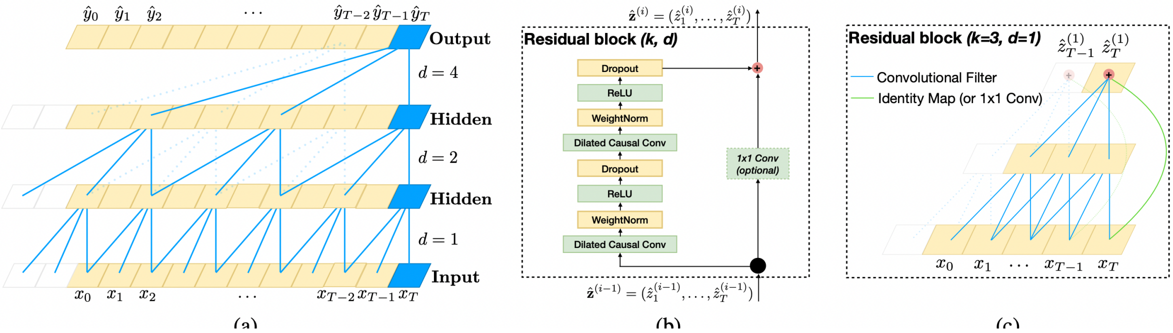
Table of Contents

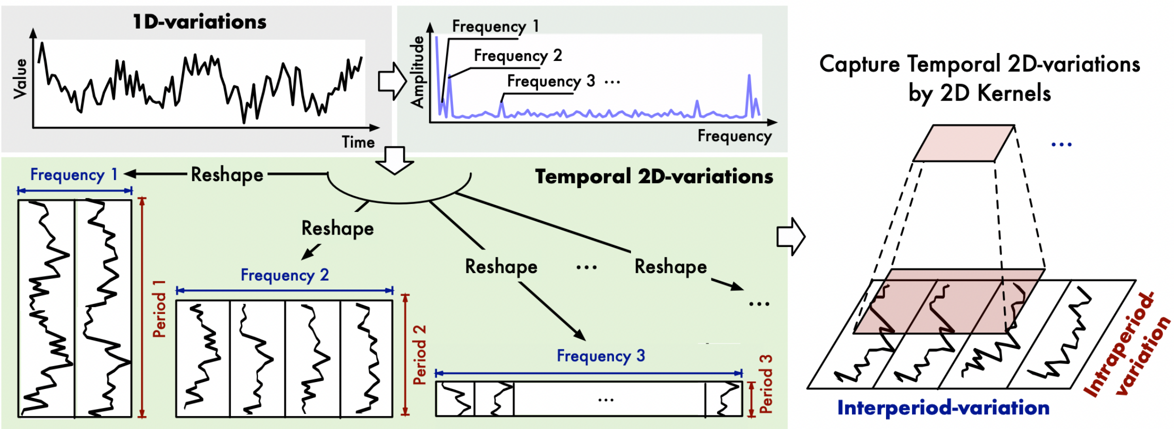
Predicting extreme events with a small window using ML on AAPL stock.

Overview

This project retrieves a specified stock (or a series of stocks) from Yahoo Finance, trains 3 different models, and evaluates the results on a requested stock. The first model is a random forest (RF) model, which is a classical machine learning model. The compute required to train and infer on this model is low, however, in this case, because of the lack of data (1 stock in this case) and low dimensionality, traditional ML models like RF are expected to be more reliable than deep learning models. The second model is a deep learning model called temporal convolutional neural network (TCNN). TCNNs are based on convolutions and due to their causal convolution architecture they prevent leakage from the future to the past and are meant to be effective even for longer sequences due to their dilated convolutions.

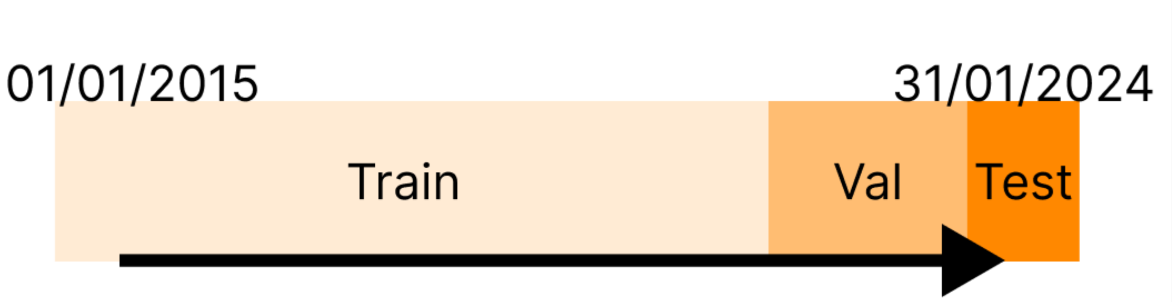


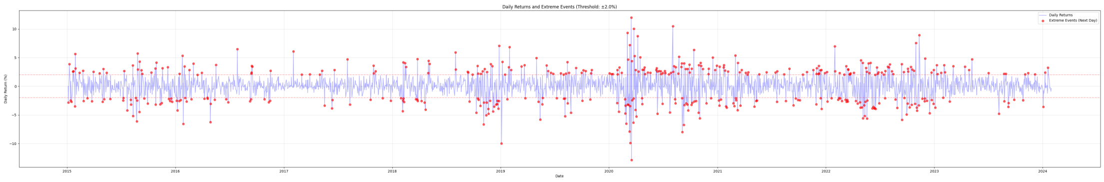
Finally, the improved architecture is a 2D convolution method where an FFT transform is used to find K frequency candidates and reshape them into a 2D space at which point a 2D CNN (or transformer) is used to extract features (ConvNext, inception , swin, etc...).



Methods

One of the most important steps for this project to generalize correctly is proper data splitting. While we could generate N sequences of length 10 across the time of the stock data, that would lead to leaks from the training set to the validation set. For that reason, the stock data was split based on the time.:

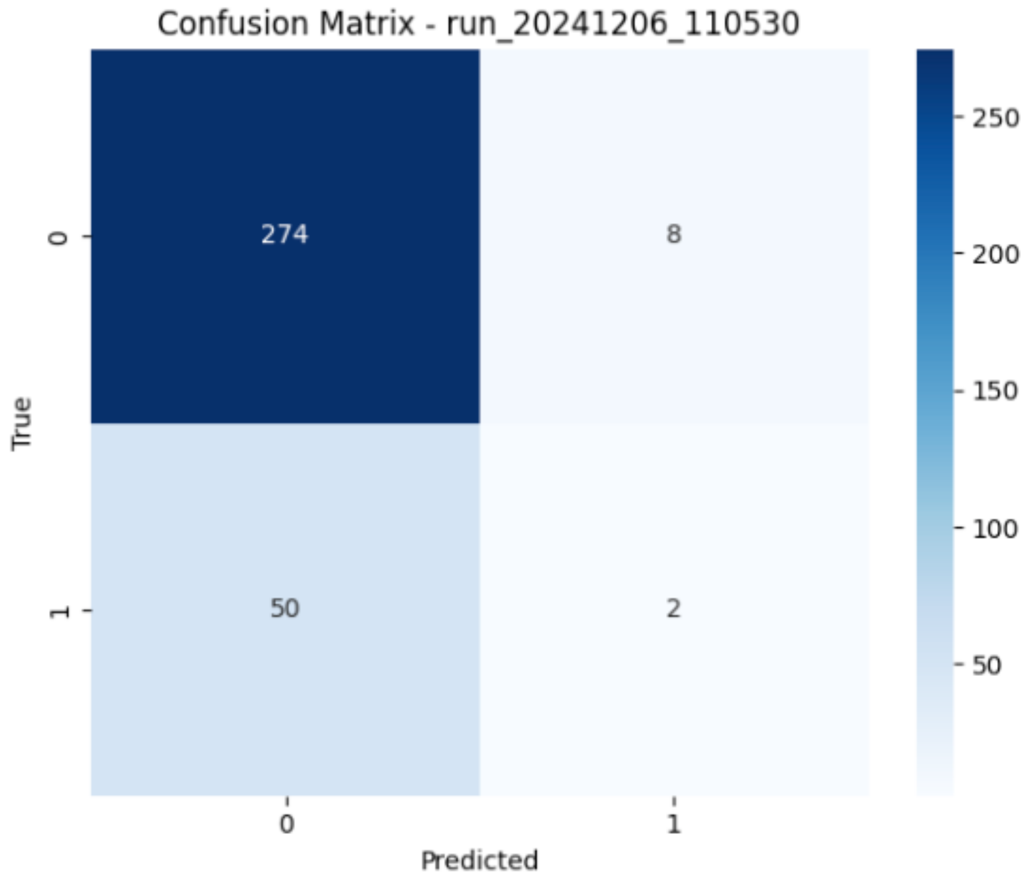


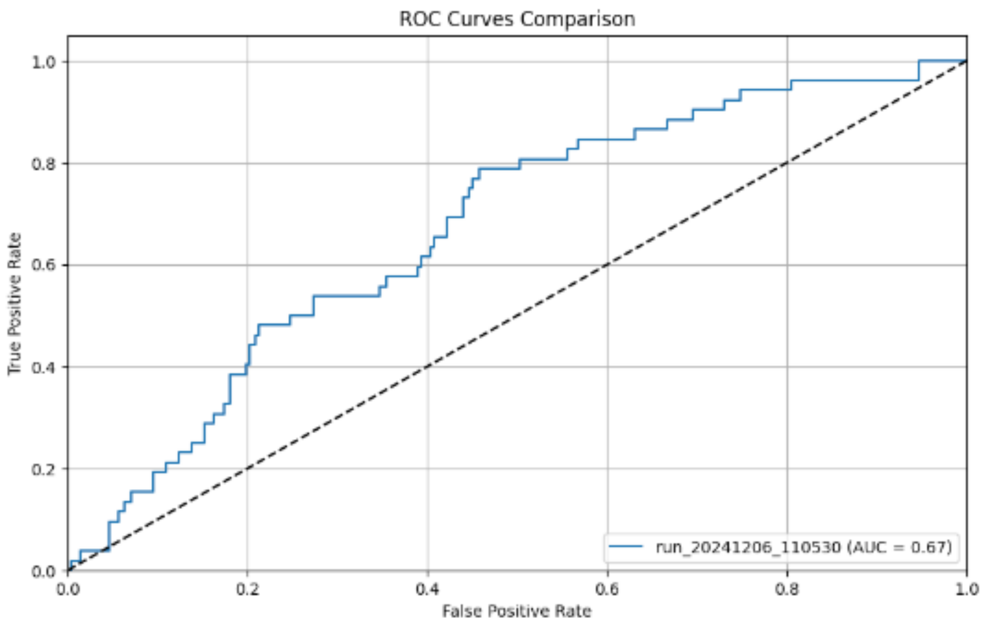


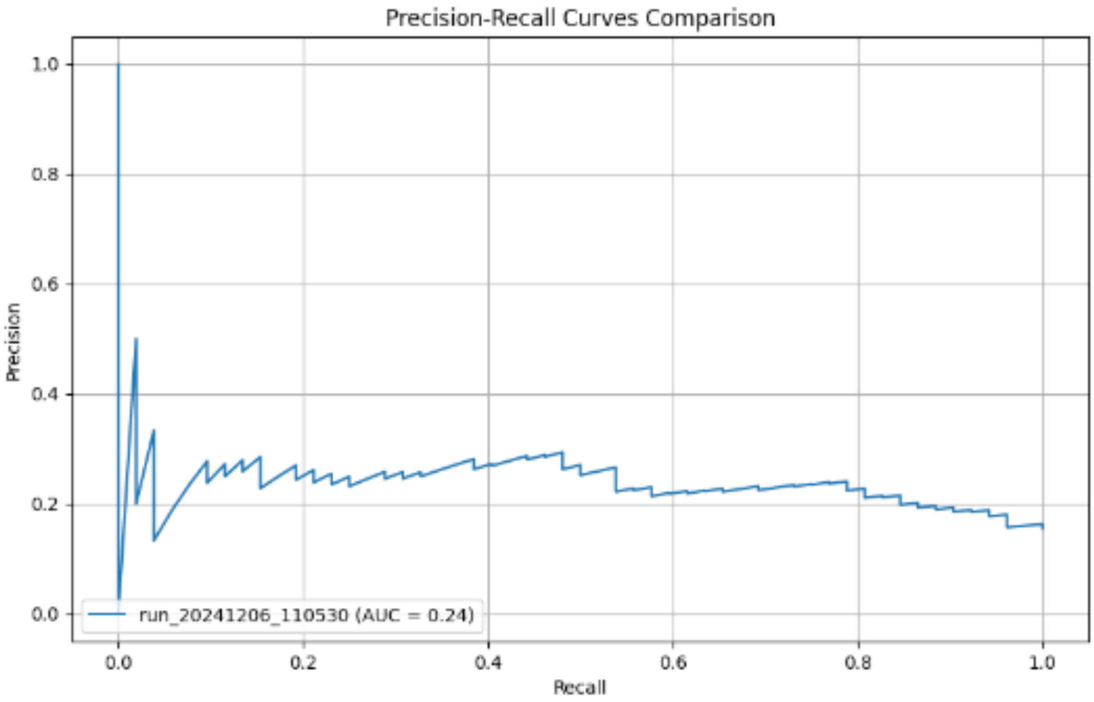
To train the deep learning models, the data had to be normalized. Because the daily return column is a percentage, I considered it to be normalized. So, I applied normalization at every batch, per channel to stabilize training. Looking at the dataset we see a clear imbalance between extreme and non-extreme labels. To solve that we have 3 options. First, we can use a weighted sampler, apply weights on the loss function, or use SMOTE to generate more data for the extreme label. The option considered the best is using a weighted sampler; however, it would be best to test each one out and compare (all of these 3 methods are implemented). Finally, we can apply PCA to remove dimensions that don't correlate with our labels and are mostly random, this will increase the performance of our smaller, RF model. Additionally, 2 types of augmentations have been implemented. First, random noise addition per channel, and then I also implemented masking the time series randomly by zeroing out points. Please keep in mind that SMOTE, PCA, and augmentations were not used in the examples below except for the case of RF where all of the training runs were executed with only one feature used (daily return) as it showed the best performance.

Model Evaluation

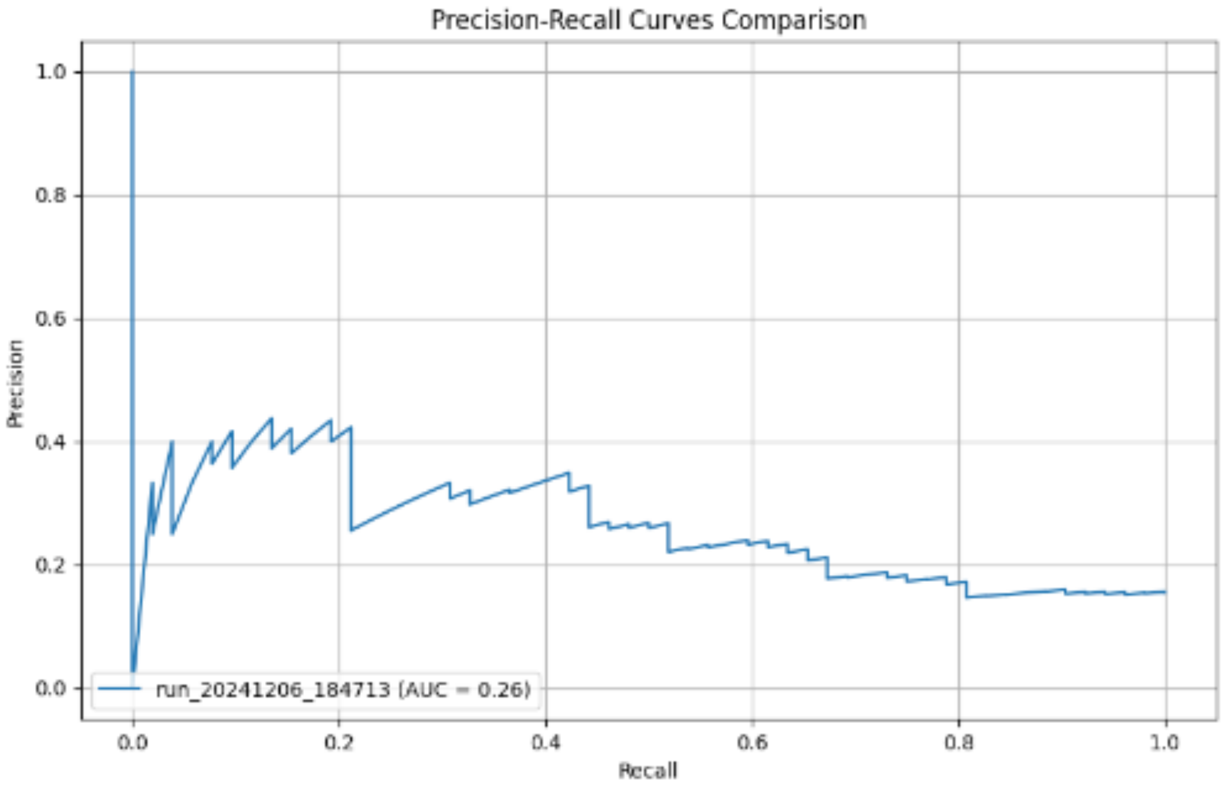
RF on AAPL data

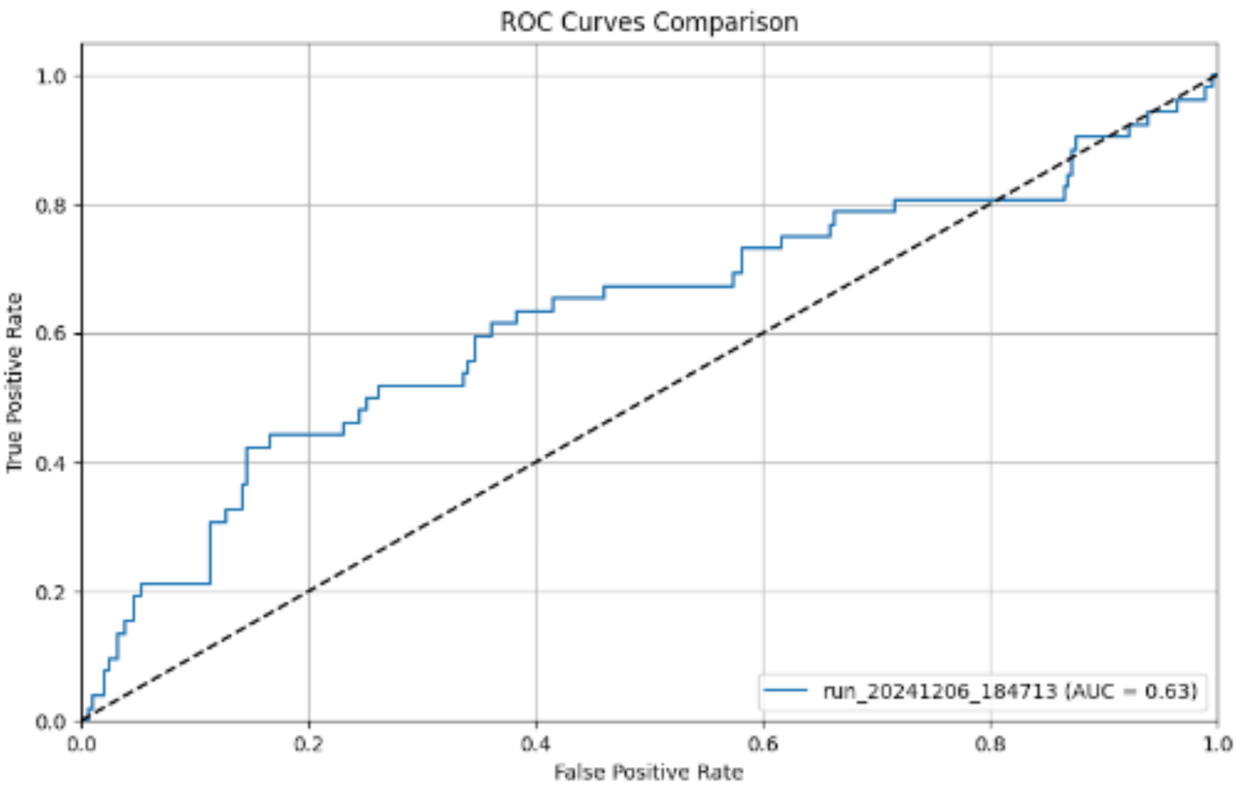


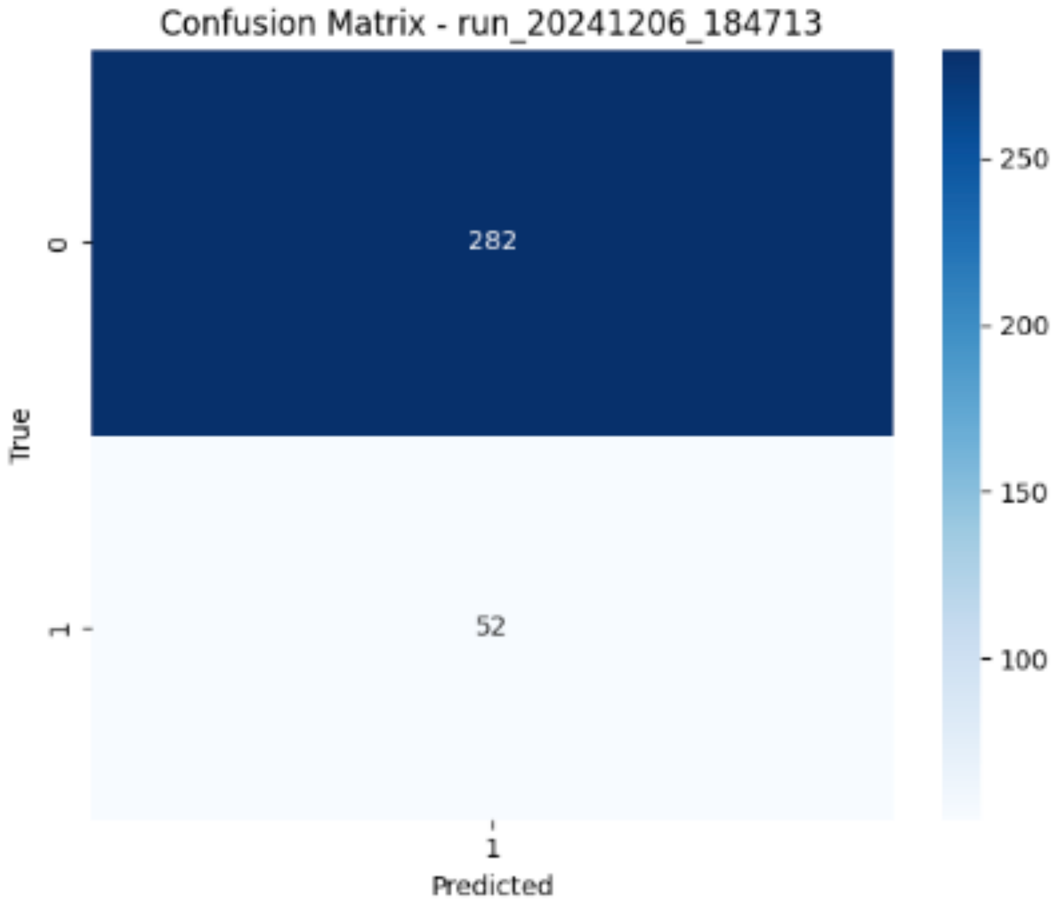



Precision-Recall Curves Comparison
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
run_20241206_110530 (AUC = 0.24)

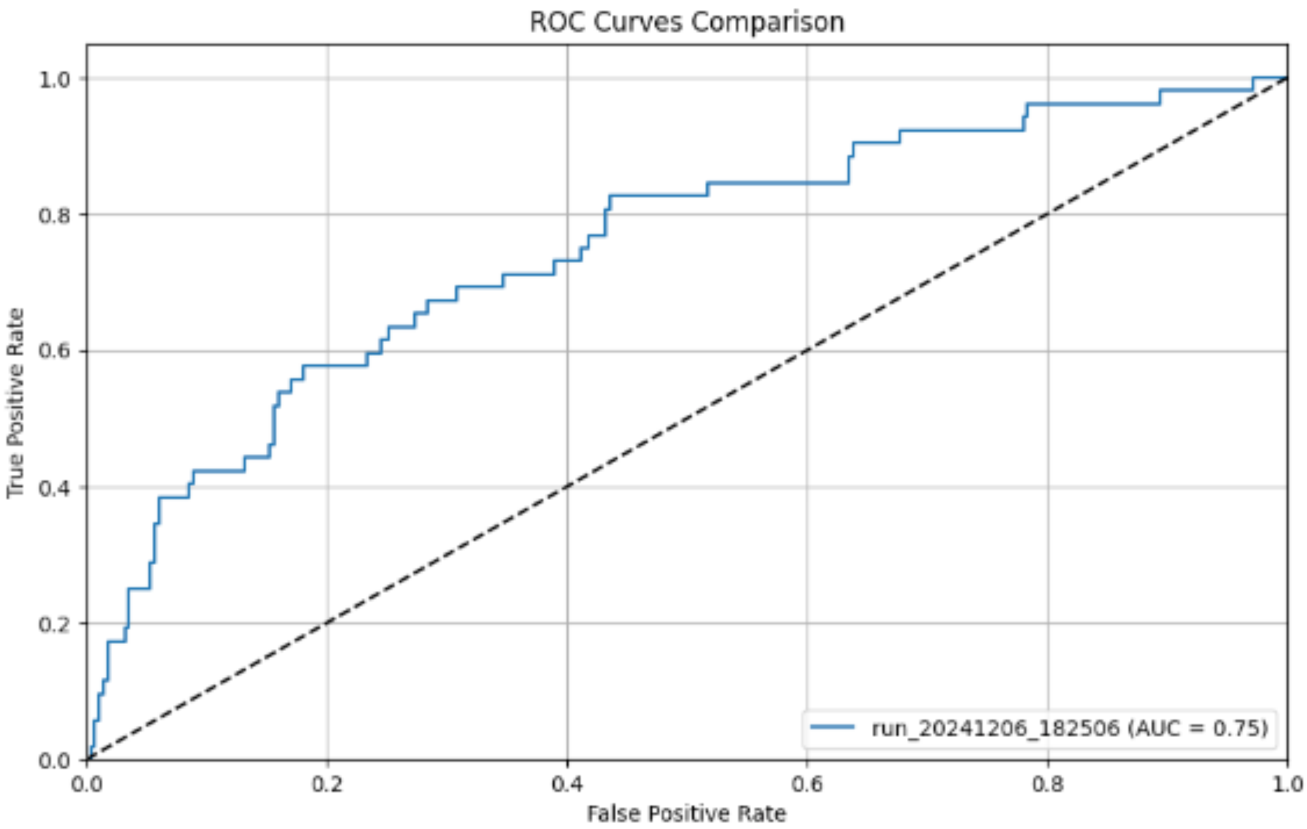
TCNN on AAPL data

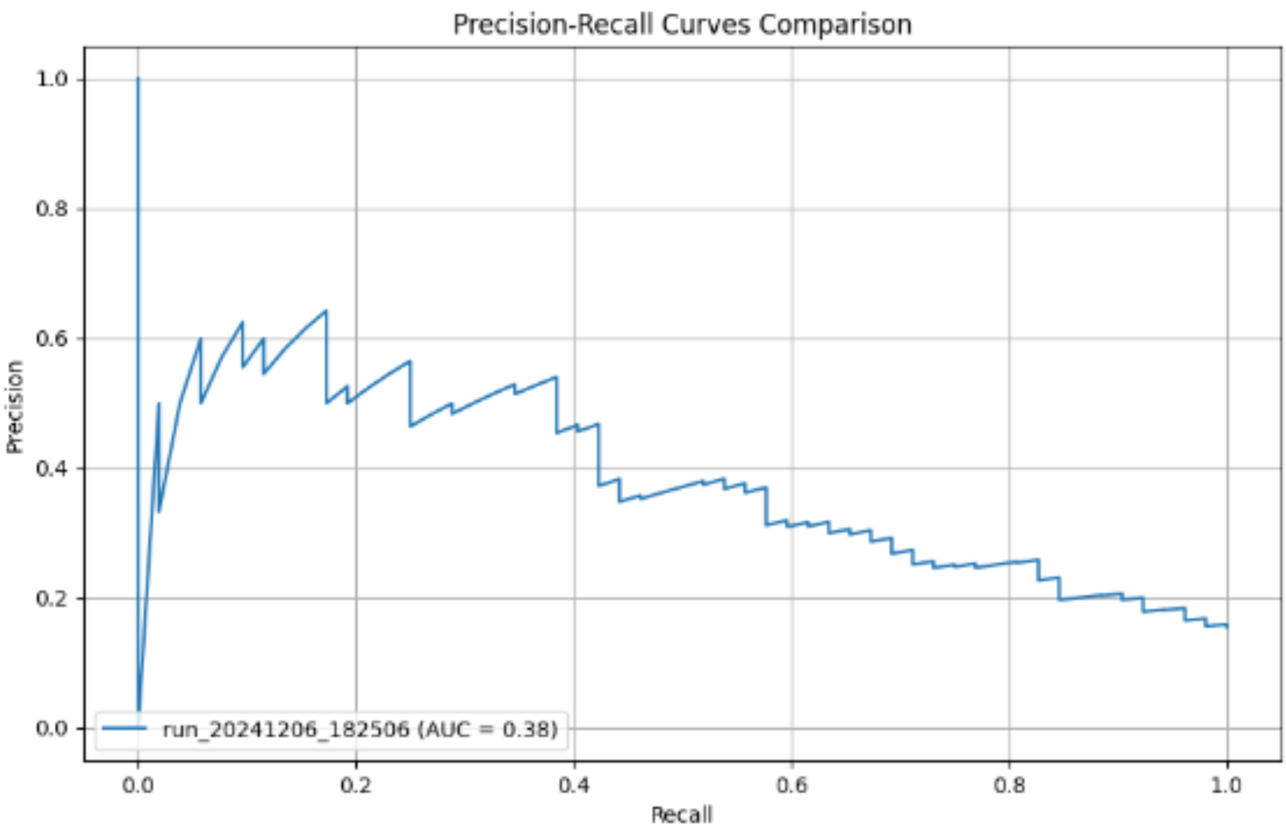


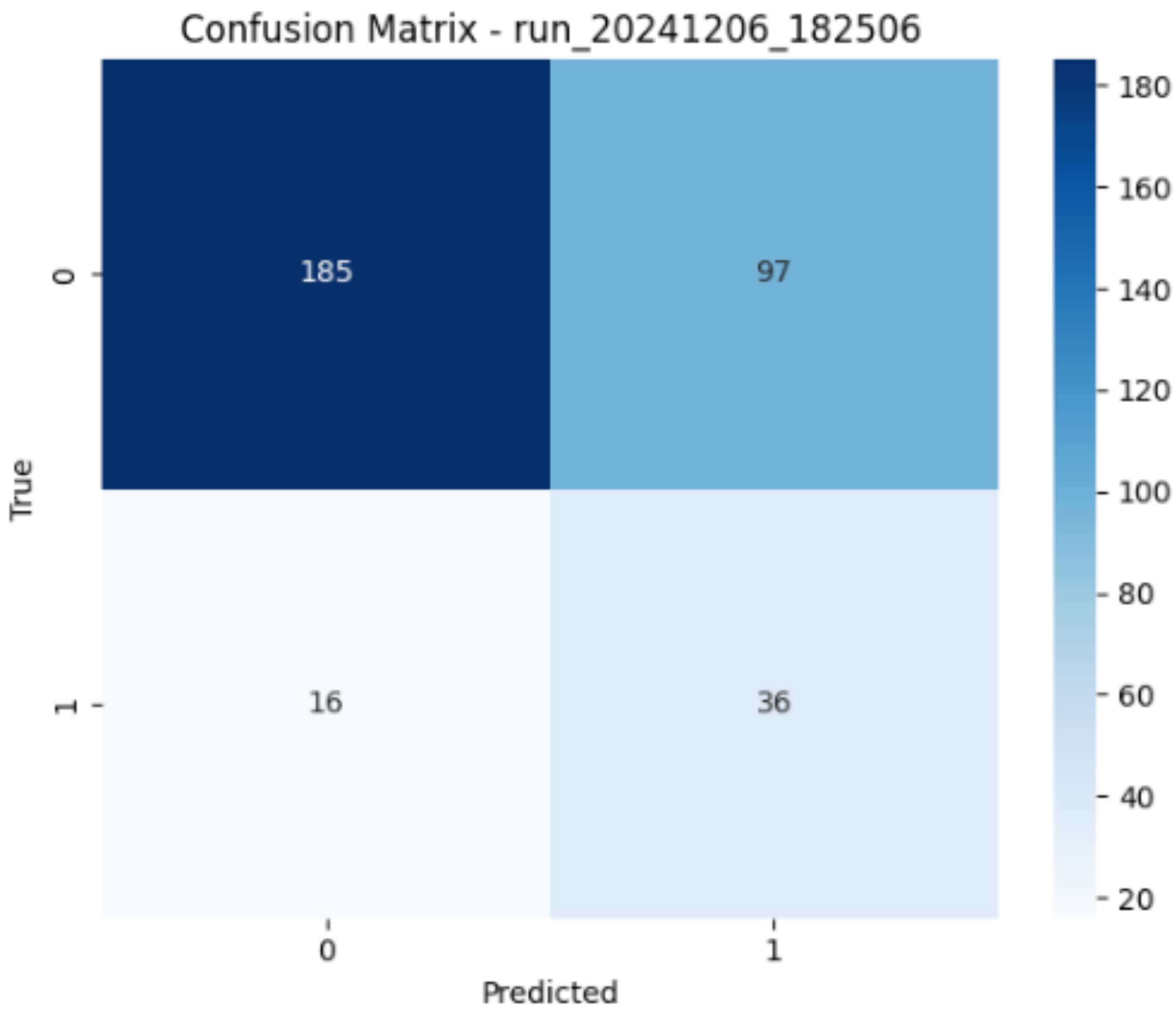



Confusion Matrix - run_20241206_184713
282
52
0
1
True
1
Predicted
250
200
150
100

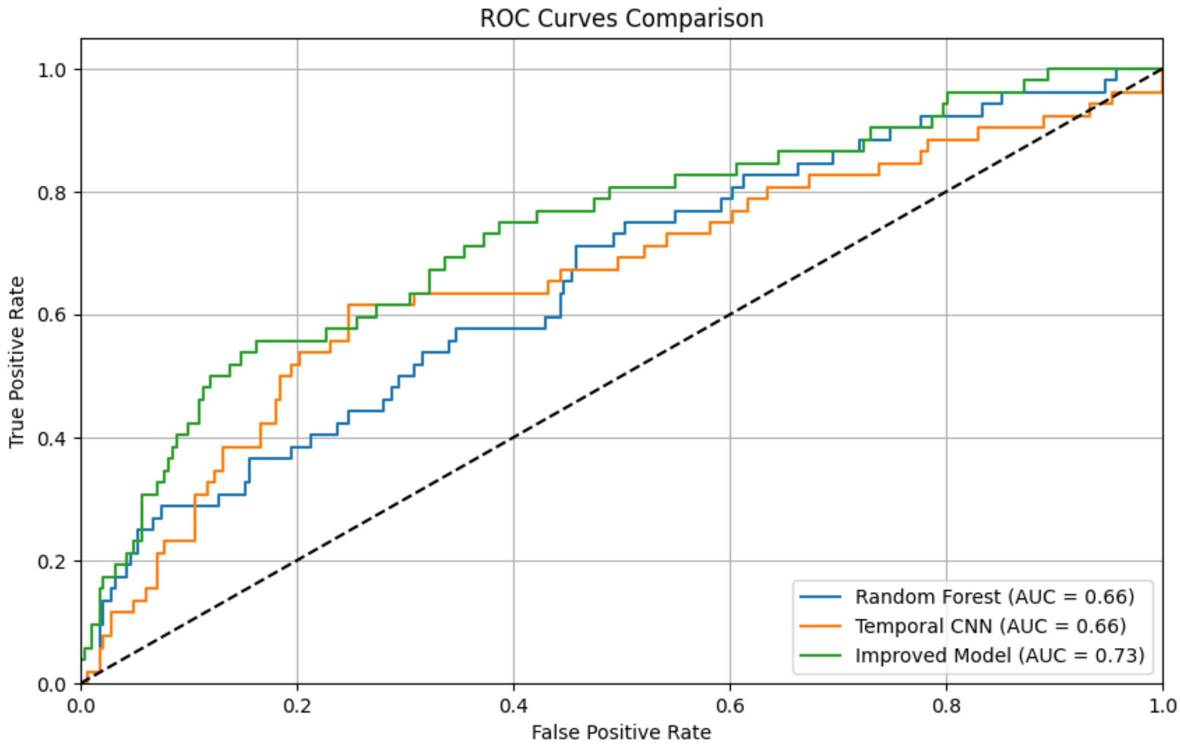
Improved on AAPL data with augmentations

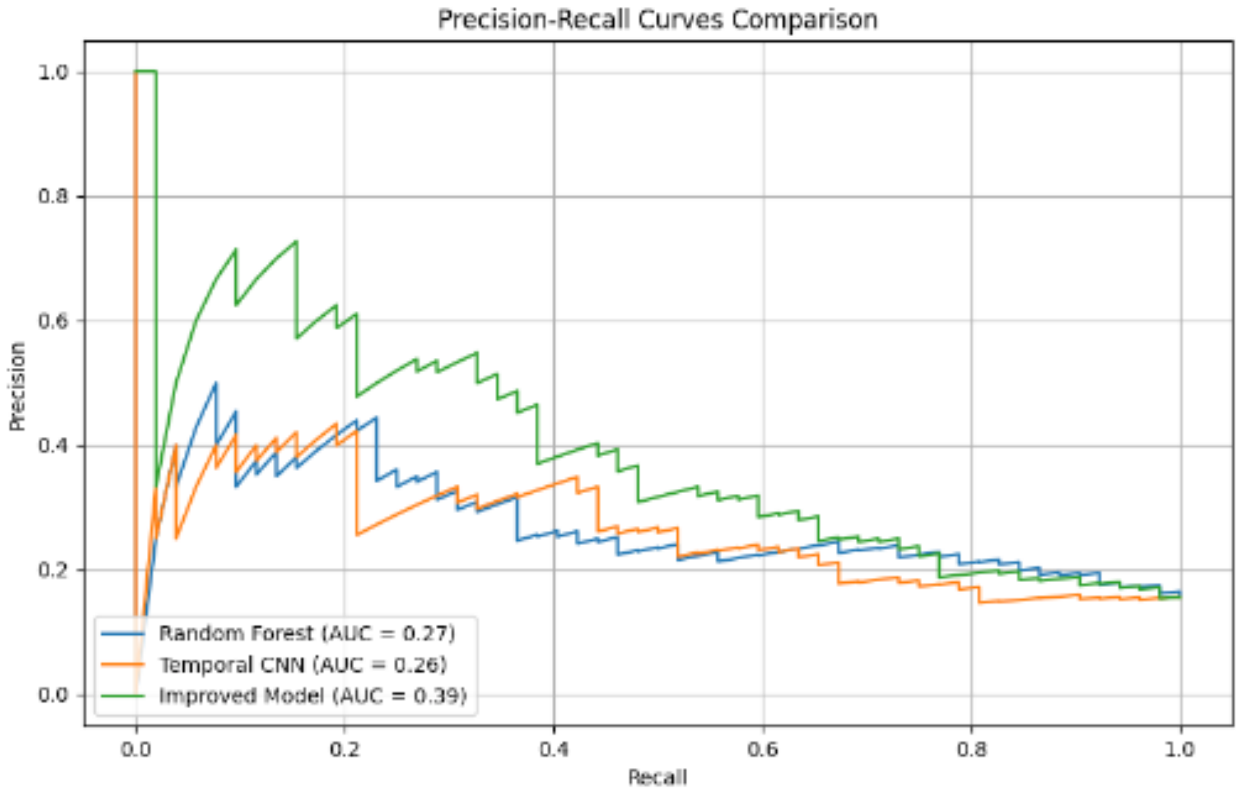





Confusion Matrix - run_20241206_182506
185
97
16
36
True
0
1
Predicted
0
1
180
160
140
120
100
80
60
40
20

Comparison RF vs TCNN vs Improved on AAPL data





Metrics For Comparison On AAPL

=== Random Forest Classification Report ===

Classification Report for Random Forest

Class: 0
Precision: 0.870
Recall: 0.947
F1-score: 0.907
Support: 282

Class: 1
Precision: 0.444
Recall: 0.231
F1-score: 0.304
Support: 52

Class: macro avg
Precision: 0.657
Recall: 0.589
F1-score: 0.605
Support: 334

Class: weighted avg
Precision: 0.803
Recall: 0.835
F1-score: 0.813
Support: 334

Overall Metrics:
Accuracy: 0.835
Macro avg F1: 0.605
Weighted avg F1: 0.813

=== Temporal CNN Classification Report ===

Classification Report for Temporal CNN

Class: 0
Precision: 0.000
Recall: 0.000
F1-score: 0.000
Support: 282

Class: 1
Precision: 0.156
Recall: 1.000
F1-score: 0.269
Support: 52

Class: macro avg
Precision: 0.078
Recall: 0.500
F1-score: 0.135
Support: 334

Class: weighted avg
Precision: 0.024
Recall: 0.156
F1-score: 0.042
Support: 334

Overall Metrics:
Accuracy: 0.156
Macro avg F1: 0.135
Weighted avg F1: 0.042

=== Improved Model Classification Report ===

Classification Report for Improved Model

Class: 0
Precision: 0.912
Recall: 0.699
F1-score: 0.791
Support: 282

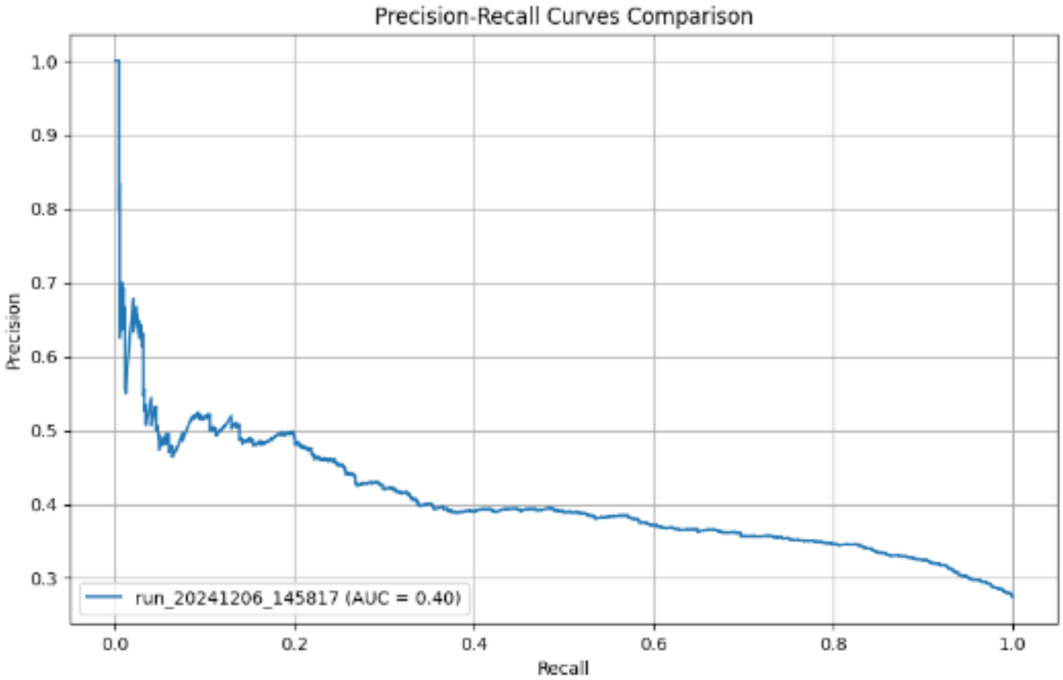
Class: 1
Precision: 0.280
Recall: 0.635
F1-score: 0.388
Support: 52

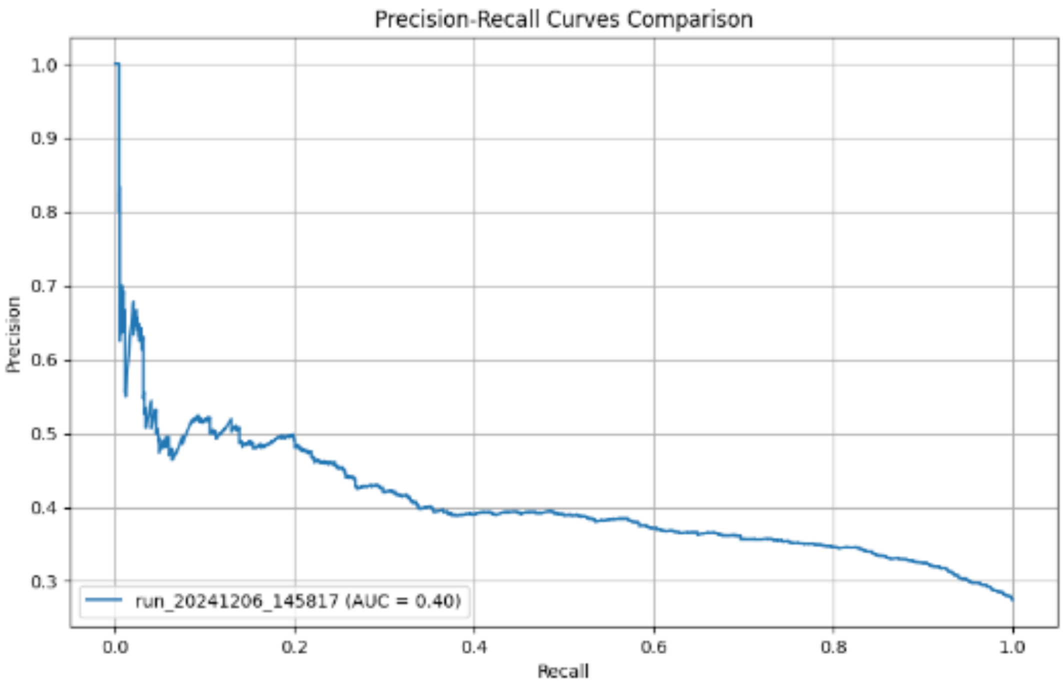
Class: macro avg
Precision: 0.596
Recall: 0.667
F1-score: 0.590
Support: 334

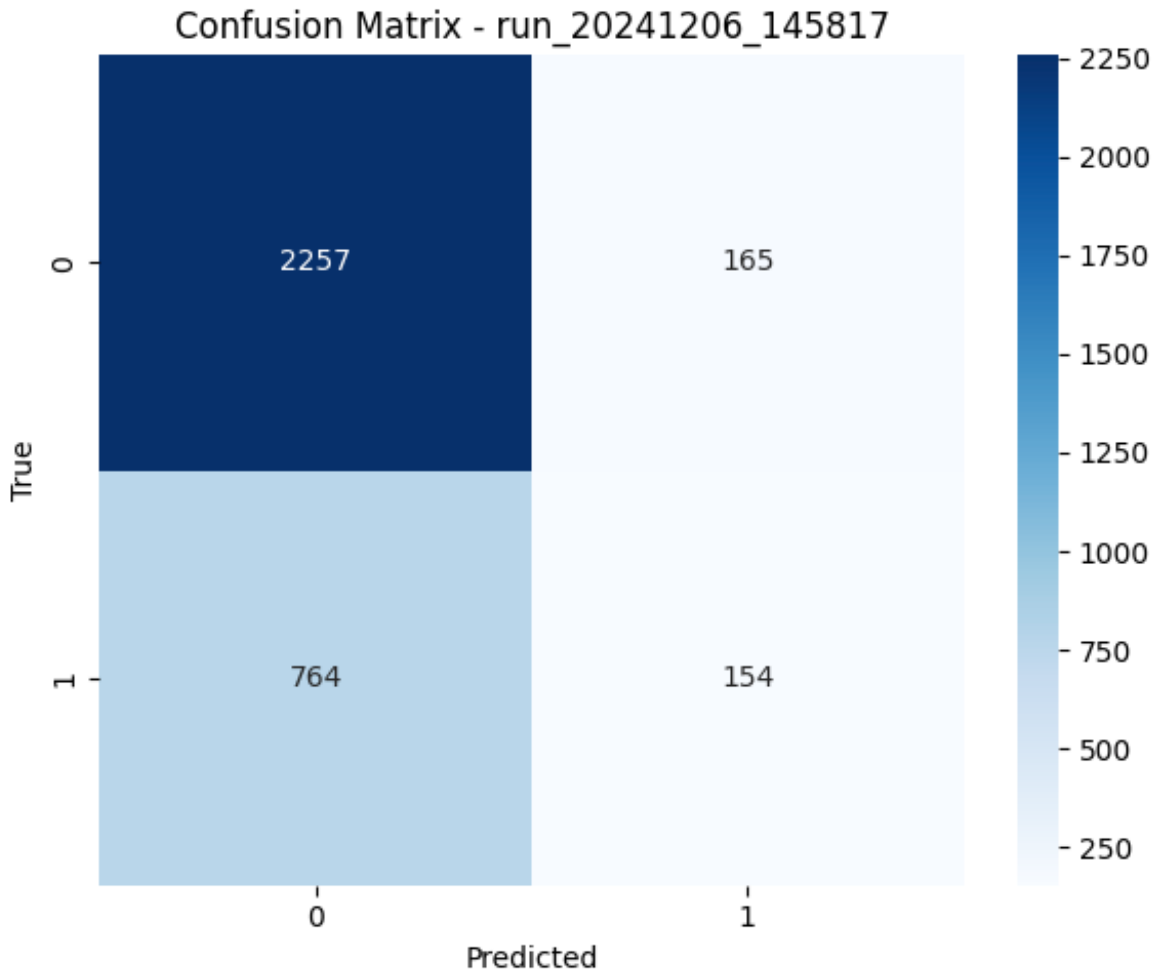
Class: weighted avg
Precision: 0.814
Recall: 0.689
F1-score: 0.728
Support: 334

Overall Metrics:
Accuracy: 0.689
Macro avg F1: 0.590
Weighted avg F1: 0.728

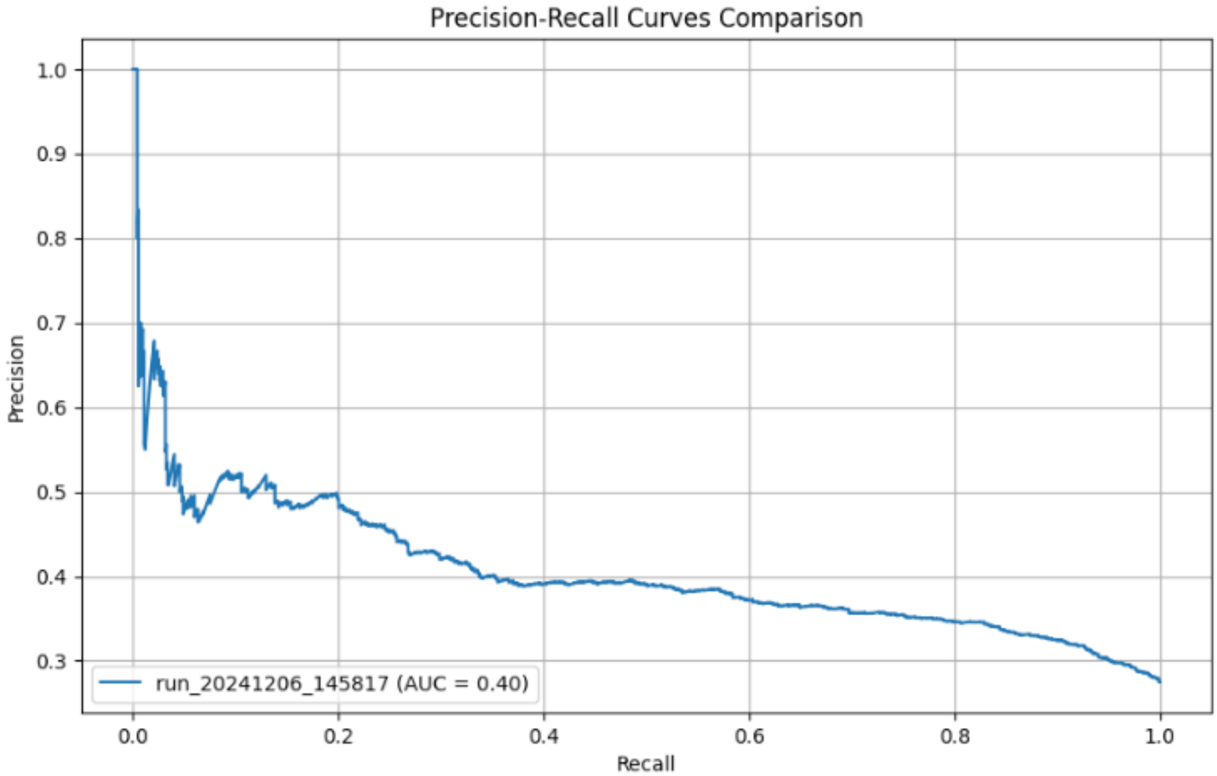
RF on SP500 TOP 10 data (S&P500 top 10 stocks)

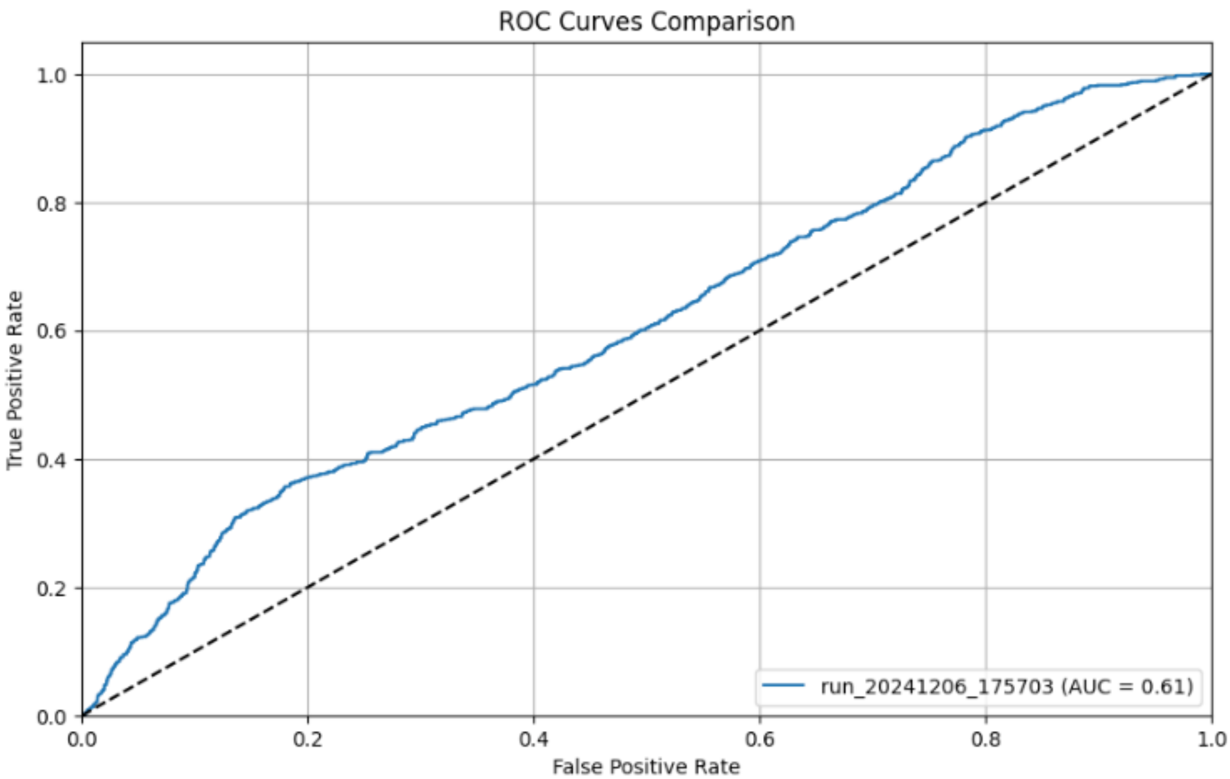


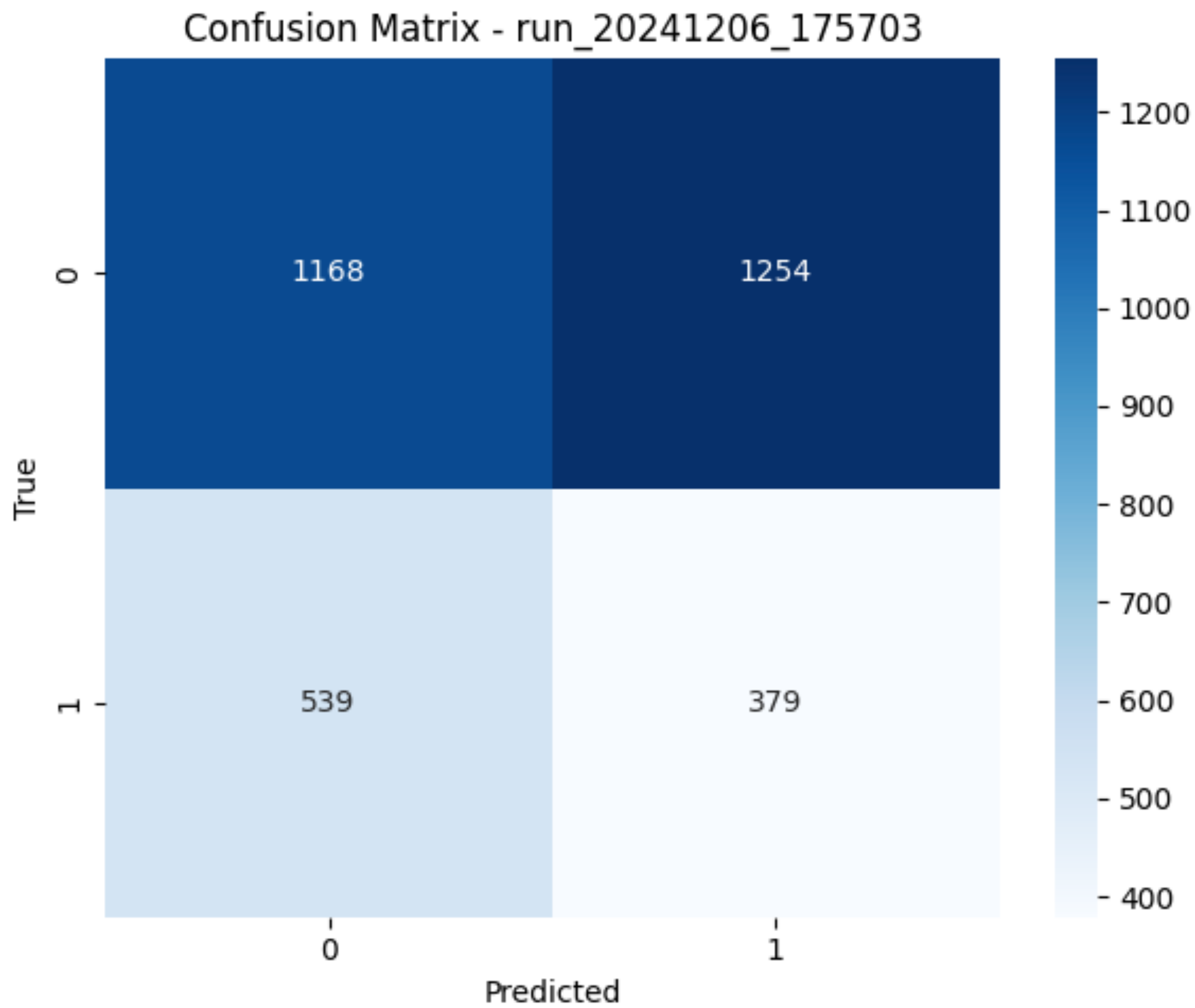



Confusion Matrix - run_20241206_145817
True
Predicted
0
1
0
1
2257
165
764
154
2250
2000
1750
1500
1250
1000
750
500
250

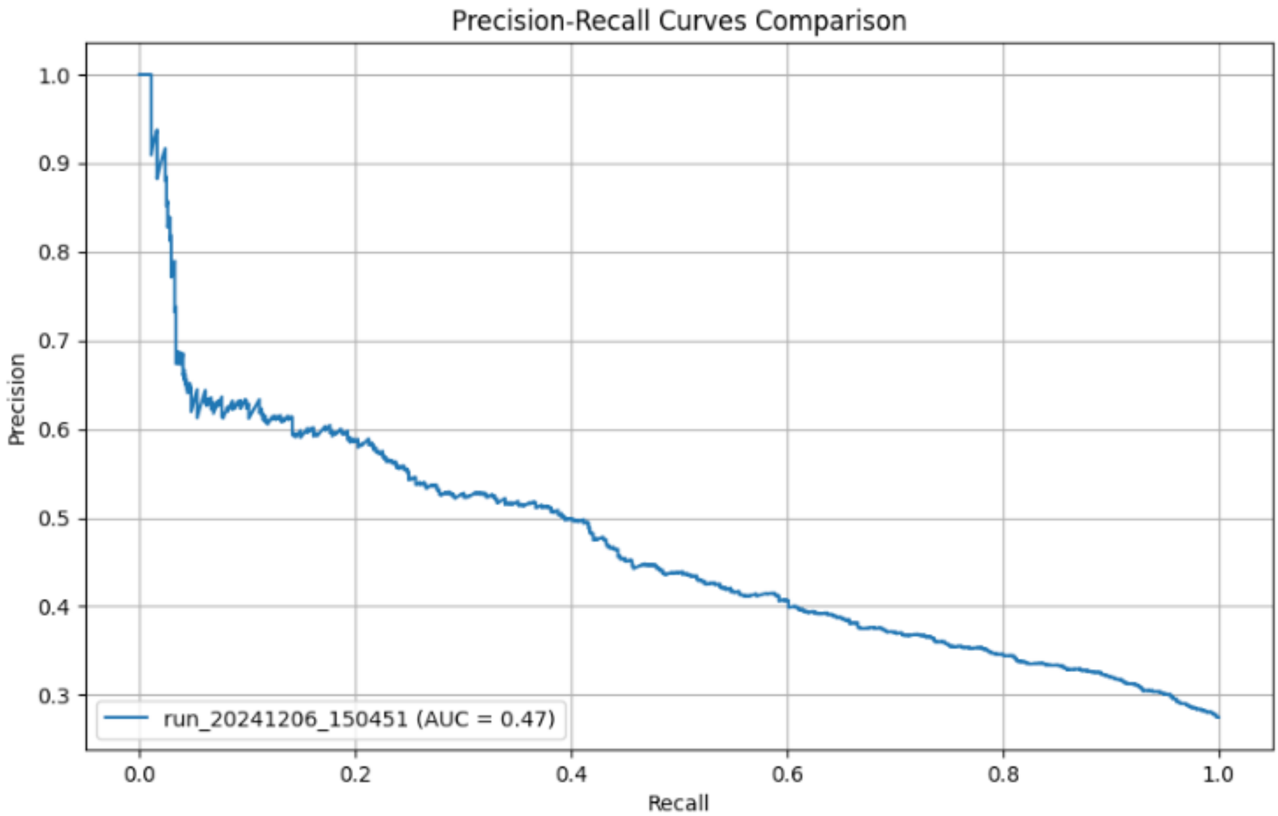
TCNN on SP500 TOP 10 (S&P500 top 10 stocks)

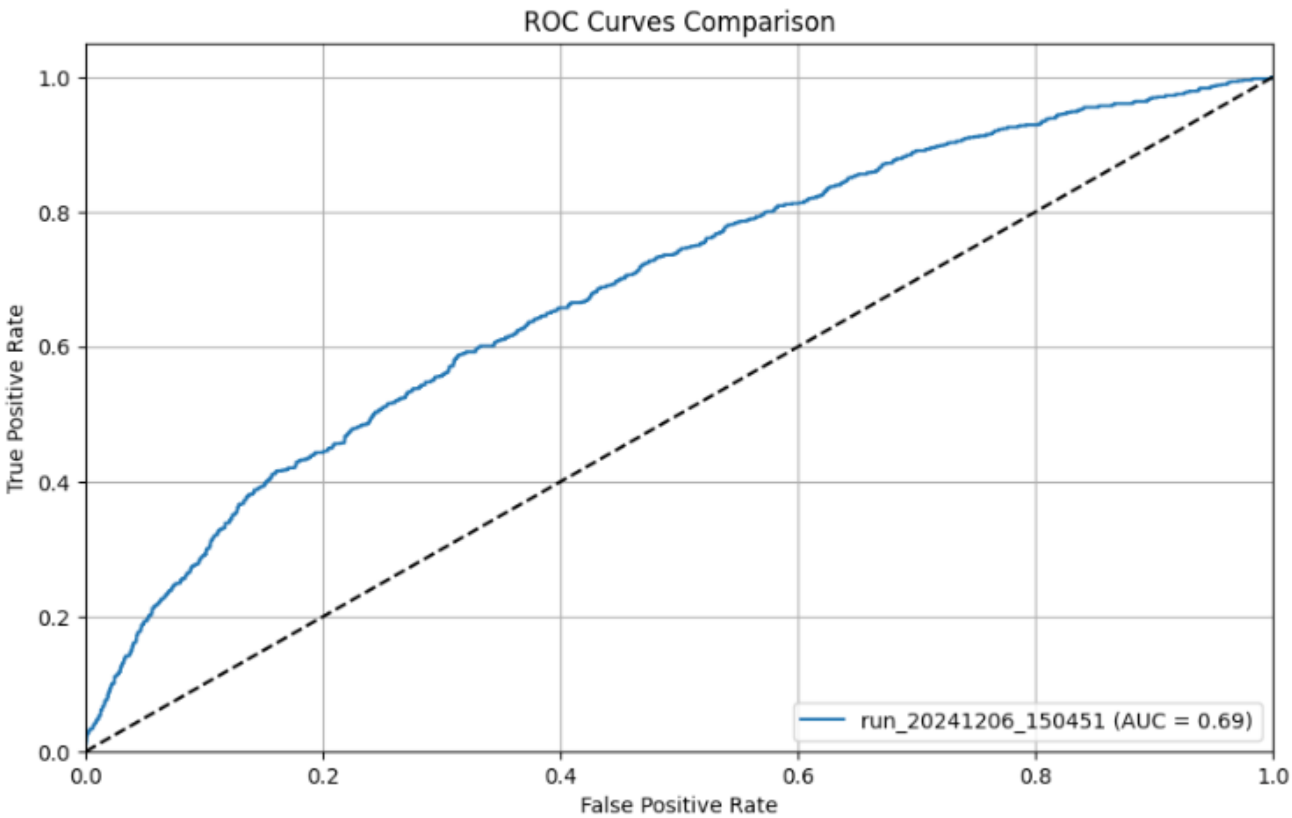


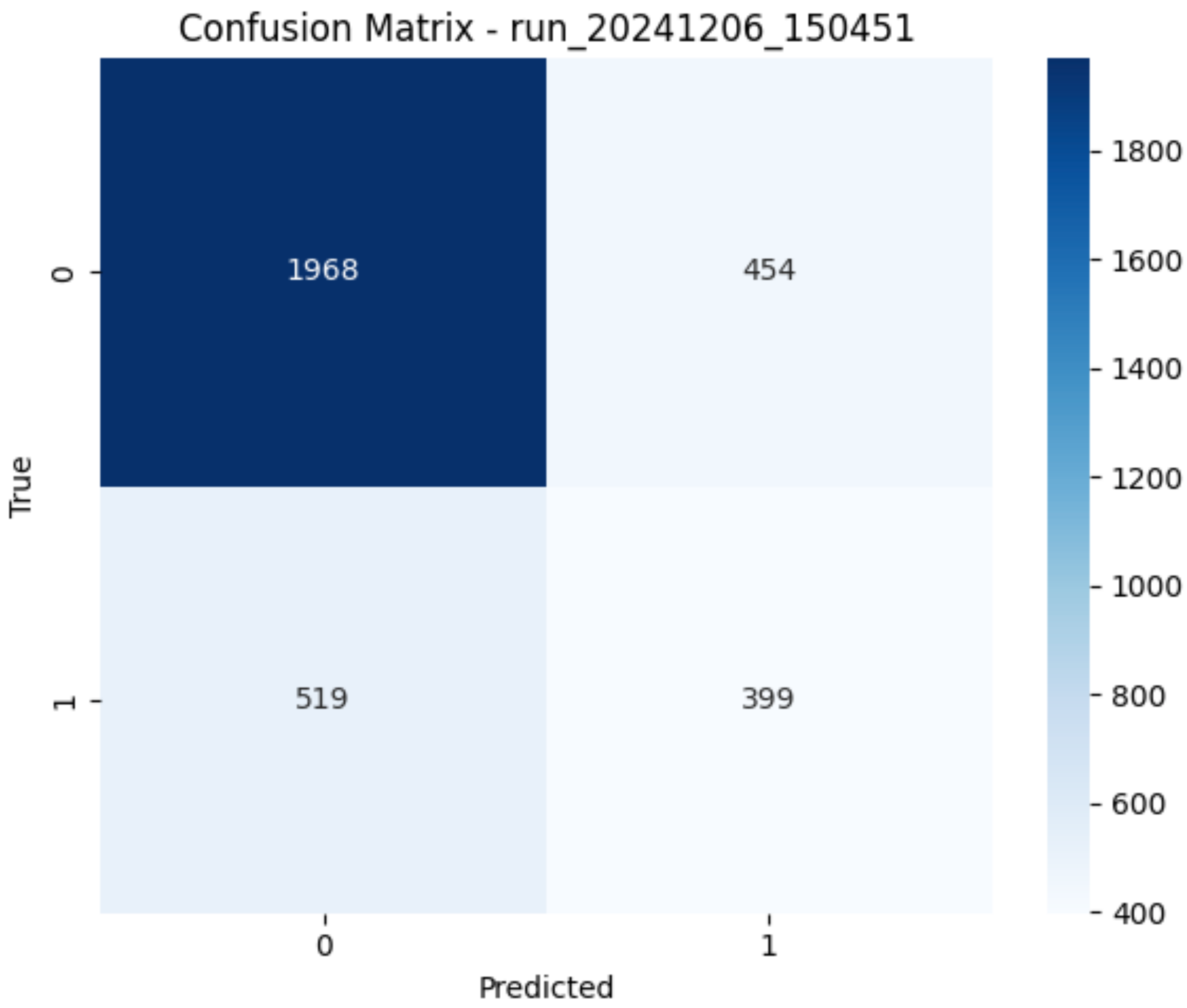



Confusion Matrix - run_20241206_175703
1168
1254
539
379
True
Predicted
0
1
0
1
1200
1100
1000
900
800
700
600
500
400

Improvement on SP500 TOP 10 (S&P500 top 10 stocks)

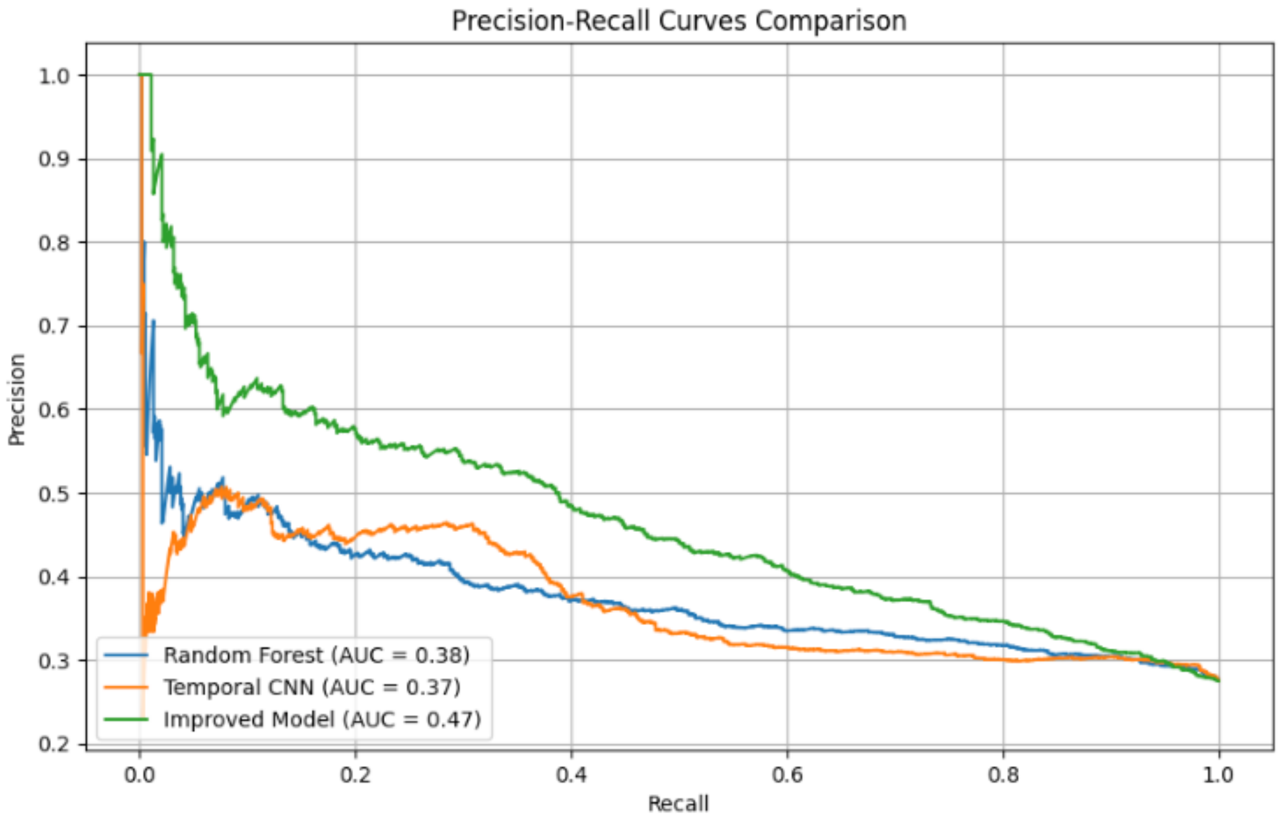


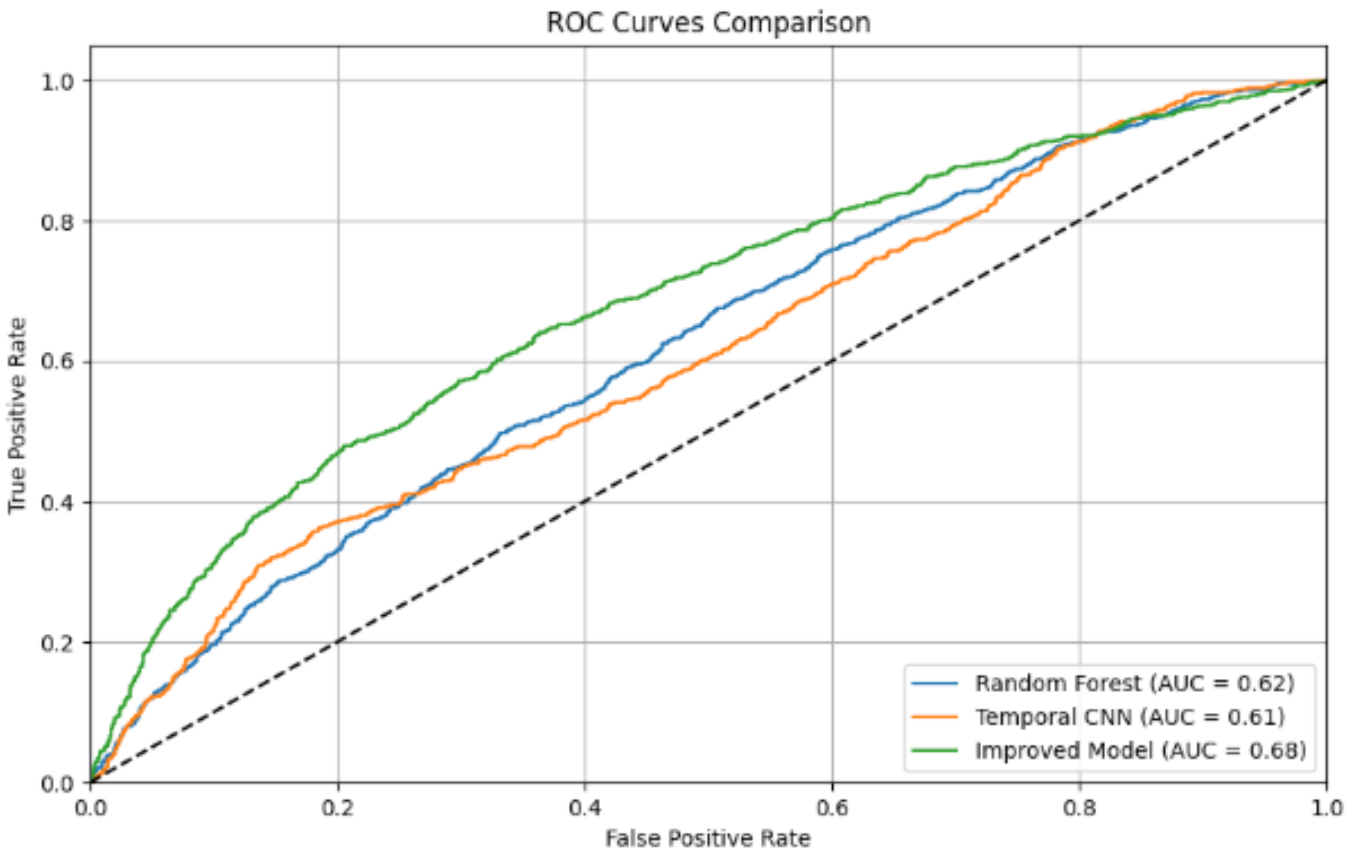




Confusion Matrix - run_20241206_150451
1968
454
519
399
0
1
True
0
1
Predicted
1800
1600
1400
1200
1000
800
600
400

Comparison RF vs TCNN vs Improved on SN500 TOP 10 data





Metrics For Comparison On SP500 TOP 10

=== Random Forest Classification Report ===

Classification Report for Random Forest

Class: 0
Precision: 0.777
Recall: 0.664
F1-score: 0.716
Support: 2422

Class: 1
Precision: 0.360
Recall: 0.499
F1-score: 0.418
Support: 918

Class: macro avg
Precision: 0.569
Recall: 0.581
F1-score: 0.567
Support: 3340

Class: weighted avg
Precision: 0.663
Recall: 0.618
F1-score: 0.634
Support: 3340

Overall Metrics:
Accuracy: 0.618
Macro avg F1: 0.567
Weighted avg F1: 0.634

=== Temporal CNN Classification Report ===

Classification Report for Temporal CNN

Class: 0
Precision: 0.684
Recall: 0.482
F1-score: 0.566
Support: 2422

Class: 1
Precision: 0.232
Recall: 0.413
F1-score: 0.297
Support: 918

Class: macro avg
Precision: 0.458
Recall: 0.448
F1-score: 0.431
Support: 3340

Class: weighted avg
Precision: 0.560
Recall: 0.463
F1-score: 0.492
Support: 3340

Overall Metrics:
Accuracy: 0.463
Macro avg F1: 0.431
Weighted avg F1: 0.492

=== Improved Model Classification Report ===

Classification Report for Improved Model

Class: 0
Precision: 0.808
Recall: 0.718
F1-score: 0.760
Support: 2422

Class: 1
Precision: 0.425
Recall: 0.549
F1-score: 0.479
Support: 918

Class: macro avg
Precision: 0.616
Recall: 0.634
F1-score: 0.620
Support: 3340

Class: weighted avg
Precision: 0.703
Recall: 0.672
F1-score: 0.683
Support: 3340

Overall Metrics:
Accuracy: 0.672
Macro avg F1: 0.620

Q&A

- Which model performs better for predicting extreme events?
Between the TCNN and RF models, the RF outperforms the TCNN across Precision, Recall, and F1-score, having a higher accuracy and a higher weighted average f1 score. Out of all the models, the best-performing one for extreme events is the Improved model with an f1 score of 0.388. When looking at the models trained on top 10 S&P500 stocks, the improved model is now the clear winner with the highest f1 score and balance between precision and recall.

- Which metric or metrics are more relevant for evaluating the performance of the methods?
The most relevant metric would be Recall (RF: 0.499 vs TCNN:0.413) for extreme events, which measures the proportion of actual extreme events that were correctly identified. F1 score (RF: 0.418 vs TCNN:0.297) for extreme events is also relevant as it can be a good indicator for a balance between precision and recall.

- Why is forecasting of such events a challenging task? Name three reasons.
1) The markets constantly adapt to change, so if we were to find the perfect algorithm today, next week the rest of the pool will have adjusted their strategies to mitigate our lead.
2)The underlying causes of some of these events are impossible to predict from time series (example: Covid).
3) Large class imbalance, which makes it difficult for models to find patterns.

- How well do the models handle class imbalance (extreme events vs no extreme events)?
As expected, all models indicate a struggle with the class imbalance, favoring the non-extreme event class 0. Overall, the improved model handles class imbalance best, TCNN struggles the most, and the RF is in between. There is definitely still room for improvement, though, through parameter tuning, better sampling, better loss, and better augmentations such as time wrap.

- **Can you assess the predictability of the models based on their performance?**
All models perform better than random chance but they all have high false positive rate across all models. The improved model catches about 55% of extreme events, the RF models 50% and the TCNN model misses 59% of the time.

- **Given the potentially low performance, would you say the models demonstrate predictive ability for extreme events in stock prices?**
These models would be better used with supervision or in an enseble with other models. More specifically, they could enable a warning system, conservative risk management and as an extra indicator.