately exploratory and educational in nature. The emphasis throughout is on clean experimental design, leakage-aware evaluation, and honest assessment of model limitations, rather than on claims of predictive edge or deployable forecasting performance.

2. Background and Modelling Context

Financial time series forecasting has traditionally relied on linear statistical models such as ARIMA and related extensions, which provide transparent representations of autocorrelation and, in some cases, conditional volatility dynamics. These approaches benefit from strong theoretical foundations and interpretability, making them useful reference points in empirical finance. However, their reliance on linear assumptions and fixed parameter structures limits their ability to accommodate regime shifts, nonlinear interactions, and time-varying relationships that characterize modern financial markets. As a result, classical models are often better suited as benchmarks than as flexible forecasting tools when multiple market covariates are involved.

Neural networks have been explored as alternatives due to their capacity to approximate nonlinear relationships without explicit specification. Early applications using feedforward architectures produced mixed results, with performance gains often failing to generalize out of sample. Subsequent developments such as recurrent neural networks and Long Short-Term Memory (LSTM) architectures were designed to better capture temporal dependencies, but empirical evidence suggests that increased architectural complexity does not consistently translate into robust predictive improvements in financial settings. In practice, apparent gains frequently diminish under stricter evaluation protocols.

More recent work has introduced attention-based components to time-series modeling, motivated by their success in other sequence learning domains. Attention mechanisms allow models to dynamically reweight information, potentially improving representation learning when signals are unevenly distributed across inputs. In this project, attention is used in a deliberately lightweight and constrained manner—as a gating mechanism within a feedforward network—rather than as full temporal attention over input sequences. This design choice prioritizes interpretability and stability over architectural sophistication.

Any attempt to model financial markets must also be interpreted through the lens of the Efficient Market Hypothesis, which posits that publicly available information is rapidly incorporated into asset prices. At short horizons, daily equity returns exhibit very weak autocorrelation, implying limited predictability and a high signal-to-noise ratio. In contrast, price-level series display strong persistence and trend behavior, making them easier to model but also more susceptible to misleading performance metrics. This distinction motivates the dual focus of the present study on both price-level and return prediction, with results evaluated relative to simple baselines and interpreted cautiously. The goal is not to demonstrate exploitable forecasting power, but to understand how different modeling choices interact with the statistical structure of financial data.

3. Data Description and Sources

The dataset consists of daily financial and macroeconomic observations spanning January 2, 2018 to January 29, 2026, covering approximately 2,015 business days prior to feature construction. The primary target series is the S&P 500 closing price, obtained from Yahoo Finance using the yfinance Python API. The S&P 500 serves as a broad benchmark for U.S. equity market performance and is widely used in both academic and practitioner research.

To incorporate macro-financial context, two additional covariates are included. The 10-year U.S. Treasury yield (DGS10), sourced from the Federal Reserve Economic Data (FRED) database, captures market expectations related to interest rates, inflation, and monetary policy. The CBOE Volatility Index (VIX), also obtained from FRED, provides a forward-looking measure of expected market volatility derived from S&P 500 option prices and serves as a proxy for aggregate market uncertainty.

All series are aligned on a business-day calendar using an inner join to ensure temporal consistency across variables. Observations with missing values—primarily arising on days when equity markets were closed but Treasury yields were reported—are removed. After alignment, lag construction, and rolling-window feature engineering, the final analysis dataset contains 1,816 complete daily observations. To preserve the temporal structure of the data and avoid look-ahead bias, the dataset is split chronologically into training, validation, and test sets. The first 70% of observations are used for model training, the subsequent 15% for validation and hyperparameter selection, and the final 15% are held out for testing. This setup mirrors a realistic forecasting scenario in which models are trained exclusively on past data and evaluated on unseen future observations. The validation set is used for early stopping and model selection, while the test set remains untouched until final evaluation.

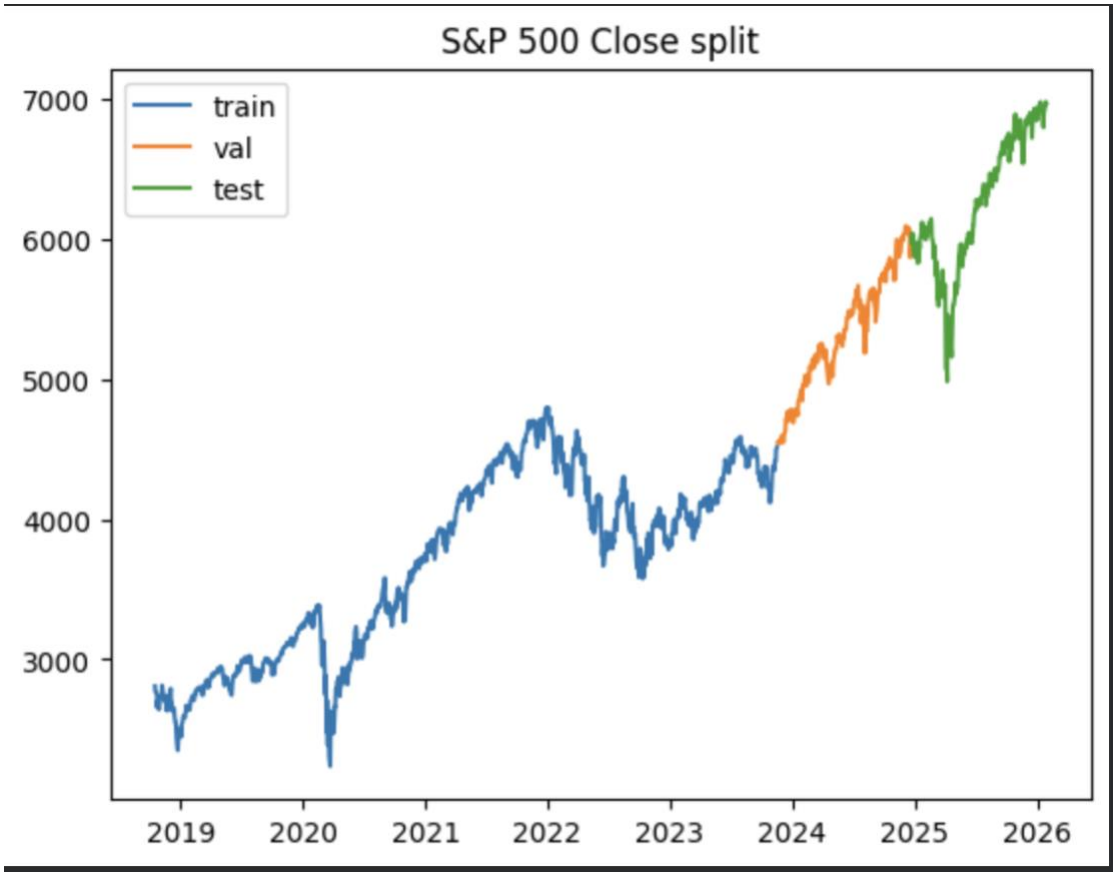


Figure 1. Train–validation–test split for S&P 500 closing prices.

4. Feature Engineering and Economic Motivation

Rather than relying solely on raw price levels—which exhibit strong trends and nonstationary behavior, the model uses a set of engineered features designed to capture complementary aspects of equity market dynamics. Feature construction is guided by standard practices in financial economics and technical

analysis, with an emphasis on transformations that are backward-looking and computable using only information available at or before each time step.

The primary price-based transformation is the logarithm of the S&P 500 closing price. Taking logarithms stabilizes the scale of the series and allows changes to be interpreted approximately as percentage movements rather than absolute differences. This transformation also aligns with the exponential growth patterns commonly observed in long-run equity index behavior. Log prices are used both directly as inputs and as the basis for computing returns and rolling statistics.

To capture short-term dynamics and momentum effects, daily log returns are computed from consecutive log prices. Returns provide a stationary representation of price movements and are used both as a prediction target in the return forecasting task and as an input feature in selected model specifications. Because daily equity returns exhibit weak autocorrelation, these features primarily serve as a baseline representation rather than a source of strong predictive signal.

Volatility-related features are included to reflect time-varying risk conditions. Rolling volatility is computed using a backward-looking window applied to daily returns, capturing volatility clustering commonly observed in financial markets. In addition, the VIX index is incorporated as an external, forward-looking market-implied volatility measure, providing complementary information about investor expectations and uncertainty.

Macroeconomic context is introduced through the 10-year U.S. Treasury yield, which serves as a proxy for prevailing interest rate conditions and broader macro-financial expectations. Changes in long-term yields can influence equity valuations through discount rate effects and are therefore included as contemporaneous covariates.

All rolling statistics and lagged features are computed using strictly historical data, with window endpoints aligned to ensure that no future information enters the feature set. Feature scaling is performed using statistics computed on the training set only and applied consistently to validation and test splits. This design ensures that the feature engineering process does not introduce look-ahead bias or target leakage.

$$log\_sp500_t = ln(sp500\_close_t)$$

Daily returns are computed to represent day-to-day price changes in a form that is approximately stationary and less dominated by long-run trends. Returns are defined as simple percentage changes in the S&P 500 closing price between consecutive trading days. This transformation shifts the modeling task from tracking persistent price levels to capturing short-horizon fluctuations, which are known to be substantially noisier and more difficult to predict.

The resulting return series exhibits low autocorrelation and pronounced volatility clustering, consistent with established empirical properties of equity returns. Figure 2 illustrates the daily return series used in the return-forecasting task. This representation provides a stringent test of model performance, as any meaningful predictive signal must be extracted from a high signal-to-noise environment.

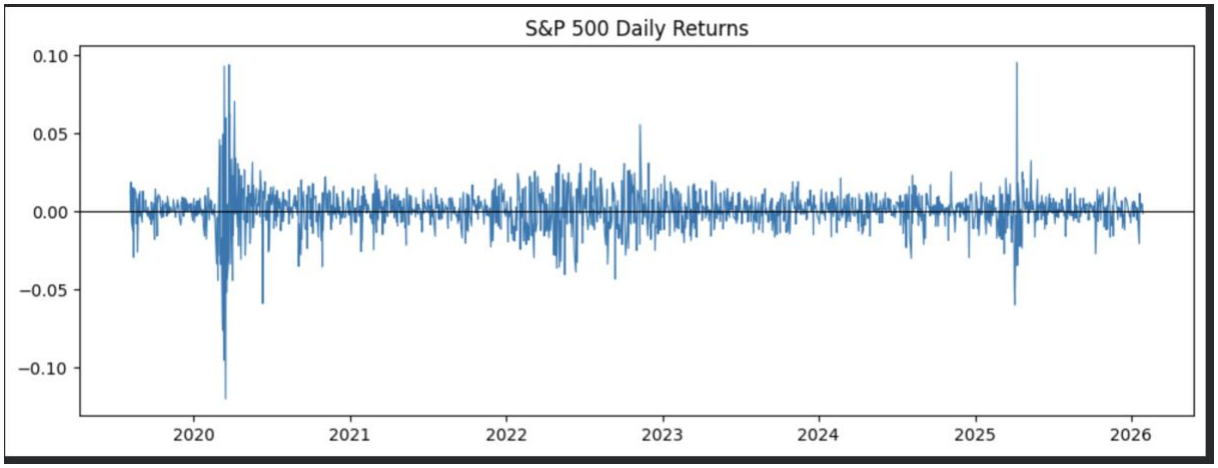


Figure 2. S&P 500 daily log returns (target series).

To capture time-varying risk conditions, realized volatility is computed as the rolling standard deviation of daily returns using a 21-day backward-looking window, corresponding approximately to one trading month. This feature reflects volatility clustering, a well-documented empirical property of financial returns. In addition, the VIX index is included as an external, market-implied measure of expected volatility.

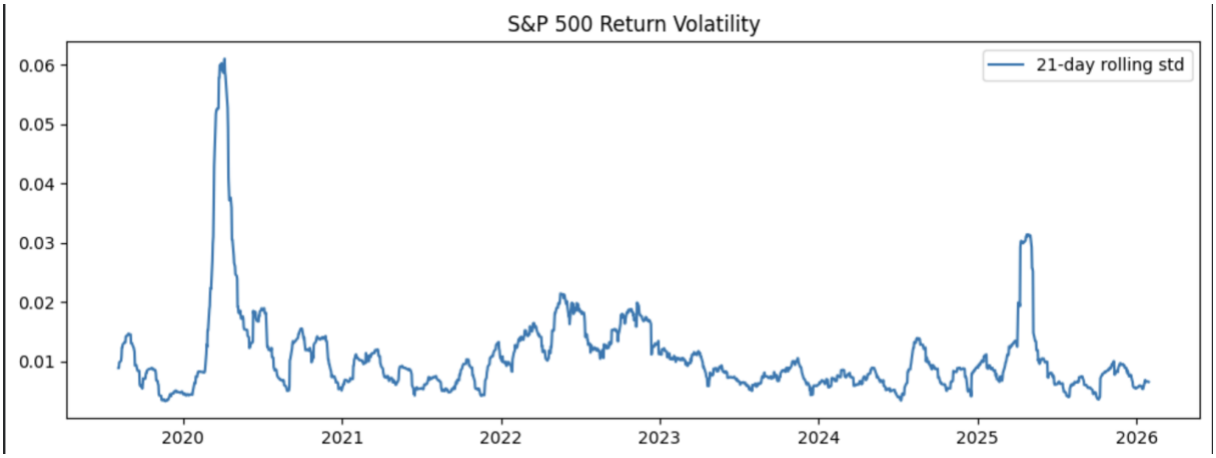


Figure 3. Rolling 21-day volatility of S&P 500 returns.

Trend and momentum information is incorporated through moving-average–based indicators that capture medium- and long-term market dynamics. Simple moving averages over 50-day and 200-day windows are computed from the S&P 500 closing price, corresponding approximately to two months and one year of trading activity, respectively. These indicators are widely used in technical analysis to characterize prevailing trends and smooth short-term fluctuations.

To normalize trend deviations, a relative strength feature is constructed as the ratio of the current price to the 200-day moving average. Values greater than one indicate that the index is trading above its long-run trend, while values below one indicate below-trend pricing. This transformation provides a scale-free measure of trend strength that is comparable across time periods.

In model variants focused on return prediction, additional short-horizon momentum features are included to explicitly encode recent price dynamics. These consist of one-day lagged returns and a five-day rolling average of returns, providing information about very recent market movements that may influence near-term behavior.

All features are standardized to have zero mean and unit variance using scikit-learn's StandardScaler. Importantly, scaling parameters are computed exclusively on the training set and then applied unchanged

to the validation and test sets. This procedure prevents data leakage and ensures that the model is evaluated only on information that would have been available at the time of prediction.

The modeling framework operates on fixed-length rolling windows of historical observations. For each time step ttt, the input to the model consists of the preceding ten days of feature vectors,

$$X_t = \{x_{t-10}, x_{t-9}, \ldots, x_{t-1}\},$$

where each x_{t-i} represents an d-dimensional feature vector (with d=8) in the price-level specification and d=10 in the return-focused specification that includes additional momentum features). A sequence length of ten trading days is chosen as a balance between providing sufficient historical context and maintaining computational efficiency. Longer sequences are explored during hyperparameter tuning but do not yield material improvements.

The prediction target yt is defined as either the next-day closing price or the next-day return, depending on the task. This formulation enforces a strict temporal ordering in which all inputs precede the target, allowing the model to learn temporal patterns without imposing an explicit linear autoregressive structure.

The neural network architecture is intentionally lightweight and interpretable. Input sequences are flattened and passed through a fully connected projection layer that maps the input into a hidden representation of dimension h=128. Batch normalization is applied after this transformation to stabilize training and mitigate internal covariate shift, followed by a ReLU activation to introduce nonlinearity.

A simplified attention-based gating mechanism is then applied. Scalar attention weights are computed via a linear transformation followed by a softmax operation,

$$\alpha = \mathrm{softmax}(W_{\mathrm{attn}} h + b_{\mathrm{attn}}),$$

and applied elementwise to the hidden representation,

$$h' = h \odot \alpha.$$

This mechanism does not perform temporal attention over sequence positions, but instead allows the model to emphasize or suppress different dimensions of the learned feature representation. Dropout with a rate of 0.2 is applied for regularization, and a final linear layer produces a scalar output corresponding to the next-day price or return forecast.

5.3 Training Procedure and Hyperparameters

The model is trained using mean squared error (MSE) as the loss function,

$$L = (1/N)\, \Sigma(y_i - \hat{y}_i)^2$$

which is appropriate for both price-level and return regression tasks. Optimization is performed using AdamW, an adaptive gradient-based optimizer with decoupled weight decay. The learning rate is set to 3×10^{-4} following experimentation over the range $[1\times10^{-4}, 5\times10^{-4}]$

To promote training stability, gradient clipping with a maximum $\ell 2$-norm of 1.0 is applied. While the model does not use recurrent layers, clipping helps guard against occasional large gradient updates that can arise when training feedforward networks on noisy financial data.

Training is conducted for a maximum of 50 epochs using a batch size of 32. During each epoch, the model parameters are updated using the training set, and performance is evaluated on a held-out validation set without gradient updates. The model checkpoint corresponding to the lowest validation loss is retained.

Final evaluation is performed exactly once on the test set using the selected checkpoint. This early-stopping procedure reduces overfitting and ensures that reported test performance reflects genuine out-of-sample behaviour rather than implicit tuning on the test data.

6. Results: Price-Level Prediction

When the target variable is defined as the S&P 500 closing price, the attention-based model achieves moderate out-of-sample accuracy on the test set, as summarized in Table 1.

Metric	Value	Interpretation
MSE	87,612.42	Mean squared error
RMSE	295.99	Root mean squared error (points)
MAE	264.14	Mean absolute error (points)
R^2	0.62	Coefficient of determination

Visual inspection of predicted versus actual prices indicates that the model closely tracks the overall level and medium-term trend of the index over the test period. Figure 4 compares the model's next-day price predictions to realized S&P 500 closing prices.

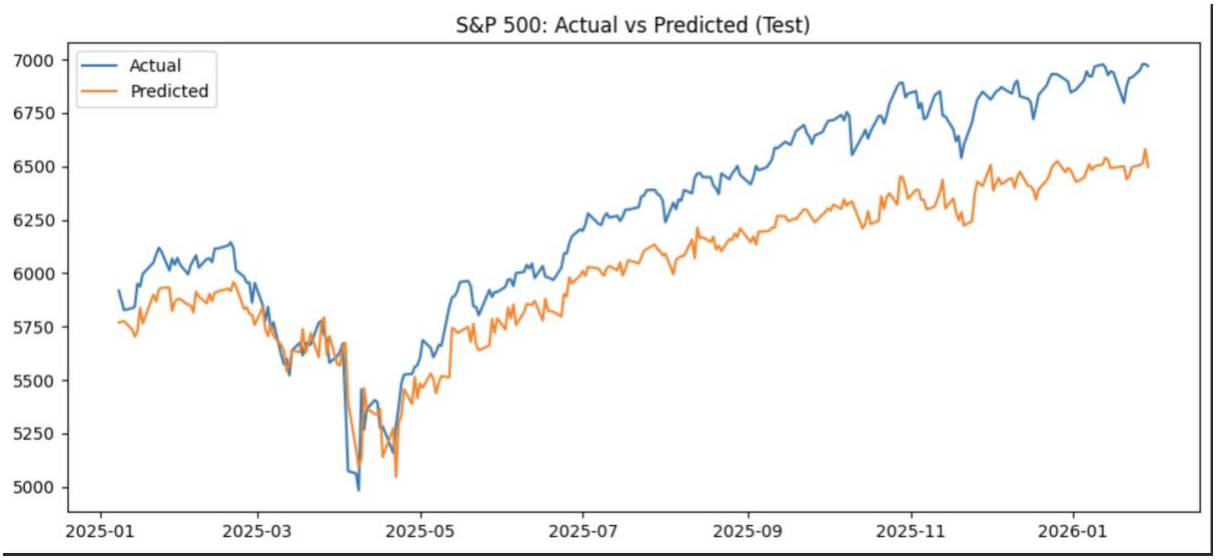


Figure 4. Actual vs predicted S&P 500 closing prices on the test set (reconstructed from predicted returns).

The relatively high R^2 value reflects the strong persistence inherent in equity price series: tomorrow's price is typically very close to today's price. As a result, models that primarily learn trend continuation or near-naive persistence can achieve apparently strong performance under conventional regression metrics. Price-level accuracy alone therefore does not imply economically meaningful forecasting ability.

For this reason, price-level prediction is treated as a diagnostic step that verifies the model's capacity to learn basic structure from the data. To assess whether the model captures information beyond persistence, it is necessary to evaluate its performance on return prediction, where trends are removed and predictability is substantially weaker. This more stringent task is examined next.

7. Results: Return Prediction and Diagnostics

When the model is retrained with daily returns as the prediction target and augmented with short-horizon momentum features (one-day lagged returns and five-day average returns), performance deteriorates substantially relative to price-level forecasting. This outcome is expected given the weak autocorrelation and high noise characteristic of daily equity returns.

Figure 5 provides context by juxtaposing S&P 500 returns with spikes in market-implied volatility as measured by the VIX. Periods of elevated volatility coincide with larger return magnitudes and increased prediction error.

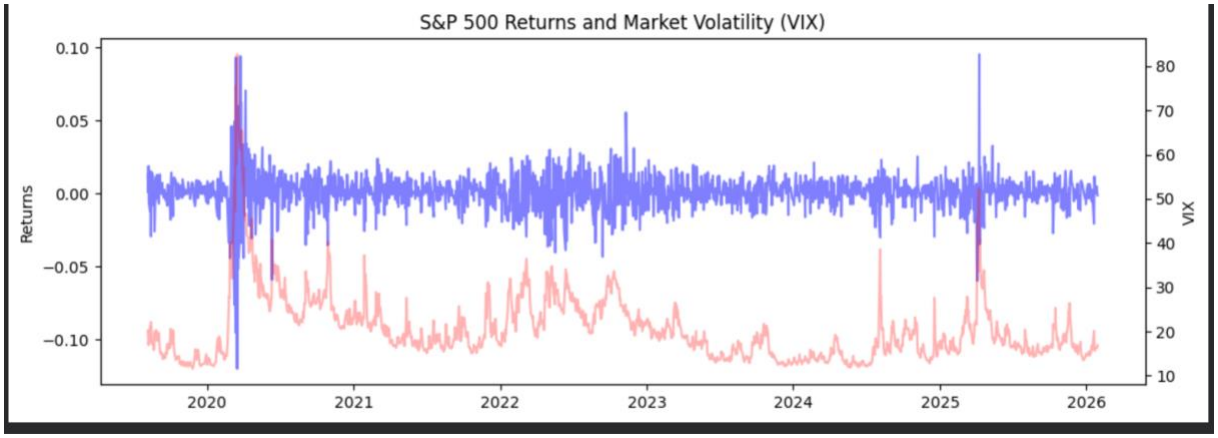


Figure 5. S&P 500 returns and VIX (market-implied volatility) over time.

Autocorrelation analysis confirms that returns exhibit near-zero serial dependence beyond very short lags, consistent with a random-walk–like structure in prices. In contrast, volatility—measured via the rolling standard deviation of returns—shows strong persistence, illustrating

volatility clustering. Figures 6–8 summarize the distributional properties of returns and the contrasting autocorrelation behavior of returns and volatility.

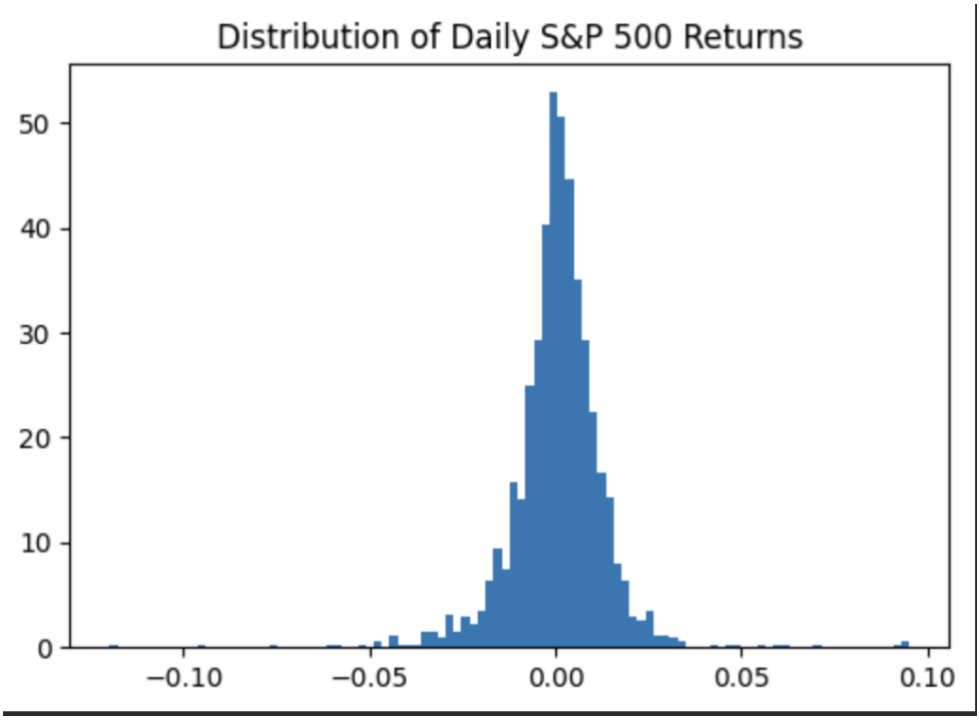
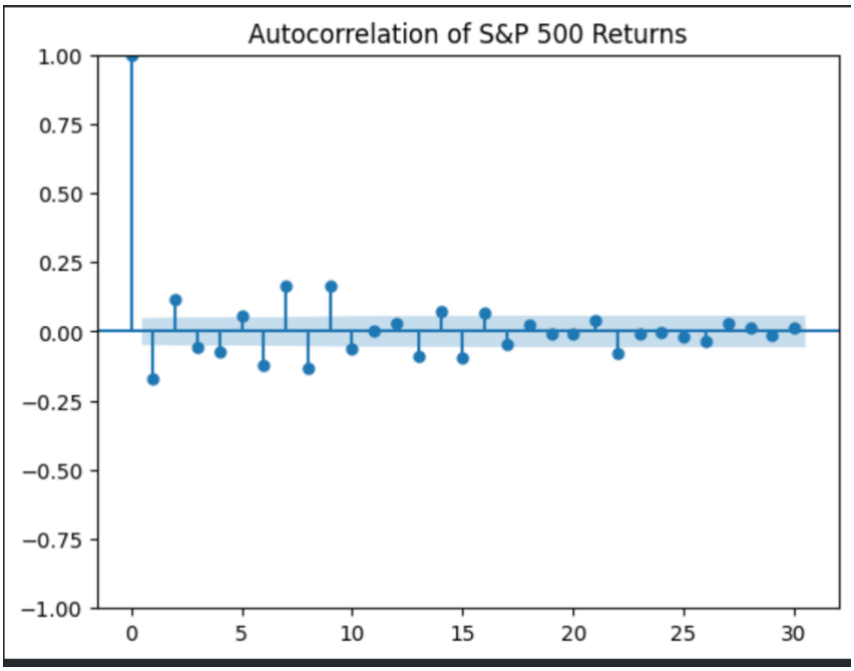


Figure 6. Distribution of daily S&P 500 returns (fat tails vs normal).

Figure 7. Autocorrelation function (ACF) of daily S&P 500 returns.



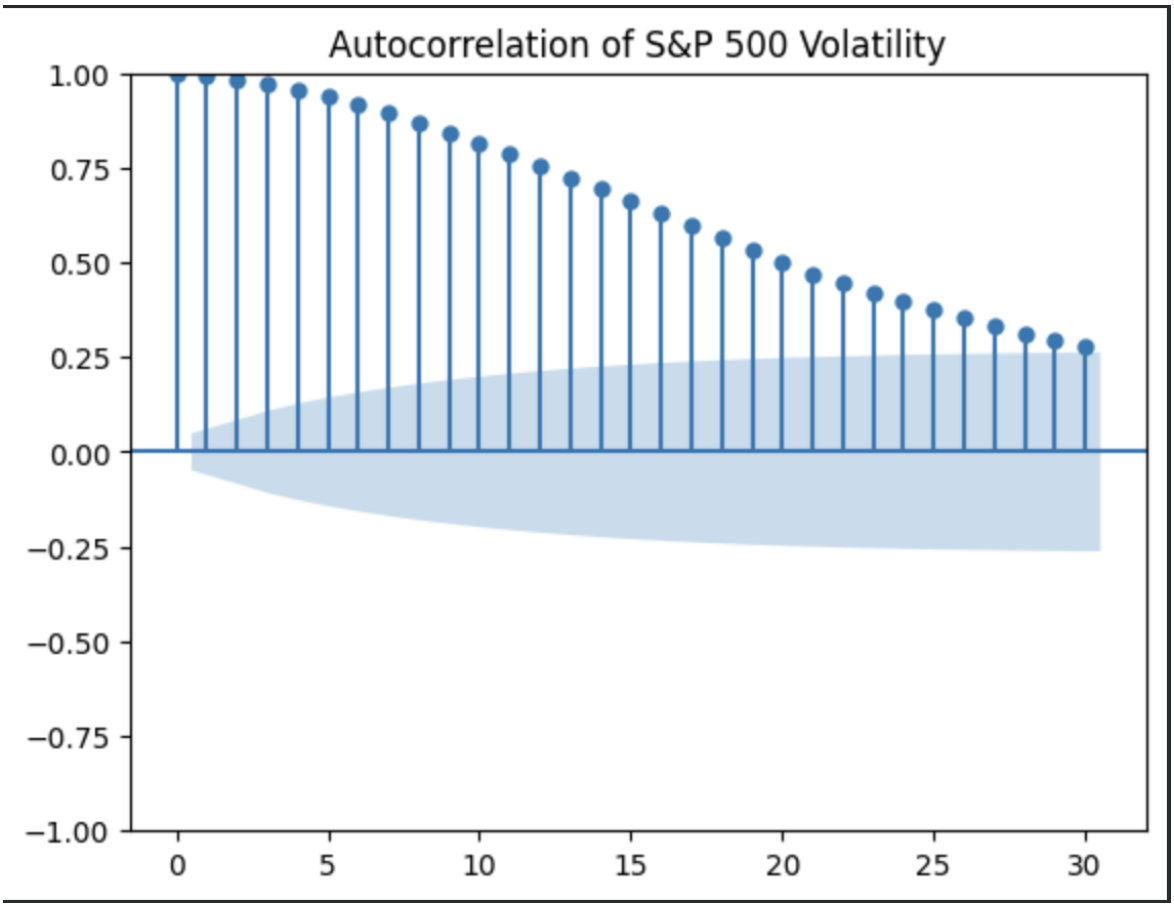


Figure 8. Autocorrelation function (ACF) of rolling S&P 500 return volatility.

Despite the inherent noise in daily returns, several qualitative patterns emerge from the return prediction analysis. Model performance varies substantially across market regimes, with larger prediction errors observed during periods of elevated volatility. In lower-volatility environments, forecasts tend to be more stable, while periods of market stress—characterized by spikes in the VIX—are associated with increased uncertainty and reduced predictability. This regime dependence is consistent with established findings in financial econometrics, where return predictability weakens during times of heightened uncertainty.

Autocorrelation analysis confirms that daily S&P 500 returns exhibit very weak serial dependence beyond extremely short horizons, while realized volatility displays strong persistence. This contrast suggests that while the *direction* of returns remains difficult to forecast reliably, the *magnitude* of price movements exhibits conditional structure. The model's behavior is consistent with this distinction: point forecasts of returns are noisy, but errors increase systematically during high-volatility regimes.

These observations suggest that the network may be capturing aspects of conditional market structure—relationships that vary with market state—even though unconditional return predictability remains limited. Importantly, this interpretation is descriptive rather than causal and does not imply the existence of a stable or exploitable forecasting signal.

Comparison with Linear Benchmarks

To contextualize the neural network's performance and assess whether architectural complexity provides tangible benefits, a Ridge regression model with L2 regularization is trained on the same flattened input sequences. Ridge regression serves as a strong linear baseline due to its robustness to multicollinearity and transparent parameterization.

Across return-based evaluation metrics, the neural network and Ridge regression exhibit broadly similar performance. This finding highlights a central lesson of financial machine learning: increased model complexity does not automatically translate into improved predictive accuracy in environments dominated by noise and weak signal strength. In this setting, much of the limited predictive structure appears to be captured by linear combinations of engineered features.

The primary value of the neural network in this context lies not in superior point forecasts, but in its flexibility to model nonlinear interactions and regime-dependent behavior. Assessing whether this flexibility yields meaningful advantages would require more granular regime-specific evaluation and alternative performance criteria.

8. Limitations and Critical Assessment

Several important limitations must be acknowledged to properly contextualize these findings. First, the attention mechanism employed is deliberately simplified and does not model temporal alignment or sequence-to-sequence dependencies in the manner of full Transformer architectures. While this design choice improves interpretability and stability, it may limit the model's capacity to capture richer temporal structure.

Second, evaluation is conducted using a single chronological train–validation–test split rather than a full walk-forward validation framework. A rolling retraining and evaluation scheme would more closely approximate real-world deployment and provide a more robust assessment of stability over time. Hyperparameter tuning is also limited in scope, and no formal statistical inference is conducted to assess the significance of differences between models.

The evaluation focuses on statistical accuracy metrics rather than economic utility. No trading strategies, transaction costs, or portfolio constraints are considered, and strong statistical performance would not necessarily translate into profitable trading outcomes. In addition, while features are economically motivated, no formal feature importance or attribution analysis is performed.

Finally, the analysis is limited to a single asset class over a period that includes major structural disruptions, including the COVID-19 market shock. Model behavior during this interval may not generalize to other assets or more typical market environments.

These limitations are consistent with the project's explicitly exploratory and educational objective. The goal is not to claim superior forecasting ability, but to understand the capabilities and boundaries of attention-based models when applied to financial time series.

9. Conclusion and Future Directions

This project examines the application of a lightweight attention-based neural network to S&P 500 forecasting under two distinct targets: price levels and returns. The results highlight a fundamental distinction between these tasks. Price-level prediction benefits from strong persistence and trend

continuation, yielding apparently strong performance under conventional metrics, while return prediction remains substantially more challenging due to weak autocorrelation and high noise.

Key takeaways include the importance of strong linear baselines, careful feature engineering, and realistic evaluation protocols. The comparison with Ridge regression demonstrates that increased architectural complexity does not guarantee superior performance in noisy financial environments. Instead, disciplined experimental design and honest interpretation play a more critical role than model sophistication alone.

Future work could extend this analysis through walk-forward validation, alternative architectures, feature attribution methods, regime-specific modeling, and uncertainty quantification. Evaluating economic utility through realistic backtesting frameworks would further clarify the practical relevance of such models. Overall, this project reinforces the need for methodological rigor and humility when applying machine learning techniques to financial markets.

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.

Caddeo, M. (2024). Predicting NASDAQ using a Time Series Attention Model built from scratch with PyTorch. *Medium*. Retrieved from https://medium.com/@ManueleCaddeo/predicting-nasdaq-using-a-time-series-attention-model-built-from-scratch-with-pytorch-60215c3742fe

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383-417.

Fama, E. F., & French, K. R. (1988). Permanent and temporary components of stock prices. *Journal of Political Economy*, 96(2), 246-273.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669.

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

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning*, 448-456.

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3), 215-236.

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, 7, 108014-108022.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *Proceedings of the International Conference on Learning Representations*.

Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*. Springer.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13(3), e0194889.

Qi, M., & Zhang, G. P. (2008). Trend time-series modeling and forecasting with neural networks. *IEEE Transactions on Neural Networks*, 19(5), 808-816.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. *Applied Soft Computing*, 90, 106181.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.