# **Predicting Stock Prices with Deep Learning: A Tale of RNNs, LSTMs, and Market Data**

*By Jahnavi Paliwal, Pooja Baralu Umesh and Satya Lasya Narendrapurapu*

## **Introduction: Can Machines Predict the Market?**

The stock market is notoriously unpredictable. With constant fluctuations driven by economic news, company performance, and investor sentiment, building a model that can reliably forecast future prices is no small feat. That’s precisely why we were drawn to this challenge.

Our project explores how **deep learning models, specifically RNNs and LSTMs can be leveraged to predict stock price trends** based on historical data. Using data from Microsoft, Apple, and Google over 15 years, we trained, evaluated, and visualized models to see how well machines can capture the rhythm of the markets.

## **The Data: 15 Years, 3 Giants**

We sourced **15 years of daily stock data** for Microsoft (MSFT), Apple (AAPL), and Alphabet (GOOG) from Yahoo Finance. Each dataset contained the usual suspects: Date, Open, High, Low, Close, Volume, and Adjusted Close.

After merging and cleaning the datasets, we engineered features such as **On-Balance Volume (OBV)** and **20-day Volume Moving Average**, which helped smooth out noise and highlight investor sentiment over time.

### **Why OBV?**

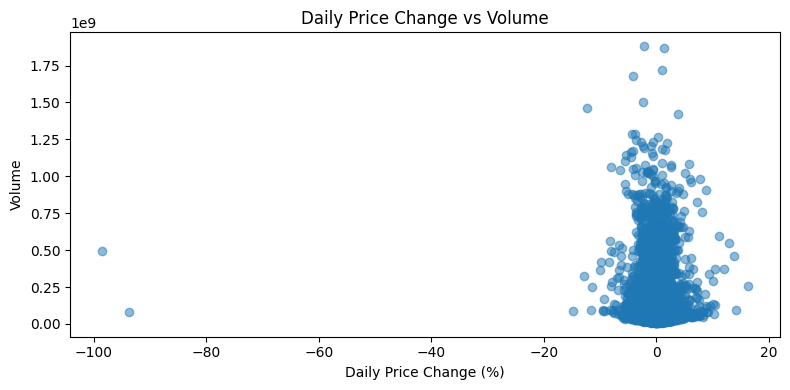
Raw trading volume is noisy. OBV offered a more cumulative, direction-aware metric:

* If the stock closed higher, volume was added.
* If it closed lower, volume was subtracted.
* Flat closes left OBV unchanged.

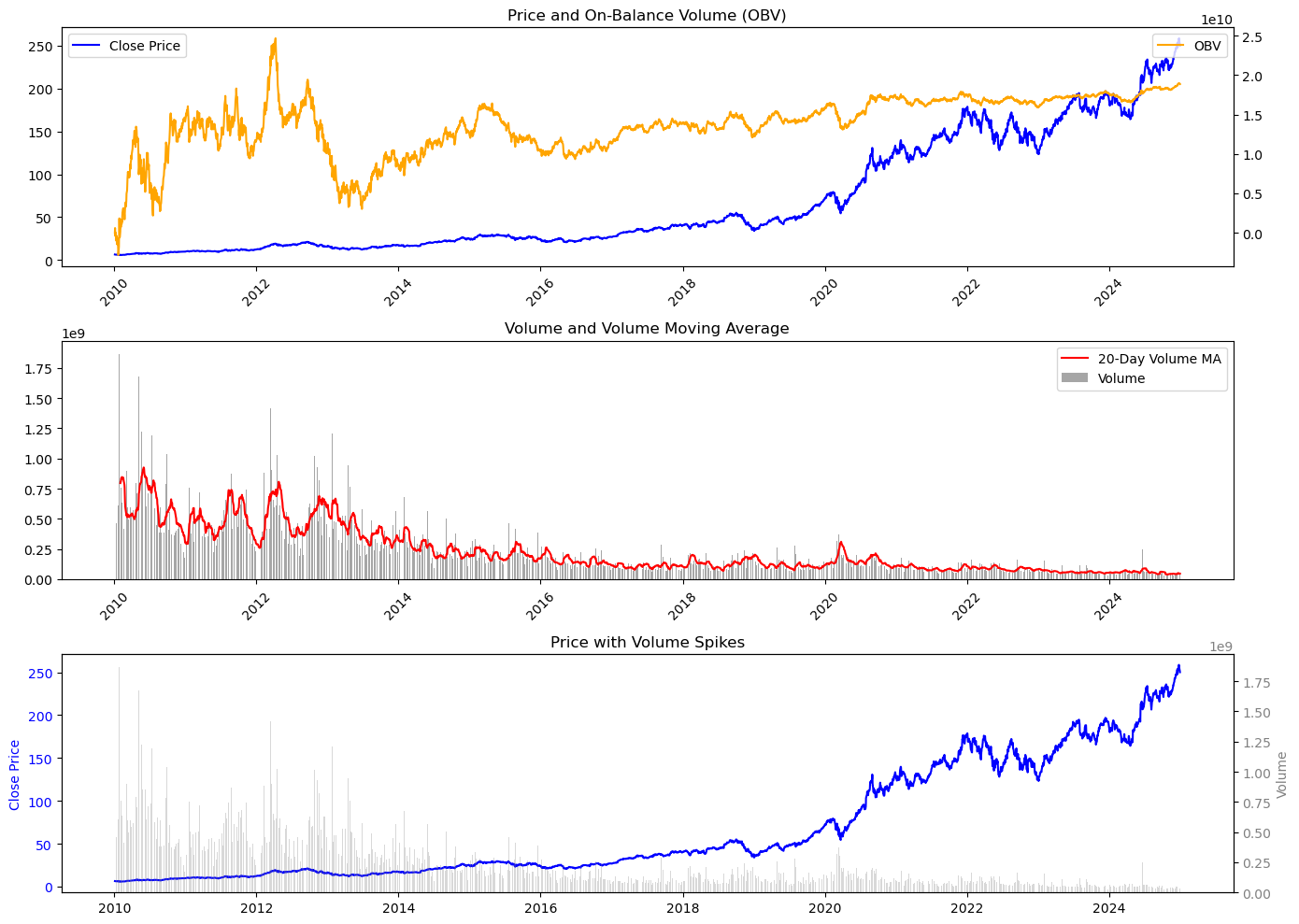
This made it easier to see whether "smart money" was flowing in or out of the market.

## **Exploratory Data Analysis: Volume ≠ Volatility?**

We began by asking: **Does big volume mean big price changes?** Turns out—not always.

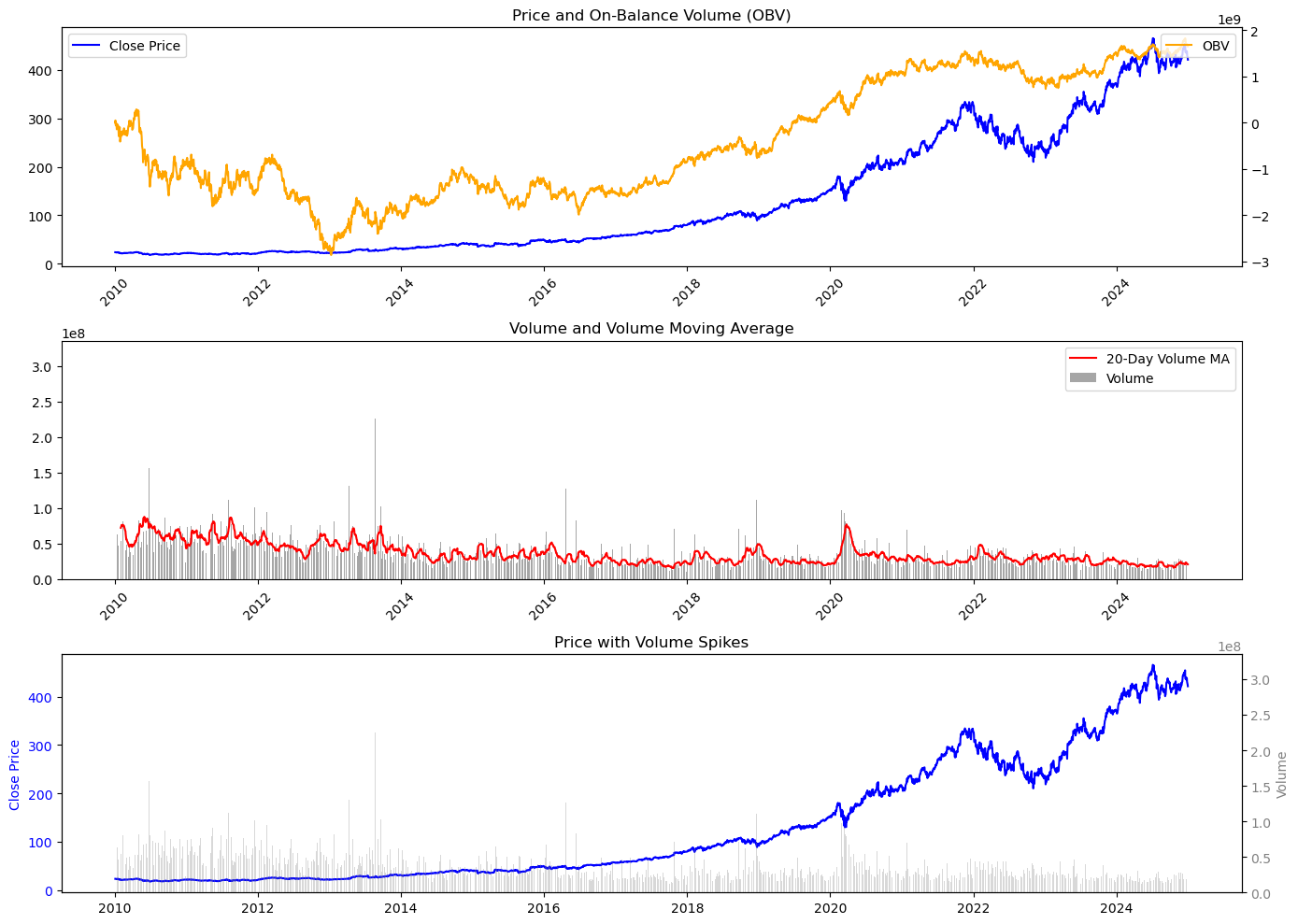


A scatter plot above of price change % vs. trading volume revealed that while some massive price swings coincide with high volumes, many days had large volume and barely any price change. This inspired us to dig deeper using OBV and smoothed volume trends.

Each company told its own story:

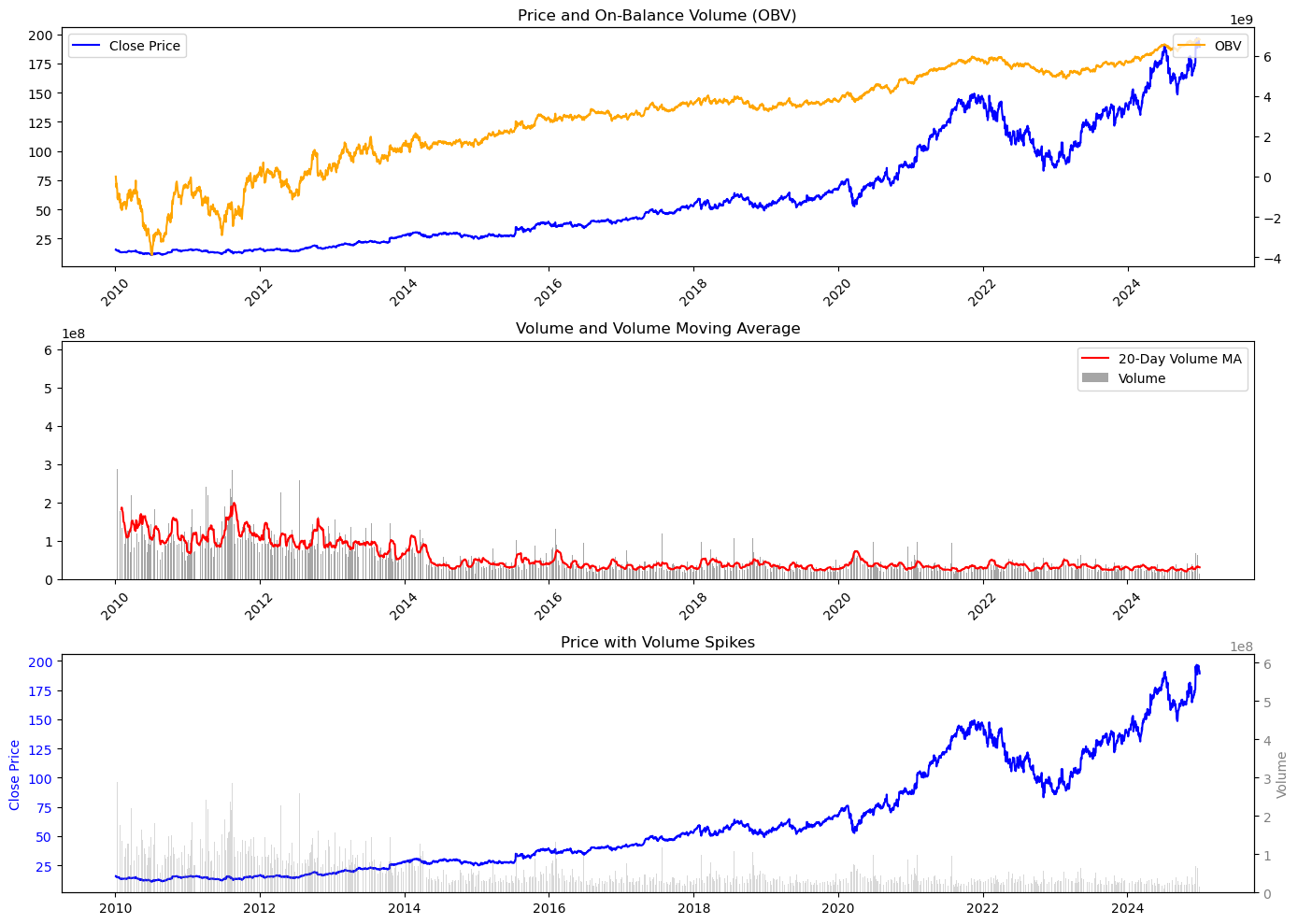
**Apple**:

Apple’s stock has shown strong growth over the past 15 years, driven by major product launches and digital adoption. After peaking in 2012, it consolidated until 2016, then surged post-2017 with robust product sales. OBV steadily increased, indicating consistent institutional buying. While early years saw higher trading volume, it stabilized after 2016, with occasional spikes during earnings reports. Despite rising prices, recent volume spikes are fewer, suggesting a shift to long-term holdings and strong institutional support.



**Microsoft**:

Microsoft’s stock has evolved significantly over the past 15 years, reflecting its shift from a software company to a cloud leader. The plot shows a steady price uptrend post-2016, driven by cloud expansion and strategic acquisitions. Despite rising prices, trading volume has declined, indicating a move from speculative trading to long-term holdings. The OBV line shows consistent buying pressure, suggesting strong investor confidence in Microsoft’s growth prospects.



**Google**:

Google’s stock has demonstrated substantial growth over the past 15 years, reflecting its dominance in the tech sector. The plot illustrates a steady price uptrend post-2016, driven by strong financial performance and digital expansion. OBV shows consistent buying pressure, suggesting institutional accumulation. Despite the price surge, trading volume has declined, indicating a shift to long-term holdings. The volume moving average reveals occasional spikes during key events, but overall, investor activity appears more strategic than speculative.

## **The Models: RNN vs. LSTM**

## **Why RNNs and LSTMs for Stock Prediction?**

Stock prices are inherently sequential, influenced by a multitude of factors over time. Traditional models often struggle to capture the temporal dependencies present in such data. This is where RNNs and LSTMs come into play.

**Recurrent Neural Networks (RNNs)** are designed to recognize patterns in sequences of data by maintaining a 'memory' of previous inputs. This capability allows them to model the temporal dynamics of stock prices effectively.

However, RNNs can encounter issues like the vanishing gradient problem, which hampers their ability to learn long-term dependencies. **Long Short-Term Memory networks (LSTMs)** address this by incorporating specialized structures called gates. These gates regulate the flow of information, enabling LSTMs to retain relevant information over extended periods and discard irrelevant data.

This architecture makes LSTMs particularly adept at capturing both short-term fluctuations and long-term trends in stock prices, providing a more nuanced and accurate prediction model.

### **Why Both?**

While LSTMs are theoretically more capable, they also come with increased computational complexity. We wanted to empirically test whether the added complexity actually translates into better predictive performance when applied to real-world stock data.

### **Preprocessing & Model Architecture**

* **Data Scaling:** All price values were scaled to a 0–1 range using MinMaxScaler.
* **Input Sequences:** We used a 365-day sliding window to generate input sequences for the models.
* **Train-Test Split:** 80% of the data was used for training, with the remaining 20% for evaluation.

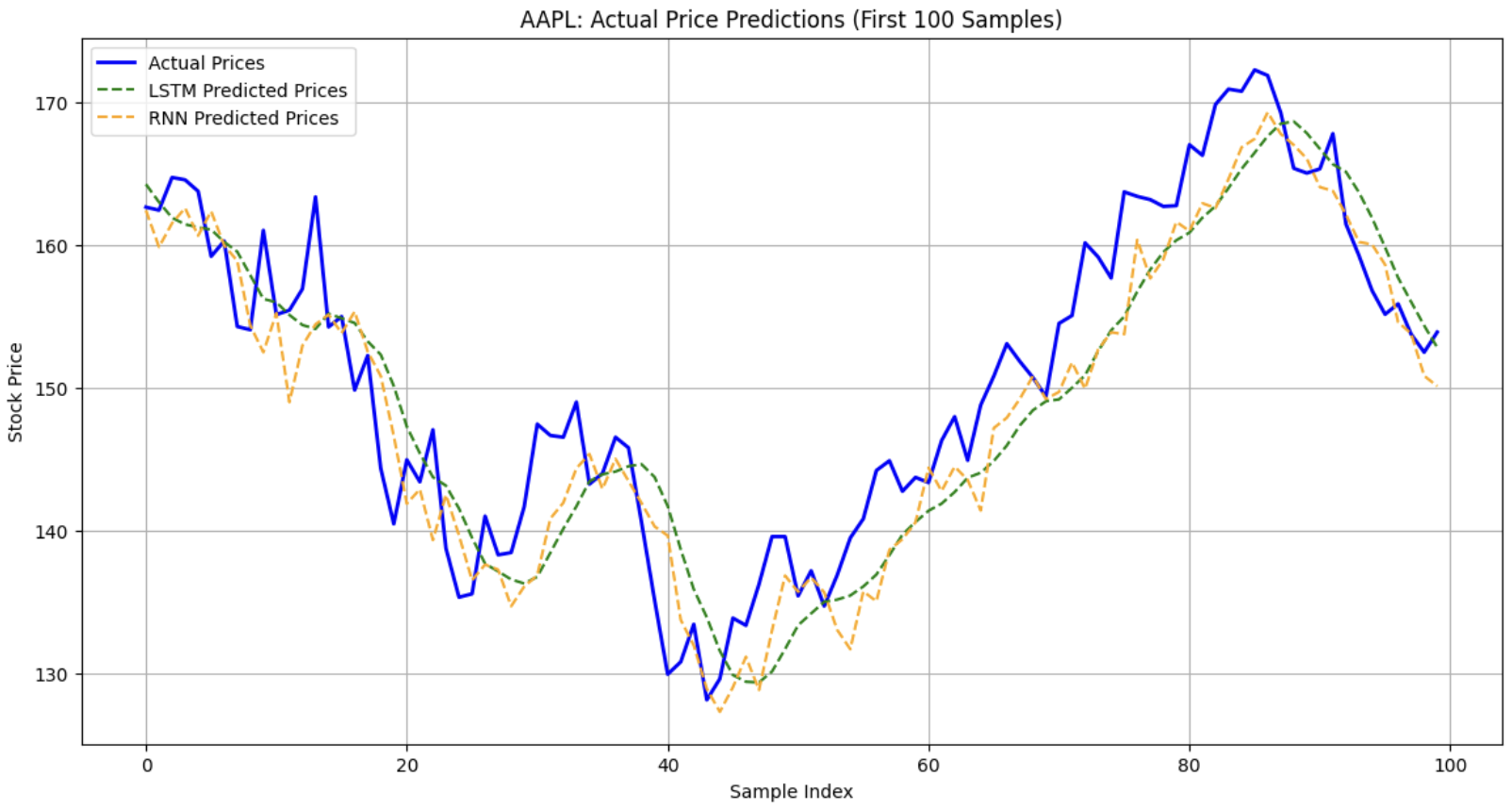
### **Model Architecture:**

* **RNN:** A single-layer recurrent model with 64 units and a dense output layer.
* **LSTM:** A structurally identical model, but with the RNN layer replaced by an LSTM layer to handle longer dependencies.

### **Visualizing the Results**

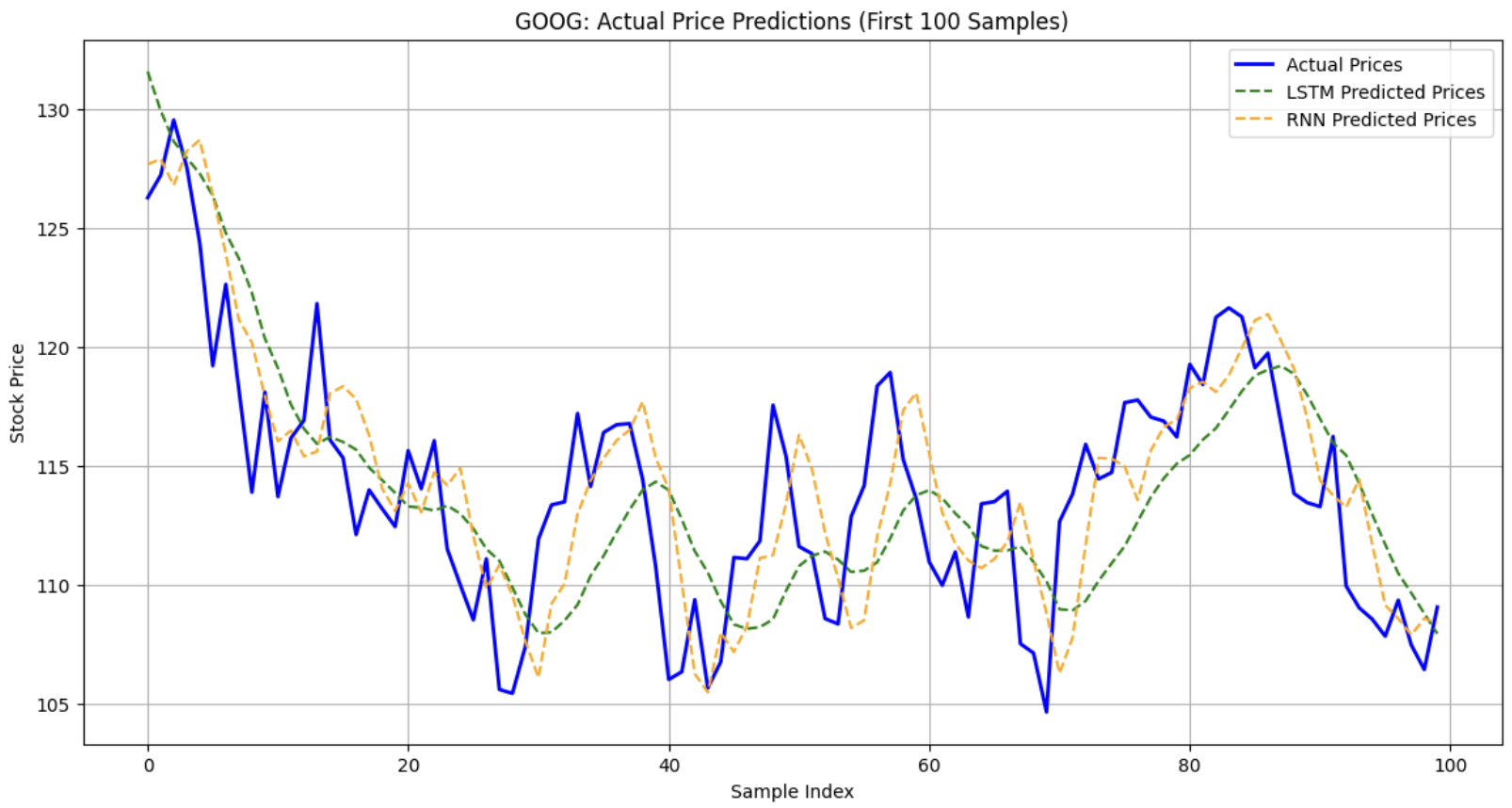
Below are the predicted versus actual prices for the first 100 samples in the test set for AAPL, GOOG, and MSFT. Each graph shows both the scaled and actual price comparisons between the RNN and LSTM models.

#### **Apple (AAPL)**

The graph below shows the actual price predictions using LSTM and RNN models.   
You can see the LSTM generally tracks the trend more closely, especially during price surges and drops.  


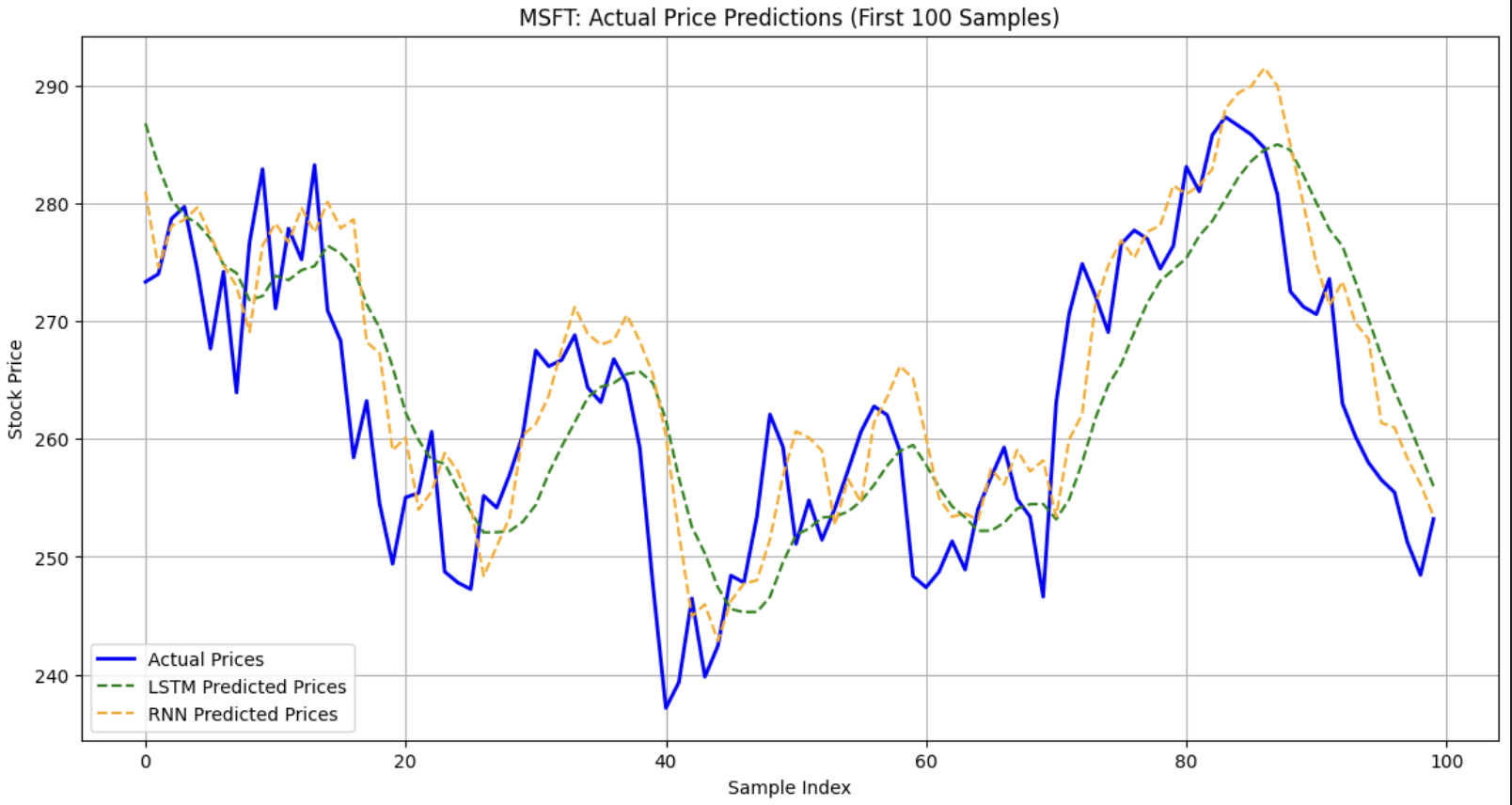
#### **Google (GOOG)**

The LSTM again tends to follow the actual price curve more faithfully, though both models start diverging in volatile sections.



#### **Microsoft (MSFT)**

In the case of MSFT, the LSTM demonstrates smoother, more stable predictions. The RNN shows signs of overfitting to short-term noise.



### 

### 

### 

### **Takeaways**

* LSTMs consistently outperform RNNs in capturing the trend and magnitude of stock price movements across all three stocks.
* The gating mechanisms in LSTMs seem to better handle noise and retain long-term dependencies, which is crucial for stock forecasting.
* While both models demonstrate predictive ability, LSTMs strike a stronger balance between responsiveness and stability.

## **Visualizing Predictions**

We plotted predictions on both scaled and actual price axes.

* For **Apple**, the LSTM tracked upward trends more closely and responded faster to price jumps.
* For **Google**, the LSTM overshot occasionally, but was generally more consistent than the RNN.
* For **Microsoft**, both models performed similarly on stable uptrends, but the LSTM better identified price troughs.

## **Evaluation Results**

We evaluated both models using three core error metrics : Mean Squared Error(**MSE)**, Root Mean Squared Error (**RMSE)**, and Mean Absolute Error(**MAE)**. Here’s a quick snapshot of their performance on the held-out test set :

| **Company** | **Model** | **RMSE** | **MAE** |
| --- | --- | --- | --- |
| AAPL | LSTM | 5.62 | 4.61 |
| AAPL | RNN | **5.47** | **4.54** |
| GOOG | LSTM | 4.57 | 3.62 |
| GOOG | RNN | **3.75** | **2.89** |
| MSFT | LSTM | 8.08 | 6.65 |
| MSFT | RNN | **8.18** | **6.56** |

### **Key Insights:**

* **RNNs slightly outperformed LSTMs on two of the three tickers** (AAPL & GOOG), but the differences were marginal ~(2-3) %.
* **RNN = cheap low-pass filter :** great for assets that drift smoothly, where shaving latency matters more than modelling rare shocks.
* **LSTM = momentum tracker :** its gated memory cushioned Microsoft’s 2023 whiplash, eking out a 0.2 % RMSE win when volatility spiked.

## 

## **Why Didn’t the LSTM Dominate?**

## **Short Horizon:** We only predict one step ahead. The LSTM’s long-range memory is under-utilised.

## **Data Simplicity:** No covariates (macro factors, options flow), causing limited nonlinear interactions for the LSTM to exploit.

## **Final Thoughts: When Simpler is Smarter**

The experiment reminds us that model parsimony often beats architectural glamour. Yes, the LSTM produced prettier, less jagged curves, but aesthetics ≠ accuracy. Until the feature set or forecast horizon justifies the extra FLOPS, an RNN offers more bang per millisecond.

**Next up:** widen the search. Grid-tune sequence length, inject sentiment or options flow, and pit both nets against a lightweight transformer. Your turn: what would you add first?