Stock Time Series Forecasting through Machine Learning

Rajakrishnan Somou

Abstract

This project tackles the challenge of predicting future stock prices using historical data, employing advanced machine learning techniques to navigate the inherent complexities of financial markets. The investigation centers on two primary model architectures: the Long Short-Term Memory (LSTM) model, serving as our baseline, and a customized Variational Auto Encoder (VAE). Through meticulous feature engineering and iterative model refinement across various stocks, our findings reveal that the customized VAE model outperforms the LSTM in short-term stock price forecasting. This enhanced performance underscores the potential of VAE structures to more effectively capture the stochastic behaviors characterizing financial time series data.

1. Introduction

In the evolving landscape of financial markets, predicting stock prices has emerged as a critical challenge and opportunity for investors, traders, and financial analysts. The inherent volatility and the multitude of influencing factors make stock price prediction a complex yet rewarding task. This project aims to leverage machine learning techniques for time series prediction, focusing on stock price movements, to offer insights and forecasting tools that can aid in decision-making processes. The approach involves collecting and pre processing historical stock price data, exploring and selecting machine learning models that capture temporal dependencies and non-linear patterns, and developing a predictive model capable of forecasting stock prices over a specified future time horizon. Our evaluation focuses on using appropriate metrics to ensure the model's accuracy and reliability in real-world scenarios.

From my findings in the last report, linear models like ARIMA didn't seem to be able to capture the non linear relationships that existed in the temporal data of the financial world. Bayesian Deep Learning architectures also seemed to bring in too much external noise into its prediction skewing the results it gave. At the end of it all and confirmed in (this paper), LSTMs seemed to have been the best performing deep learning architecture. This makes sense as its strengths lay in its abilities to capture complex temporal patterns through its input. This study extends this inquiry by exploring the efficacy of stochastic models, particularly Variational Auto Encoders (VAEs), to better address the uncertainties inherent in stock market predictions. Through rigorous testing and feature engineering, our results indicate that VAEs can indeed surpass LSTMs in the domain of short-term stock price forecasting, providing a promising direction for future research in financial modeling.

2. Related Work

In the landscape of stock price prediction, diverse methodologies have been explored over the past decade, revealing the complexity and dynamism of financial markets. According to a thorough review by MDPI, several forecasting techniques, such as rough sets, genetic algorithms, neuro-fuzzy systems, and support vector regression, have been scrutinized. The review highlights the efficacy of neuro-fuzzy hybrid systems and Takagi–Sugeno fuzzy models when combined with support vector regression, showing a nuanced approach to navigating stock price volatilities.

Building on these foundations, a groundbreaking study from Emerald Insight introduces the Residual-CNN-Seq2Seq (RCSNet) model. This hybrid deep learning framework combines ARIMA, CNN, Seq2Seq LSTM, and fully connected layers, effectively capturing both linear and nonlinear dependencies in stock data. RCSNet stands out for its ability to handle pattern dependencies, promising enhanced accuracy in stock price forecasting.

In parallel, recent innovations in diffusion models have been encapsulated in "DiffSTOCK: Probabilistic Relational Stock Market Predictions using Diffusion Models," published in Papers With Code, 2024. These models, particularly Denoising Diffusion Probabilistic Models (DDPMs), tackle the intrinsic uncertainties of financial datasets. By simulating multiple potential market scenarios, DDPMs offer a sophisticated probabilistic approach that significantly advances the depiction of complex market dynamics.

Comparatively, traditional models like ARIMA continue to be used due to their reliability in certain contexts, yet the shift towards advanced machine learning techniques is undeniable. Models such as RCSNet and DDPMs illustrate

this trend, incorporating both generative and probabilistic elements to better represent the multifaceted nature of stock markets. The VAE-LSTM method, blending generative and discriminative strategies, uniquely addresses the capture of temporal dependencies and latent structures in stock data, showcasing a distinctive approach within the evolving field of financial predictions.

As this field progresses, it is crucial to engage with recent projects and publications that share similar objectives or methodologies. Analyzing these works can provide insights into common challenges and the effectiveness of various predictive frameworks. By comparing these models to newer approaches like RCS Net and diffusion models, we can better understand their relative strengths and limitations, thus guiding future innovations in stock market forecasting.

3. Dataset and Evaluation

Dataset: 3 different stocks were used this time around to measure the robustness of these algorithms: **NVDA, AMD and CVX**.

For each stock, time series data from 2019 to now was collected through the yfinance python library. When requested, a data frame was generated containing 5 columns: *'Open', 'High', 'Low', 'Close', 'Volume'* and had **1344** rows corresponding to daily values of each of these columns.

Further feature engineering was tested as these new features were added:

- Smoothed Daily Percentage Change: This measures the percentage change in the closing price from one day to the next. It can provide insights into the volatility of the stock. Due to variability, the variable was smoothed out over 15 days to reduce noise.
- Moving Averages: These are averages of different lengths (5-day and 10-day were computed) and are used to smooth out price data to identify trends.
- Relative Strength Index (RSI): This is a momentum indicator that measures the speed and change of price movements. Typically, it helps in identifying overbought or oversold conditions.

Final features: 'Smoothed Daily Percentage Change', '5 Day Moving Average', '10 Day Moving Average', 'Close', 'Volume' , 'Relative Strength Index'

The data was then pre-processed through the creation of time lagged instances over all 6 input features dictated by the *time_steps* variable (=60). The labels were values extending from the input's end into the future dictated by the *future_horizon* variable (=1). The data was then split into 90% train and 10% test/validation to retain the last 130 days to forward test the models in new forecasting and guided forecasting.

The MSE (Mean Squared Error) evaluation metric was used to see how accurate my prediction was with the true closing price values. Two methods were used to forecast, one was guided with the actual closing prices and the second was auto regressive as it used its own prediction plus the closest known data to generate the next 30 days.

4. Methods

Models: 2 refined models were tested for this time series problem coming out of my last report. The baseline was a Long Short Term Memory model (LSTM), a type of recurrent neural network capable of learning long-term dependencies, as its best suited for time series data like stock prices and because it was the best architecture from previous results.

Model Architecture for LSTM:

- First LSTM Layer: takes in a **'time_steps' x 6** dimensional matrix that represents the last **'time_steps'** days with 6 features each. The layer outputs **50** units and returns a sequence of outputs to the next layer due to return_sequences=True.
- Second LSTM Layer: Also with **50** units, this layer condenses the sequence into a single output vector from the final time step, as return_sequences=False.
- Dense Layer: A single-unit layer that outputs the next day

The theoretical approach was a Customized Variational Auto Encoder. The VAE-LSTM constructed combines a encoder and decoder into a single model, learning to reconstruct the input data while also learning a latent representation that follows a Gaussian distribution.

Model Architecture for VAE-LSTM:

- Encoder:
  - The encoder network takes the input time series data (in this case, historical stock data) and maps it to a latent space representation.
  - In this implementation, the encoder consists of an LSTM layer followed by Dense layers.
  - The LSTM layer processes the input sequences and captures temporal dependencies in the data.
  - The output of the LSTM layer is passed through Dense layers to produce two vectors:
    - z_mean: Represents the mean of the latent distribution.

    - ∗ z_log_sigma: Represents the log-variance of the latent distribution.
  - These vectors collectively define a Gaussian distribution in the latent space.

- Sampling:
  - The sampling function takes the z_mean and z_log_sigma vectors as input and generates samples from the corresponding Gaussian distribution.
  - It introduces stochasticity into the model, allowing it to generate diverse representations for the input data.

- Decoder:
  - The decoder network takes the samples from the latent space and reconstructs the input data.
  - In this implementation, the decoder consists of an LSTM layer followed by TimeDistributed Dense layers.
  - The LSTM layer processes the latent samples and generates sequences representing the forecasted values for each feature.
  - TimeDistributed Dense layers ensure that each time step in the output sequence is independently decoded.

- Loss Function: The loss function of the VAE consists of two parts: the reconstruction loss (measuring how well the model reconstructs the input sequence) and the KL divergence (measuring how closely the learned latent distribution matches the target distribution). These are combined to form the overall VAE loss.

Dataset Augmentation: As mentioned in the previous section, I construct 4 different features ('Smoothed Daily Percentage Change', '5 Day Moving Average', '10 Day Moving Average', 'Close', 'Volume' , 'Relative Strength Index') to extend off of the features already given to us to enhance the performance of the models tested here by introducing new variables that capture additional insights about the data.

Hyper parameter Tuning: No automated grid search was applied to these models as it would've been computationally infeasible, instead manual tuning was performed where 'time_step' (number of days used in input to predict next day) and 'future_horizon' (number of days we want to predict into the future) were of focus.

- 'future_horizon': After testing a variety of values for 'future_horizon', I converged on 1. The reason is as you extend this parameter, the harder it is for models to learn the increasingly complex non linear relationships that might arise in considering more future days in relation to the past feature vectors. This led to a universal decrease in performance as this was increased so 1 sufficed.

- 'time_steps': In sampling a selection of values for 'time_steps', I converged on 60. The reasoning is when you have less than 15 days of data the model doesn't have enough intermediate data to perform future forecasting as more complex relationships arise with more days considered. Having more than 70 days made computations difficult, had marginal benefits as it increased and seemed to over fit heavily. Therefore as a medium of the 2 extremes, 60 seemed optimal.

5. Experiments and Results

Here are some of the graphs and results generated through the methods outlined.

The experiments consist of training the LSTM and VAE on each stock separately, taking each model and performing inference in 2 ways:

1. 'Guided' which uses the ground truth closed prices of the stock for the model to predict the next day for the rest of the test data horizon

2. 'Forecasting' which auto regressively uses its own input and the previous 30 days up to that point to forecast 30 days into the future without given closing price data.

After generating the curves, MSE was calculated at the end for comparative measures.

Model	**Test Data (MSE)**	**Forecasting (MSE)**
NVDA LSTM	1.9205	0.2331
AMD LSTM	0.3214	0.1207
CVX LSTM	0.0494	0.4867
NVDA VAE	0.9334	0.2552
AMD VAE	0.9193	0.2412
CVX VAE	0.1750	0.0936

Table 1. Comparison of MSE for Test Data Guiding and Forecasting

Figure 1. Long Short Term Memory model on NVDA

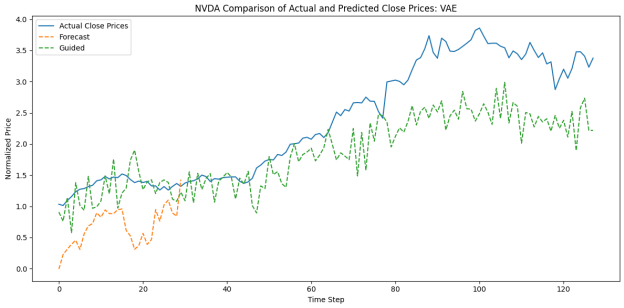

Figure 4. Variational Auto Encoder on NVDA

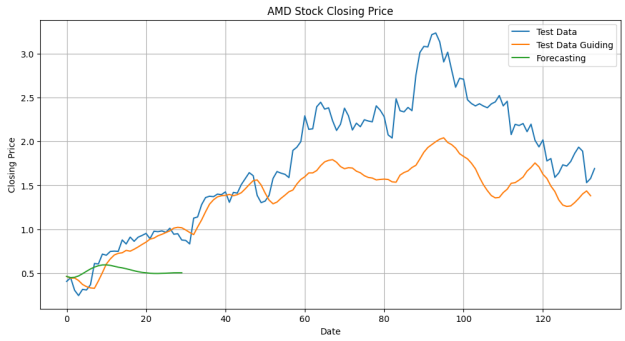

Figure 2. Long Short Term Memory model on AMD

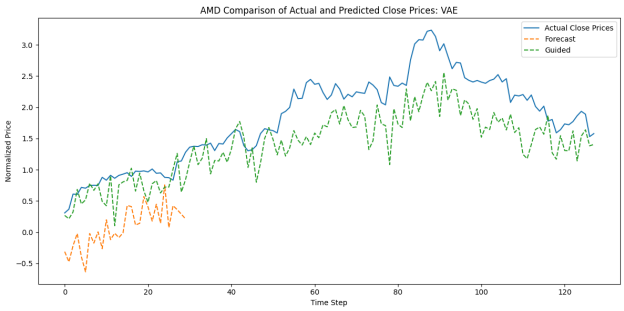

Figure 5. Variational Auto Encoder on AMD

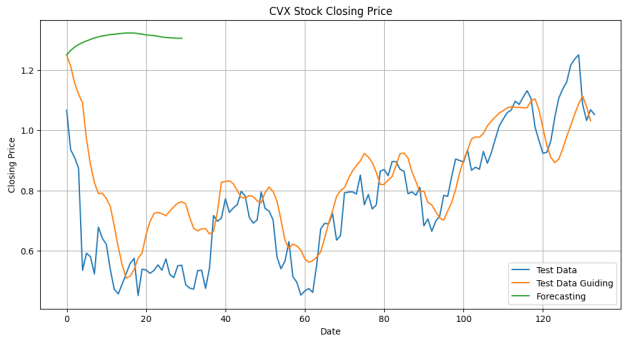

Figure 3. Long Short Term Memory model on CVX (Chevron)

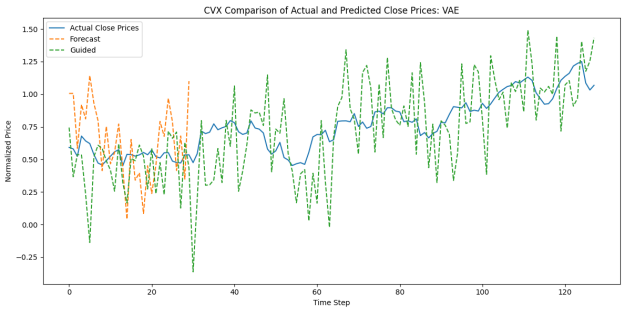

Figure 6. Variational Auto Encoder on CVX (Chevron)

6. Discussion

Quantitative Metric: With an average MSE of 0.19 for VAE forecasting over all 3 stocks and a average MSE of 0.27 for LSTM forecasting over all 3 stocks, the VAE was better at predicting future data.

Qualitative Metric: The VAE was more expressive and better alligned with structure of future data. The LSTM on the other hand predicted a flat line for all 3 stocks.

Given the results in the graphs and table, its self evident that the VAE-LSTM constructed a representation of the complex relationships that existed in the stock data better than the LSTM. As shown in the average MSE values generated, the VAE-LSTM performed better in terms of forecasting but when it came to guided forecasting it performed worse. A sound reason could be that the LSTM over fitted to the structure of the data in training and so when guided it was able to reconstruct the data but not generalize when it had to auto regress.

When visually analyzing, the LSTM converged on a flat line in every stock when asked to forecast while the VAE-LSTM predicted the overall direction of the stock accurately. I hypothesize that this resulted in MSE values that didn't well represent the model's ability to predict. In the future, I plan to analyze the structure of the output through geometric means such as predicting its overall slope and comparing it to the ground truth to better indicate the predictive power of these models.

Provided with the immense difficulty of the stock price prediction problem, the constructed VAE-LSTM model performed better than I expected. The model presents a compelling approach for stock price prediction due to its ability to capture both the complex temporal dependencies inherent in financial time series data and the underlying latent structures that govern market dynamics. By combining the strengths of Variational Autoencoders (VAEs) and Long Short-Term Memory (LSTM) networks, the VAE-LSTM architecture offers a holistic framework for learning meaningful representations of the input data while simultaneously addressing the challenges of uncertainty and non-linearity. The VAE component enables the model to learn a latent space representation that follows a Gaussian distribution, facilitating the capture of underlying patterns and relationships within the data. Meanwhile, the LSTM component excels at capturing long-term dependencies, crucial for predicting the intricate dynamics of stock prices over time. This hybrid approach not only enhances the model's ability to generate accurate forecasts but also provides valuable insights into the underlying factors driving market behavior, making it a robust and effective tool for stock price prediction tasks.

7. Conclusion

Throughout this project, the exploration of advanced machine learning techniques for predicting stock prices has yielded significant insights into the effectiveness of different architectures in handling the complexities of financial time series data. The comparative analysis between LSTM and VAE-LSTM models underscores the nuanced capabilities of these approaches in capturing temporal and latent dependencies. The VAE-LSTM model, in particular, demonstrated a superior ability to predict future stock prices by effectively modeling the stochastic nature of financial markets, which was reflected in its lower mean squared error scores and more dynamic predictions compared to the traditional LSTM approach.

This project not only confirmed the limitations of simpler linear models and highlighted the challenges posed by external noise in Bayesian frameworks but also reinforced the potential of combining generative and discriminative models like VAE and LSTM. The success of the VAE-LSTM in forecasting underlines the importance of integrating multiple model strengths to address the non-linearity and uncertainty inherent in stock data.

From a methodological standpoint, the project emphasized the importance of feature engineering and the strategic selection of hyper parameters in enhancing model performance. It also demonstrated the practical implications of model choice on forecasting accuracy and the interpretability of results in real-world scenarios. In the future, I plan to use a broader selection of hyper parameters like exponential moving averages or market sentiment to better inform my model.

Moving forward, it would be beneficial to explore additional combinations of machine learning techniques and delve deeper into hybrid models that could further refine predictive accuracy. Additionally, considering alternative performance metrics that capture the directional accuracy and the ability to predict market trends might provide a more comprehensive assessment of a model's practical utility.

In conclusion, this project has not only advanced my understanding of machine learning applications in financial markets but has also highlighted the promising potential of hybrid models like VAE-LSTM in navigating the complexities of stock price prediction. These insights pave the way for future investigations into more sophisticated models and their application in broader financial contexts.

Github Repository: https://github.com/rsomou/Stock-Time-Series-Forecasting