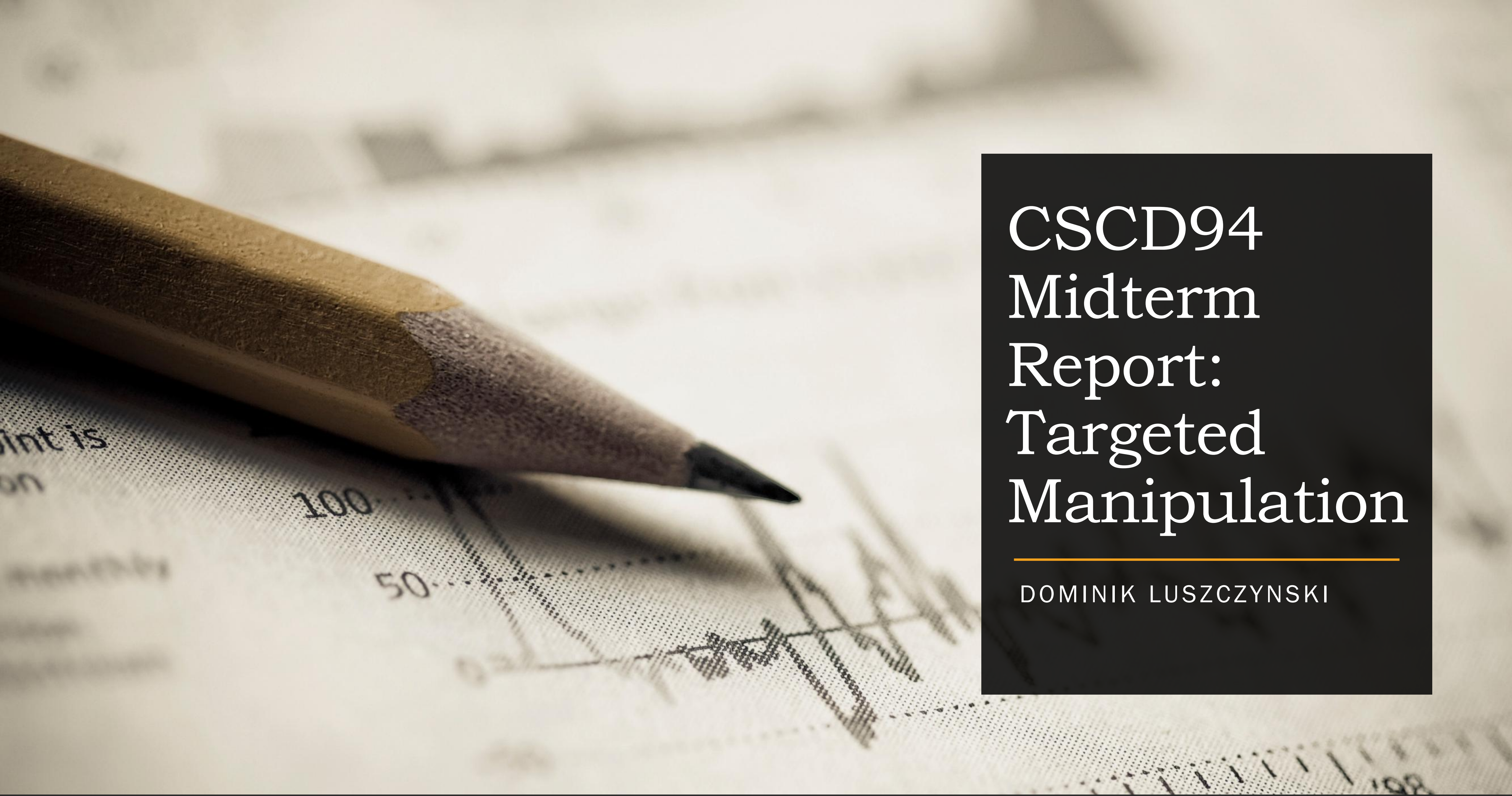


CSCD94 Midterm Report: Targeted Manipulation

DOMINIK LUSZCZYNSKI

Adversarial Attacks

- Adversarial attacks occur when an attack slightly modifies the input to a model, typically by adding noise, which causes a model to produce an incorrect result [2].

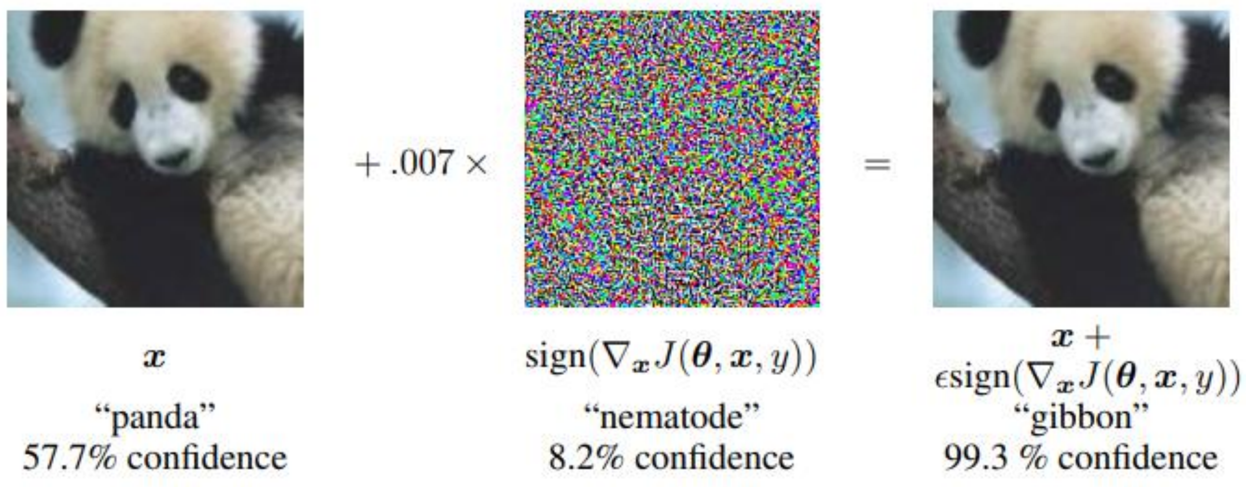

$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Image Source: J. Shlens, I. J. Goodfellow and C. Szegedy [2]

White Box vs Black Box

- White box attack occurs when the attacker has full access to all information about the model, including parameters, which enables the attacker to exploit gradient information [3].
- Black-box attack occurs when the model is hidden from the attacker, and they do not have any knowledge about the structure or parameters [3], [4].

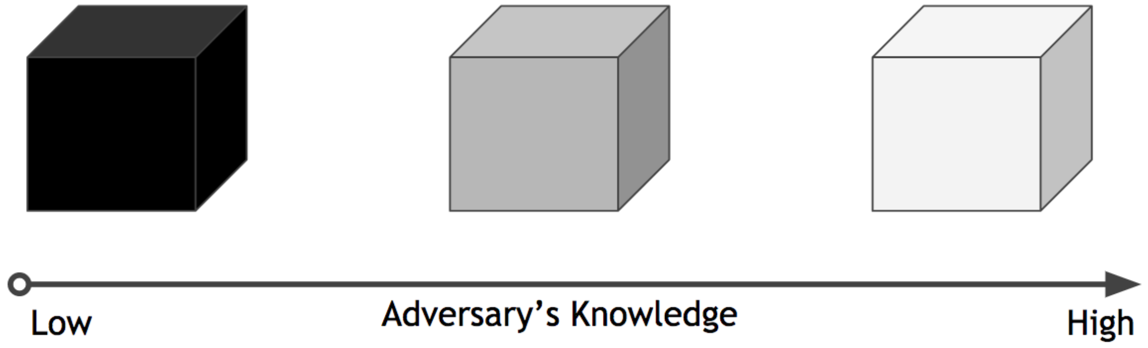


Image Source: https://secml.github.io/class1/

The Basis of an Attack

- When gradient descent is performed, the weights are moved in the opposite direction of the gradient.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda \nabla E(\mathbf{w}_t)$$

Change the – into a +

Fast Gradient Sign Method (FGSM)

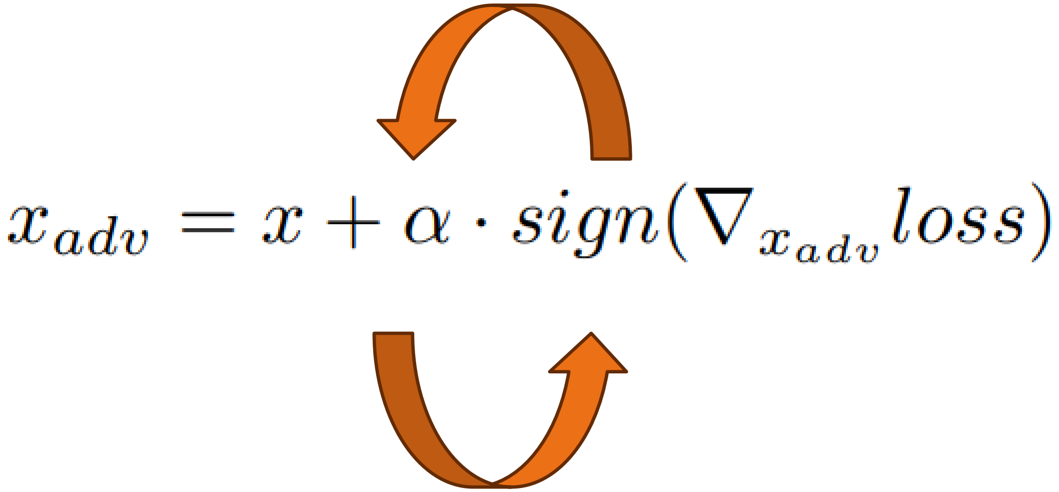
- The FGSM involves computing the loss with respect the predictions, then adjusting the adversarial attack x_{adv} in the direction that maximizes the loss [17].

- Here x_{adv} is our adversarial input, x is the original input, and ε is our maximum perturbation size.

$$x_{adv} = x + \epsilon \cdot sign(\nabla_{x_{adv}} loss) \quad [17]$$

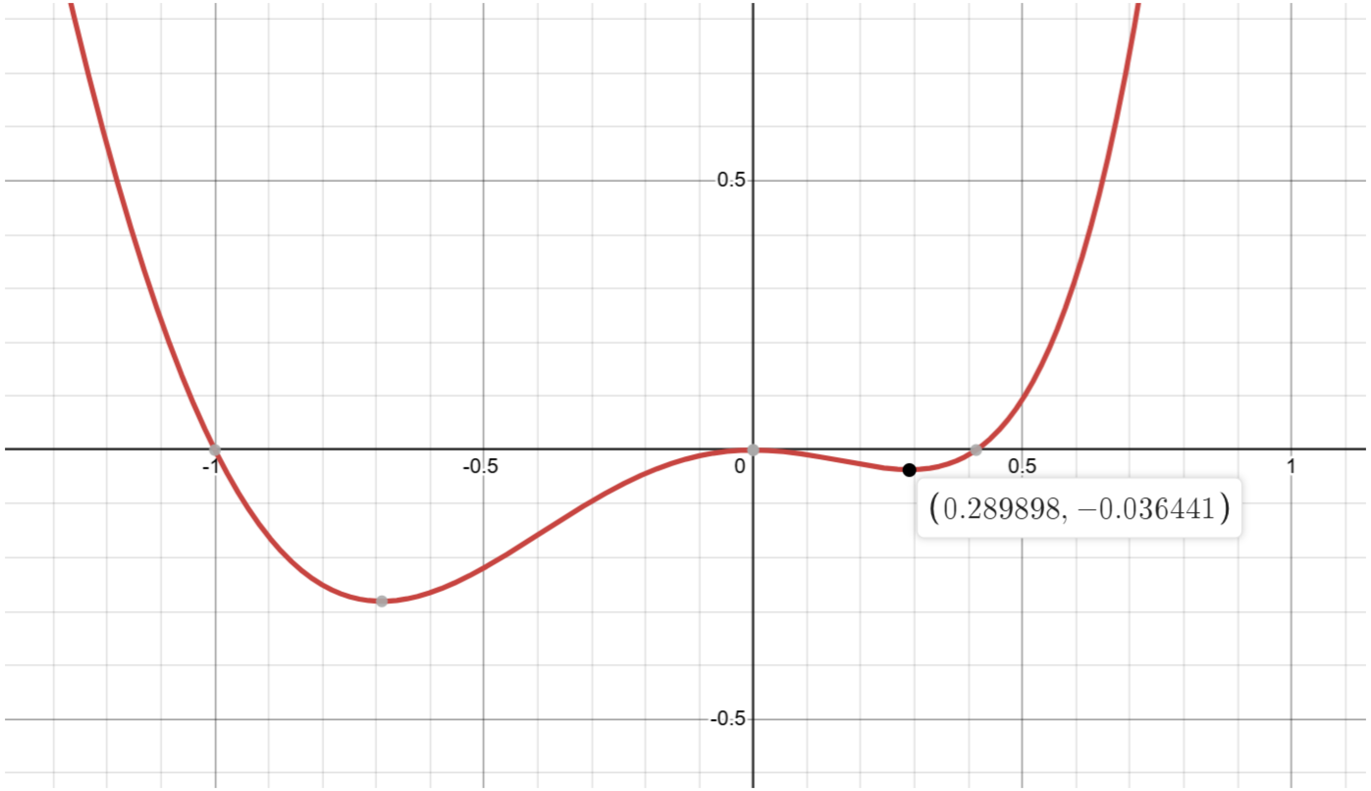
Basic Iterative Method (BIM)

- Rather than performing the attack in a single step, update x_{adv} iteratively, for some number of desired iterations, with some step size α.
- However, to ensure that we stay within the ε-neighbourhood we need to clip x_{adv} to x + ε and x – ε.



$$x_{adv} = x + \alpha \cdot sign(\nabla_{x_{adv}} loss)$$

The Dreaded Local Minima



😠

Momentum Iterative FGSM (MI-FGSM)

- Many optimization algorithms may struggle with getting stuck in a local minima.
- Similar to the momentum technique used to accelerate gradient descent algorithms, the MI-FGSM uses the same technique to help avoid getting stuck [1].
- Initialize $g_0 = 0$, then

$$g_{t+1} = \mu \cdot g_t \cdot \frac{\nabla_{x_{adv}} loss}{||\nabla_{x_{adv}} loss||_1}$$

and we the update x_{adv} in the direction of g_{t+1}

$$x_{adv} = x_{adv} + \alpha \cdot sign(g_{t+1})$$ [1]

The Basis of an Attack

- When gradient descent is performed, the weights are moved in the opposite direction of the gradient.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda \nabla E(\mathbf{w}_t)$$

Targeted Iterative Method (TIM)

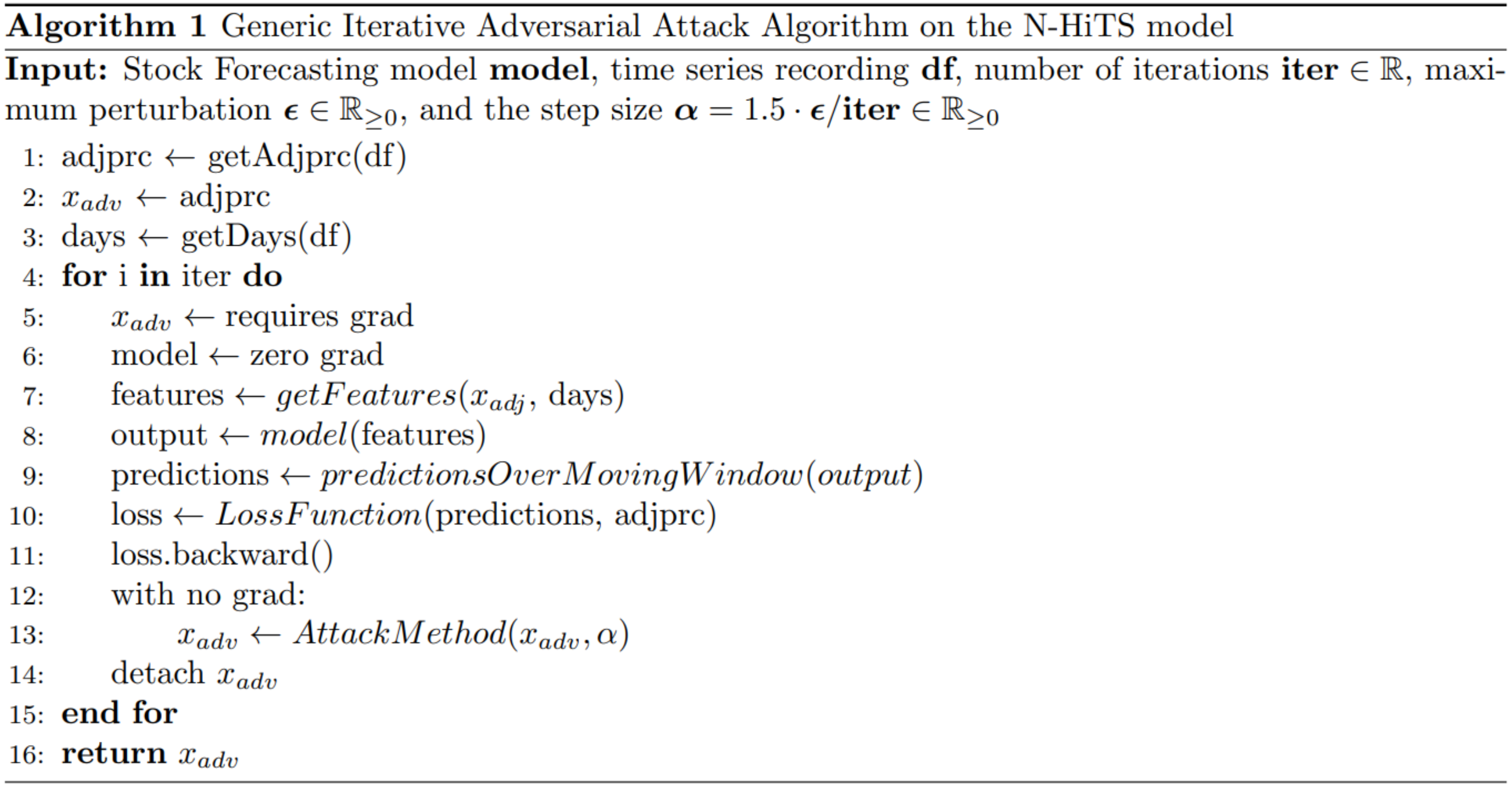
- Although the goal for many adversarial attacks is to maximize the error of the model, in realistic scenarios the attacker would want to choose the direction in which the model fails.
- The attack requires a parameter d and γ which represent the direction and margin, and calculates the new target as:

$$tar = \mathrm{adjprc} + d \cdot \gamma \qquad [3]$$

then we the update x_{adv} as:

$$x_{adv} = x_{adv} - \alpha \cdot sign(\nabla_{x_{adv}} loss) \qquad [3]$$

How Do We Make an Adversarial Attack?



Algorithm 1 Generic Iterative Adversarial Attack Algorithm on the N-HiTS model

Input: Stock Forecasting model **model**, time series recording **df**, number of iterations **iter** $\in \mathbb{R}$, maximum perturbation $\boldsymbol{\epsilon} \in \mathbb{R}_{\geq 0}$, and the step size $\boldsymbol{\alpha} = 1.5 \cdot \boldsymbol{\epsilon}/\textbf{iter} \in \mathbb{R}_{\geq 0}$

```
adjprc ← getAdjprc(df)
$x_{adv}$ ← adjprc
days ← getDays(df)
for i in iter do
    $x_{adv}$ ← requires grad
    model ← zero grad
    features ← $getFeatures(x_{adj}$, days)
    output ← $model$(features)
    predictions ← $predictionsOverMovingWindow(output)$
    loss ← $LossFunction$(predictions, adjprc)
    loss.backward()
    with no grad:
        $x_{adv} \leftarrow AttackMethod(x_{adv}, \alpha)$
    detach $x_{adv}$
end for
return $x_{adv}$
```

C&W Attack

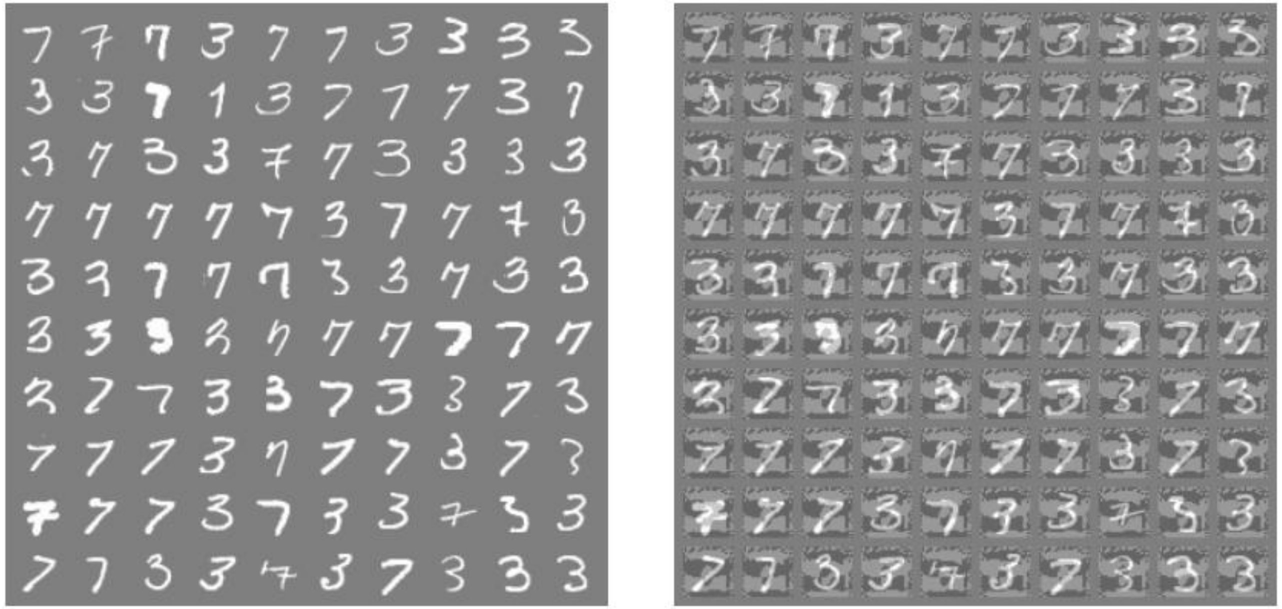
- The Carlini and Wagner attack (C&W) deviates from the general approach to adversarial attacks. Rather than iteratively modifying x_{adv}, a C&W attack aims to find the optimum noise vector η that is added to the adjprc [8].

- The goal is to solve the following optimization problem:

$$\min ||\eta||_2 + f(x_{adv} + \eta)$$ [8]

PROBLEM

The majority of existing adversarial research has been done on image and text classification, while attacks on time-series data is still in its infancy [3].



(c)

(d)

Image Source: J. Shlens, I. J. Goodfellow and C. Szegedy [2]

Current Research

- There have been recent advancements in the time series domain, with Gallager et al., attacking a simple three layered Convolutional Neural Network (CNN) forecasting the Google Stock from 2006 to 2018 with Fast Gradient Sign Method [5].
- Rathore et al., applied the Fast Gradient Sign Method and the Basic Iterative Method on a classification model trained on 54 different time series datasets related to healthcare, vehicle sensors and electrical equipment [6].

PROBLEM

The introductory research in the time series domain typically focus on classification problems, reapplying image/text based adversarial attacks on time series data, and do not consider the believability of an adversarial example [3], [7].

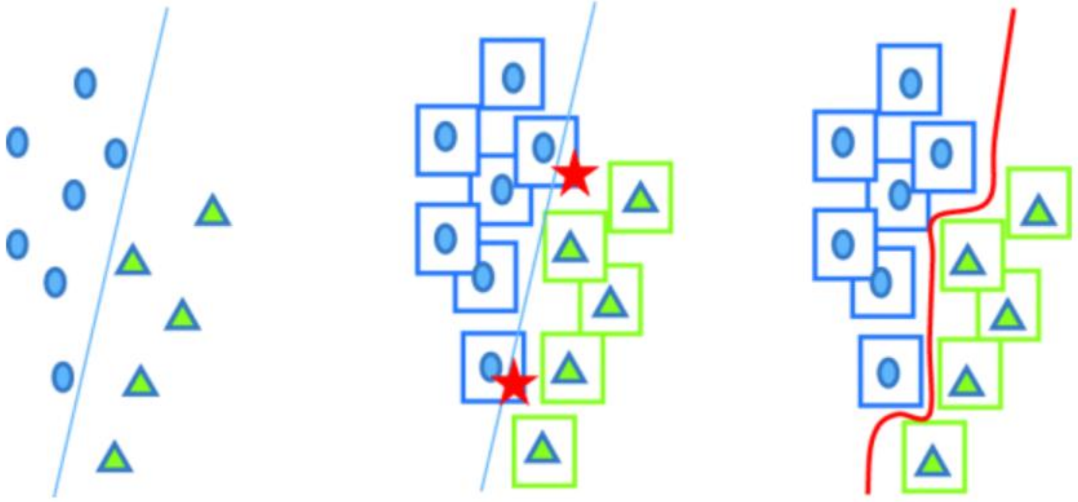


Image Source P. Rathore et al [6]

Stealthy Iterative Method (SIM)

- Shen and Li introduced Stealthy attacks which rely on using cosine similarities to ensure temporal characteristics of time series remain intact [3].
- After performing an iteration of the standard BIM, compare the cosine similarity between x and the new x_{adv} vs x with $x + \varepsilon$ and finally with x and $x – \varepsilon$:
  - If the cosine_similarity(x, x_{adv}) < cosine_similarity(x, $x + \varepsilon$) then $x_{adv} = x + \varepsilon$
  - If the cosine_similarity(x, x_{adv}) < cosine_similarity(x, $x - \varepsilon$) then $x_{adv} = x – \varepsilon$
  - Otherwise, x_{adv} is kept the same [3].

PROBLEM

A lot of the research described in the previous slides were performed on shallow neural networks.

- SIM (Shen and Li):
  - 3-layered CNN with hidden dimension of 60 and a fully connected output layer [3].
  - 3-layered LSTM with hidden dimension of 100 and a full connected output layer [3].
  - 3-layered GRU with hidden dimension of 100 and a full connected output layer [3].

- Gallager et al.:
  - 3-layered CNN [5].

Let’s Fool a More Complex Model

N-HiTS

- N-HiTS is a novel projection model which builds upon the N-BEATS architecture, simultaneously improving computational performance and accuracy by sampling the time series at different rates [14].
- The N-HiTS architecture uses Multi-layer Perceptrons (MLPs) for each block of the time series to estimate coefficients for the backcasts and forecasts [14].
- However, the N-HiTS model introduces novel components to each block, like using MaxPool with some kernel size k_b for each block b, claiming that it helps the MLP focus on low frequency and large-scale contents of the time series, which allows it to sample different rates [14].
- After the pooling layer, a non-linear regression is applied to estimate the backcasts and forecasts, and to eventually generate the final predictions, hierarchal interpolation is performed [14].

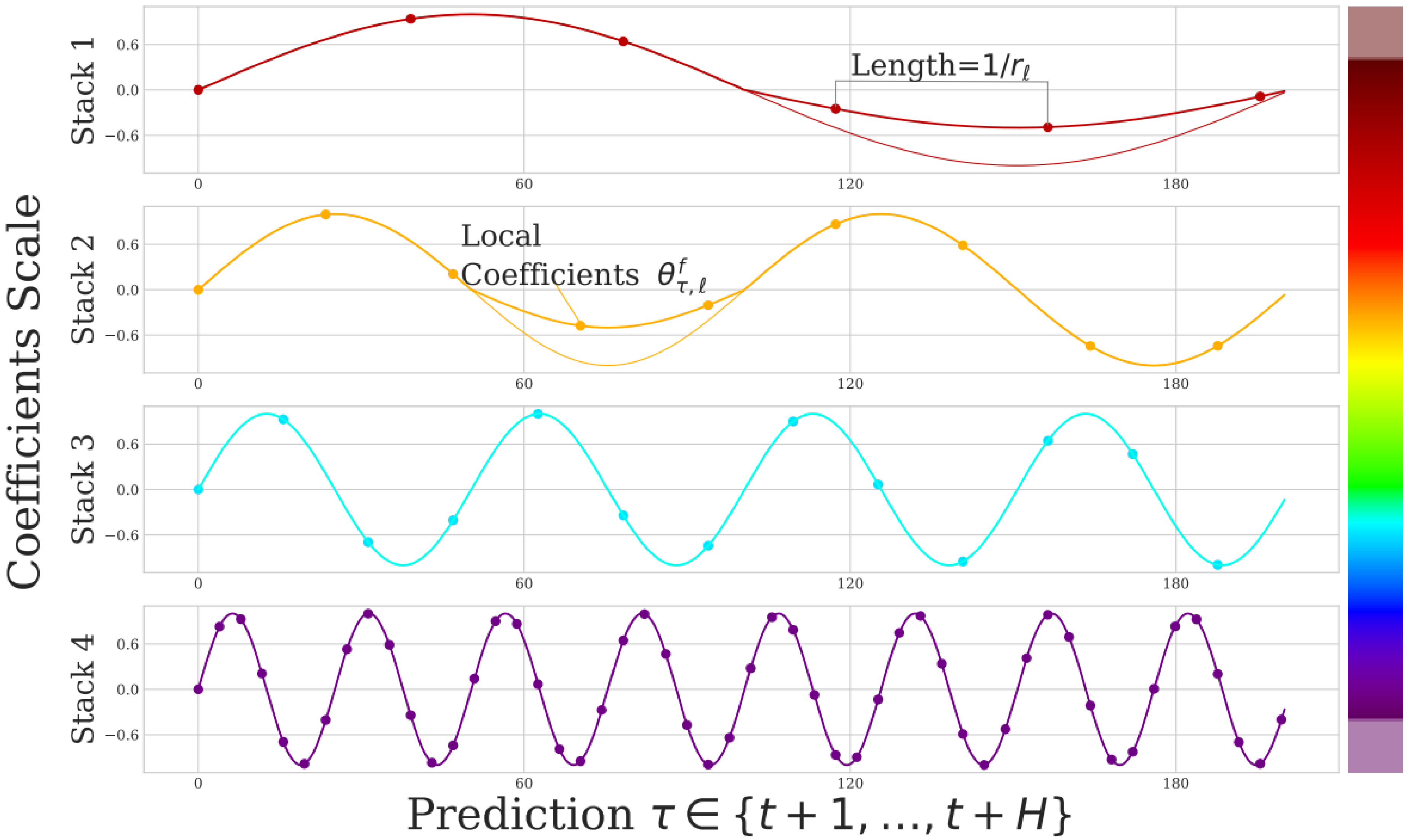


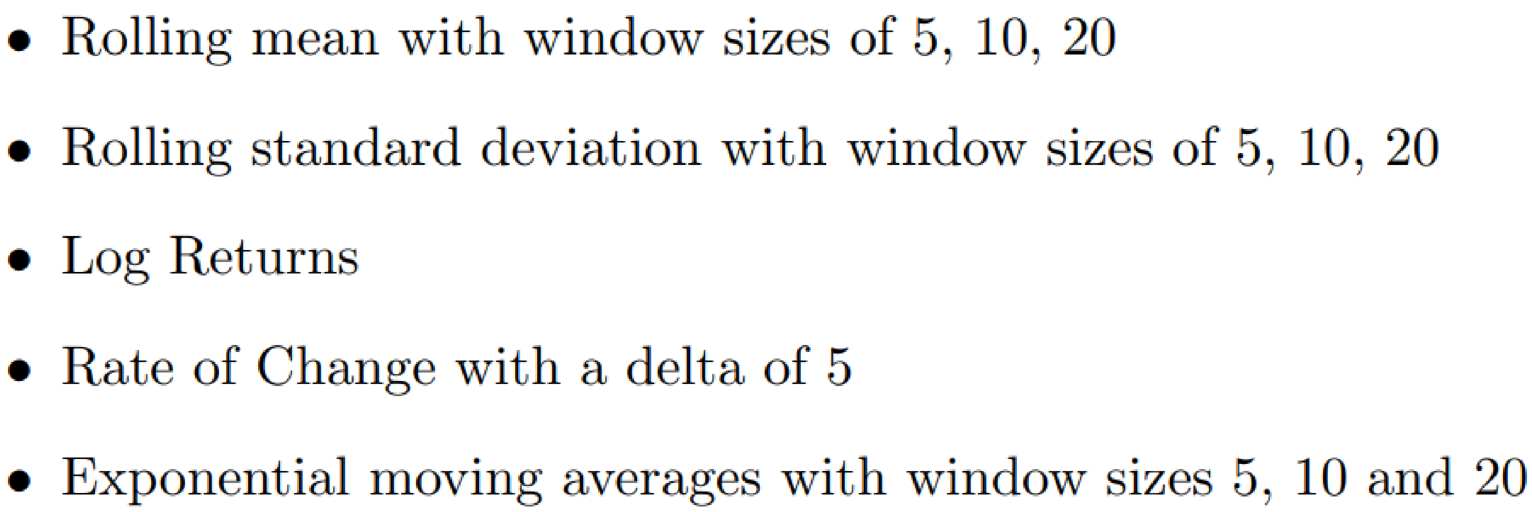
Image Source: C. Challu et al. [14]

Dataset

- The daily adjusted price (adjprc) of all S&P 500 stocks was collected from Center for Research Security Prices (CRSP) for the 2015:05:01- 2025:05:01 period.
- Rather than training on a single stock, to improve generalizability the N-HiTS model was trained on 360 different stocks from the S&P 500. The validation set included 48 stocks while the test set contained 72 stocks.
- The split was determined by a stratified split on the stock price collected from [15].

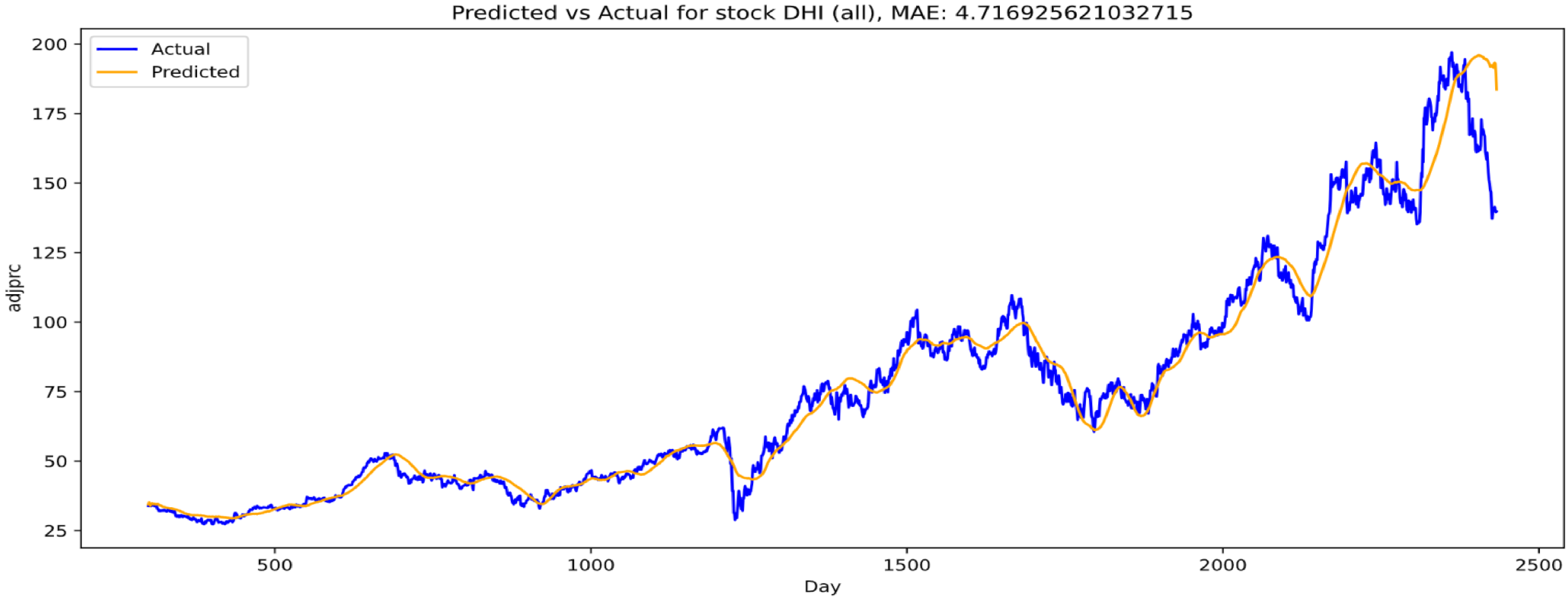
Dataset

- Several Features were derived from the adjprc such as:



- Rolling mean with window sizes of 5, 10, 20
- Rolling standard deviation with window sizes of 5, 10, 20
- Log Returns
- Rate of Change with a delta of 5
- Exponential moving averages with window sizes 5, 10 and 20

- In addition, the day of the week was extracted from the date, as it has been shown that there are differences in volatility between the beginning and end of the trading week [16].



N-HiTS Performance

Average MAE: **6.62**

Average RMSE: **9.43**

MAPE: **5.29%**

N-HiTS Hyperparameters

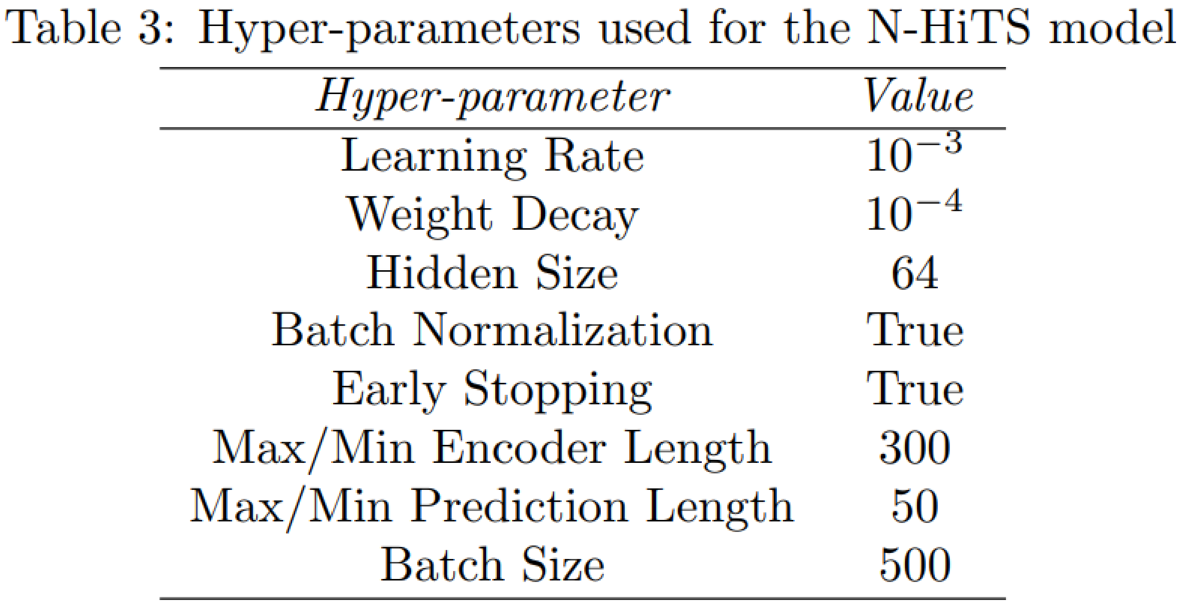


Table 3: Hyper-parameters used for the N-HiTS model

Hyper-parameter	*Value*
Learning Rate	10^{-3}
Weight Decay	10^{-4}
Hidden Size	64
Batch Normalization	True
Early Stopping	True
Max/Min Encoder Length	300
Max/Min Prediction Length	50
Batch Size	500

Adversarial Attacks

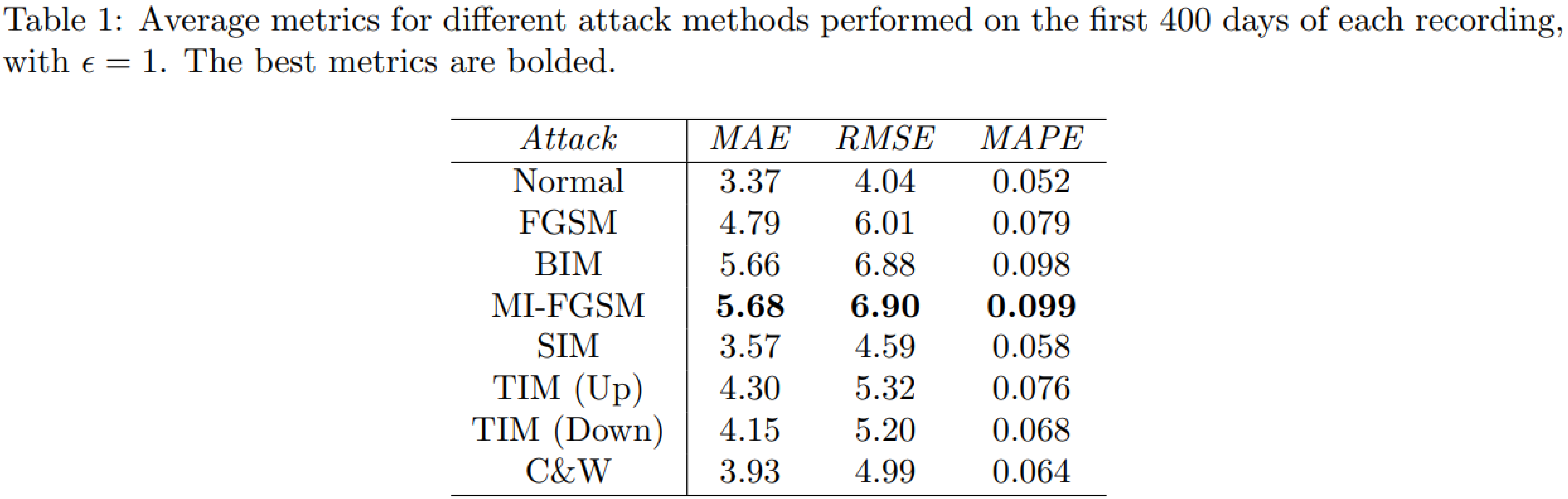


Table 1: Average metrics for different attack methods performed on the first 400 days of each recording, with $\epsilon = 1$. The best metrics are bolded.

Attack	*MAE*	*RMSE*	*MAPE*
Normal	3.37	4.04	0.052
FGSM	4.79	6.01	0.079
BIM	5.66	6.88	0.098
MI-FGSM	**5.68**	**6.90**	**0.099**
SIM	3.57	4.59	0.058
TIM (Up)	4.30	5.32	0.076
TIM (Down)	4.15	5.20	0.068
C&W	3.93	4.99	0.064

Attacks With Various Epsilon Sizes

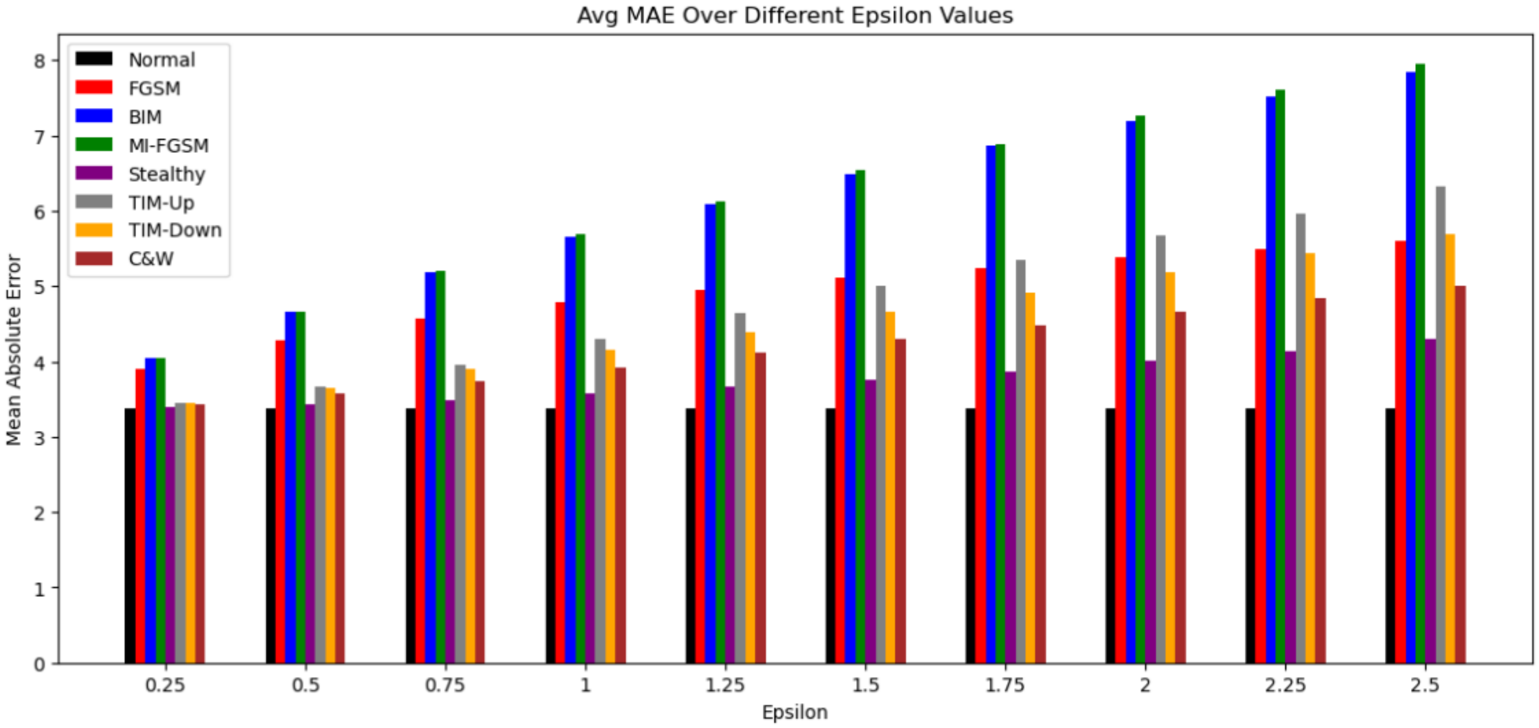


Figure 3: Average MAE for different attack methods performed on the first 400 days of each recording, with varying ϵ sizes.

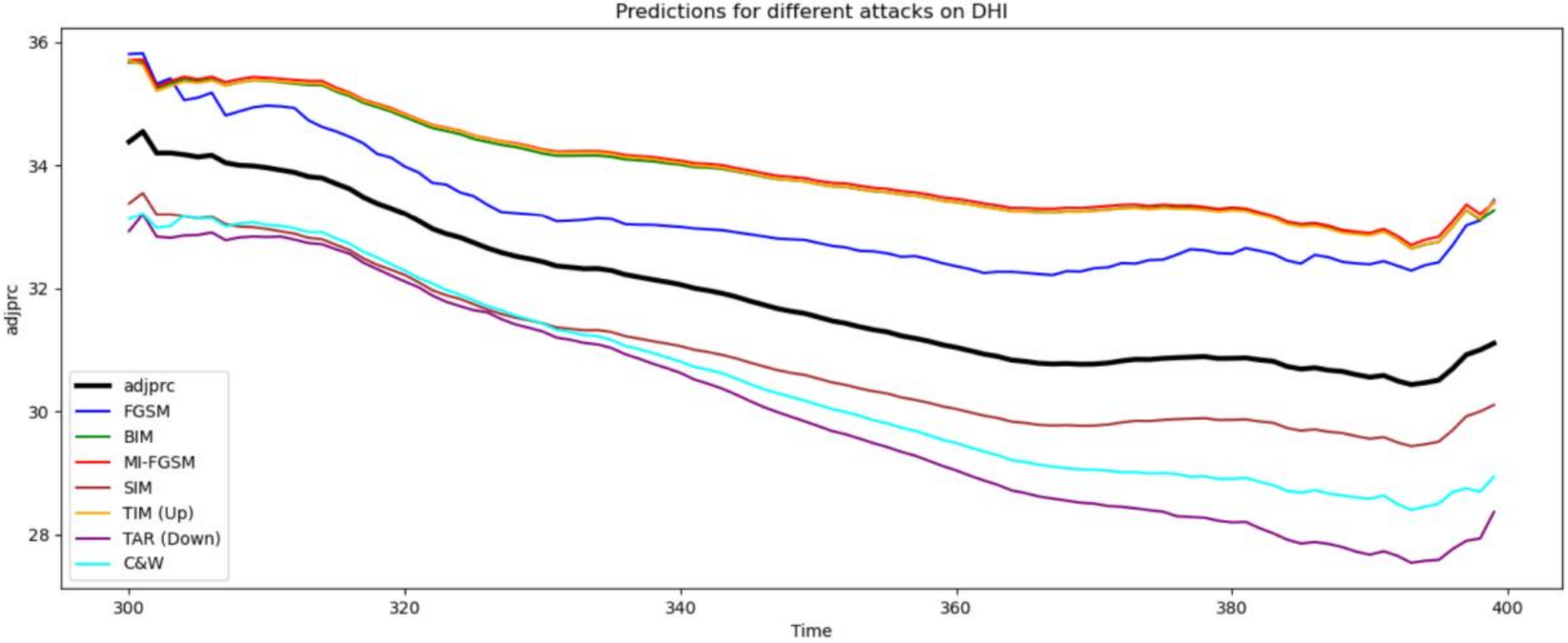


Figure 5: Various attacks performed on the stock *DHI*. Attacks were performed on the first 400 days of the recording.

PROBLEM

There is a large tradeoff between making an adversarial attack look believable and the amount of error we can inflict on the model.

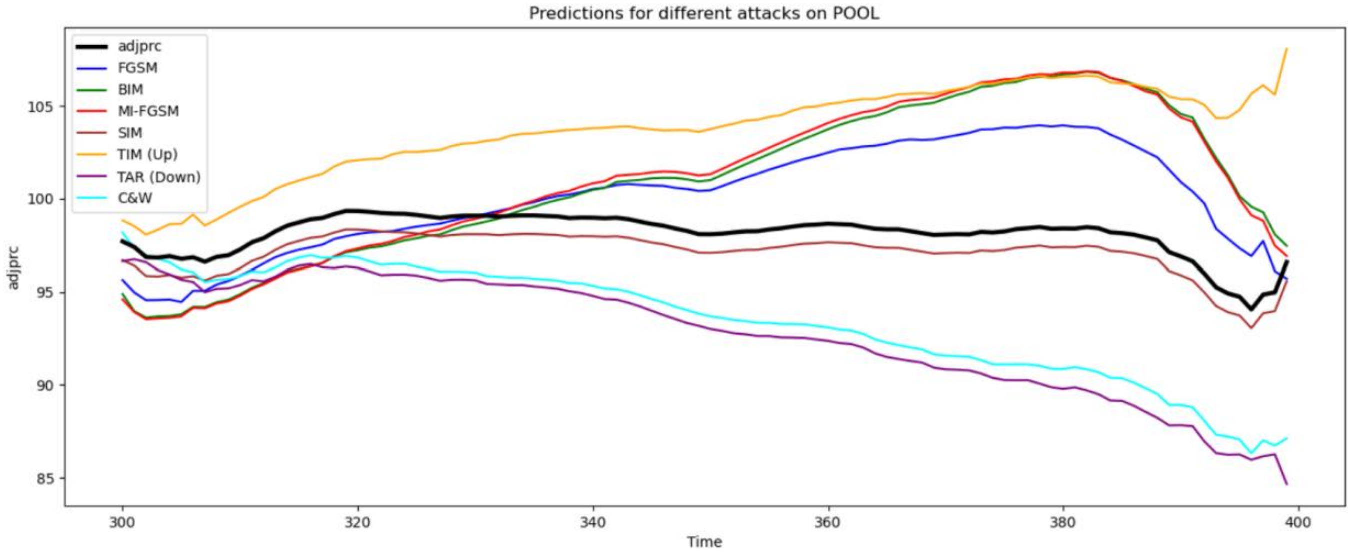
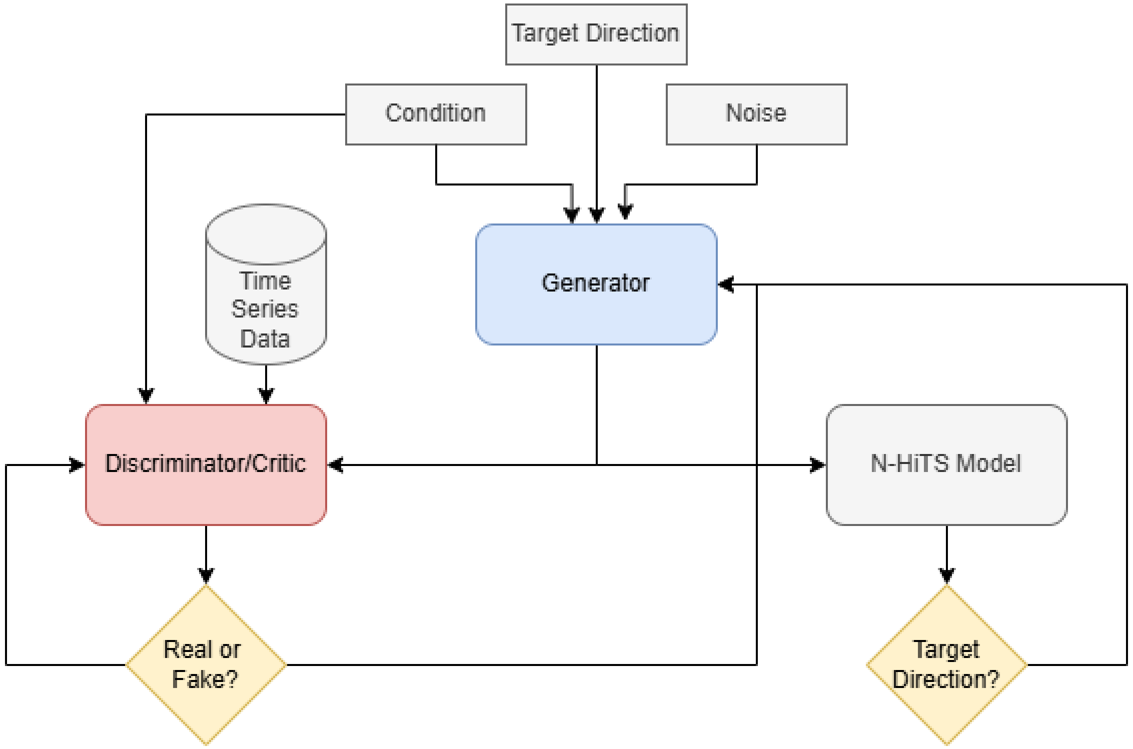


Figure 6: Various attacks performed on the stock *POOL*. Attacks were performed on the first 400 days of the recording.

Generative Adversarial Networks (GANs)

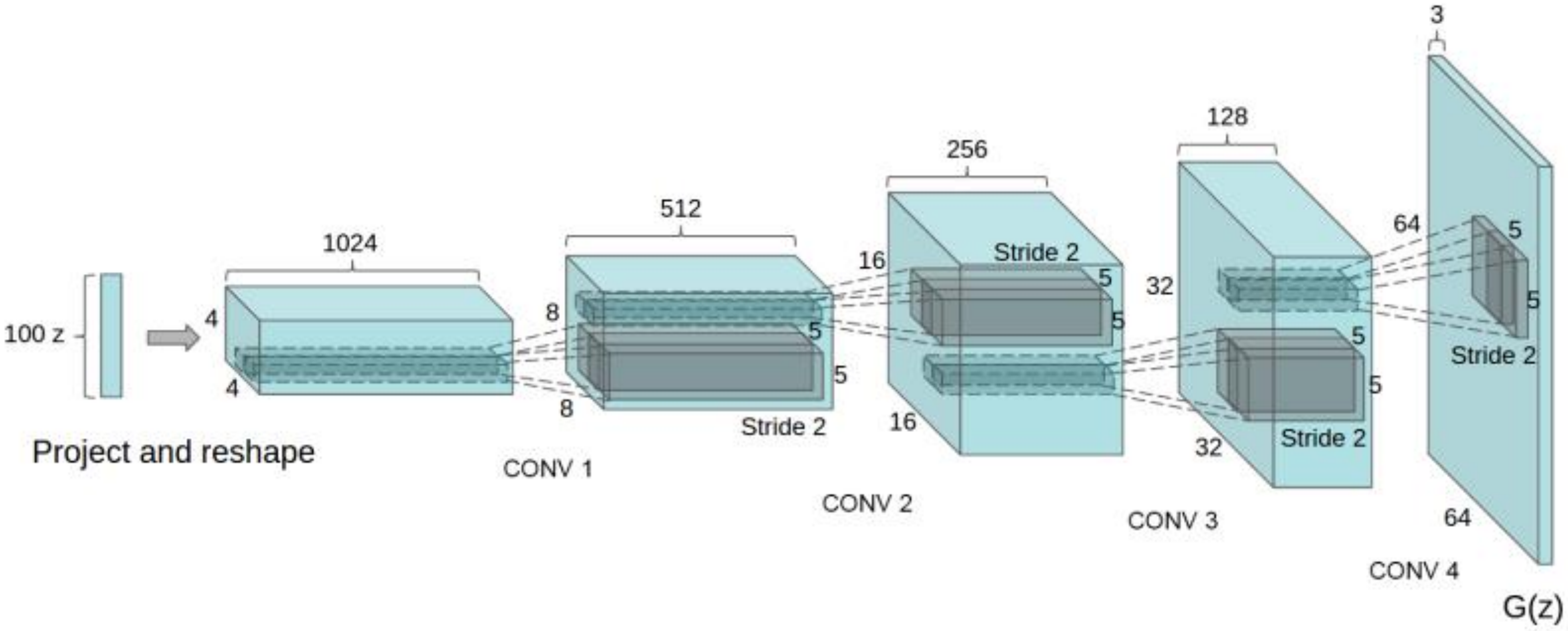
- Researchers have experimented with adding a second critic to the GAN architecture, which is the model that the attacker is aiming to fool [9].
- However, similar to adversarial attacks, the majority of research on GANs have been on images, and there have been even fewer studies using GANs to generate adversarial examples for time series forecasting [9], [11], [12].



This Research Aims to Make Both Powerful and Believable Adversarial Attacks on Time Series Data

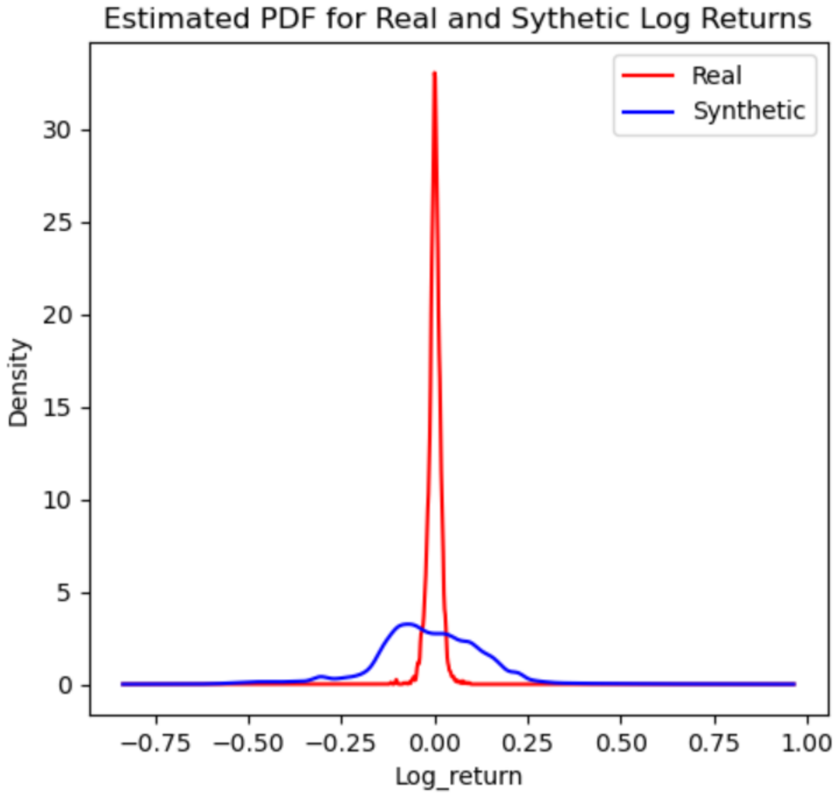
In order to become familiar with GANs, multiple variations were implemented

DCGAN

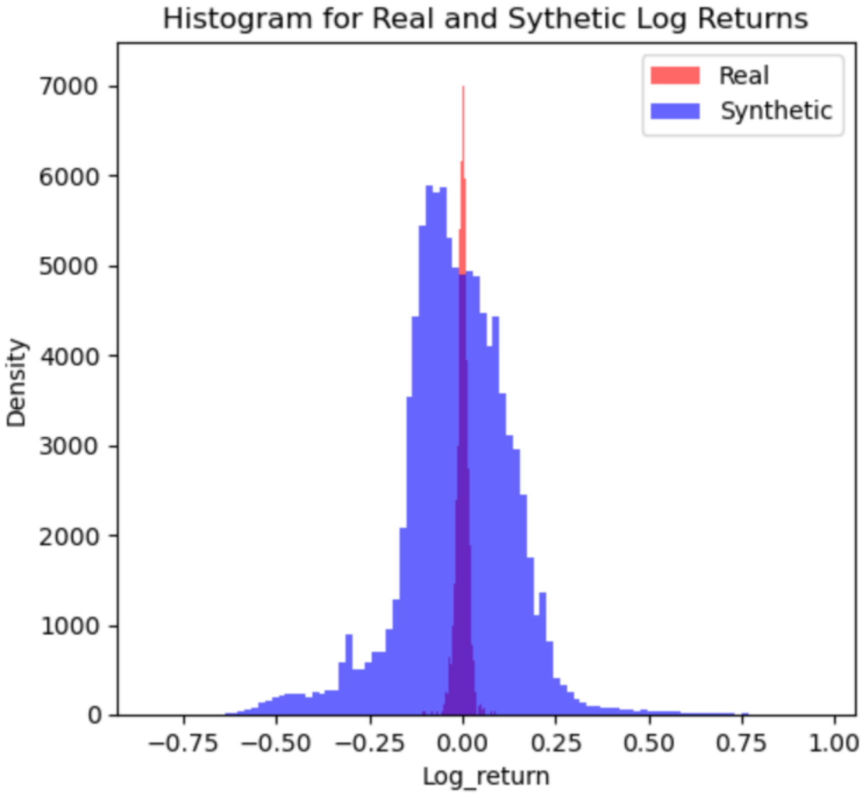


- The current implementation of the DCGAN, contains four blocks of strided transpose 1D convolutions and strided 1D convolutions for the generator and discriminator respectively.

DCGAN

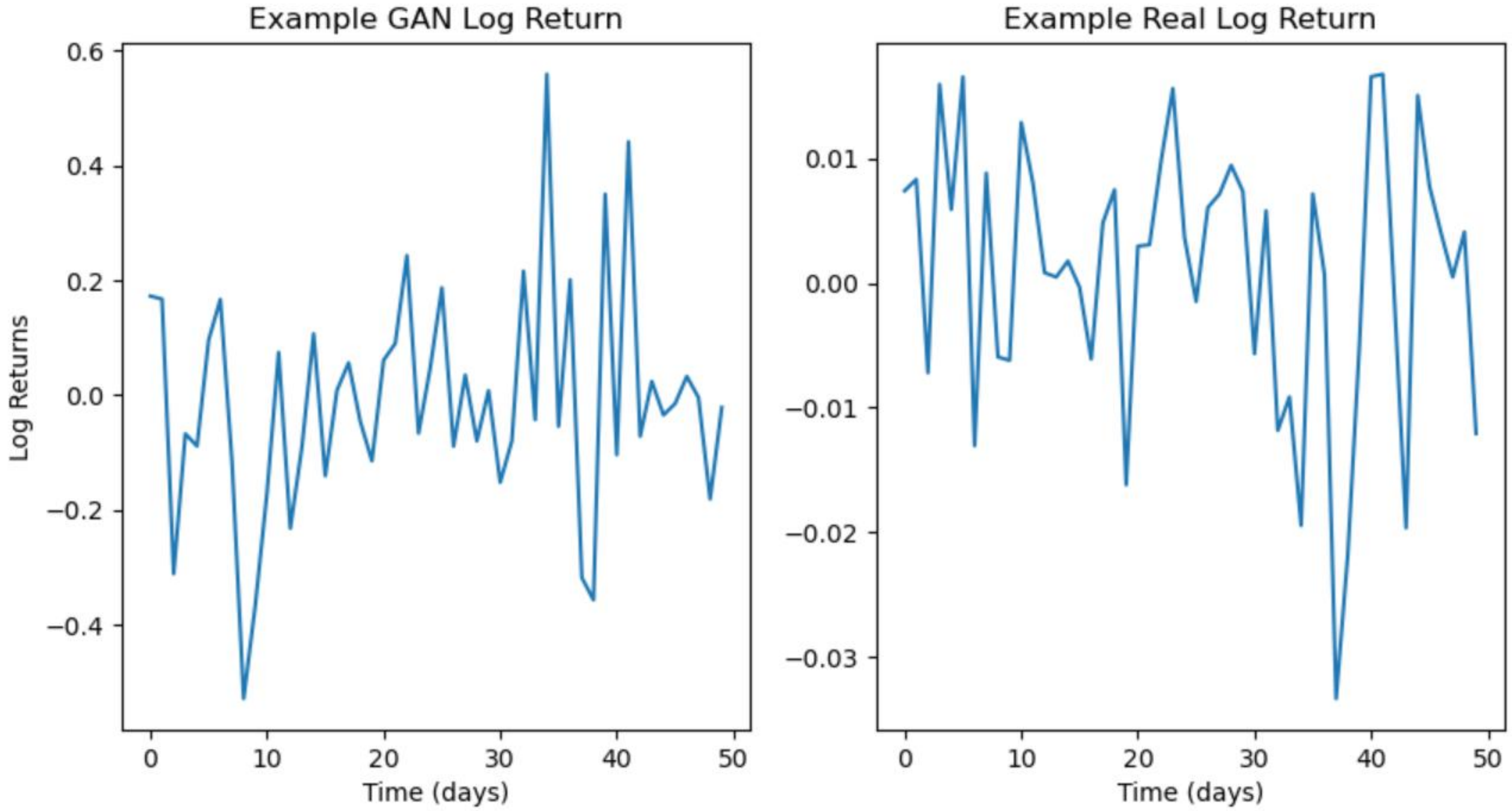


Estimated PDF for Real and Sythetic Log Returns



Histogram for Real and Sythetic Log Returns

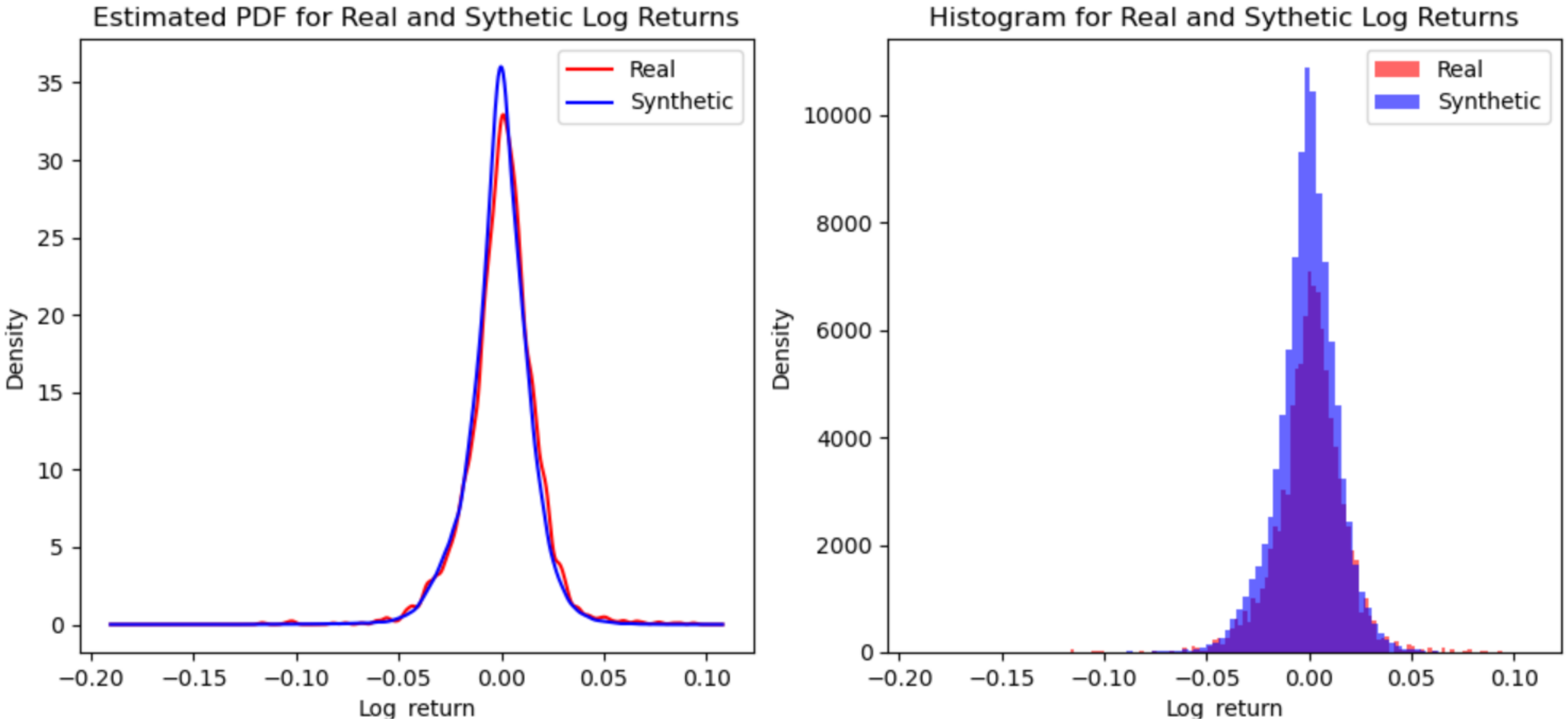
DCGAN


Example GAN Log Return
Example Real Log Return
Log Returns
0.6
0.4
0.2
0.0
−0.2
−0.4
0.01
0.00
−0.01
−0.02
−0.03
0
10
20
30
40
50
Time (days)
Time (days)

WGAN

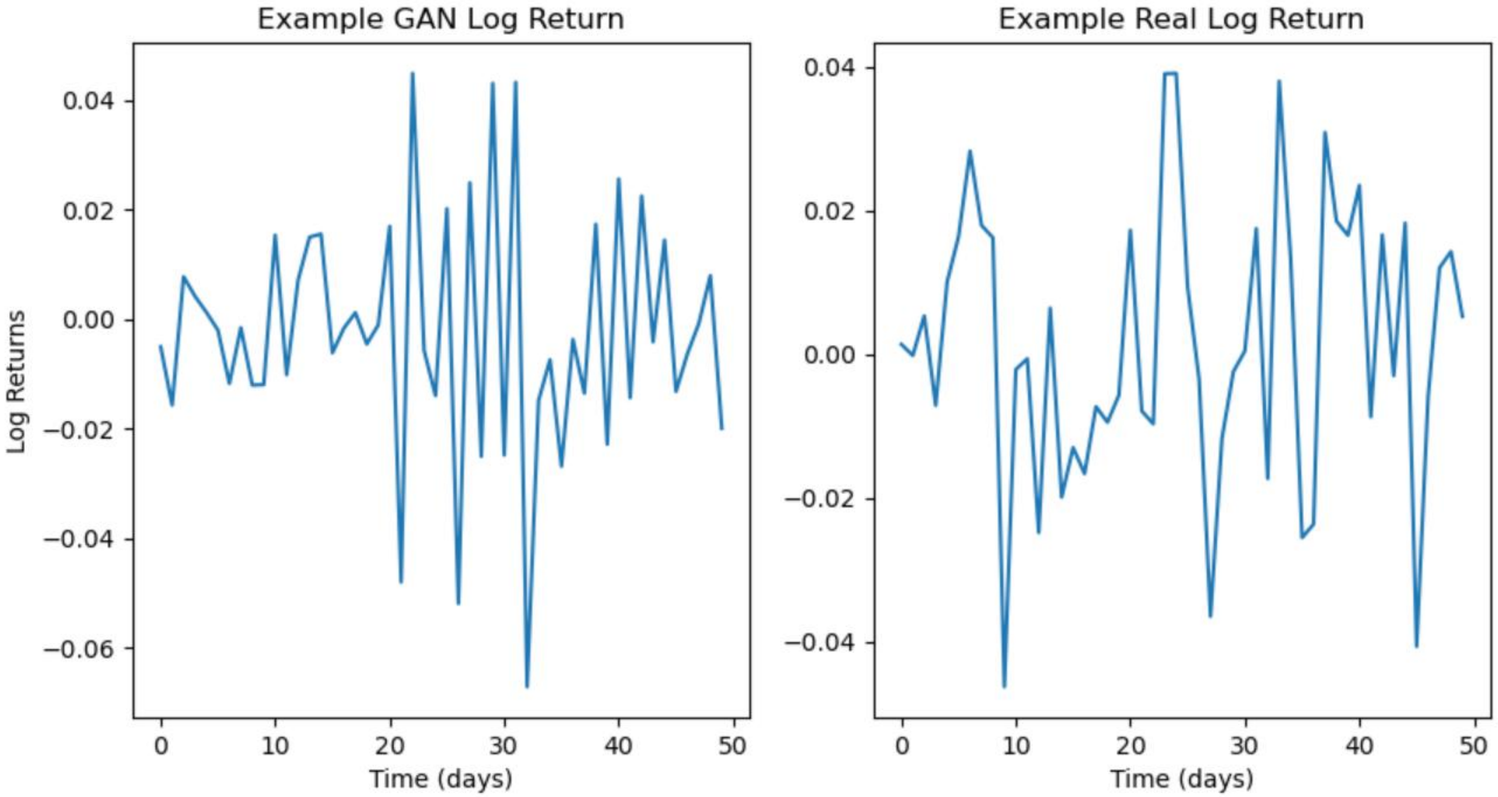
- Moving away from the DCGAN architecture, rather than increasing the dimensionality of a small noise vector, the noise vector was initialized to the same size as the desired synthetic data (50 days).
- Then, the noise vector was passed through generator with a 4-layered Temporal Convolutial Network (TCN) architecture.
- The critic (former discriminator) has a hybrid architecture made up of alternating TCN and Gated Recurrent Network (GRU) blocks.

WGAN


Estimated PDF for Real and Sythetic Log Returns

Histogram for Real and Sythetic Log Returns

WGAN

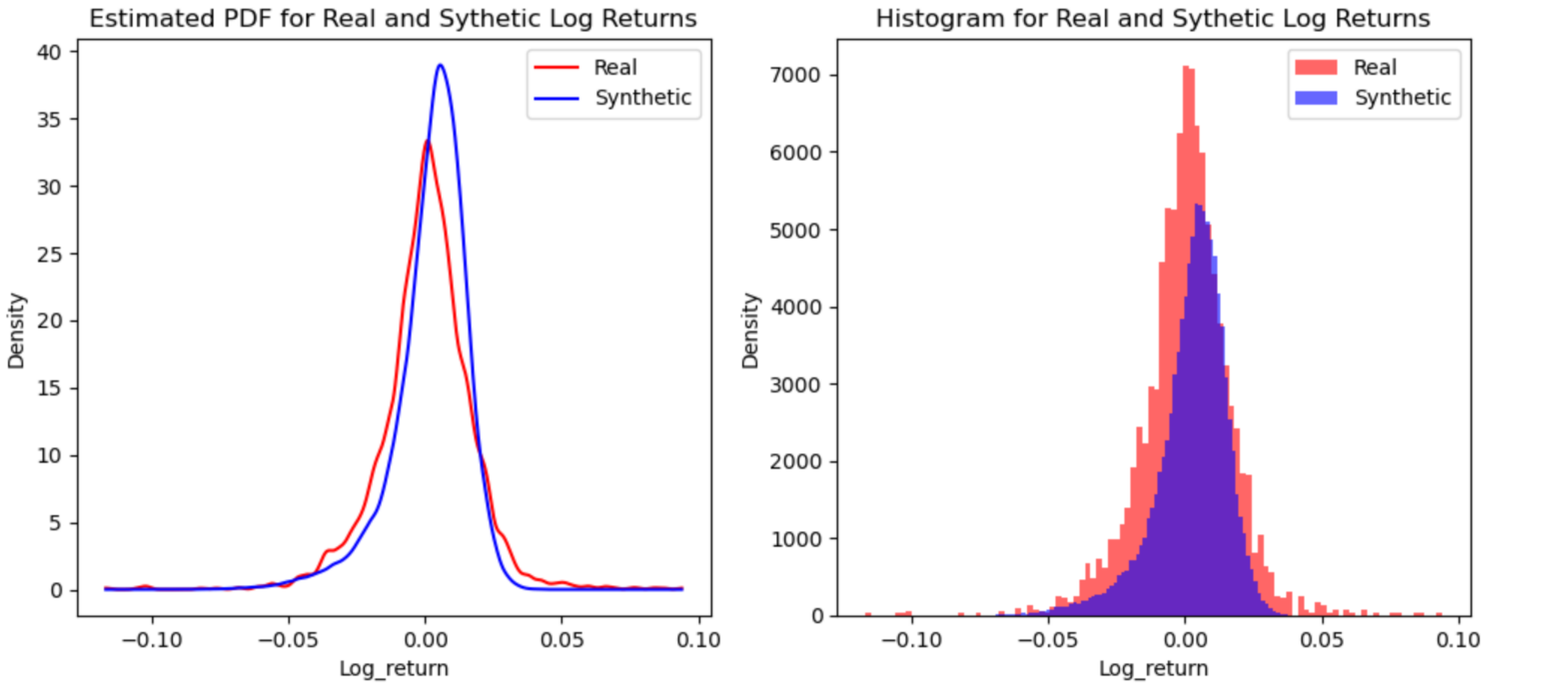


Example GAN Log Return
Log Returns
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
10
20
30
40
50
Time (days)
Example Real Log Return
0.04
0.02
0.00
−0.02
−0.04
0
10
20
30
40
50
Time (days)

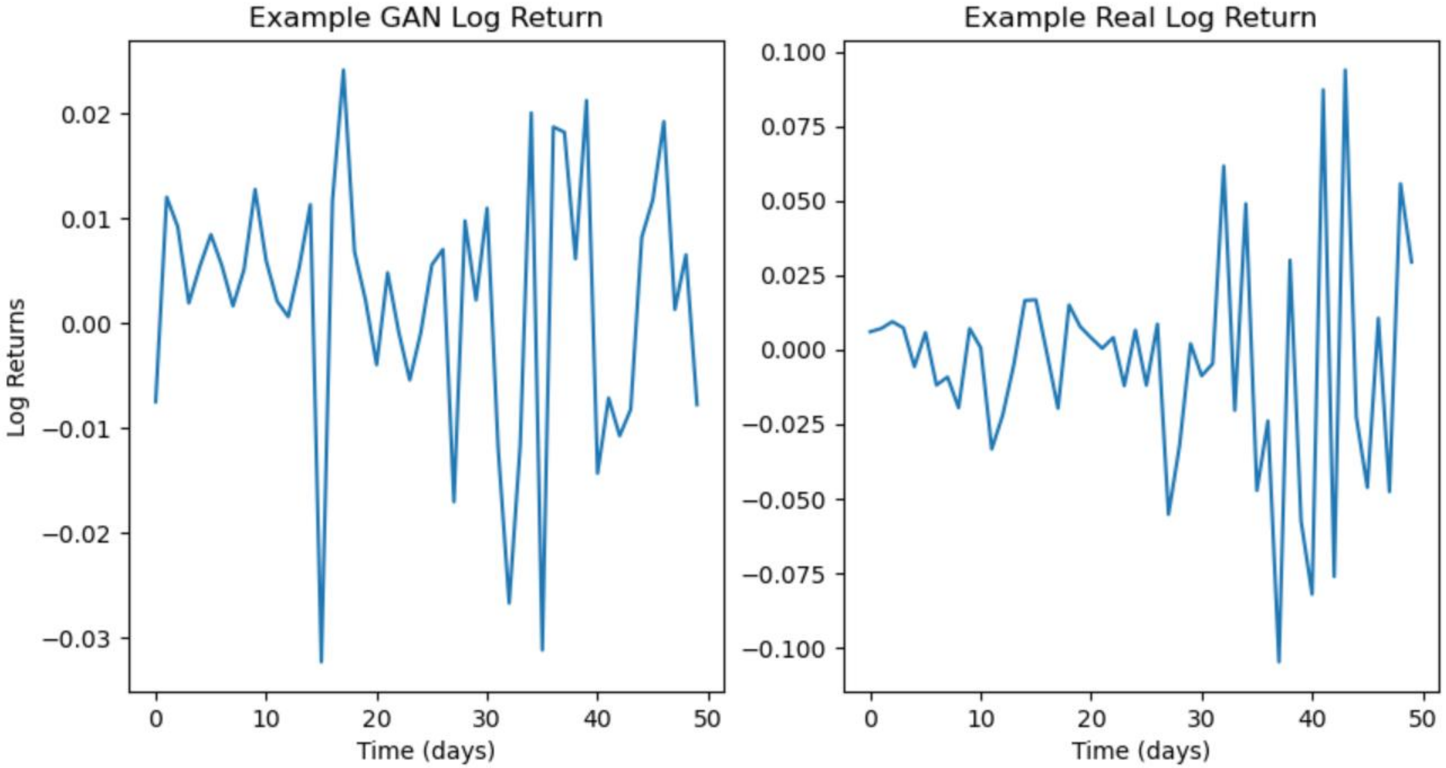
C-WGAN

- A further extension to the WGAN is to include a condition on the generated data [20].
- To condition the WGAN, the previous 50 days of log returns is concatenated with the noise vector in the feature dimension for both the generator and the critic.
- The same architecture as the WGAN was used.

C-WGAN


Estimated PDF for Real and Sythetic Log Returns
Real
Synthetic
Density
40
35
30
25
20
15
10
5
0
−0.10
−0.05
0.00
0.05
0.10
Log_return
Histogram for Real and Sythetic Log Returns
Real
Synthetic
Density
7000
6000
5000
4000
3000
2000
1000
0
−0.10
−0.05
0.00
0.05
0.10
Log_return

C-WGAN


Example GAN Log Return
Log Returns
0.02
0.01
0.00
−0.01
−0.02
−0.03
0
10
20
30
40
50
Time (days)
Example Real Log Return
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
0
10
20
30
40
50
Time (days)

GAN Comparison

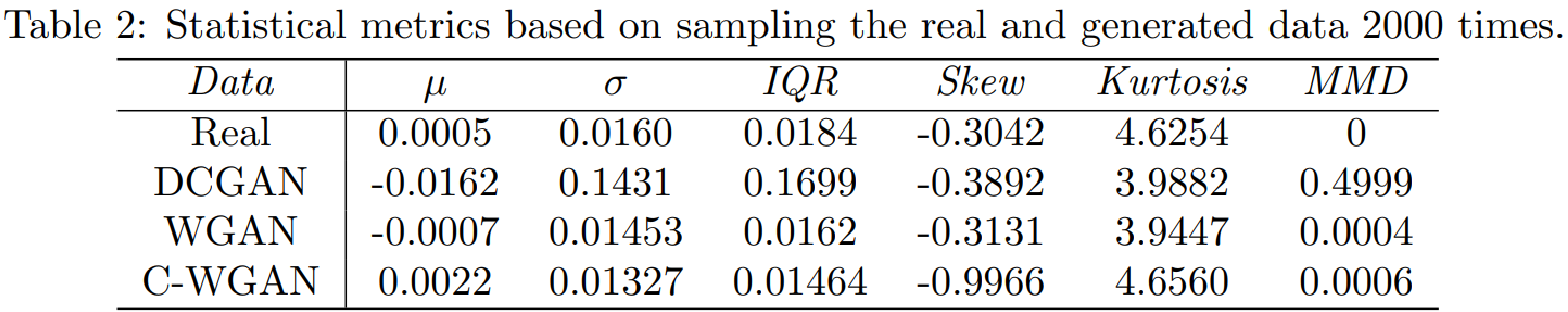
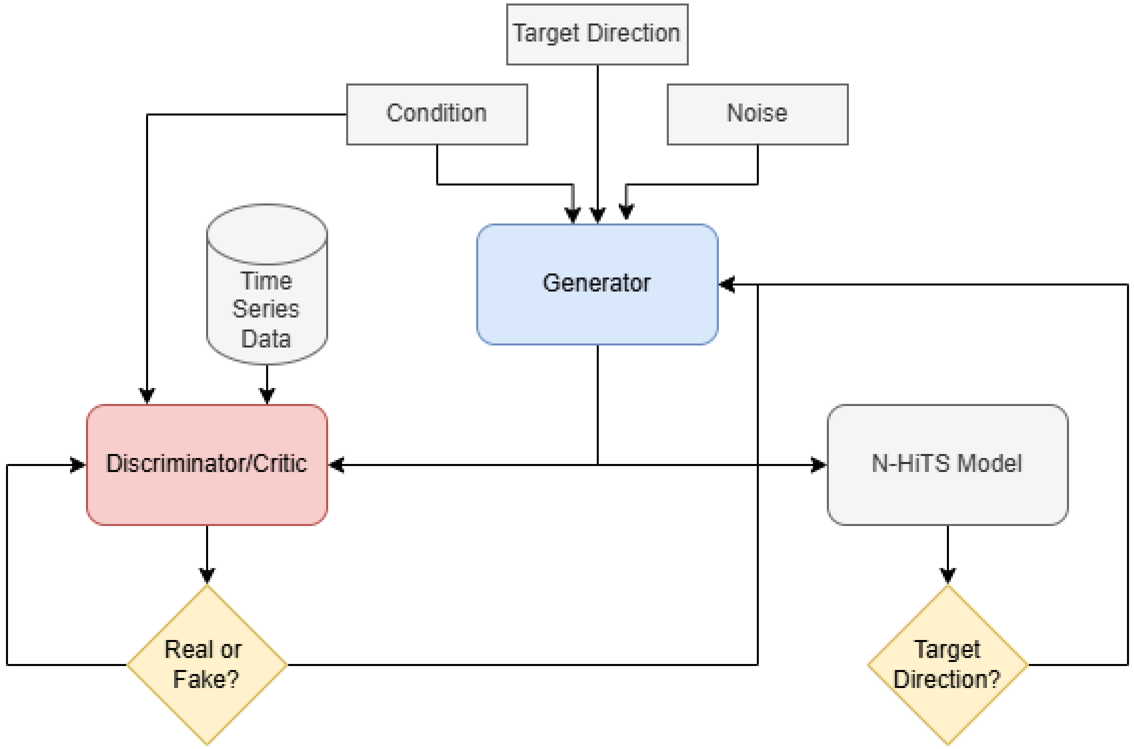


Table 2: Statistical metrics based on sampling the real and generated data 2000 times.

Data	μ	σ	*IQR*	*Skew*	*Kurtosis*	*MMD*
Real	0.0005	0.0160	0.0184	-0.3042	4.6254	0
DCGAN	-0.0162	0.1431	0.1699	-0.3892	3.9882	0.4999
WGAN	-0.0007	0.01453	0.0162	-0.3131	3.9447	0.0004
C-WGAN	0.0022	0.01327	0.01464	-0.9966	4.6560	0.0006

Future Work

- In the upcoming weeks, the research will focus on adding a second critic to the GAN, which is the N-HiTS model to be fooled.
  - Ideally, we would conduct experiments in a white box and black box setting, to evaluate the effectiveness of the attack method.
  - Furthermore, given that the N-HiTS model requires at least 300 days of data, experimentation will need to be performed to determine how we would generate 300 days of data.
- Perform hyperparameter tuning on the W-GAN/C-WGAN to maximize performance. In addition, experimentation with the TimeGAN and Sig-WGAN could be performed.
- Implement more black box attacks to compare as a baseline, as currently we have only white box methods.



This Research Aims to Make Both Powerful and Believable Adversarial Attacks on Time Series Data

References

[1] Y. Dong et al., “Boosting adversarial attacks with momentum,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018. doi: 10.1109/CVPR.2018.00957.

[2] I. J. Goodfellow, J. Shlens and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014.doi: 10.48550/arxiv.1412.6572.

[3] Z. Shen and Y. Li, “Temporal characteristics-based adversarial attacks on time series forecasting,” Expert systems with applications, vol. 264, no. 125950, 2025. doi: 10.1016/j.eswa.2024.125950.

[4] N. Ghaffari Laleh et al., “Adversarial attacks and adversarial robustness in computational pathology,” Nature communications, vol. 13, no. 1, 2022. doi: 10.1038/s41467-022-33266-0.

[5] M. Gallagher et al., “Investigating machine learning attacks on financial time series models,”Computers & security, vol. 123, no. 102933, 2022. doi: 10.1016/j.cose.2022.102933.

[6] P. Rathore et al., “Untargeted, targeted and universal adversarial attacks and defenses on timeseries,” in 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207272.

[7] J.Zhang et al., “Are time-series foundation models deployment-ready? a systematic study of adversarial robustness across domains,” 2025. doi: 10.48550/arxiv.2505.19397.

[8] 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.

[9] J. Chen et al., “Mag-gan: Massive attack generator via gan,” Information sciences, vol. 536, pp. 67–90, 2020. doi: 10.1016/j.ins.2020.04.019.

[10] J. Chen et al., “Time series data augmentation for energy consumption data based on improved timegan,” Sensors, vol. 25, no. 2, 2025. doi: 10.3390/s25020493.

[11] L. Wang and K.-J. Yoon, “Psat-gan: Efficient adversarial attacks against holistic scene understanding,” IEEE transactions on image processing, vol. 30, no. 9524508, 2021. doi: 10.1109/TIP.2021.3106807.

[12] S. Wu H. Sun and L. Ma, “Adversarial attacks on gan-based image fusion,” Information fusion, vol. 108, no. 102389, 2024. doi: 10.1016/j.inffus.2024.102389.

[13] S. Baluja and I. Fischer, “Adversarial transformation networks: Learning to generate adversarial examples,” 2017. doi: 10.48550/arxiv.1703.09387.

[14] C. Challu et al., “N-hits: Neural hierarchical interpolation for time series forecasting,” 2022. doi:10.48550/arxiv.2201.12886.

[15] StockAnalysis. “A list of all stocks in the s&p 500 index.” (2025), [Online]. Available: https://stockanalysis.com/list/sp-500-stocks/ (visited on 06/15/2025).

[16] Y. Lai J. Zhang and J. Lin, “The day-of-the-week effects of stock markets in different countries,”Finance research letters, vol. 20, pp. 47–62, 2017. doi: 10.1016/j.frl.2016.09.006.

[17] J. Sen and S. Dasgupta, “Adversarial attacks on image classification models: Fgsm and patchattacks and their impact,” 2023. doi: 10.48550/arxiv.2307.02055.

[18] I. Goodfellow A. Kurakin and S. Bengio, “Adversarial examples in the physical world,” 2016. doi:10.48550/arxiv.1607.02533.

[19] I. J. Goodfellow et al., “Generative adversarial networks,” 2014. doi: 10.48550/arxiv.1406.2661.

[20] J. Liao et al., “Sig-wasserstein gans for conditional time series generation,” Mathematical finance, vol. 34, no. 2, pp. 622–670, 2024. doi: 10.1111/mafi.12423.

[21] S. L. Hyland C. Esteban and G. Ratsch, “Real-valued (medical) time series generation with recur-rent conditional gans,” 2027. doi: 10.48550/arxiv.1706.02633.

[22] L. Metz A. Radford and S. Chintala, “Unsupervised representation learning with deep convolutionalgenerative adversarial networks,” 2015. doi: 10.48550/arxiv.1511.06434.

[23] I. Gulrajani et al., “Improved training of wasserstein gans,” 2017. doi: 10.48550/arxiv.1704.00028