Q1. *(15 points) Experiment with different window sizes and steps. Train the model using 3 different combinations of window size and step. Evaluate the Mean Squared Error (MSE) for each configuration. Report the MSEs using a table and analyze the results. (Approximately 100 words.)*

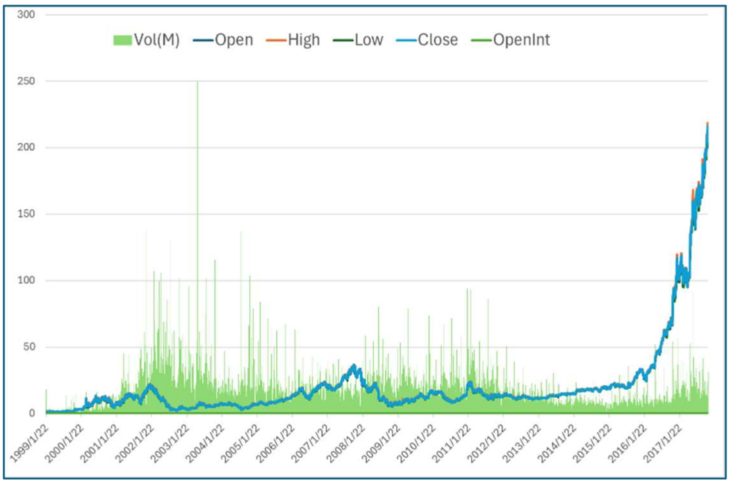
While smaller windows and fewer steps yield more accurate predictions for the near future (evidenced by Trial#1 and Trial#4), they may not be optimal for long-term forecasting. Such configurations, which focus on immediate price movements, can be valuable for short-term trading strategies. However, they may overlook longer-term trends crucial for predicting further out. Conversely, Trial#6, with a larger window, could potentially incorporate these broader trends, but at the cost of increased MSE, indicating a trade-off between short-term accuracy and long-term foresight. Tailoring model parameters to the investment horizon is key: striking a balance between capturing imminent price actions and longer-term market dynamics.

Trial#	Window Size	Steps	Best Val Loss (MSE)
Trial#0	10	15	185.2501
Trial#1	10	5	2.096
Trial#4	5	2	1.8013
Trial#6	30	10	106.6195

Q2. *(Approximately 200 words.)*

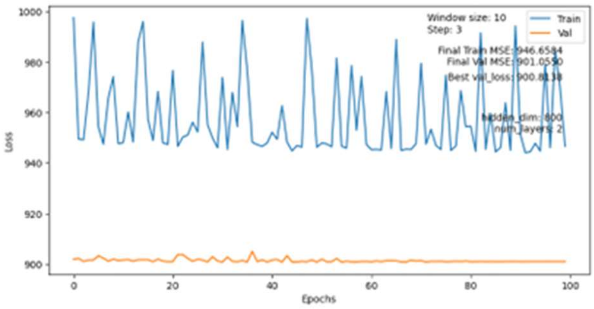
(i) (15 points) Include 'Volume' as an additional input feature in your model. Discuss the impact of incorporating 'Volume' on the model's performance.

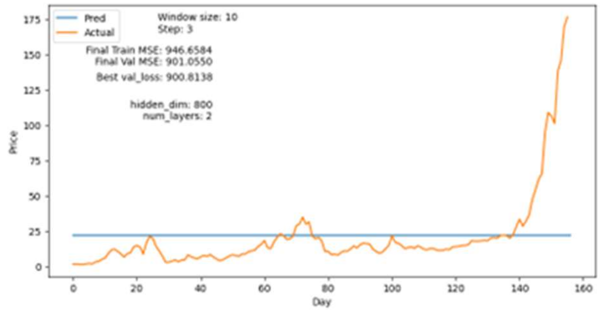
Adding Volume as an input to stock price prediction model without adjusting its scale can lead to challenges due to the vastly different magnitudes between volume and price data. Volume reflects the number of stocks traded and typically exhibits large, volatile numbers, which can dominate the model's processing capabilities, overshadowing more stable price movements. This disparity in scale can affect the model's ability to converge and accurately interpret the data. Without appropriate adjustments to the volume data, models might struggle to effectively integrate this variable, potentially degrading their predictive accuracy and limiting their capacity to capture essential market dynamics.



(ii) (15 points) Explore and report on the best combination of input features that yields the best MSE. Briefly describe the reasons of your attempts and analyze the final, optimal input combination.

While ensuring overlapping window data (i.e. step < window size), various combination of window size ranging from 5 to 50 and step ranging from 2 to 10 were tuned but unsuccessful. The performance ended up with horizontal prediction line where MSE is at the range of 900 to 1400. Such MSE level is totally unacceptable. The best result that can be achieved is with window size of 10 and step of 3 but the MSE is still high at 900.8138, which again, is still unacceptable.



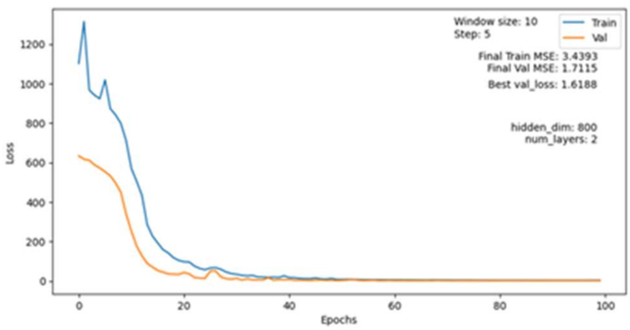


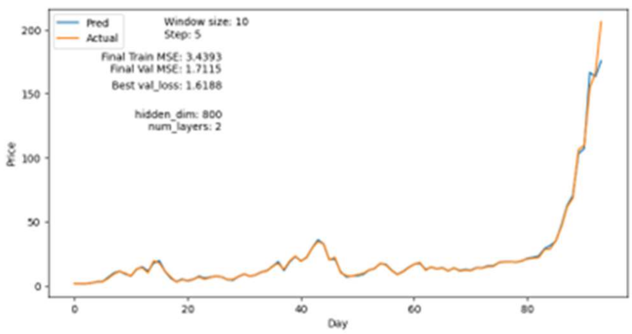
Q3. *(15 points) Analyze the performance of the model with and without normalized inputs in Lab 4. You can use experimental results or external references (which must be cited) to support your conclusions on whether normalization improves the model's performance. (Approximately 100 words.)*

Without normalization, the model struggles with large scale discrepancies between volume against the price variables, resulting in a very high MSE and poor predictive accuracy. This is primarily because the algorithms perform better when numerical input variables have similar scales and distributions. By normalizing the data—scaling down the volume by a factor of a million and applying a logarithmic transformation (log1p)—the skewed distribution of volume data is adjusted, making it comparable with the other price data. This significantly drives down the MSE in normalized case to below 1.6188. Alternatively, this reference paper "Impact of data normalization on deep neural network for time series forecasting (2018)" describes different normalization methods used.

Q4. *(10 points) Why should the window size be less than the step size in Lab 4? Do you think this is correct? If you use external sources, please include references to support your response. (Approximately 50 words.)*

It is incorrect. Using an LSTM with a window size smaller than the step can miss critical trends, leading to skipped data and temporal discontinuities. This results in inefficient learning and higher errors. The complexities of such configurations are discussed in "Deep learning with long short-term memory networks for financial market predictions" by Fischer and Krauss (2018).





Q5. *(15 points) Describe one method for data augmentation specifically applicable to time-series data. Cite references to support your findings. (Approximately 100 words.)*

Synthetic Minority Over-sampling Technique (SMOTE) is effective in generating synthetic samples by interpolating between positive instances in the minority class. This can significantly improve the performance of classifiers by balancing the class distribution. It is especially efficient in scenarios where the minority class is underrepresented, helping models to better generalize from limited data. However, its efficiency can diminish if the minority class examples have a lot of noise, as SMOTE might generate unhelpful synthetic samples. Reference paper: "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining".

Q6. *Discuss how to handle window size during inference in different model architectures (approximately 150 words):*

(i) (5 points) Convolution-based models
(ii) (5 points) Recurrent-based models
(iii) (5 points) Transformer-based models

(i) **Convolution-based Models:** Utilize fixed window sizes to enhance accuracy in time-series tasks such as electricity demand forecasting, ensuring effective noise minimization and feature extraction. Overlapping windows provide better data coverage and improve model predictions by capturing subtle changes, albeit at a higher computational cost. This balance ensures models perform optimally for applications requiring detailed temporal analysis.

(ii) **Recurrent-based Models:** These are effective with variable window sizes, allowing for flexible adaptation to diverse time-series data, such as financial market trends. Their stateful processing capability maintains continuity across sequences, enhancing predictive accuracy while efficiently managing computational resources. This makes them particularly suitable for applications like real-time stock price forecasting, where data continuity is crucial.

(iii) **Transformer-based Models:** Apply fixed or sliding windows for complex time-series predictions, such as weather pattern forecasting. Sliding windows help manage long sequence data without overwhelming computational resources, while advancements like Efficient Transformers optimize accuracy and computational efficiency. This makes transformers ideal for handling extensive time-series data requiring nuanced understanding over long periods.