Final Report

Contents

3.1. Introduction/Background:

3.1.1. Literature Review:

With an abundance of papers written about various methodologies for taming the unpredictable stock market, there is no shortage of concise, effective papers to use as a basis for our models. For example, Barra et al.[1] proposed these parallel computations on existing pre-trained models in order to be able

to utilize strong CNN models without developing new ones from scratch or using 1D-CNN models. To balance the work of CNN, we also have evaluated the work of Farrell & Correra [2] as an application of GPR for stock valuation. Finally, we have the work by Lin et al.[3] for ResNet applied to stock evaluation.

3.1.2. Dataset Description:

A collection of stock price values from NYSE containing 514 stocks since IPO. Includes daily and hourly highs, lows, start, and end price information on each stock.

3.1.3. Dataset Link:

https://www.kaggle.com/datasets/olegshpagin/usa-stocks-prices-ohlcv

3.2. Problem Definition

3.2.1. Problem:

With the volatile post-COVID economy and popularity of trading apps targeted at retail investors, accurately being able to predict the stock value based on historical data is not an easy task leading to uninformed decision making.

3.2.2. Motivation:

Training a comprehensive model to accurately predict the price of a given stock to help reduce risk to inexperienced investors and encourages more sustainable investment practices.

3.3. Methods

3.3.1. Data Preprocessing Methods:

The data processing methods were kept constant for each model in order to ensure a fair comparison for each. Different from the midpoint, not all windows were used, instead it was a random selection. Additionally, the individual stock price information was normalized to ensure that stock was evaluated on its relative movements and not on the actual value of the stock (especially since this differed from stock to stock).

The sliding window approach analyzed the stock data that was in 5-minute intervals and looked for a significant change over an interval of 1 hour. The data from the 8 hours leading up to the window was logged as well as the relative change. The philosophy for this came from the research before the implementation. If there was an event that would push the stock such as an

earnings call, it would take time for the stock to change in price. The threshold for “significant change” was defined as any change between 3% and 1% for a positive change and -3% and -1% for negative direction changes in stock price.

In order to preserve good training practice and model visibility, data during the window was obscured from all models during training since this directly influences the calculation for the change it was trying to predict.

4 machine learning approaches were attempted: a 1D Resnet, Gaussian Process Regression (GPR) Ensemble, Decision Tree, and Convolutional Neural Networks with varying results.

3.3.2. ML Algorithms/Models:

The 1D ResNet model used in this project is a simple architecture designed for sequential data, which was used as the baseline in this project. The model takes a single channel input representing the time data, though no actual time stamps were passed in. The first layer is a convolutional layer with a 7-sized kernel, stride of 2, and padding of 3, which maps the input to 16 feature channels while halving the temporal resolution. The next layer is a batch

normalization layer to stabilize the training process and a ReLU function is used to introduce non-linearity. A max-pooling operation further reduces the temporal resolution by half.

The core of the model is made up of three sets of residual layers with each residual block containing two convolutional layers with a 3-sized kernel, stride of 1, and padding of 1. The first set of residual layers maintains the temporal resolution, while the second and third sets halve the resolution and increase the number of feature channels to 32 and 64, respectively.

After the residual layers, an average pooling layer reduces the temporal dimension to 1. The output of this layer is flattened then passed through a fully connected layer to produce the final output, which in this instance was the relative predicted change over the next hour after the given 8 hours of data from inference. This architecture was designed to leverage residual learning, which mitigates the vanishing gradient problem and allows for efficient pattern recognition for its size. However, this does come with the caveat that its simplistic architecture might not be able to notice the more nuanced patterns present in this financial dataset.

The Convolutional Neural Network model was built along more of a straightforward architecture designed on the same sequential data as the 1d ResNet model. As this used the

same uniform dataset, the input was a single channel representing time data. The first layer being the convolutional layer had a 5-sized kernel, stried of 1, and padding of 2, which mapped the input to 32 feature channels while preserving the input of 32 feature channels. This was followed up with a ReLU function to introduce non-linearity. The second convolutional layer used a 3-sized kernel, stried of 1, and padding of 1 increasing the feature channels to 128. With the last dilated convolutional layer using a 3-sized kernel, stried of 1, padding of 2, and dilation of two furthermore increasing the feature channels to 256. Finally a gloabl average pooling layer was than used to reduce the temporal dimension to 1 to produce the final output.

The Gaussian Process Regression (GPR) model applied in this project is utilizing a common probabilistic approach which is mentioned to model complex relationships in data. It used each data point from the 8 hours of data (96 points) after normalization and preprocessing as features. This model inherently relies on a kernel function knows Radial Basis Function (RBF) due to the it’s ability to capture smooth relationship.

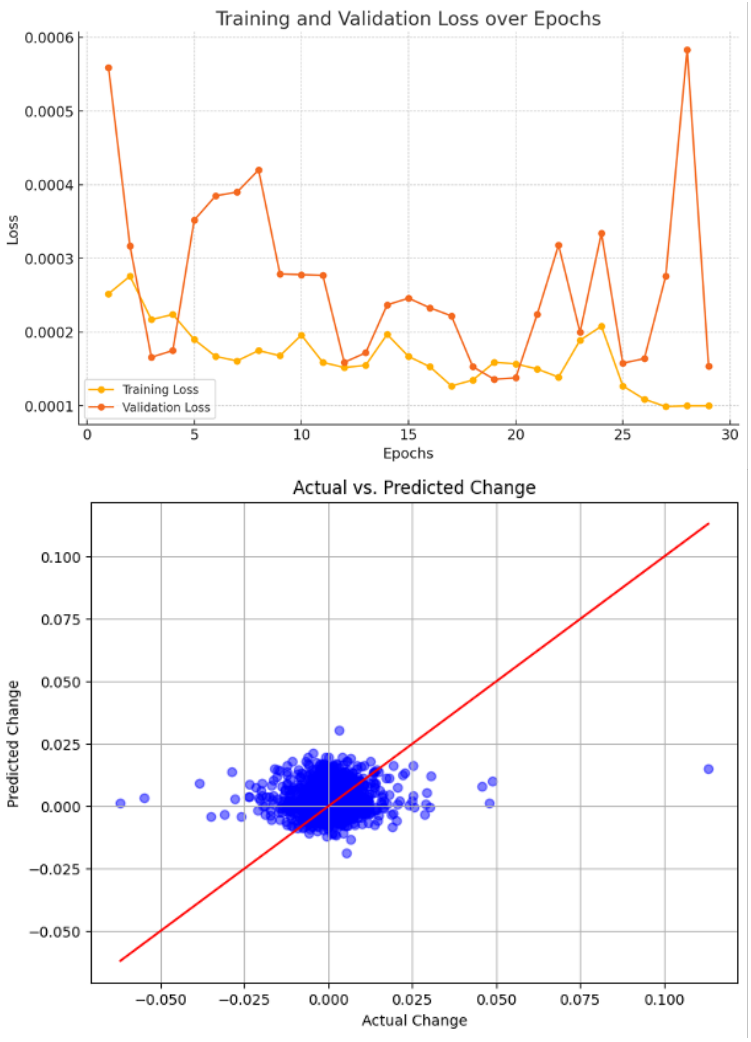
The training process involves optimizing the marginal log-likelihood, as well adjusting hyperparameters of the kernel based on the first and second moments of these optimizations. The GPR model then produces predictions by

conditioning the process on the training data, and it then computes predictions and confidence intervals for all of the test data.

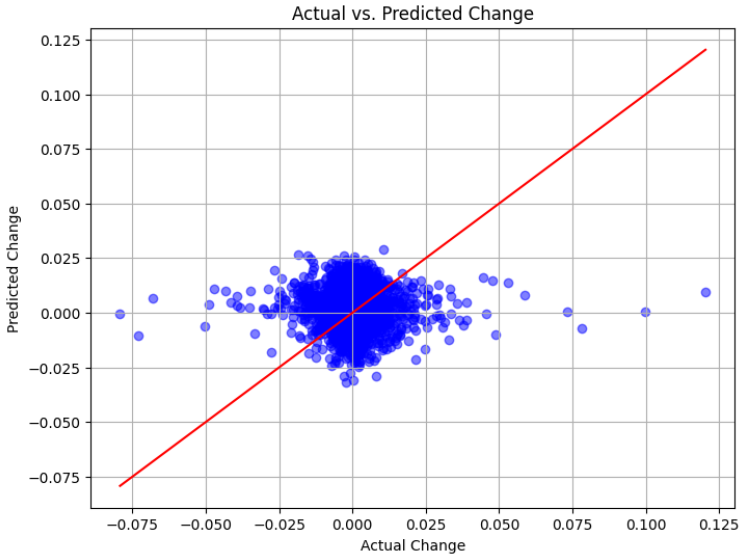
3.4. Results and Discussions

3.4.1. Visualizations:

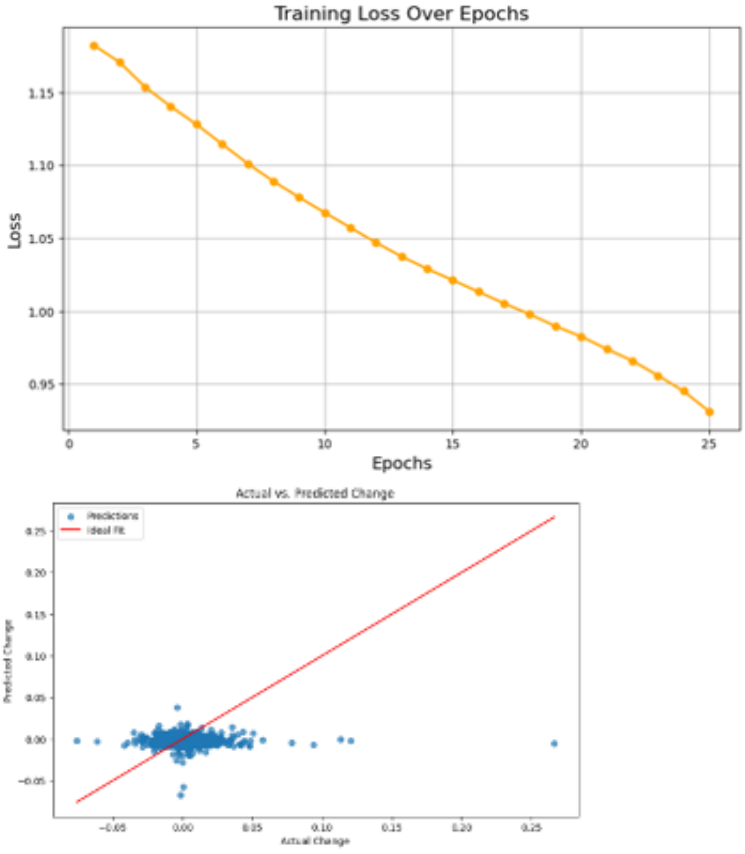
3.4.1.1. 1D Resnet:



3.4.1.2. CNN:



3.4.1.3. GPR:



3.4.2. Quantitative Metrics:

The quantitative metrics that were collected help us gain a more comprehensive evaluation of our 1D Res-Net model's. The Test Loss and Mean Squared Error (MSE) both stand at 0.000206, indicating that the model's predictions are closely aligned with the actual stock values, although with small deviations. The Mean Absolute Error (MAE) of 0.013682 means that in general the prediction made nby the model is off from the actual value by approximately 1.37%. With this in mind however, the R-squared value of -0.000599 means that the model doesn't account for the variability of the stock prices well. Specifically the fact that the model performs worse than a horizontal line, more commonly known as the mean of the data. This is extremely significant as it means that the model struggles to capture the complex patterns and variability inherent in stock price movements.

For the CNN the Test Loss was 0.000080 which indicates that the model's prediction were somewhat aligned with the actual stock values. This is further supported by our 0.000160 Mean Squared Error. Lastly the Mean Absolute Error of 0.009283 suggests that on average the model's predictoins are off from the actual values by approximately 0.93%.

With this being said, the R^2 values of

-0.964676 does state that the model is not able to account for the variability of the stock prices well. Honing in on the negative portion of this value helps to explain that the model really struggles to capture the complex patterns in stock pricing.

For the GPR, the mean squared error was 0.000134, while the mean absolute error was 0.007002. When dealing with such low numbers, these weren't great. The R^2 value was -0.268, indicating a bad correlation prediction of the model.

3.4.3. Analysis of 3+ Algorithm/Model:

The ResNet model did not perform to expectations. Despite hyperparameter tuning and introducing anti-overfitting techniques such as early stopping with training epochs, the 1D Resnet model failed to learn the underlying patterns in the data. Even with a lower learning rate, the model's performance did not significantly improve. The MSE showed that the model learned to perform well on the training data, however, the R-squared value indicates that the model overfit heavily. Based on the results of predicted and actual, it was apparent that the model was very risk adverse, predicting most changes to be closer to 0 and much less likely to predict large movements in stocks. In

the future, more noise or adding other features could help the model to learn more underlying patterns.

Below are the metrics:

- Test Loss (MSE): 0.000051
- Mean Squared Error: 0.000099
- Mean Absolute Error: 0.007093
- R-squared (R^2): -0.521279

The CNN tried its best to capture the multivariate variability that comes with stock pricing. A lot of hyperparameter tuning took place as well as breaking the training epochs early to try and stop overfitting however this only helped us improve our model's performance slightly. The issue is that our model overfit extremely well to our training data and was not able to make inferences when tested outside of this environment as seen in the R^2 value. As seen in the visualization graphic, the CNN model predicted

Metrics for CNN:

- Test Loss (MSE): 0.000080
- Mean Squared Error: 0.000160
- Mean Absolute Error: 0.009283
- R-squared (R^2): -0.964676

For GPR, the model was also unfortunately not able to sufficiently find a fit to the data. After

many tweaks and hyper parameter tuning methodologies, the model was only finding a fit for the training data and could not extrapolate to the test data set. The GPR has a few weaknesses that were on display during the development of this model. Firstly, for the number of features we had, the dataset was rather large and included many similarly valued results, and GPR performs poorly with imbalanced and biased data. Unfortunately attempts to cull the dataset for better results did not improve the model. In addition, GPR has a tendency to vary heavily based on initial hyperparameter settings.

Below are the metrics:

- Mean Squared Error: 0.000134
- Mean Absolute Error: 0.007002
- R-squared (R^2): -0.268565

3.4.4. Comparison of 3+ Algorithm/Model:

3.4.5. Next Steps:

3.5. References:

[1] S. Barra, S. M. Carta, A. Corriga, A. S. Podda, and D. R. Recupero, “Deep learning and time

series-to-image encoding for financial forecasting," IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 3, pp. 683–692, May 2020, doi: https://doi.org/10.1109/jas.2020.1003132.

[2] T. Farrell and A. Correa, "Gaussian Process Regression Models for Predicting Stock Trends," ResearchGate, Jan. 2007. https://www.researchgate.net/publication/249769503_Gaussian_Process_Regression_Models_for_Predicting_Stock_Trends (accessed Oct. 05, 2024).

[3] H. Lin, J. Zhao, S. Liang, and H. Kang, "Prediction model for stock price trend based on convolution neural network," Journal of Intelligent & Fuzzy Systems, vol. 39, no. 4, pp. 4999–5008, Oct. 2020, doi: https://doi.org/10.3233/jifs-179985.

3.6. Gantt Chart

https://docs.google.com/spreadsheets/d/1BfBJEoKF4znHZFGCZvamXogz6soi_jKy/edit?usp=sharing&ouid=112451109936451276976&rtpof=true&sd=true

3.7. Contribution Table:

Name	Contributions
Ankit Agrawal	GPR, Video Presentation
Namkhang Le	Website, CNN
Mason Rein	
Joshua Muehring	Website, CNN
William Pardo	Data Preprocessing, 1D Resnet Model