CSCD94 Week 11 Update

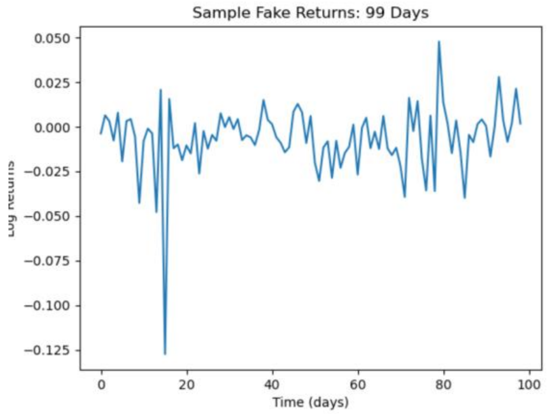
Dominik Luszczynski

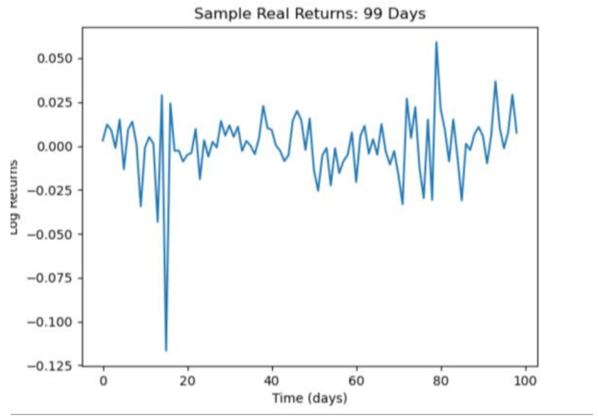
Last Week – Slope Based Attack

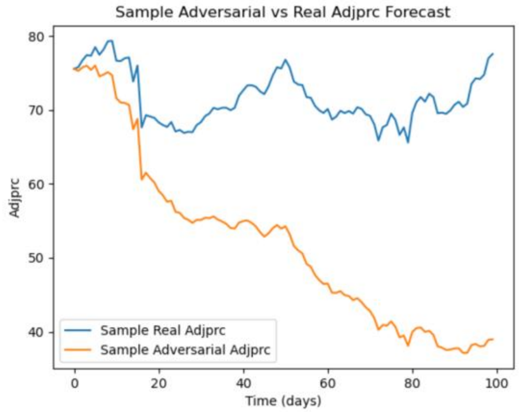
1. x_i = adjprc
2. For i in range of num_iterations:
    1. x_i.requires_grad_(True)
    2. model.zero_grad()
    3. pred = get_predictions(x_i)
    4. **slope = (pred[-1] – pred[0]) / len(pred)**
    5. loss = slope_loss(slope)
    6. loss.backward()
    7. With no gradients:
        1. grad = loss.grad.data
        2. sign_grad = grad.sign()
        3. noise = step_size * sign_grad
        4. x_i = x_i – noise # Want to move in direction of gradient to minimize the loss
        5. x_i = clamp(x_i, adjprc – epsilon, adjprc + epsilon)
    8. x_i.detach()

Problems

1. We can't directly use **slope = (pred[-1] – pred[0])/ len(pred)** in the GAN.
    a) The gradients will only be non-zero at pred[-1] and pred[0], which does not give the GAN enough information on how to alter its generated sequences.
2. Given the forecasting model works in price space, we shouldn't be generating intervals in the log return space since tiny fluctuations can drastically affect the adjprc.







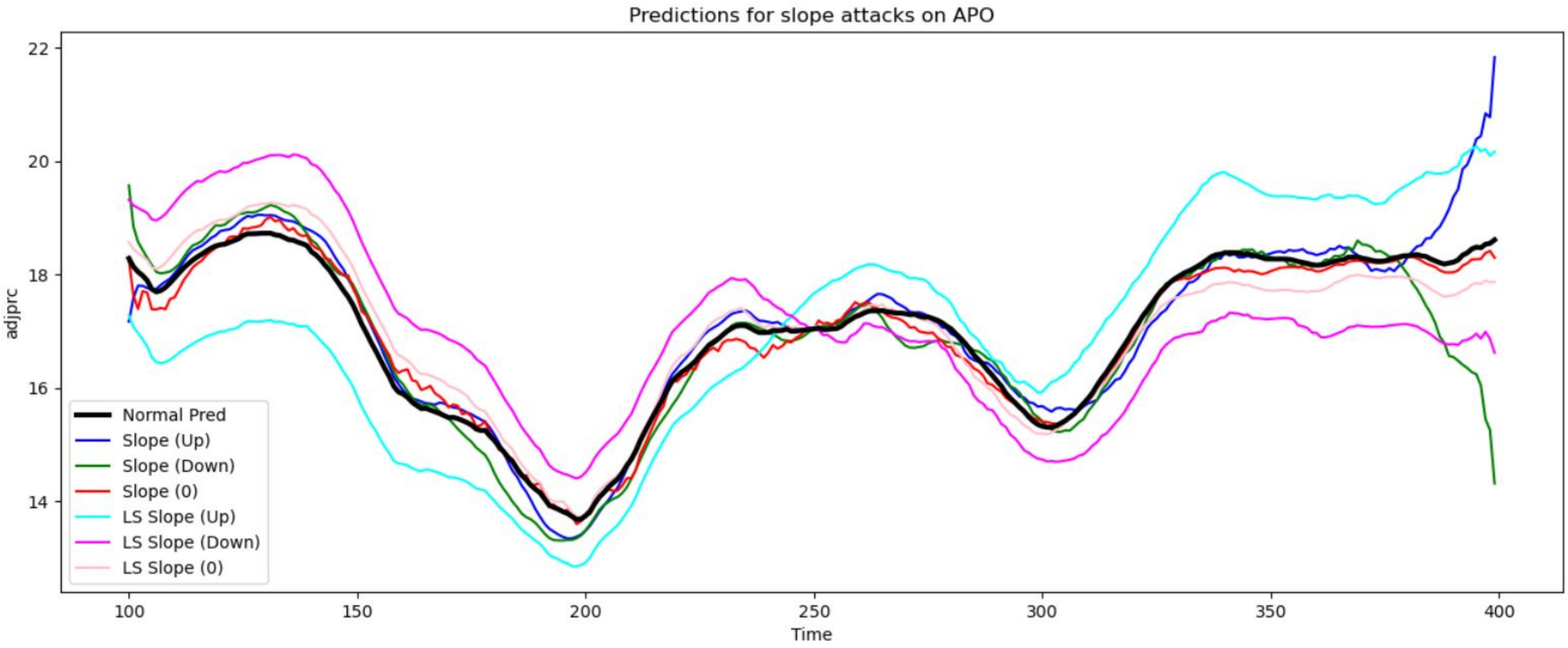
Solutions

- Rather than computing the slope using the endpoints of the interval, we should try to capture the overall slope/trend of the forecast.

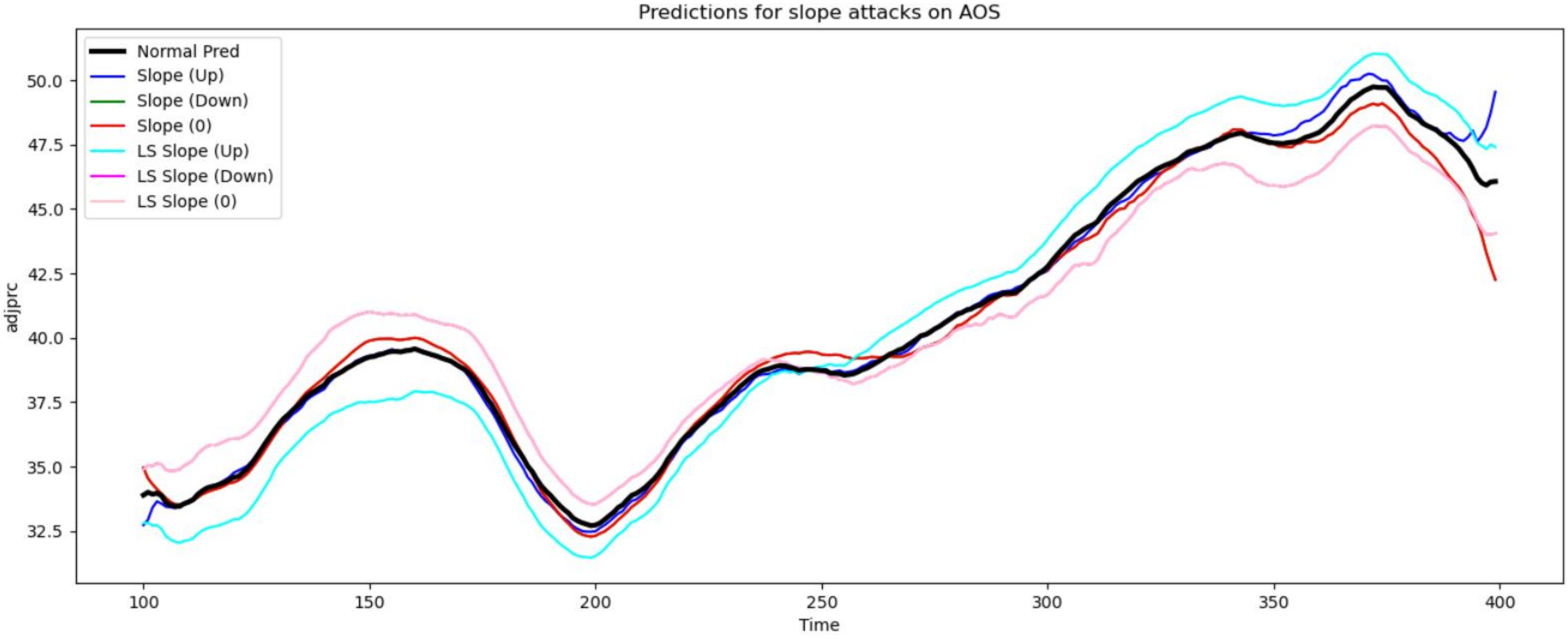
$$f(x) = wx + b$$

$$w^* = \frac{\sum_i (y_{i-}\bar{y})(x_i - \bar{x})}{\sum_i (x_i - \bar{x})^2}$$

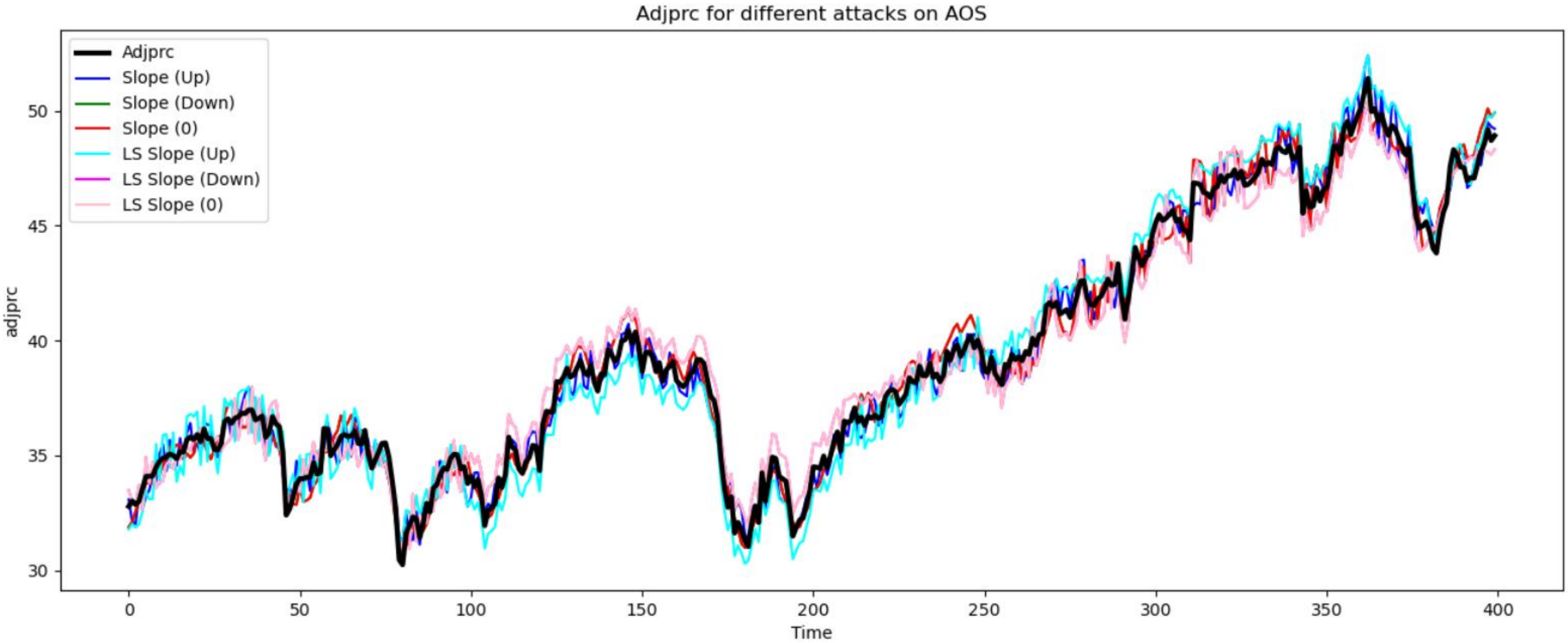
Adding the LS-Slope to the BIM



Adding the LS-Slope to the BIM



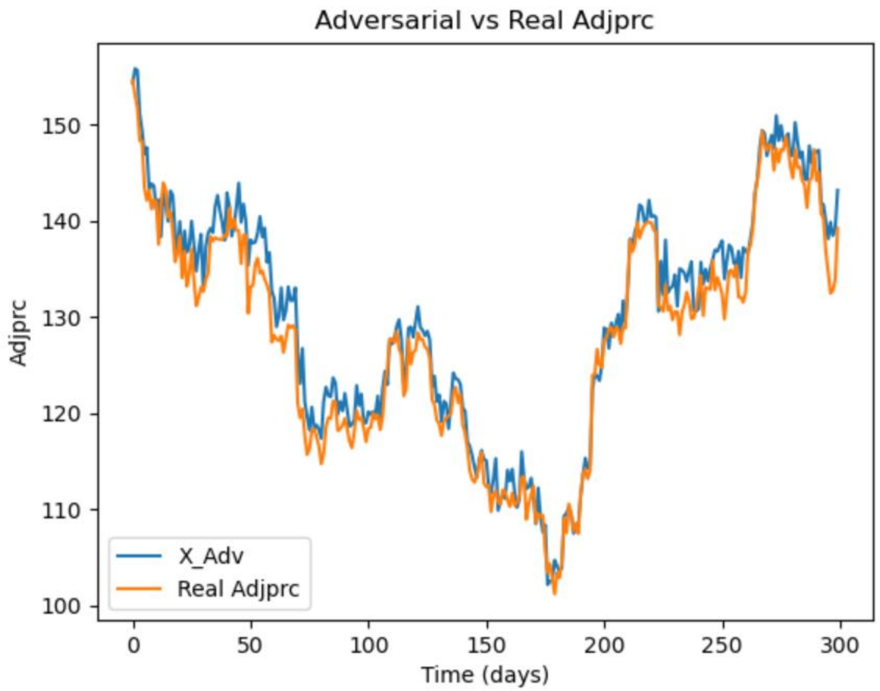
But The Attacks Look More Noticeable

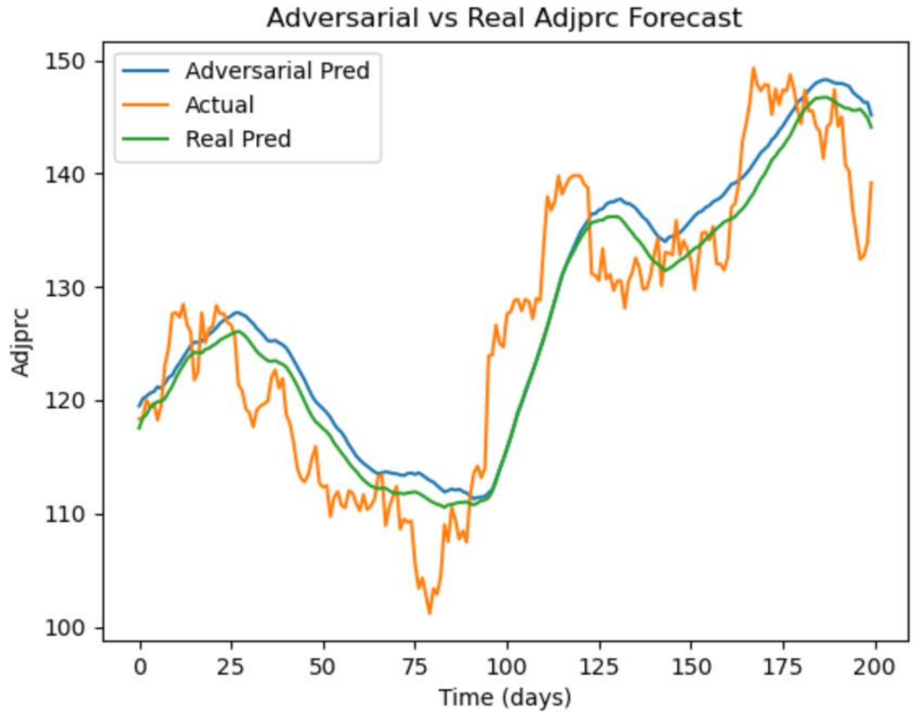


What about the GAN?

- Changes (some due to the need to run locally):
  - Removed GRU blocks in critic (full TCN)
  - Generate adjprc sequences rather than log returns
  - Added a norm loss (2-norm) to help the generator generate realistic predictions.
  - 20-day forecasts might not be enough time to get meaningful slope, so rather than generating 100 days, we generate 300.
  - Added the LS-Slope rather than the y2-y1/x2-x1 version
  - Training is done in stages
    - Train for 50 epochs at a time, increasing the scale of the adversarial loss, save the best model
    - In final iteration, disable the critic and just have the generator learn from the norm loss and adversarial loss (for now).

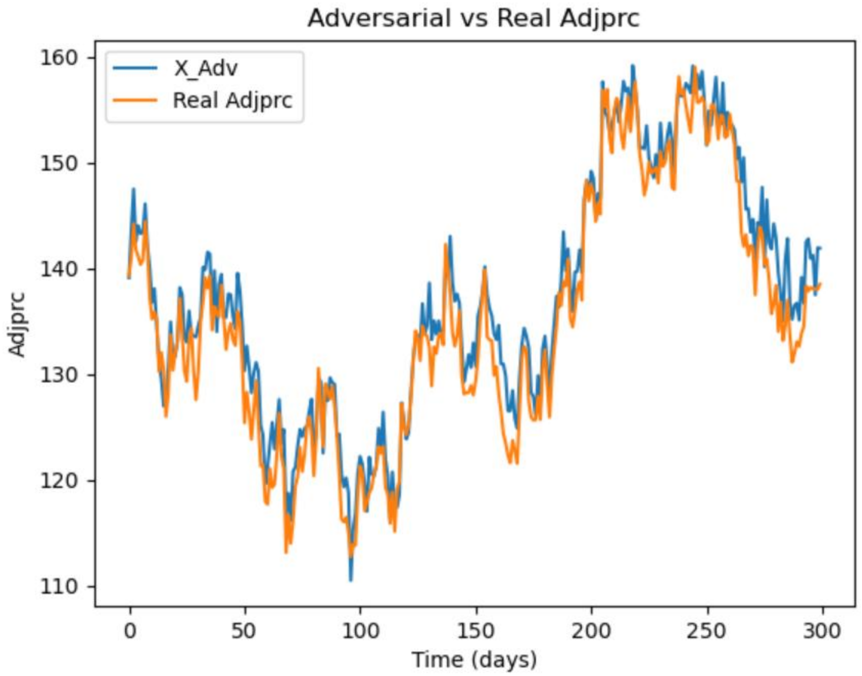
Its Definitely “Stealthy”!

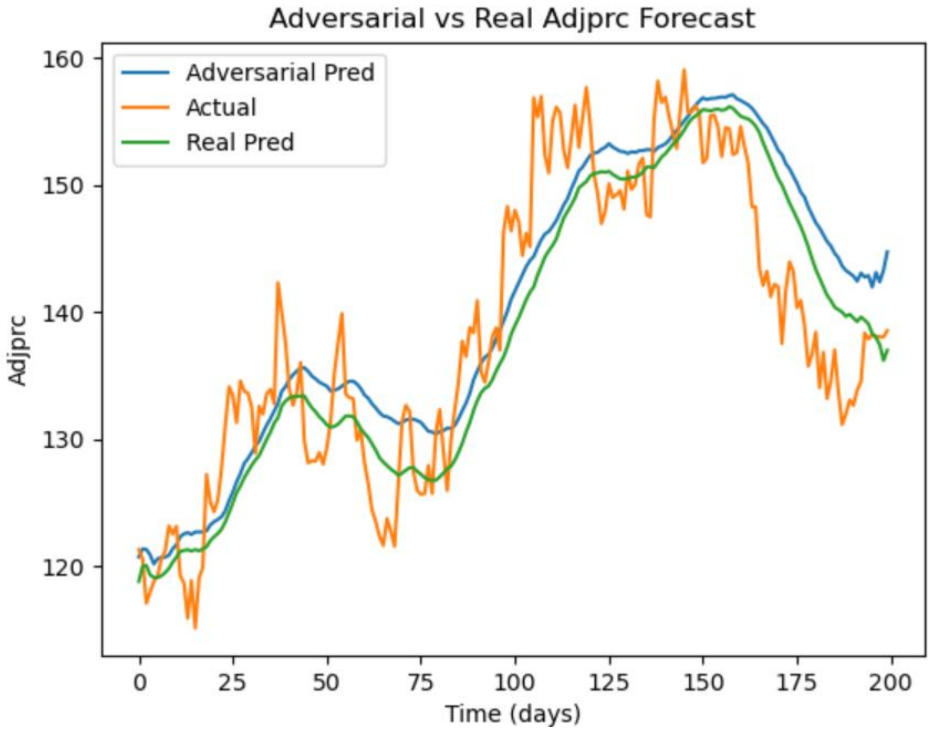





```
=====================================================
Adv Slope: tensor([0.1495], grad_fn=<DivBackward0>)
Real Slope: tensor([0.1463], grad_fn=<DivBackward0>)
2-norm Loss: 3.9383656939436045
=====================================================
```

Its Definitely “Stealthy”!






```
===============================================================
Adv Slope: tensor([0.1666], grad_fn=<DivBackward0>)
Real Slope: tensor([0.1609], grad_fn=<DivBackward0>)
2-norm Loss: 3.745380896057849
===============================================================
```

Its Not Great…But There Is Some Promise

1. In both the log return and adjprc case, the GAN is a bit afraid to deviate from the condition, and increasing the scale of the adversarial loss would cause the model to completely ignore the condition.
   a) Remove the condition, and just try to learn a distribution of sequences that mess up the model
2. Rather than generating the adjprc, we could generate do something similar to the C&W attack and just generate the perturbation.
3. Gradients for the adversarial loss are still extremely small, mainly due to the long prediction length
   a) Reduce the generated sequence
   b) Compute loss over subwindows

“Why Should We Care?”

Malware

- Eventually, people will be downloading ML/AI models like any other apps, libraries, frameworks etc. (HuggingFace).
- Every model will have some sort *predict* function or script to use the model.
- However, if a malware has access to the script, then they could theoretically inject the adversarial attack before the script calls *model(x).*
- The malware could either be attached to the model (in its library), or can do a high-level search of your computer looking for any .pt or pickled files.
- In this scenario, black vs white box attacks no longer matter.

Sample Malware

- I have made a general script to run the model given a csv file with the adjprc for some ticker.
- This week I will create the simple malware (just injecting the adversarial attack before doing model(x))

Malware Pseudocode (will be attached to model library)

- Can be disguised as __init__ file which is automatically called when imports are made.
- The file will simply read the *predict* file/script, find the location of the *model(x)* (if PyTorch is used, then we can easily search for some *with torch.no_grad()* call).
- Inject the adversarial attack on x, then proceed with the prediction.

Adversarial Attack Papers

- If domain specific like time-series, they typically just introduce their attack methods with some sort of narrative.
  - For example, “Most adversarial attacks on time series typically derive from image-based attacks, however these do not consider the patterns and temporal characteristics of time series data” [1]. (2025 paper)
  - Another example is the Pialla et al. paper where they discuss how adapting image-based adversarial attacks to time series is not trivial as they could be easily detected by human-eye (then they propose their own attacks) [2]. (2025 paper)
    - Pialla et al, also showed the effectiveness of adversarial training [2].
- Hu and Pang (2025) went heavy into the defences in addition to their GAN-based attack (on images).
  - Here they used unconditional GAN.

Next Steps

- Try to improve the GAN with the 3 solutions discussed before.
- Finish the malware.
- Implement basic adversarial training (time constraint).
- Implement C&W versions of the slope attacks.
- Start to write final report:
  - Narrative would be similar to [1] with some extra steps:
    - Discuss attacks on time series and where the pitfalls are with implementing image-based adversarial attacks on time-series.
    - Discuss how these “new” attack methods are typically performed on simple models like 3-layered CNNs, and how N-HiTS would be a better baseline since it is more production-ready.
    - Discuss new slope-based attacks.
    - Discuss the adversarial GAN (hopefully it works)
    - Discuss how easy it would be to inject malware, and how adversarial attacks need to be taken more seriously.
    - Effects of adversarial training.

References

- [1] Z. Shen and Y. Li, “Temporal characteristics-based adversarial attacks on time series forecasting,” *Expert systems with applications*, vol. 264, Art. no. 125950, 2025, doi: 10.1016/j.eswa.2024.125950.
- [2] G. Pialla *et al.*, “Time series adversarial attacks: an investigation of smooth perturbations and defense approaches,” *International journal of data science and analytics*, vol. 19, no. 1, pp. 129–139, 2025, doi: 10.1007/s41060-023-00438-0.
- [3] H. Hu and J. Pang, “Fooling machine learning models: a novel out-of-distribution attack through generative adversarial networks: Fooling machine learning models: a novel out-of-distribution attack,” *Applied intelligence (Dordrecht, Netherlands)*, vol. 55, no. 5, 2025, doi: 10.1007/s10489-024-05974-1.