Diffusion VAE for handling Stochasticity in Multi-Step Regression Stock Price Prediction

Pratham Kheskwani
Department of Electrical Engineering
IIT Bombay
Mumbai, India
Email: 21d070051@iitb.ac.in

Sahil Dharod
Department of Electrical Engineering
IIT Bombay
Mumbai, India
Email: 210070026@iitb.ac.in

Chanakya Varude
Department of Electrical Engineering
IIT Bombay
Mumbai, India
Email: 210070092@iitb.ac.in

***Abstract*—Predicting stock prices over an extended period is vital in financial contexts. It aids in volatility forecasting for derivatives, risk assessment in trading, and compliance with regulatory liquidity requirements. However, forecasting multi-step stock prices is intricate due to their stochastic nature. Existing methods focus on single-step predictions and lack representation capability. To address this, we propose a novel approach—combining hierarchical VAE and diffusion probabilistic techniques for seq2seq stock prediction. Our model learns intricate stock variables and handles stochasticity via progressive noise addition. Augmenting target sequences with noise and denoising enhance model generalizability. The D-Va model outperforms existing solutions, showing improved accuracy and variance. Our project elucidates the components' contributions, reducing data noise for better accuracy. Moreover, multi-step outputs aid in forming stock portfolios, as demonstrated by the Sharpe ratio metric improvement. Our Model predicts Closing prices for the Next 5 days using previous 12 days.**

I. Introduction

In stock prediction, current research mainly focuses on single-day forecasts, neglecting multi-day predictions. However, forecasting over a longer period is crucial for volatility estimation, essential in financial applications like derivatives pricing and risk quantification in banks. Regulatory liquidity requirements demand a 5/10/20-day horizon for investors to exit risky assets without market impact. Limited studies address multi-step prediction, so our work fills this gap by proposing a method. Challenges include stock price stochasticity and the difficulty in modeling target sequences. Existing techniques involve VAE-based continuous variable learning and adversarial perturbations, but they're constrained to single-step prediction. Multi-step regression exacerbates this issue, as direct prediction of target sequences reduces generalizability at test-time. We implemented this Model on the Apple Stock, and got appreciable results.

To address these challenges, we introduce our D-Va model, merging hierarchical VAEs and diffusion probabilistic techniques for seq2seq stock prediction. The hierarchical VAE enriches the stock price distribution, learning intricate variables. Simultaneously, the diffusion model trains against stock price stochasticity by progressively adding noise to input data. Augmenting the target series with noise via a coupled diffusion process handles stochastic target sequences. Denoising processes refine predictions, reducing uncertainty and improving accuracy.

II. Methodology

We divided our work into 4 major chunks:
1)Feature Extraction
2)Heirarchical VAE Structure
3)Diffusion Modelling
4)Denoising Score-Matching Structure

A. Feature Extraction

The significance of input features in deep learning models is paramount. Features serve as the foundation on which models build their understanding of the data. High-quality and relevant features not only enhance model performance but also facilitate better generalization to unseen data. They help the model discern patterns, relationships, and representations within the data, enabling more effective learning. In contrast, inadequate or irrelevant features may hinder the model's ability to extract meaningful insights, leading to poor performance and overfitting.

1) OHLCV: The OHLCV (Open, High, Low, Close, Volume) features are fundamental in stock price prediction. They encapsulate vital information about market sentiment, price volatility, and trading activity. We normalize OHLC by previous day's closing price and volume by last 5 days rolling mean of volume.

2) Volatility: Volatility, represented by the standard deviation of trading volume over a specified time window, offers insights into market dynamics and uncertainty. Higher volatility often accompanies periods of increased trading activity or unpredictability in the market, indicating potential shifts in stock prices. Incorporating volatility as a feature in stock price models can provide additional context for understanding market behavior.(Very important for confidence score and incorporating technical chart trends)

3) Relative Strength Index: The Relative Strength Index (RSI) is a momentum indicator used in stock analysis to evaluate the magnitude of recent price changes. It measures the strength and speed of price movements, indicating potential overbought or oversold conditions in a stock. RSI values range between 0 and 100, with readings above 70 often considered overbought and readings below 30 indicating oversold conditions. Integrating RSI as a feature in stock price models provides insights into the stock's momentum and potential reversal points, aiding in decision-making for traders and investors.

4) Moving Average Convergence Divergence: The Moving Average Convergence Divergence (MACD) is a widely used technical indicator in stock analysis. It calculates the relationship between two moving averages of a stock's price, we implemented it by a 5-day exponential moving average (EMA) and a 9-day EMA. The MACD line represents the difference between these moving averages, while the signal line is a 3-day EMA of the MACD line. Crossovers and divergences between the MACD and signal lines can signal potential buying or selling opportunities, providing insights into price trends and momentum shifts. Integrating MACD as a feature in stock price models enables the assessment of trend strength and potential reversals, aiding in decision-making for traders and investors.

5) Stochastic Oscillator: The Stochastic Oscillator, a momentum indicator in stock price analysis, assesses a current closing price's position relative to its price range over a specified period. It consists of two lines: the

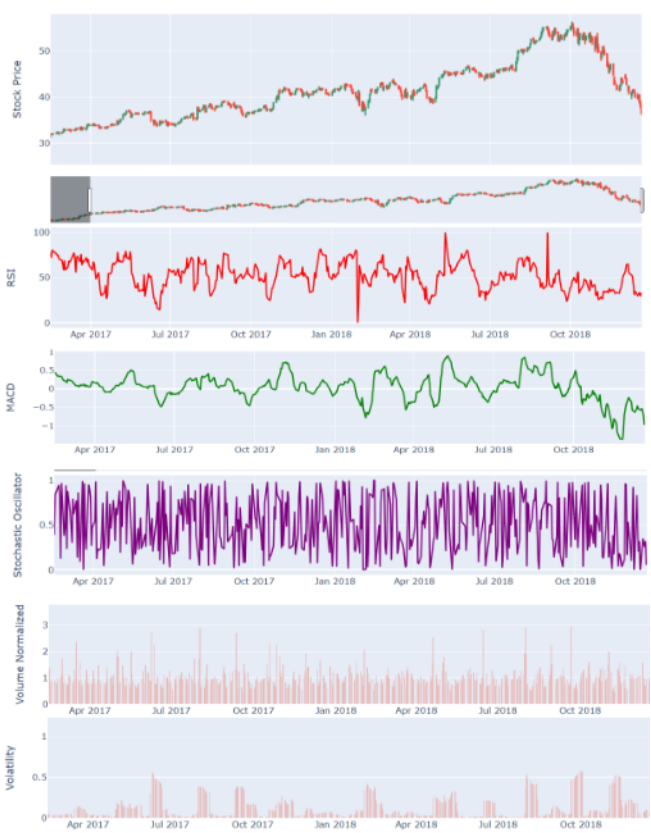

Fig. 1. features

6) Periodic Wave terms: Elliott Wave periodic features obtained via Fourier transformation involve decomposing stock price data into periodic components. This technique aims to uncover repetitive patterns or cycles within the data, aligning with the Elliott Wave Theory. Fourier transformation helps extract these periodic features, revealing potential market cycles that could aid in predicting future price movements. Integrating these periodic features into stock price models offers insights into recurring patterns and market cycles, potentially enhancing the models' predictive capabilities.

B. VAE Structure

To enhance modeling of continuous latent factors impacting stock prices, we employ deep hierarchical Variational Autoencoders (VAEs). Our primary model, the Noveau Variational AutoEncoder (NVAE), initially designed for image generation, is repurposed as a seq2seq prediction model. The NVAE architecture involves generative and encoder networks with decoder and encoder residual cells. These cells employ batch normalization, Swish activation, convolution layers, and a Squeeze-and-Excitation (SE) layer, experimentally enhancing the VAE's performance. The decoder cells include depthwise separable convolution layers to capture long-range dependencies in the data while managing computational complexity. Leveraging the stack of latent variables and depthwise separable convolutions allows us to grasp intricate, low-level dependencies in stock price data, enhancing prediction accuracy significantly.

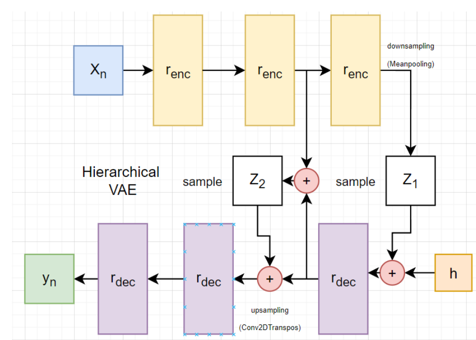

Fig. 2. Heirarchical VAE Structure

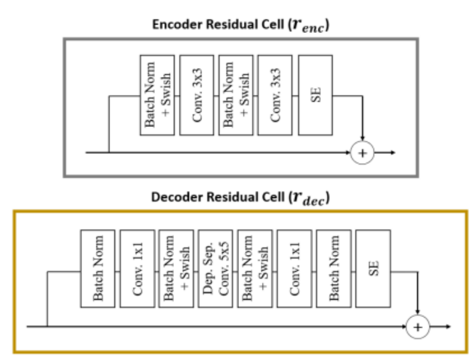

Fig. 3. Encoder and Decoder structures

C. Diffusion Modelling

Input Sequence Diffusion:To facilitate learning from stochastic stock data, we incrementally introduce random noise into the input stock price sequence using diffusion probabilistic models. This involves defining a Markov chain of diffusion steps that gradually adds Gaussian noise to the input sequence X, generating noisy samples $X_1, X_2, ..., X_n$. The noise variance at each step is controlled by a variance schedule $\beta_t \in [0, 1]$. Utilizing the reparameterization trick, we obtain tractable closed-form samples X_t at arbitrary steps. This gradual augmentation of the input series trains the model to predict the target sequence amidst varying noise levels, resulting in more robust and generalizable predictions.

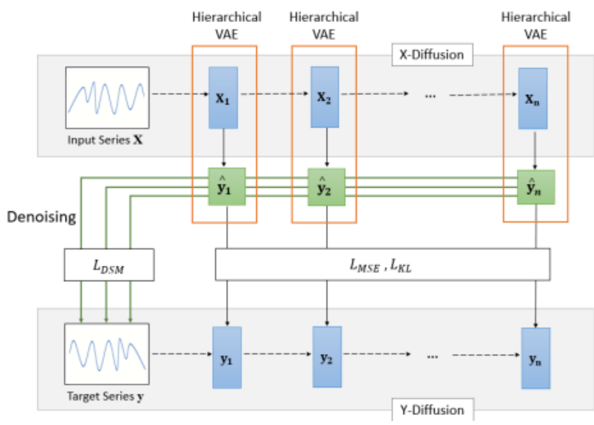

Fig. 4. Overall Generating process

Output Sequence Diffusion:Furthermore, as demonstrated, the introduction of diffusion noise concurrently to sequences X and y, aligning the distributions learned from the generative model and the diffusion process, effectively diminishes the overall uncertainty arising from both the generative model and the intrinsic data noise (aleatoric uncertainty). The relationship is formulated as the KL divergence between noise distributions after the diffusion process and those from the generative process, showcasing uncertainty reduction after augmentation. This coupling of generative and diffusion processes notably decreases the model's prediction uncertainty.

Building upon this insight, we incorporate coupled Gaussian noise to target sequences y, obtaining noisy samples $y1, y2, ..., y_n$. This addition of noise at each step follows a scaling factor $\beta1_t = \gamma * \beta_t$ for y, where γ serves as a hyperparameter. The formulation involves a similar approach as before to obtain samples at arbitrary steps, allowing us to generate the predicted sequence yˆ with reduced uncertainty, simulating stochasticity in the target series akin to the approach applied to input sequence X.

D. Denoising Structure

In D-Va, we replaced the standard diffusion model's reverse process with a predictor for y, eliminating the need for denoising diffused samples. During testing, input sequence X is fed into the hierarchical VAE model to predict the target sequence y. However, y was previously defined as a stochastic series, $y = y_r + \epsilon_y$, not ideal for complete recovery. To capture the "true" sequence yr lying on the actual data manifold, an additional denoising step akin to removing residual noise is performed on the final sample y. This step aims to remove intrinsic noise y from the generated target sequence, reducing aleatoric uncertainty in the series prediction. This denoising score-matching (DSM) process, adapted from standard diffusion probabilistic models, matches gradients to learn an energy function, aiding in recovering yr from a corrupted y sequence with Gaussian noise. At test-time, a one-step denoising jump is executed to obtain the final predicted sequence yfinal, effectively reducing estimated aleatoric uncertainty resulting from data stochasticity.

III. Training and Testing Results

$$L = \mathrm{MSE} + \zeta \cdot \mathrm{KL} + \eta \cdot \mathrm{DSM},$$

$$L_{DSM,n} = \mathbb{E}_{q(\hat{\mathbf{y}}_n|\mathbf{y})}\sigma_n \left\|\mathbf{y} - \hat{\mathbf{y}}_n + \nabla_{\hat{\mathbf{y}}_n} E(\hat{\mathbf{y}}_n)\right\|^2.$$
$$L_{KL} = D_{KL}(p(\hat{\mathbf{y}}_n)||q(\mathbf{y}_n)),$$

where ζ and η denote tradeoff parameters. The MSE term calculates the overall Mean Squared Error between predicted sequence $\hat{y_n}$ and diffused sequence y_n across all diffusion steps $t \in [0, n]$, forming the overarching loss function L for our model.

During training, we apply the coupled diffusion process to both input and target sequences X and y to generate diffused sequences X_n and y_n. The hierarchical VAE is trained to generate predictions $\hat{y_n}$ from the diffused input sequence X_n, aligned with the diffused target sequence y_n. Simultaneously, we train a denoising energy function $\delta(\hat{y})$ to derive "clean" predictions $\hat{y}$ from $\hat{y_n}$.

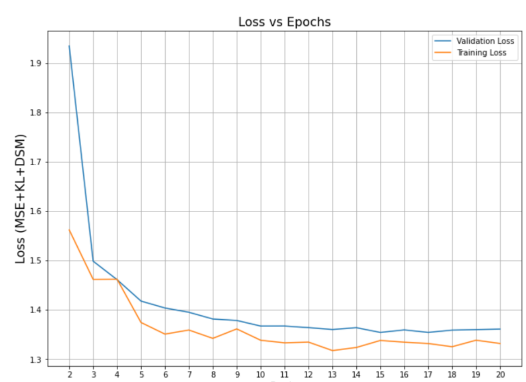

Fig. 5. Training and Validation Loss vs Epochs

For inference, the trained hierarchical VAE generates predictions $\hat{y}$ from input sequences X. These predicted sequences undergo a one-step denoising jump to eliminate estimated aleatoric uncertainty, resulting in our final predicted sequences $\hat{y_{true}}$. For Testing using the model we predicted the Mean of VAE and subtracted it with gradx

$$\hat{\mathbf{y}}_{final} = \hat{\mathbf{y}} - \nabla_{\hat{\mathbf{y}}} E(\hat{\mathbf{y}}),$$

to get our predictions, we got these for all the sequences, we took every 5th predicted sequence to construct the plot.

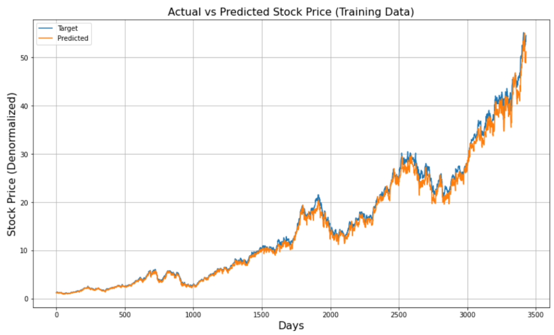

Fig. 6. Prediction vs Actual (Training Dataset AAPL)

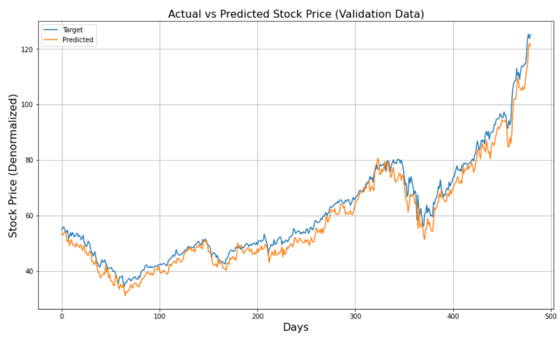

Fig. 7. Prediction vs Actual (Validation Dataset AAPL)

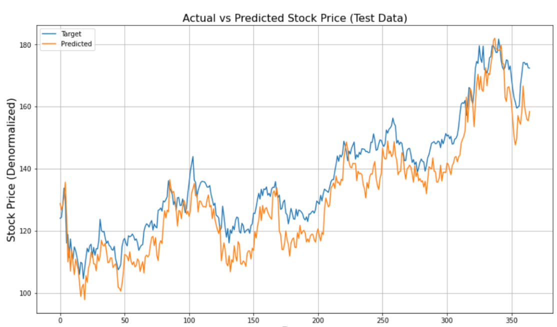

Fig. 8. Prediction vs Actual (Test Dataset AAPL)

IV. Conclusion

In this project, we underscore the significance of multi-step regression in stock price prediction, often overlooked in current literature. We identify challenges related to handling stochastic noise in input data and noise within the target sequence. To address these, we propose D-Va, a framework combining hierarchical VAE and diffusion probabilistic techniques. Extensive experiments show D-Va outperforms existing methods in prediction accuracy and variance. This work suggests avenues for future research: exploring diverse data augmentation techniques(like path shadowing Monte-Carlo), integrating alternative data sources like text or audio, and investigating how these additional sources alleviate prediction uncertainty in stochastic data like stock prices. Additionally, leveraging D-Va's sequence predictions, exploring advanced portfolio strategies like HRP and PEPS could unveil further synergies in financial applications.

References

1) Diffusion Variational Autoencoder for Tackling Stochasticity:
https://arxiv.org/pdf/2309.00073.pdf
2) Diffusion basics:
https://arxiv.org/pdf/2301.03028
3) NVAE:
https://arxiv.org/pdf/2007.03898.pdf
4) NVAE implementation:
https://github.com/GlassyWing/nvae/blob/master/nvae/common.py
5) KL Divergence:
https://stats.stackexchange.com/questions/7440/kl-divergence-between-two-univariate-gaussians
6) Denoising and Gradient estimation steps:
https://arxiv.org/abs/1907.05600v3
7) Denoising Score Matching:
https://www.activeloop.ai/resources/glossary/denoising-score-matching/: :text=Denoising
8) Weight initialization:
https://365datascience.com/tutorials/machine-learning-tutorials/what-is-xavier-initialization/: :text=For
9) Learning Rate Scheduler:
https://towardsdatascience.com/a-visual-guide-to-learning-rate-schedulers-in-pytorch-24bbb262c863