CSCD94 Week 9 Update

Dominik Luszczynski

Completed Tasks

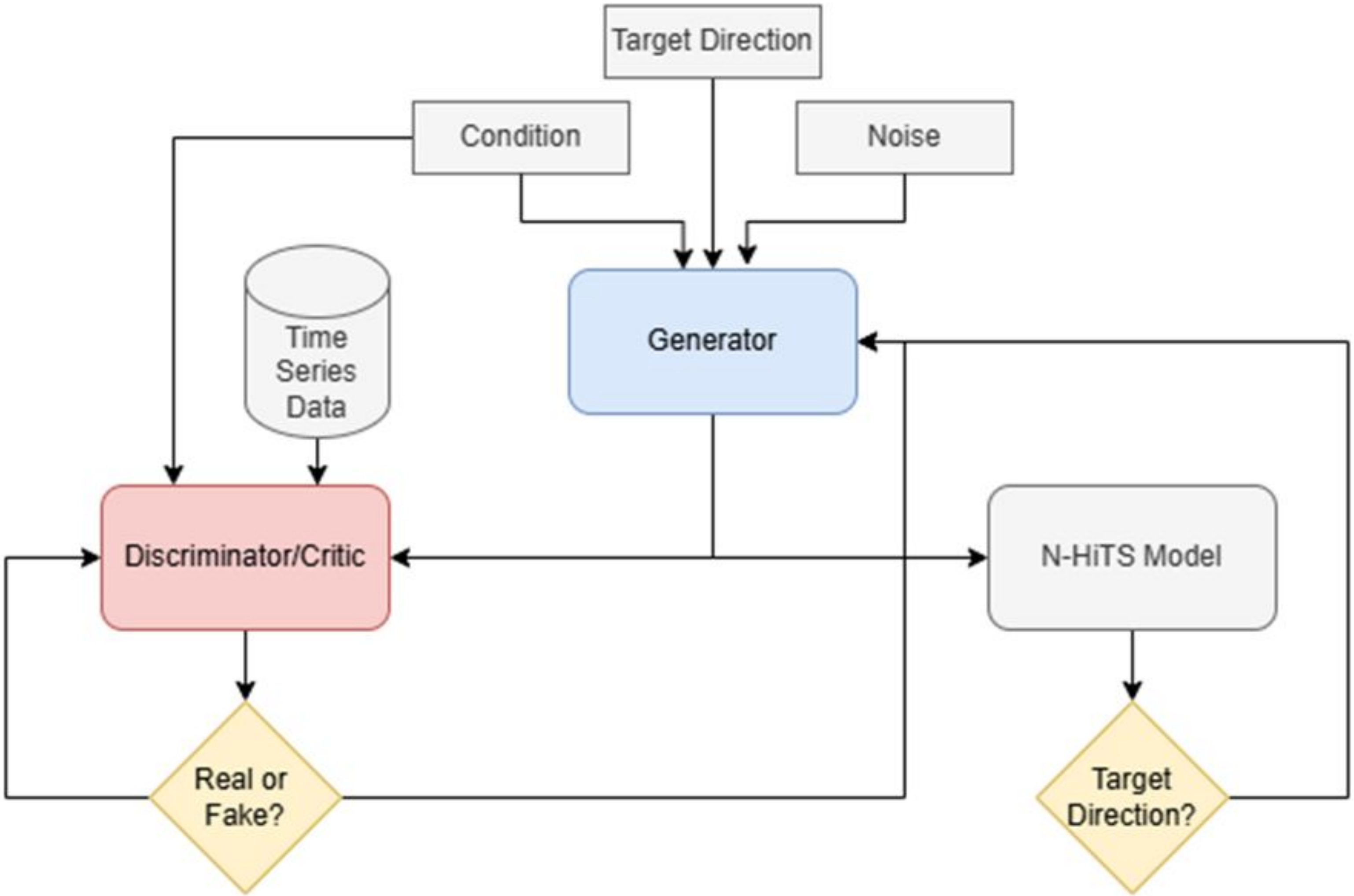
- Extend the conditional WGAN with a second critic being the N-HiTS model.

Problem

- Niagara/Mist does not support pytorch-forecasting, so I need to train locally which takes around 10 hours.

Adv-CWGAN-GP

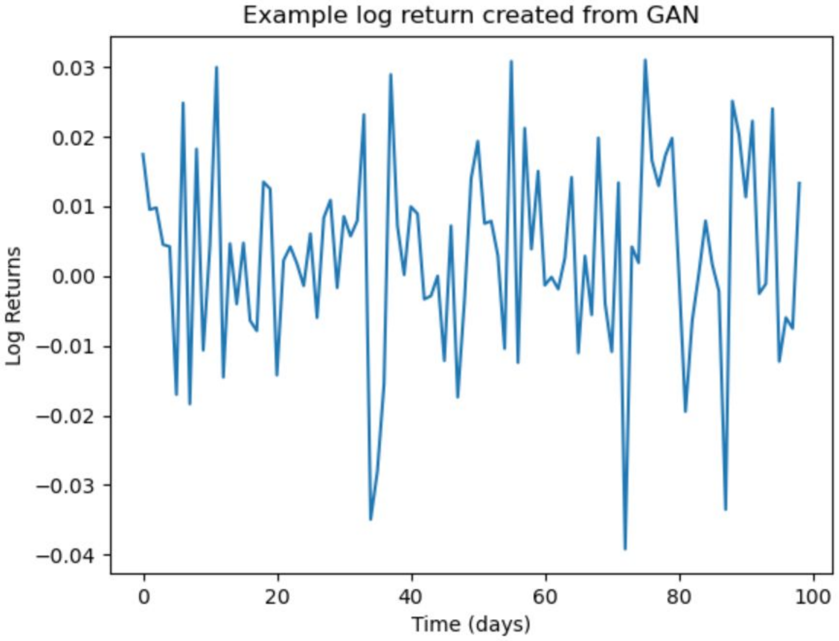
- Conditioned on 99 days of real log returns, we generate 99 days of synthetic data.
  - We condition in the feature dimension (we want to match the real log returns as much as possible).
- The two player game between the generator and discriminator remains the same.
- Computing the adversarial loss (l1-loss) (generator step):
  1) Compute the loss between the next 20 days of real adjprc with the 20 day predictions based on the synthetic data (current implementation).
     a) GOAL: Maximize error
  2) Compute predictions from both the real and synthetic data, and get the difference in the loss.
     a) GOAL: Maximize difference between real predictions and fake predictions.
  3) Targeted attack (next week): Either choose a target or desired slope, and compute loss based on that

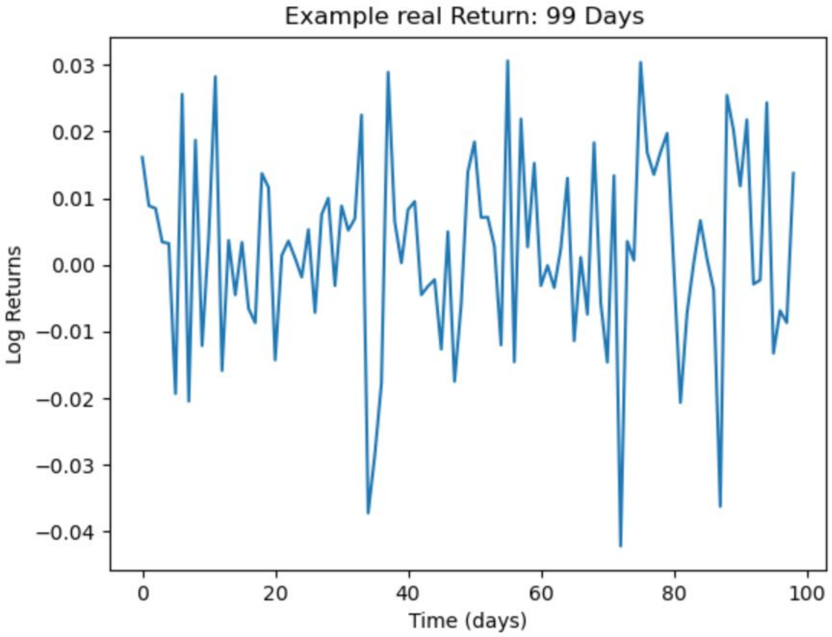

Target Direction
Condition
Noise
Time Series Data
Generator
Discriminator/Critic
N-HiTS Model
Real or Fake?
Target Direction?

Incorporating the Adversarial Loss

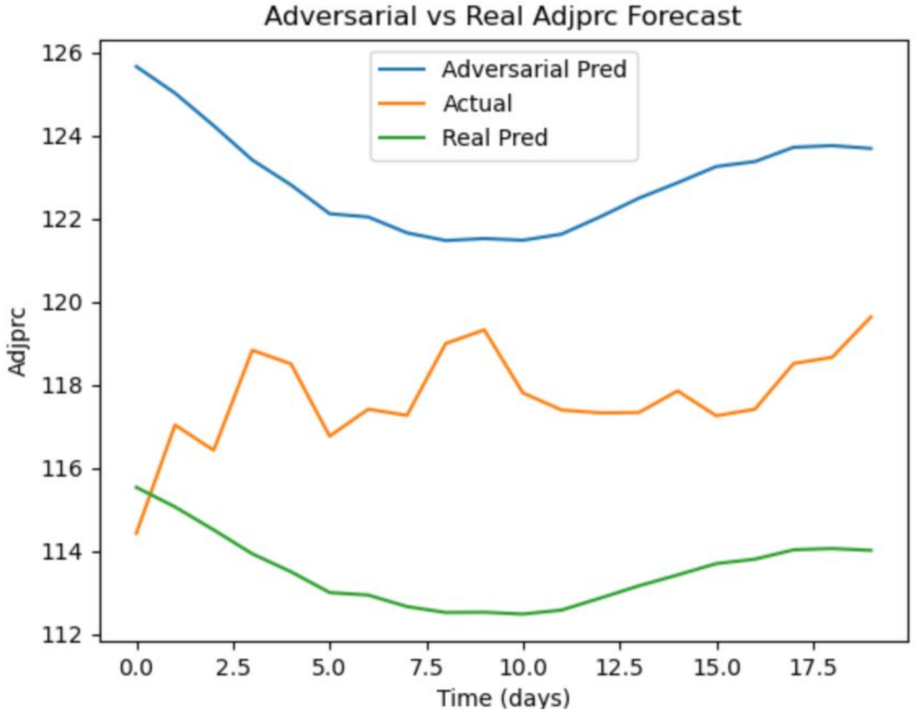
- Similar to a learning rate scheduler, we increase the scale (beta) of the adversarial loss after a set number of epochs.
- This allows for the GAN to initially generate realistic data before we aim to generate adversarial data.

Example


Example log return created from GAN
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
Log Returns
0
20
40
60
80
100
Time (days)

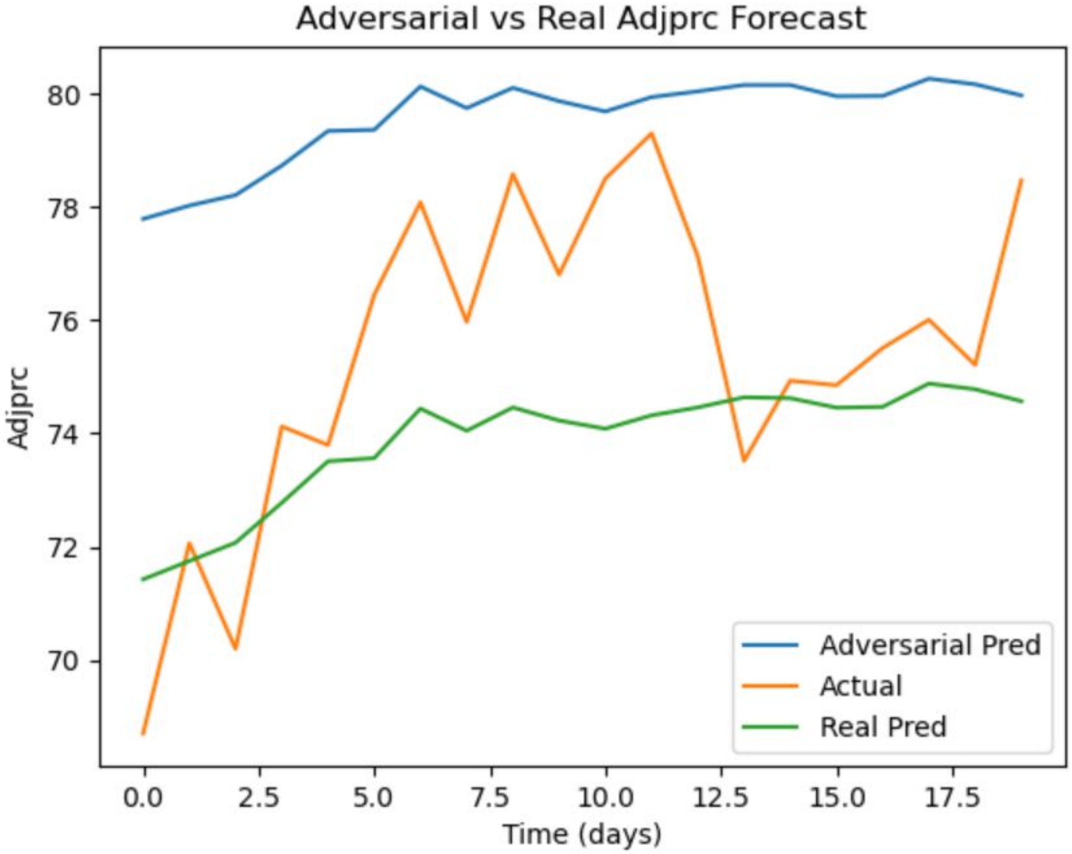

Example real Return: 99 Days
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
Log Returns
0
20
40
60
80
100
Time (days)

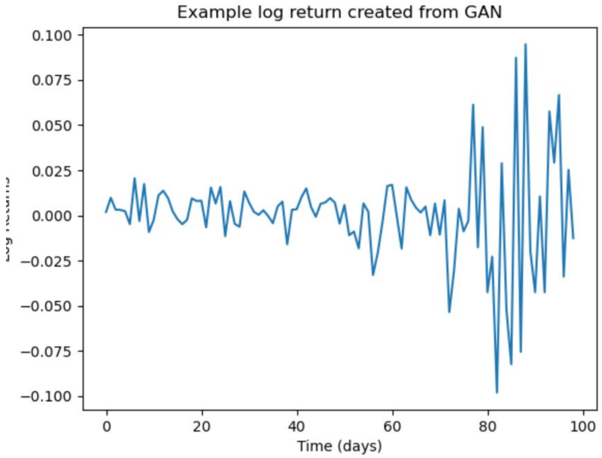
By nature of the GAN (and log returns), the synthetic data already cause

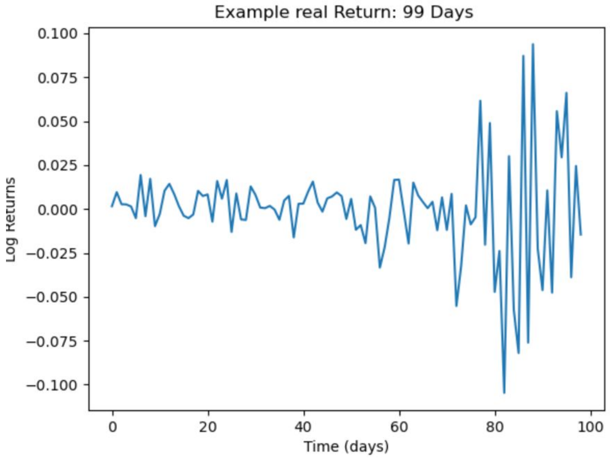


Epoch: 175, Beta: ~0.24

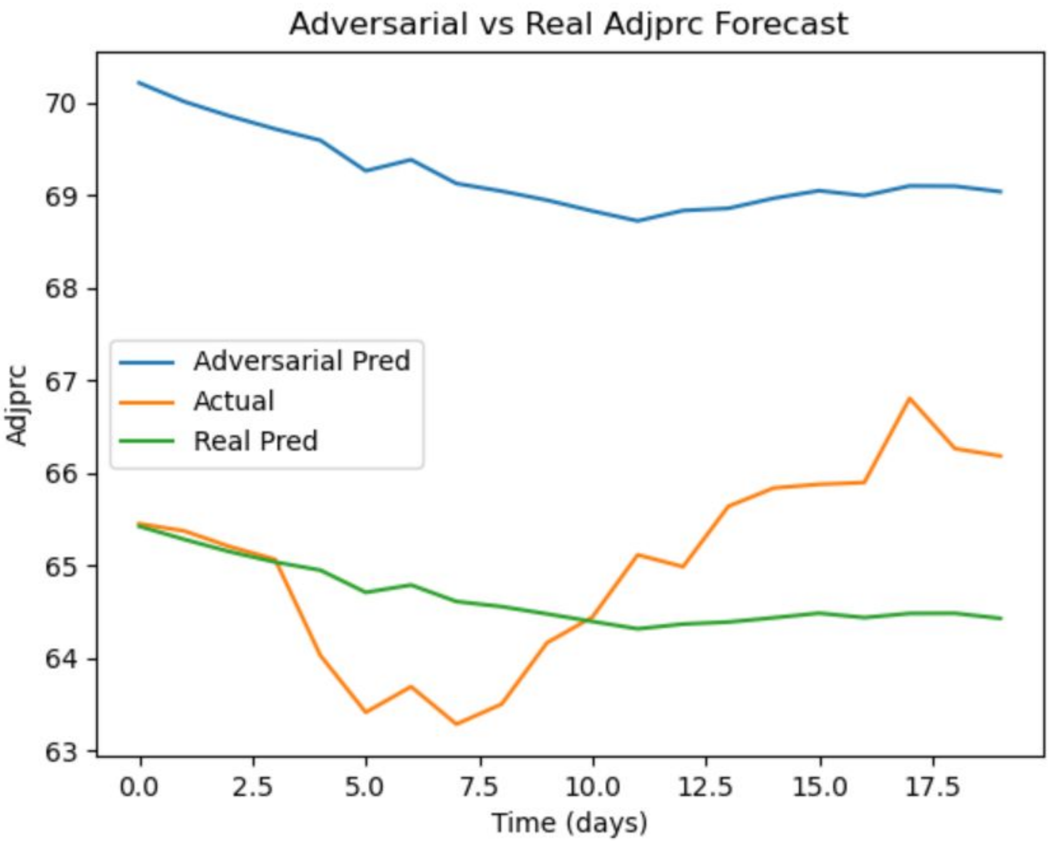
Epoch: 171, Beta: ~0.24

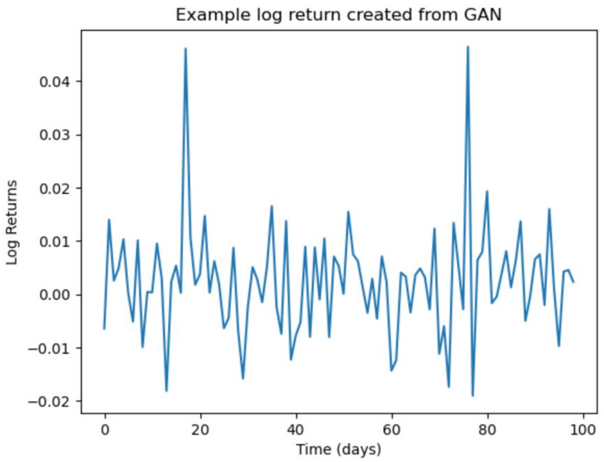


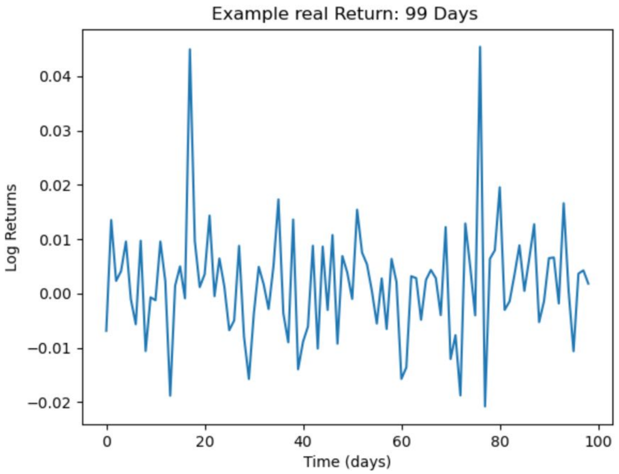




Epoch: 171, Beta: ~0.24







Why is the adversarial prediction always slightly higher?

```
==================================================
Sample Mean: 0.000551883065317805, Fake Mean: 0.001460602420302118 7
Sample Stdev: 0.016635720814690176, Fake Stdev: 0.01619742999326766 2
Sample IQR: 0.018717247656138185, Fake IQR: 0.018063817046204047
Sample skew: -0.28708057394500924, Fake skew: -0.2820244639347584 4
Sample kurtosis: 5.544345445377175, Fake kurtosis: 5.686985435413195
Similarity Loss 0.8480635855704955
Adversarial Loss 2.02350829425191
Final Loss 1.3282515889916033
==================================================
```

The mean is slightly higher

Therefore, we need to have a target

- If synthetic log returns already naturally cause a difference in performance, even though they visually look identical, then we should towards defining a target for the GAN (which was the original idea).
- Rather than simply having a target, I would like to try defining a “slope-like” loss, where rather than defining a target where we would shift the entire time prediction up or down, we would want to start off at a similar adjprc, then either plummet or skyrocket.

Next Steps

- Plot adjprc to see if differences are more noticeable than log returns.
- Add targeting to the adversarial model.
- Perform hyperparameter tuning for the “Adversarial Scheduler.”