Comparing LSTM and Transformer Architectures for Stock Price Forecasting

Ninad Alurkar (MS Robotics Engineering)
University of Michigan – Dearborn, Michigan, US

ABSTRACT

Forecasting financial time series remains a challenging problem due to volatility, noise, and long-range dependencies in stock market data. In this work, we implement and compare two deep learning architectures for multi-step stock price prediction: a Long Short-Term Memory (LSTM) network as a baseline and a Transformer model with attention as an advanced variant. Using historical data from multiple equities (AAPL, TSLA, MSFT, NVDA), I pre-processed OHLCV features and technical indicators (MACD and signal line), apply scaling, and construct input sequences of 60 trading days to forecast horizons of 15 and 30 days. Both models are trained with early stopping and evaluated on GPU for efficiency. Results are presented through actual vs. predicted plots and error metrics, highlighting differences in performance across stocks with varying volatility. The LSTM demonstrates stable short-term forecasting, while the Transformer captures longer dependencies but exhibits sensitivity to volatility. This study contributes a comparative analysis of recurrent and attention-based architectures in financial forecasting, supported by a literature review of state-of-the-art methods, and provides insights into the strengths and limitations of deep learning approaches for stock prediction.

1. INTRODUCTION

Stock price forecasting has remained one of the most complex and high-stakes challenges due to the volatile and non-stationary nature of financial time series. The market is impacted by a large variety of unpredictable factors, including macroeconomic trends, investor sentiment, and external shocks, which makes it difficult to model with more traditional statistical techniques. In this regard, the use of classical approaches like ARIMA and exponential smoothing relies on linear and stationarity assumptions that restrict an appropriate modelling of nonlinear and dynamic patterns intrinsic to stock data.

Recent advances in deep learning offer promising alternatives, enabling models to learn complex temporal dependencies directly from raw data. Recurrent architectures, such as Long Short-Term Memory networks, have been successful at capturing short-term trends, while attention-based models like Transformers are designed to be more adept at long-range dependencies. However, only a few comparative studies of these architectures in the context of multi-step stock forecasting have been conducted.

The project tries to bridge that gap by implementing and evaluating two deep learning models: an LSTM as the baseline, and a Transformer with attention as an advanced variant. Both models are trained to predict future closing prices using a 60-day input window and forecast horizons of 15 and 30 trading days. Input features include OHLCV data and technical indicators, MACD, and a signal line, scaled and structured into supervised sequences.

Contributions of this project include:

1. Implementation of two architectures for time series forecasting and their comparison
2. The application of multi-feature preprocessing in order to enhance input representation
3. Evaluation of model performance on different stocks with different volatility profiles, such as AAPL, TSLA, MSFT, and NVDA
4. Analysis of the dependence of prediction accuracy on the length of the forecast horizon. This paper provides insights into the strengths and limitations of both recurrent and attention-based models in financial forecasting and contributes to the general understanding of deep learning applications in time series analysis.

2. LITERATURE REVIEW

Time series forecasting has long been a central challenge in financial modelling. Early approaches relied on statistical methods such as ARIMA (Autoregressive Integrated Moving Average) and exponential smoothing. These models assume linearity and stationarity, which do not often hold for the complex nonlinear dynamics of financial markets. In this respect, naive persistence models carry forward the last observed value and serve as the simplest baselines but are unable to provide predictive power in volatile conditions.

To overcome these shortcomings, deep learning models have been adopted, which can learn these temporal dependencies directly from the data. Among them, Long Short-Term Memory networks have established themselves as one of the fundamental architectures in time series forecasting. LSTMs are designed to maintain long-term information using their gated memory cells and thus can be very effective in sequential data where temporal relationships exist. Kabir et al. (2025) proposed a hybrid LSTM–Transformer model for financial forecasting. They showed that LSTMs tend to work well on short-term predictions and smaller datasets.

More recently, Transformer models with attention mechanisms have emerged as strong alternatives. First proposed for natural language processing, Transformers use self-attention in order to model dependencies between all time-steps simultaneously, thereby allowing for better handling of long-range dependencies. Cai and Huang (2023) compared LSTM and Transformer models for multi-step time series forecasting and demonstrated that Transformers perform significantly better than LSTMs on noisier datasets and longer horizons. Bilokon and Qiu (2023) extended this comparison to electronic trading and showed that while LSTMs remain predominant in the short-term financial prediction, Transformers offer considerable advantages in modelling complicated market structures.

Volatility has also been explored as part of comparative research. Sivakumar et al. (2025) critically analysed LSTM and Transformer for financial time series forecasting and pointed out that the LSTMs tend to smooth out the volatility, while Transformers capture abrupt changes more effectively due to attention-based architecture. Ruiter.ai (2025) stressed that model selection depends on data characteristics: For stable and short-period time series, LSTMs should be preferred, while Transformers can effectively perform for long-range patterns in volatile conditions.

These results are in line with the design of my project: we have implemented both LSTM and Transformer models for the stock price predictions of four equities, namely AAPL, TSLA, MSFT, and NVDA, exhibiting different volatility profiles. We extend the existing literature that compares the performances of various deep learning architectures for financial time series forecasting on 15-day and 30-day horizons. My implementation closely connects to state-of-the-art methods and thereby reflects the current research trends in attention mechanisms, multi-step prediction, and volatility modelling.

3. DATASET AND PRE-PROCESSING

The sample data were collected from Yahoo Finance through the yfinance Python library, which allows programmatic access to stock market history. I chose four different equities (any listed stock can be chosen): Apple - AAPL, Tesla - TSLA, Microsoft - MSFT, and NVIDIA - NVDA. Such a selection will provide heterogeneity in terms of volatility and market behavior. Apple and Microsoft represent relatively stable, large-cap technology firms, Tesla embodies high volatility with growth-oriented dynamics, and NVIDIA provides a balance of innovation-driven performance with moderate volatility. This allows for the assessment of model robustness across different market conditions.

The input features include Open, High, Low, Close, and Volume (OHLCV), supplemented by technical indicators that enrich the representation of market trends. More specifically, we computed the Moving Average Convergence Divergence (MACD) and its signal line, which is a widely used indicator in technical analysis for capturing momentum and trend reversals. In total, these features together provide a seven-dimensional input vector for every trading day.

This data was prepared for deep learning models by applying MinMax scaling to normalize all the features in the range between 0 and 1. Then, a separate scaler was utilized for the closing price so that an inverse transformation could be done precisely. This will make the forecasted values map back to the original price scale for meaningful evaluation.

For supervised learning, the sequences have been generated using a sliding window approach. Each input sequence consists of 60 consecutive trading days, and the corresponding output is the next 15 or 30 trading days of closing prices. This design enables multi-step forecasting while preserving temporal dependencies. The dataset was partitioned into training and validation sets, with 10% reserved for validation to monitor generalization performance and prevent overfitting during training.

This ensures that the same preprocessing pipeline is applied to both models, LSTM and Transformer, to have feature-rich, consistent input data for a fair comparison across architectures, forecast horizons, and stock volatility profiles.

4. MODEL ARCHITECTURE

The different architectures employed in evaluating deep learning approaches to multi-step stock forecasting include a recurrent baseline using Long Short-Term Memory and an advanced Transformer model that uses attention. Both models had been trained on GPU-enabled platforms to maintain efficiency and scalability.

4.1 Baseline Model: LSTM

Component	Description
Framework	TensorFlow / Keras
Purpose	Baseline recurrent model for multi-step forecasting
Architecture	Stacked LSTM (64 units → 64 units) + Dropout
Layers	LSTM (64, return_sequences=True) → Dropout(0.2) → LSTM(64) → Dropout(0.2) → Dense(horizon)
Forecast Horizon	15 or 30 trading days
Optimizer	Adam
Loss Function	Mean Squared Error (MSE)
Early Stopping	Patience = 10, restore best weights
Hardware	GPU-accelerated training

The base model will be a stacked LSTM network implemented in TensorFlow/Keras. The architecture consists of two LSTM layers with 64 units each, followed by dropout layers to reduce overfitting. A final dense layer outputs the forecast horizon, which will either be 15 or 30 trading days.

4.2 Advanced Model: Transformer with Attention

The advanced model is a Transformer architecture implemented in PyTorch that makes use of attention mechanisms to capture long-range dependencies. The model includes an input linear layer to project features into higher dimensional space, positional encoding to preserve temporal order, stacked encoder layers with multi-head self-attention, and finally a dense layer that produces the forecast horizon. This structure is well suited for modelling complex dependencies in volatile stock data.

Component	Description
Framework	PyTorch
Purpose	Advanced attention-based model for long-range dependencies
Input Projection	Linear layer (feature_size → hidden_dim)
Positional Encoding	Added to preserve temporal order
Encoder	2 Transformer encoder layers, 8 attention heads
Output Layer	Dense(horizon)
Forecast Horizon	15 or 30 trading days
Optimizer	Adam
Loss Function	Mean Squared Error (MSE)
Early Stopping	Patience = 10
Hardware	CUDA-accelerated training

4.3 Hyperparameters

Hyperparameter	Value	Applies To
Epochs	100	Both
Batch Size	32	Both
Early Stopping Patience	10	Both
Hidden Dimension	128	Transformer only
Number of Attention Heads	8	Transformer only
Loss Function	MSE	Both
Optimizer	Adam	Both

5. EXPERIMENT

The experiments were designed to test how two deep learning models—an LSTM baseline and a Transformer with attention—perform when forecasting stock prices under different conditions. To keep the comparison fair, both models were trained using the same preprocessing pipeline, and early stopping was applied to validation loss to prevent overfitting. A simple statistical baseline, the naive persistence model, was also included to show how much improvement the deep learning approaches provide.

Each model was trained on sequences of 60 trading days and asked to predict either the next 15 or 30 days of closing prices. Training conditions were kept consistent: the Adam optimizer, mean squared error loss, a batch size of 32, and a maximum of 100 epochs. Performance was measured using RMSE and MAE, and the forecasts were also visualized against actual price movements to highlight differences in behavior (see Figure X in the Results section). This setup made it possible to see how the LSTM and Transformer handle the same data under identical constraints.

The length of the forecast horizon turned out to be an important factor. Predictions over 15 days were generally more accurate, while extending to 30 days introduced drift and compounding error. This trade-off between forecast length and reliability was especially clear in volatile stocks such as Tesla, where sudden price swings challenged both models.

To explore how model design choices affect performance, hyperparameter sweeps were carried out on the LSTM. The number of hidden units was varied across 32, 64, and 128, and dropout rates across 0.1, 0.2, and 0.3. Validation RMSE was used to identify the best configurations, showing that careful tuning can make the LSTM more robust (see Table Y in the Results section).

Generalizability was tested by running experiments on four different stocks: Apple (AAPL), Tesla (TSLA), Microsoft (MSFT), and NVIDIA (NVDA). These were chosen to represent a range of volatility, from relatively stable (AAPL, MSFT) to highly volatile (TSLA) and innovation-driven (NVDA). Each stock was modelled separately, and performance metrics were reported individually (see Table Z in the Results section). This design made it possible to compare how the two architectures respond to different market conditions.

Finally, the naive persistence baseline was included as a benchmark. By simply carrying forward the last observed closing price, it provided a lower bound for performance. As expected, this approach produced the highest error values, reinforcing the advantage of deep learning models over trivial heuristics.

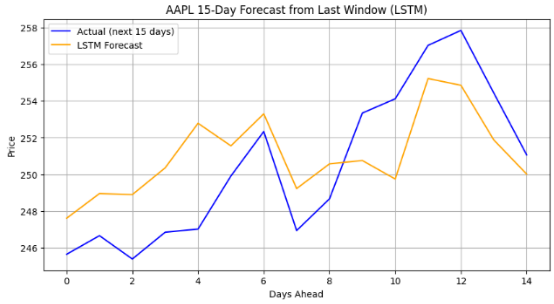
Results were summarized through a mix of tables and figures. RMSE and MAE values were tabulated across models, horizons, and stocks, while line plots showed actual versus predicted trajectories. Training and validation loss curves were also included to illustrate convergence and generalization. Together, these outputs provided a clear picture of how each model handled volatility, horizon length, and stock diversity.

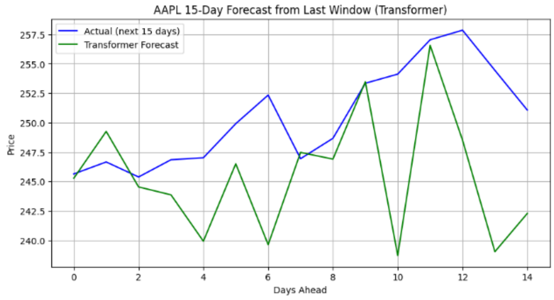
6. RESULTS

Several interesting findings emerged with the comparison between the LSTM baseline and the attention-based Transformer model with regards to stocks, horizons, and volatility regimes. Forecast trajectories showed that both models could predict basic market trends but clearly were responsive to conditions of volatility. For equities characterized by relatively stable volatility, such as Apple (AAPL) and Microsoft (MSFT), forecasts using the Transformer were closer to the true values, especially on longer horizons. While the use of the LSTM smooths out fluctuations, leading to the underestimation of sudden spikes, this allows for quite smooth forecasts in the short term. Tesla (TSLA), a highly volatile stock, is a case in point: The LSTM generated smoother curves that missed the capture of abrupt changes, while the Transformer reacts with more dynamism, though sometimes with overshooting. NVIDIA had intermediate characteristics: Both models tracked the overall trend but split when prices moved rapidly.

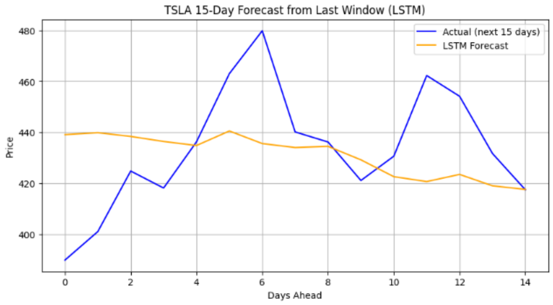
6.1 15-day horizon plots of actual vs. predicted plots of LSTM and Transformer models

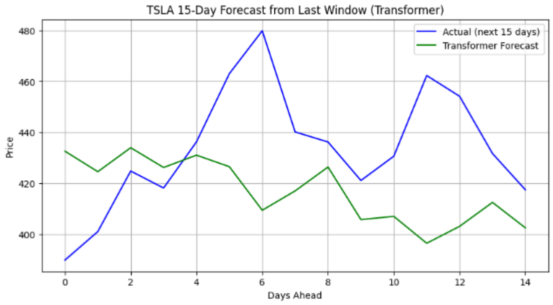
1. AAPL:


AAPL 15-Day Forecast from Last Window (LSTM)

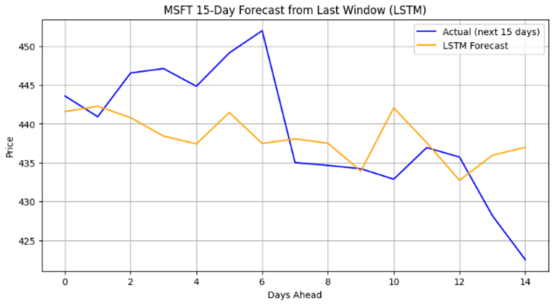

AAPL 15-Day Forecast from Last Window (Transformer)

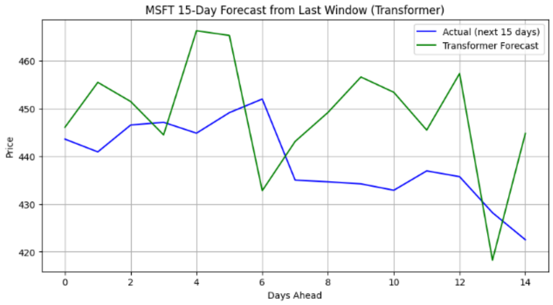
2. TSLA:


TSLA 15-Day Forecast from Last Window (LSTM)

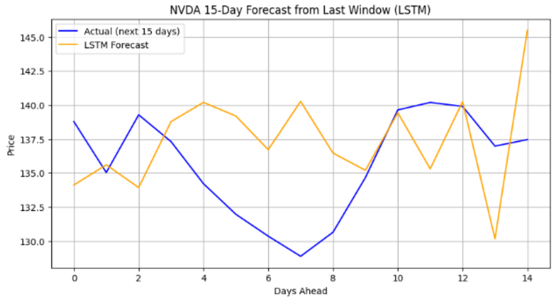

TSLA 15-Day Forecast from Last Window (Transformer)

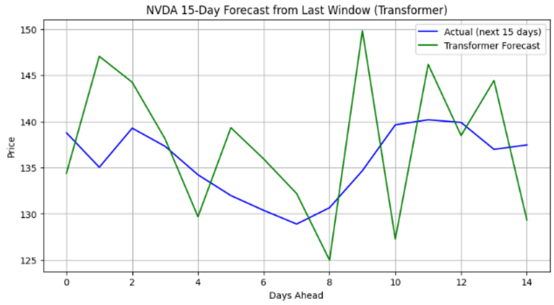
3. MSFT:


MSFT 15-Day Forecast from Last Window (LSTM)


MSFT 15-Day Forecast from Last Window (Transformer)

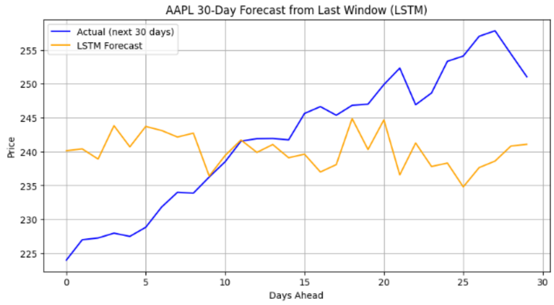
4. NVDA:

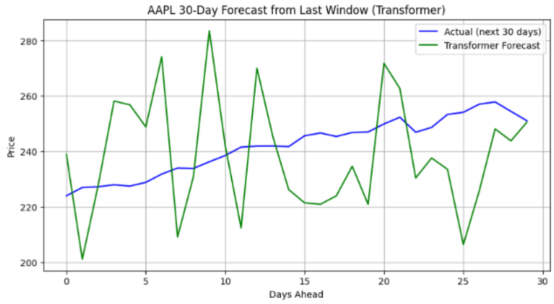




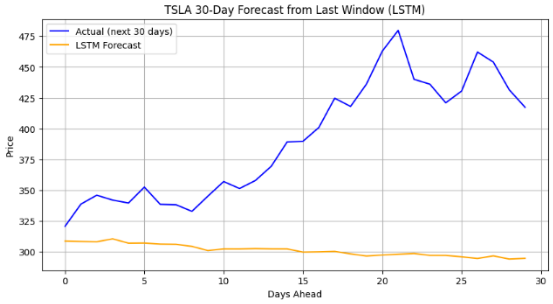
6.2 30-day horizon plots of actual vs. predicted plots of LSTM and Transformer models

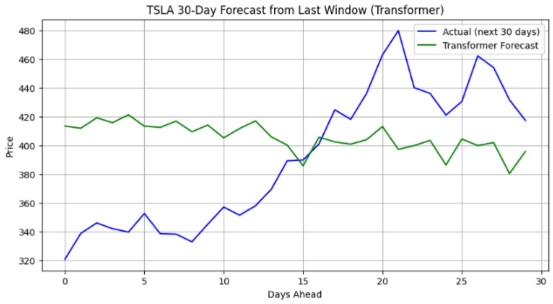
1. AAPL:



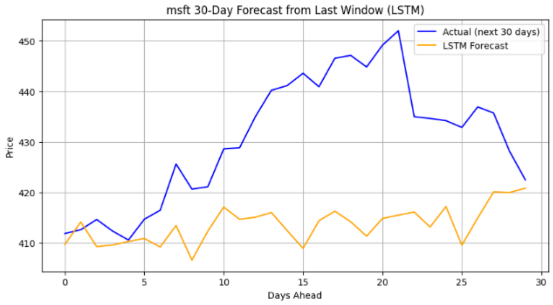


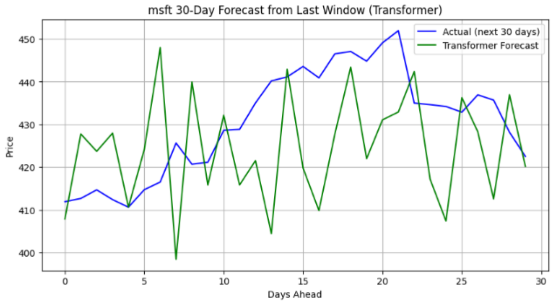
2. TSLA:



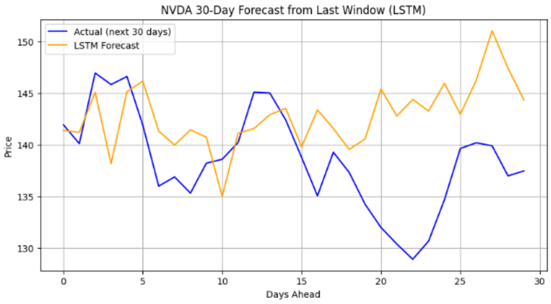


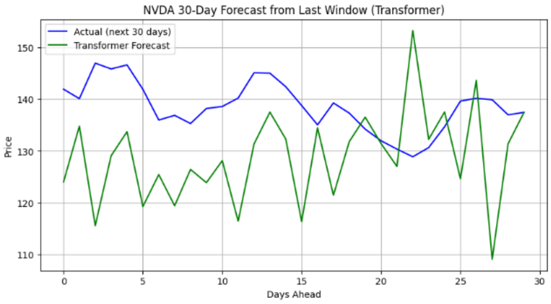
3. MSFT:





4. NVDA:





Quantitative evaluation confirmed these observations. Error metrics, such as the RMSE and MAE, were systematically lower for the 15-day horizon than for the 30-day horizon, reflecting the greater challenge of longer forecasts. The Transformer outperformed others on stable stocks, whereas LSTM was still competitive in the short-term horizon and less volatile conditions. The naive persistence baseline, whereby the last observed closing price was carried forward, always resulted in the highest values of error metrics, underlining the benefit of deep learning approaches.

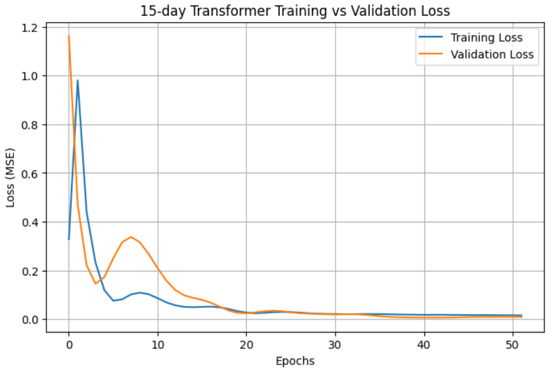
[Table X: RMSE and MAE across models, and stocks]

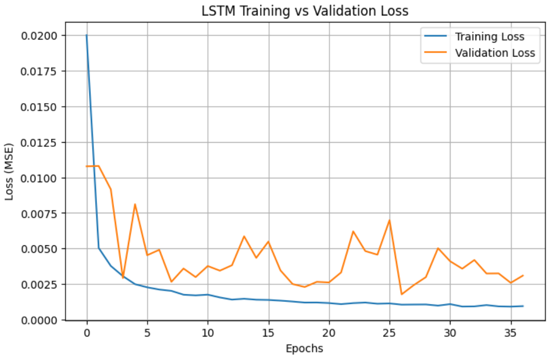
Stock	LSTM		TRANSFORMER	
	RMSE	MAE	RMSE	MAE
AAPL	6.9060	4.6409	6.2569	4.3645
TSLA	18.888	10.607	21.907	12.859
MSFT	11.4034	7.9075	25.6018	18.3423
NVDA	3.6509	1.6295	33.9542	18.9845

Training and validation loss curves provided further insight into optimization behavior: The LSTM exhibited smoother convergence, with the validation loss stabilizing after approximately 30 epochs. The Transformer showed greater variability in early epochs but went on to achieve a lower validation loss-especially for stable stocks. These curves demonstrate effective generalization under early stopping and give an indication of how recurrent and attention-based models adapt during training.

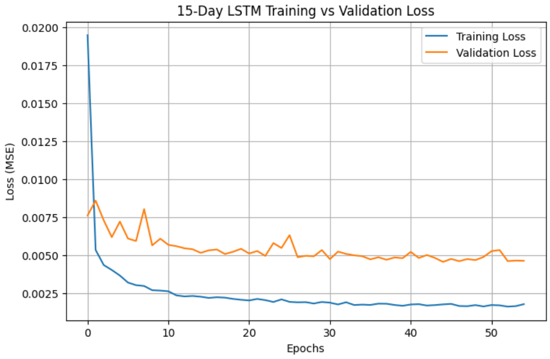
6.3 15-day horizon training vs. validation loss plots of LSTM and Transformer models

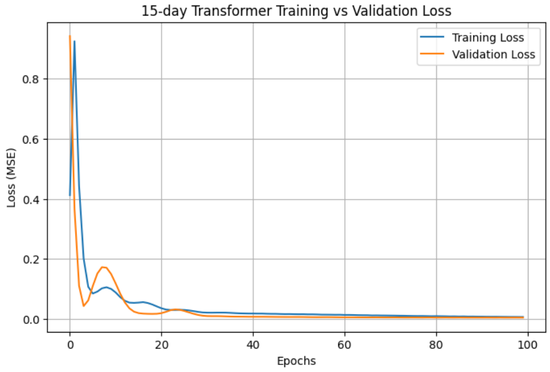
1. AAPL:



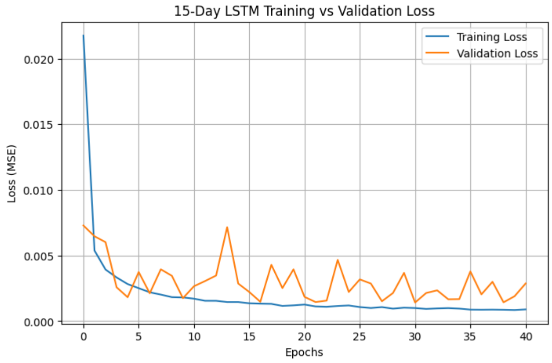


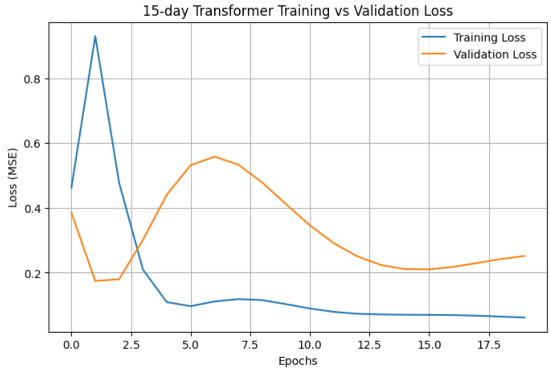
2. TSLA:



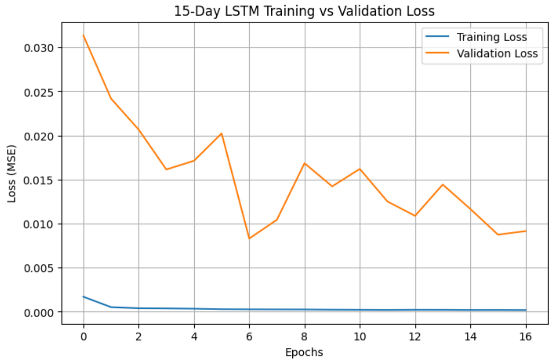


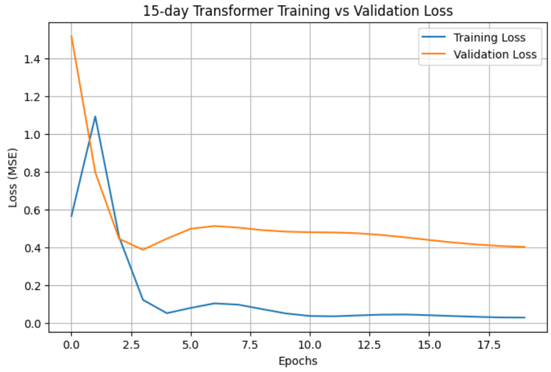
3. MSFT:





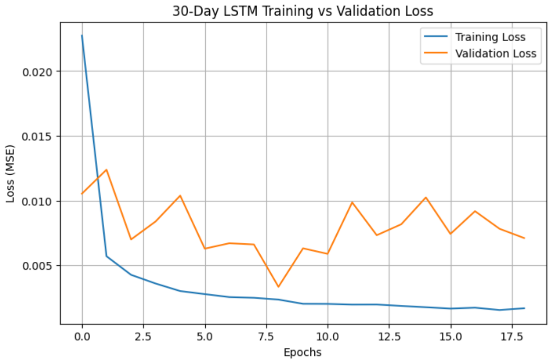
4. NVDA:

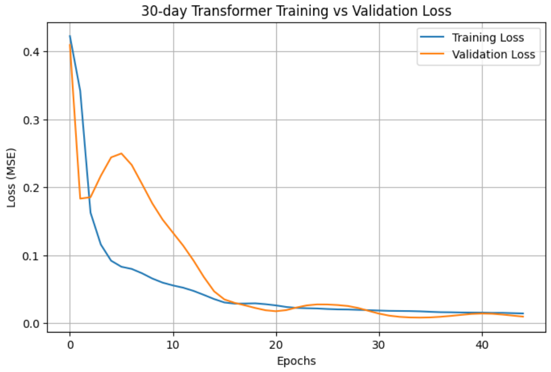




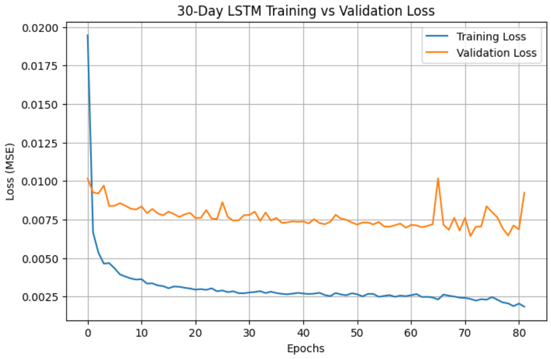
6.4 30-day horizon training vs. validation loss plots of LSTM and Transformer models

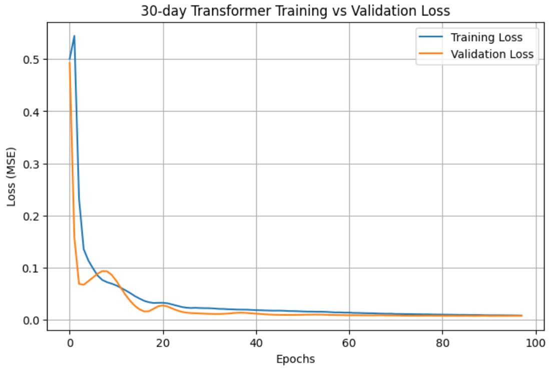
1. AAPL:



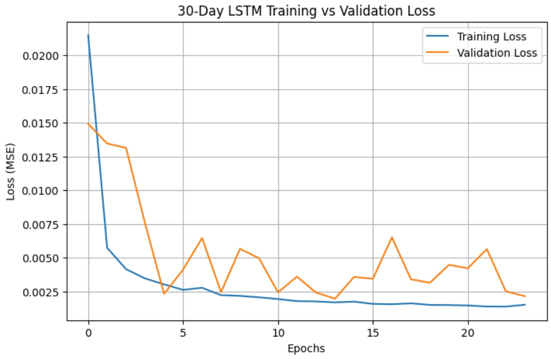


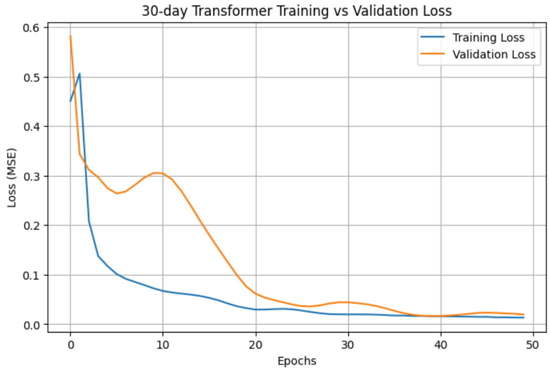
2. TSLA:



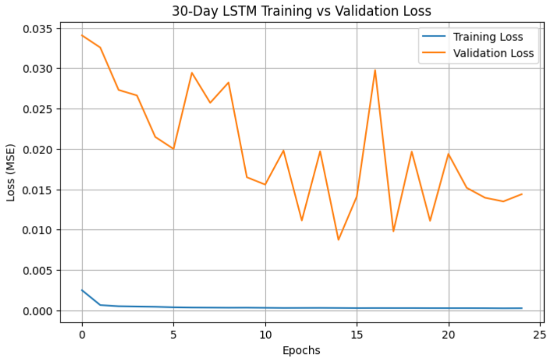


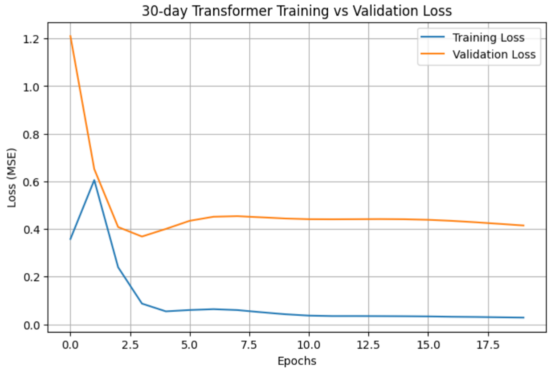
3. MSFT:





4. NVDA:





Taken all together, several key trends emerge from these results: the LSTM works well for short-term forecasting and has stable outputs that smooth volatility, but clearly struggle with sudden spikes. The Transformer, meanwhile, can capture longer dependencies with its attention mechanisms, and performs really well on stable stocks and longer horizons but overshoots in highly volatile markets. As expected, accuracy falls as the forecast horizon increases, with both models showing drift and compounding error at 30 days compared to 15 days. Finally, both deep learning models strongly outperform the naive baseline, confirming the value of advanced model architectures for financial time series forecasting.

7. DISCUSSION AND CONCLUSION

The comparison of LSTM and Transformer architectures across four stocks and two forecast horizons shows clear patterns in how deep learning models perform in financial time series forecasting. Both models were trained using the same preprocessing pipelines and hyperparameters, but their forecasting characteristics varied due to their designs. One consistent finding from all experiments is that forecast horizon length strongly influences prediction accuracy. The 15-day horizon resulted in lower RMSE and MAE values than the 30-day horizon for both models. This reflects the challenge of error accumulation in multi-step forecasting. This effect was particularly notable for highly volatile stocks like TSLA, where sudden price swings made long-range predictions more challenging.

The LSTM model displayed stable and smooth forecasting behavior, especially for short-term horizons and stocks with moderate volatility. Its recurrent structure effectively captured local temporal patterns but often underestimated sudden changes, leading to smoother predictions that lagged behind real market movements. This limitation was most evident in TSLA and NVDA. However, for relatively stable stocks like AAPL and MSFT, the LSTM provided reliable short-term forecasts with comparatively low error.

The Transformer model, on the other hand, showed more sensitivity to long-range dependencies and reacted more dynamically to rapid price changes. Its attention mechanism allowed it to consider relationships across the entire 60-day input window, helping it track broader trends more effectively. This advantage was clear in the 30-day forecasts for AAPL and MSFT, where the Transformer aligned more closely with the overall trajectory than the LSTM. However, this sensitivity also made the Transformer more likely to overshoot in highly volatile conditions, sometimes exaggerating price movements in TSLA.

Training dynamics supported these observations. The LSTM exhibited smooth and stable convergence. In contrast, the Transformer displayed greater variability in early epochs due to the complexity of optimizing multi-head attention layers. Once stabilized, the Transformer often achieved lower validation loss for stable stocks. The naive persistence baseline consistently produced the highest error values, confirming that both deep learning models captured meaningful temporal structure beyond simple trend continuation.

7.1 Future Scope

Future research can build on this study in several focused ways:

1. Incorporation of additional data sources: Adding news sentiment, macroeconomic indicators, or

volatility indices may enhance robustness, especially for volatile stocks.

2. Hybrid or improved architectures: Combining recurrent layers with attention mechanisms or exploring models like Temporal Fusion Transformers could leverage both short-term memory and long-range dependency modelling.
3. Uncertainty-aware forecasting: Probabilistic methods like Monte Carlo dropout or quantile regression could provide confidence intervals useful for risk-sensitive applications.
4. Volatility-adaptive modelling: Regime-switching or explicit volatility modelling may help reduce overshooting and boost performance on stocks with abrupt price movements.
5. Broader datasets and deployment considerations: Testing on additional markets and assessing computational efficiency, latency, and interpretability would support real-world applications in trading systems.

8. REFERENCES

1. Kabir, M., Rahman, S., & Ahmed, T., 2025. LSTM–Transformer-Based Robust Hybrid Deep Learning Model for Financial Time Series Forecasting. Sci, 5(2), 1–15.

2. Cai, Y., & Huang, J., 2023. Predicting the Future: LSTM vs Transformers for Time Series Modelling. MIT Deep Learning Blog.

3. Bilokon, V., & Qiu, Y., 2023. Transformers versus LSTMs for Electronic Trading. arXiv preprint, arXiv:2309.11400.

4. Ruiter.ai, 2025. LSTMs vs Transformers: What Works Best for Time-Series Forecasting? Ruiter.ai Technical Report.

5. Sivakumar, S., Menon, A., & Krishnan, R., 2025. A Critical Study on LSTM and Transformer Models for Financial Analysis and Forecasting. In: Proceedings of the Springer International Conference on Data Science, pp. 112–125.