GROUP WORK PROJECT # 3 MScFE 642: Deep Learning for Finance

GROUP NUMBER: 7634

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Chih wei Ng	Malaysia	ncw388@gmail.com	
Bharat Swami	India	bharatswami1299@gmail.com	
Pole Okuttu Stephen	Uganda	poleoks@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).	
Team member 1	**Chih wei Ng**
Team member 2	**Bharat Swami**
Team member 3	**Pole Okuttu Stephen**

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed. **Note:** You may be required to provide proof of your outreach to non-contributing members upon request.

Group Number: 7634

This report is done with reference to the Colab file.

Step 1

For this project, we selected the time series of JP Morgan's stock price. The period chosen is from Jan 1, 2017 to Dec 31, 2023 (1760 observations, on a daily basis). A simple plot of the stock price is shown below.

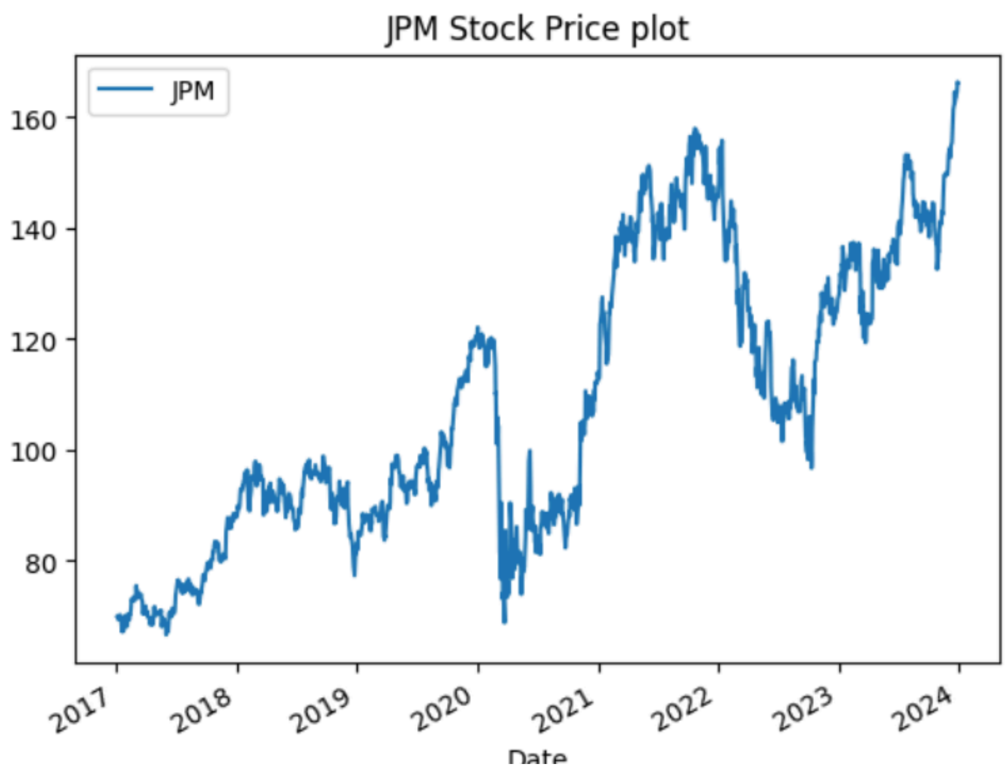

Fig 1: A plot of the time series of the stock price of JP Morgan from Jan 1, 2017 to Dec 31, 2023.

Predictive models are then built using three deep learning models, namely MLP, LSTM and CNN, using a single train/test split. The labels of the models are arranged such that there would be some information leakage between the training and test samples of the dataset. Price returns of JPM are used for the predictive models.

Group Number: 7634

MLP model

The general structure of the MLP model is as follows:

1) 3 hidden layers with 30, 20 and 10 units respectively with a single unit output layer.
2) Dropout layer of 20% after each hidden layer.
3) Optimizer is Adam, and Loss function used is Binary cross entropy.
4) Activation function is Relu for all hidden layers and sigmoid for the output layer.

For the input features, different lags were used (15 in total, from the returns over the past 2 days up till the returns over the past 30 days, skipping a day in between every 2 days). As for the output feature, the +5 days returns was used. However, to introduce leakage, the +5 days returns was calculated using the rolling past 10 days return.

The backtests were done using three strategies.

1) Long only - If the predicted return is positive, we will have a long position. If the return is zero or negative, we will not put on a position.
2) Long/short - If the predicted return is positive, we will have a long position, while if the return is zero or negative, we will put on a short position.
3) Buy and hold - We will put on a long position from the beginning of the period and hold the position till the end.

The results for the trading strategies are as below.

Strategy	Returns (%)
Long only	51.92
Long/Short	55.83
Buy and hold	44.32

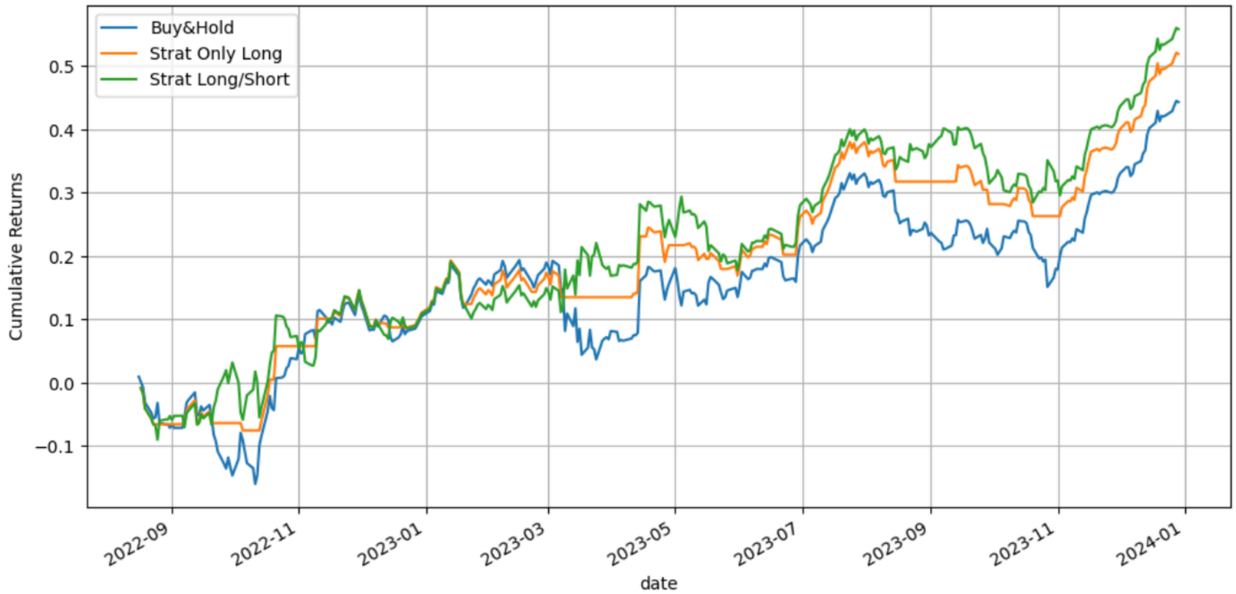

Fig 2: Plot of the cumulative returns of the different strategies over time.

<u>LSTM model</u>

The LSTM model with leakage (5 days) was used to strategize on whether to sell, buy or hold the asset (JPM). Long/short strategy was the best strategy, followed by long position and lastly the Buy and hold position as shown below. All the 3 strategies outperformed the market benchmark

Table: Results from the 3 strategies

Strategy	Returns (%)
Long	1.23
Long/Short	1.2
Buy&Hold	1.18

Group Number: 7634

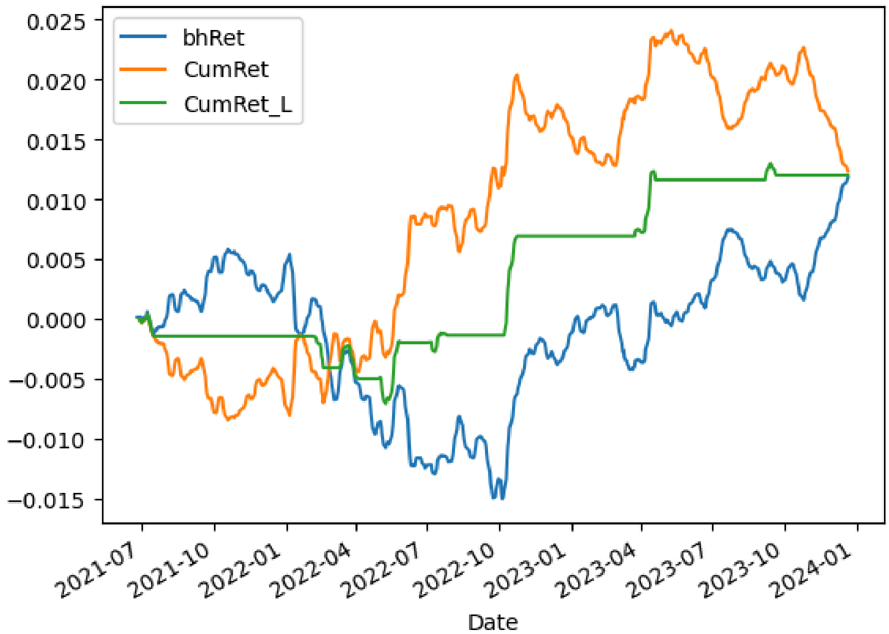

Fig 3: Plot of the cumulative returns for the 3 strategies

CNN model

Random CNN model with GAF to set the base for our best model. This model will create the basic skeleton for our main optimized CNN model.
Random model Summary:

Model : “squential”

Layer (type)	Output Shape	Parameters #
conv2d (Conv20)	(None, 15, 15, 32)	320
max_pooling2d (MaxPooling2D)	(None, 7, 7, 32)	0
conv2d_1 (Conv20)	(None, 7, 7, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 64)	0
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 128)	73856
dropout (Dropout)	(None, 128)	0

dense_1 (Dense)	(None, 1)	129

- Total params: 92,801 (362.50 KB)
- Trainable params: 92,801 (362.50 KB)
- Non-trainable params: 0 (0.00 B)

After training the model we predict the output for out text data and below are the results we get (in form of confusion matrix plot)

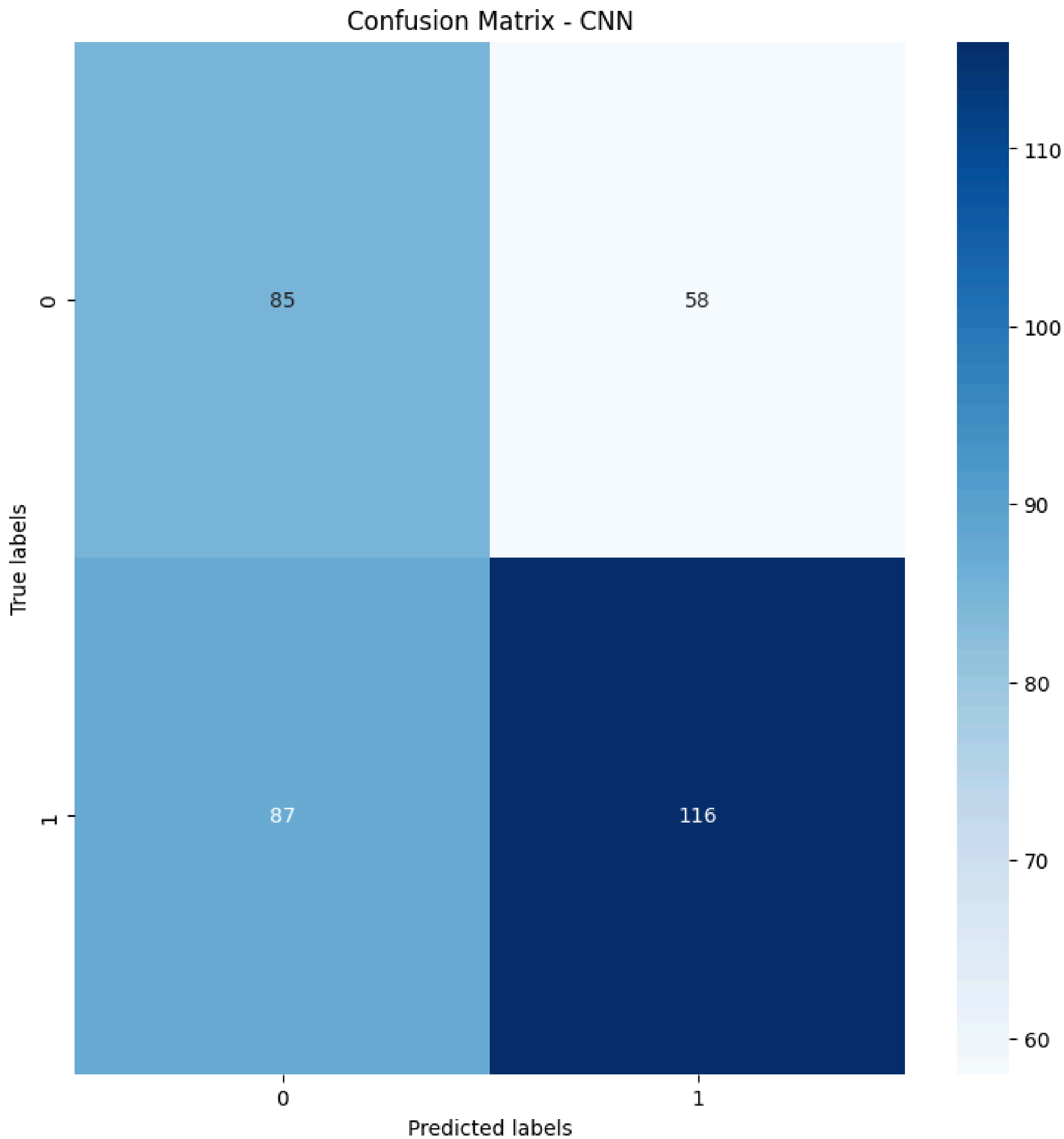

Fig 3: confusion matrix for base CNN model

As we can see from the plot it is not very accurate. Below are the results we can conclude from above plot-

- Model does predict the true labels more than false label for both class.
- Data is slightly imbalanced towards class 1 being more. Which may lead to overfitting.

But overall we can conclude we are in correct direction of CNN model with GAF.

The results for the trading strategies are as below. (we are using the same strategy as we used in step 1 part a, to compare all the models on same dataset and strategy)

Strategy	Returns (%)
Long only	44.19
Long/Short	39.39
Buy and hold	44.32

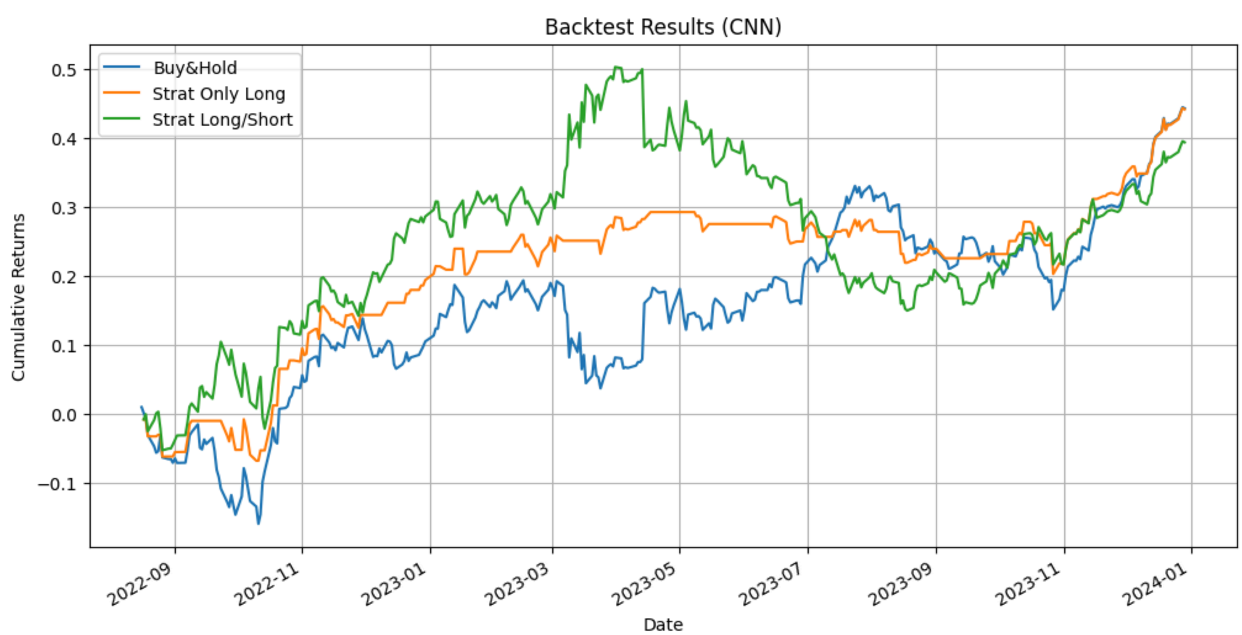

Fig 4: Plot of the cumulative returns of the different strategies over time for base CNN model

As we can see from the result our model is not very good but we can use this as a base model. Now we are going to use hyperparameter optimization (HPO) to increase our returns based on these strategies. Basically we want our two strategies, Only Long and Long/Short to be better than Buy & Hold.

Now for HPO we are going to use choices as mentioned below-

layer	Choices
Conv2D layer 1	[16, 32, 64] with l2 regularization parameters 0.0001 to 0.01
dropout layer 1	0.2 to 0.5 with step 0.1
Conv2D layer 2	[32, 64, 128] with l2 regularization parameters 0.0001 to 0.01
dropout layer 2	0.2 to 0.5 with step 0.1
Dense layer 1	[32, 64, 128] with l2 regularization parameters 0.0001 to 0.01

Add we are using Adam optimizer from tensorflow keras class, binary cross entropy loss function and accuracy as metric for compiling.

After the HPO we get following hyperparameters for our best model with above defined skeleton.

Best Hyperparameters:

- Filters for the first Conv2D layer: 32
- Filters for the second Conv2D layer: 128
- L2 Regularization factor: 0.00033
- Dropout rate for first Conv2D layer: 0.3
- Dropout rate for Dense layer: 0.4
- Units in the Dense layer: 64
- Learning rate: 0.00984

Finally we run our best model on the dataset with same split on GAF introduced dataset and below is the confusion matrix plot we get-

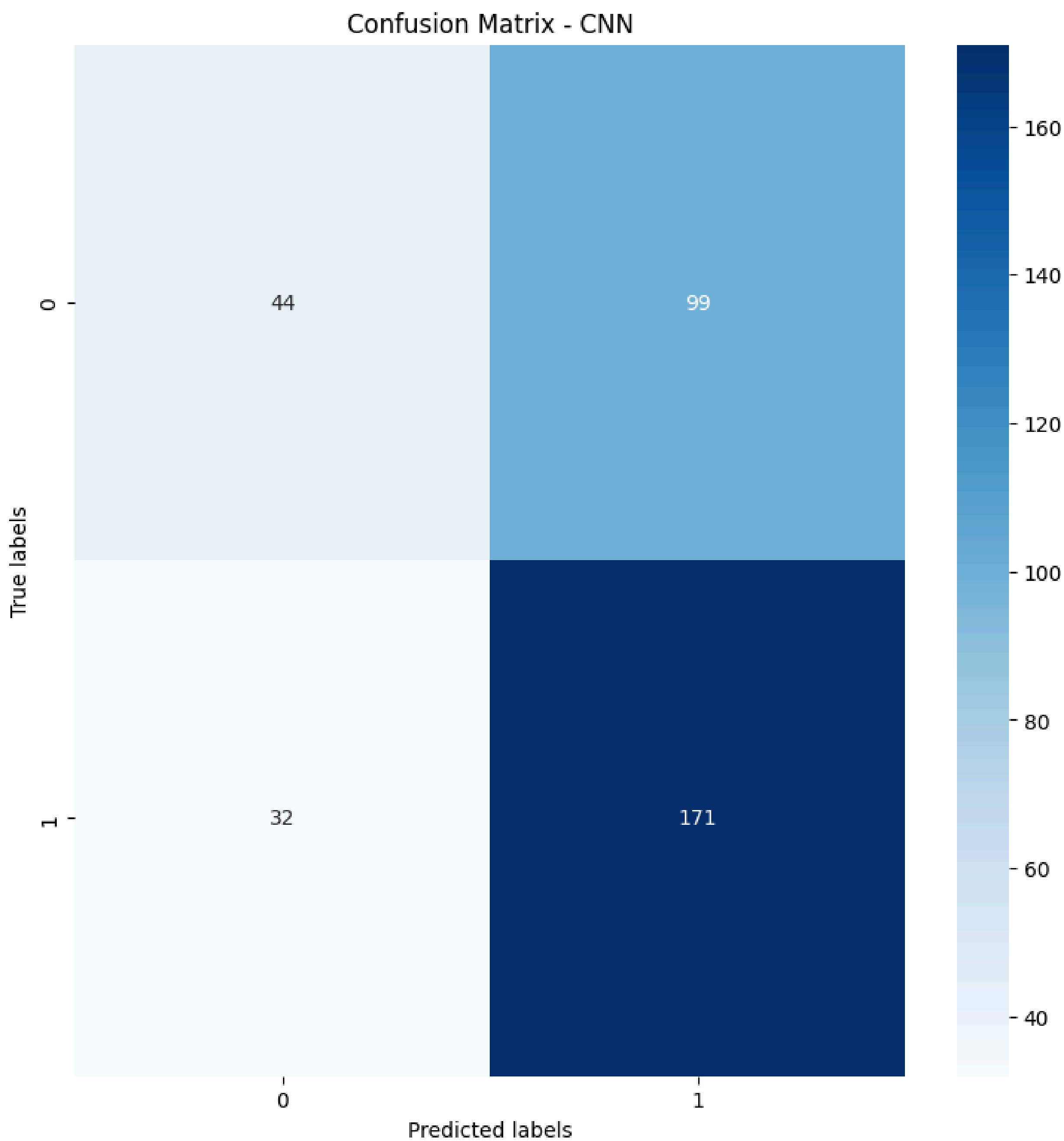

Fig 6: Confusion matrix for best CNN model after HPO

as we can see out best model did a great job on the class 1 but perform poorly on class 0. It predicts the class 1 more than then other, and this is maybe because of imbalance in out data we started with.

Group Number: 7634

The results for the trading strategies are as below. (we are using the same strategy as we used in step 1 part a, to compare all the models on same dataset and strategy)

Strategy	Returns (%)
Long only	66.30
Long/Short	89.42
Buy and hold	44.32

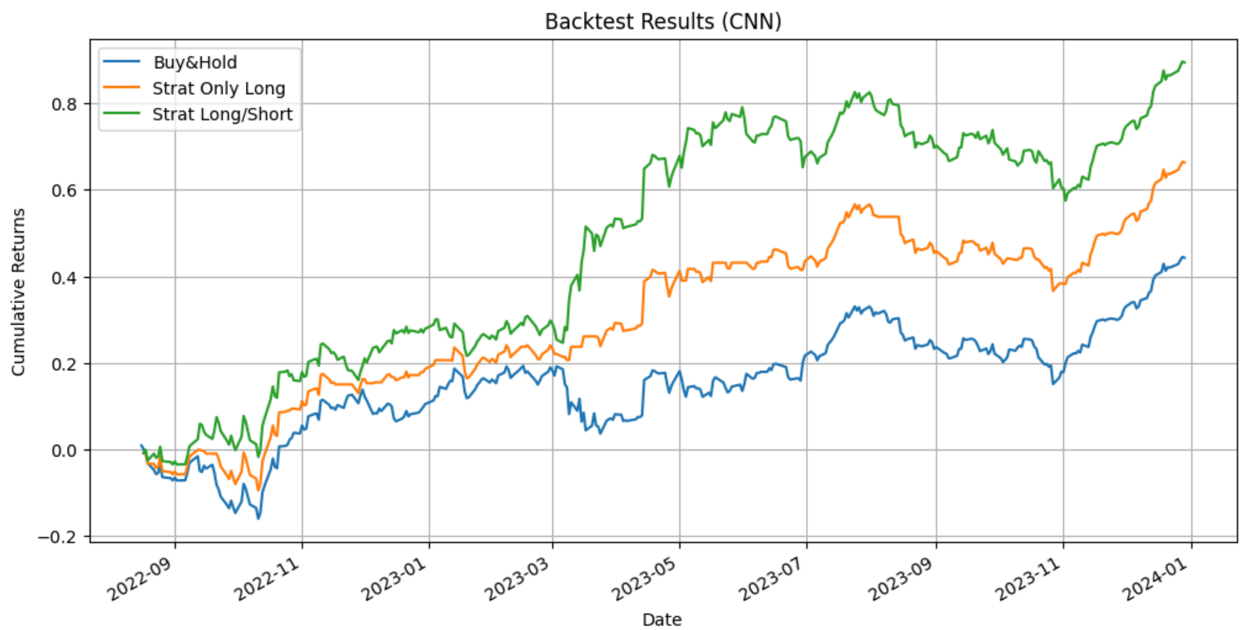

Fig 7: Plot of the cumulative returns of the different strategies over time for best CNN model after HPO

As we can see out best model perform better on our strategies which is good sign for further improvement in this direction.

Step 2

MLP model

a) Results using non-anchored walk forward method, with train/test split of 500 observations in each set.

Strategy	Returns (%)
Long only	178.64
Long/Short	176.16
Buy and hold	93.73

b) Results using non-anchored walk forward method, with train/test split of 500 observations in training set, and 100 observations in test set.

Strategy	Returns (%)
Long only	188.37
Long/Short	242.44
Buy and hold	93.73

Fig 3: Plot of the cumulative returns of the different strategies over time for 500/500 train/test split.

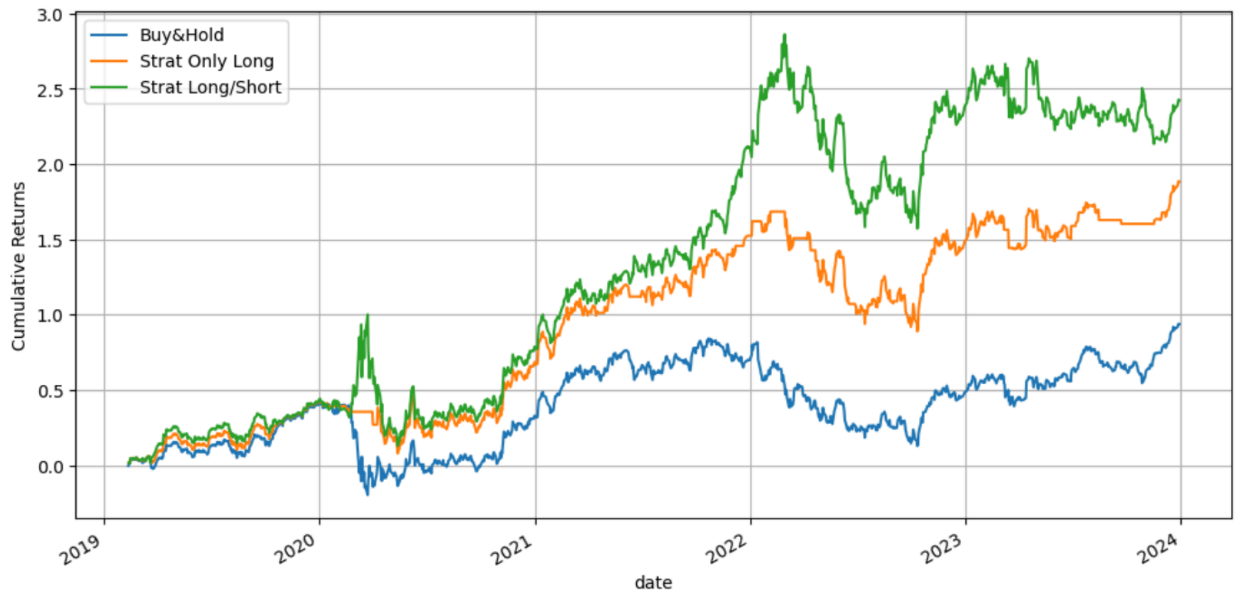

Fig 4: Plot of the cumulative returns of the different strategies over time for 500/100 train/test split

Compared to the results in Step 1, both the backtest results in parts a and b in this step are superior. One of the reasons could be that in Step 1, we only did a single train/test split of the dataset, which limits the amount of information that the model can learn from both the training part, and the test part. The single train/test split also biased our test results on a certain part of the time series (the latest part).

The results in Step 2 are superior, with one of the reasons being that the walk forward method was used for training/testing. In this way, different sequences of the dataset were used for training and testing, using a rolling walk forward time window. The walk forward was done in one direction though, which retains the time sequence of the dataset. As a result, a much higher portion of the dataset could be used for testing, which likely increased the robustness while decreasing the potential bias towards the part of data at the end of the time period (as in the single train/test split in Step 1). Therefore, better results were obtained.

As for the comparison between parts a and b of Step 2, we can see that the results in part b are better than those in part a (other than the buy and hold strategy, which is expected to be the same). It is possible that there could have been backtest overfitting due to leakage, as leakage was intentionally introduced in the training/test data, just as in Step 1. Another reason could be that as the walk forward window for part b is smaller than that of part a, it helped to make the model more robust, due to the higher number of training/testing during the walk forward.

Group Number: 7634

LSTM model

a)

Strategy	Return (%)
Long	0.0
Long/Short	-1.16
Buy and Hold	1.18

b)

Strategy	Return (%)
Long	0.0
Long/Short	-1.16
Buy and Hold	1.18

CNN model

Now in this part we are continuing using the CNN model we HPO in step 1.

a) Results using non-anchored walk forward method, with train/test split of 500 observations in each set.

Strategy	**Returns (%)**
Long only	63.71
Long/Short	22.77
Buy and hold	93.73

Group Number: 7634

b) Results using non-anchored walk forward method, with train/test split of 500 observations in training set, and 100 observations in test set.

Strategy	Returns (%)
Long only	73.02
Long/Short	44.22
Buy and hold	93.73

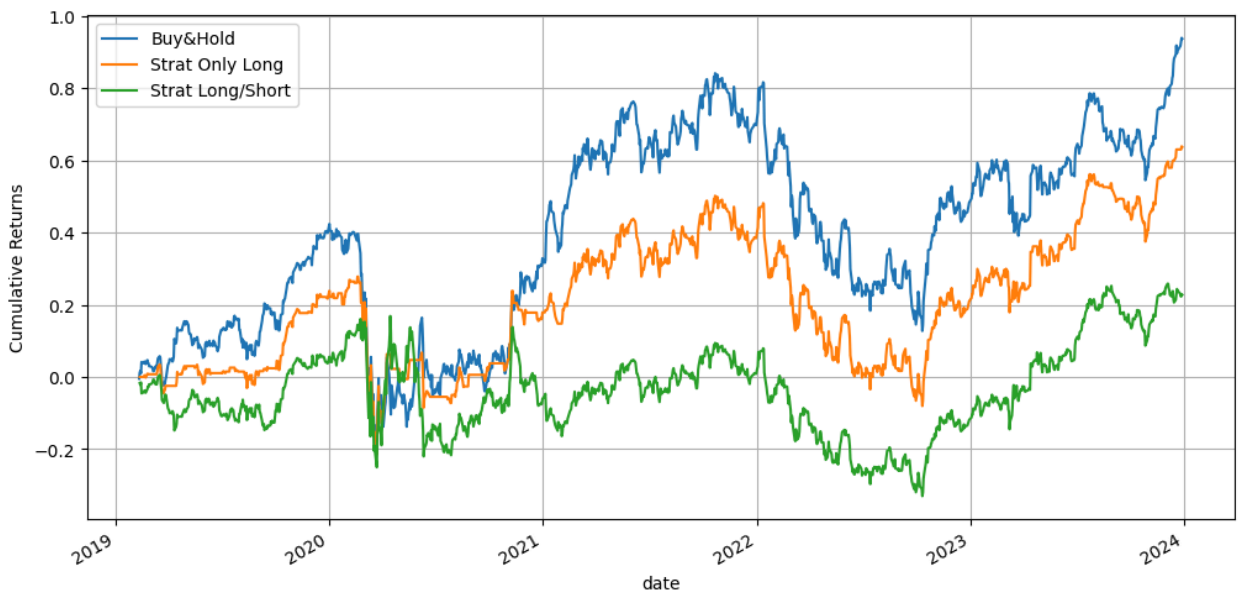

Fig 9: Plot of the cumulative returns of the different strategies over time for 500/500 train/test split for CNN model

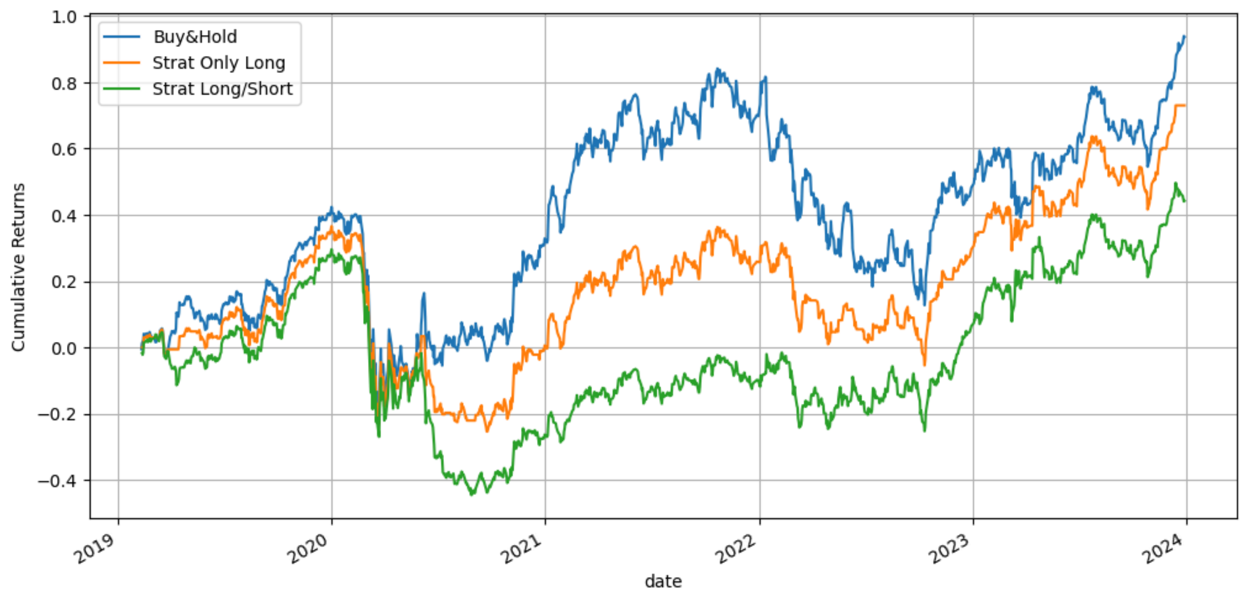

Fig : Plot of the cumulative returns of the different strategies over time for 500/100 train/test split for CNN model

As we can see from the results, returns for the part b are better than part a in CNN model where test split are 100 observation in b as compared to a with 500 observations. We can also see that in part a model does not perform better than step 1 model for strategy "Long only" but in part b it perform better. And as of strategy "Long/Short" model did not perform better than step 1.
And in this step 2, if we compare our model to MLP model it perform very worse. Returns are no where near to the MLP results.
The reason for this behavior of our CNN model is because of overfitting on the imbalance data. We choose the best performing model based on accuracy metric with binary cross entropy loss function. which from results gives best accuracy but on cost of overfitting and which is not desired at all. So in future we must work on a overfitting and imbalance class problem because it lead to very poor growth of next step complex models.

Group Number: 7634

Step 3

MLP model

The same process as in Step 2 was done, but with the leakage purged. This was done by using the 20 days forward prediction as the output, but with the forward prediction using only the prices of the previous 10 days. This means that there is a gap of 10 days between the end of the training period and the beginning of the test period.

b) Results using non-anchored walk forward method, with a train/test split of 500 observations in each set.

Strategy	Returns (%)
Long only	62.50
Long/Short	0.25
Buy and hold	93.73

c) Results using non-anchored walk forward method, with a train/test split of 500 observations in the training set, and 100 observations in the test set.

Strategy	Returns (%)
Long only	122.54
Long/Short	75..34
Buy and hold	93.73

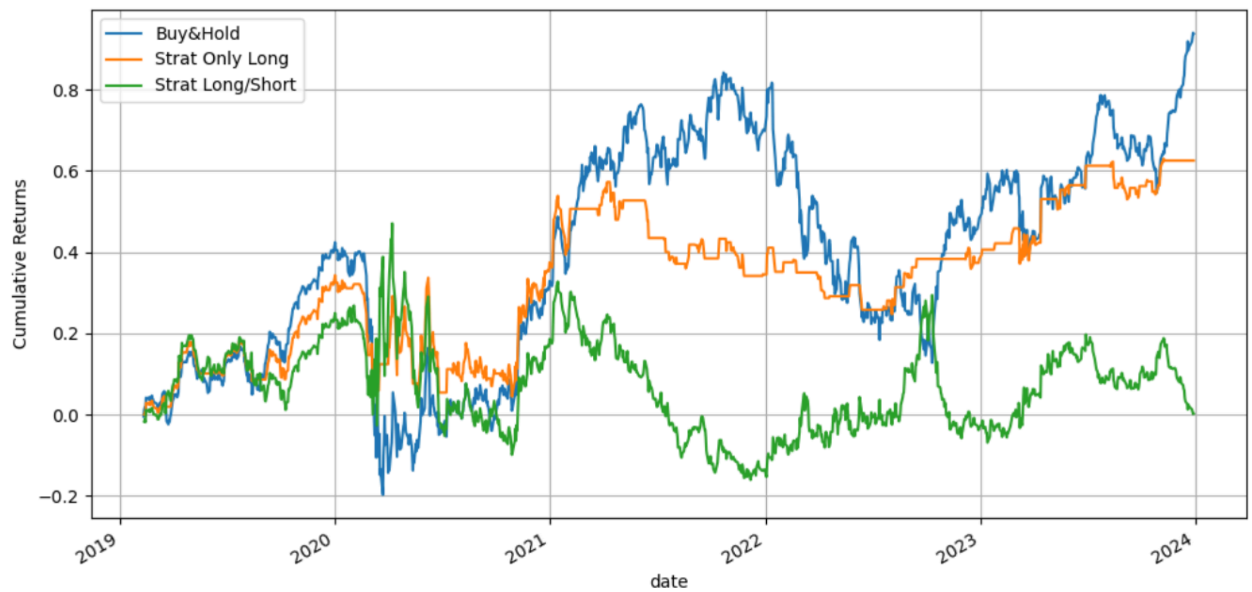

Fig 5: Plot of the cumulative returns of the different strategies over time for 500/500 train/test split

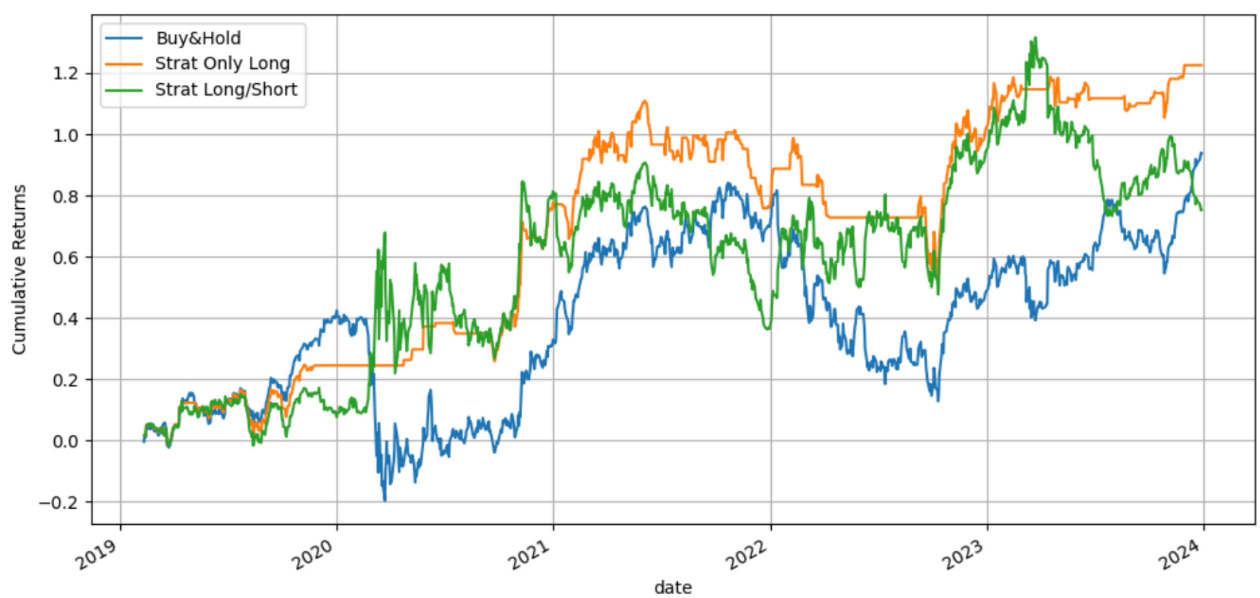

Fig 6: Plot of the cumulative returns of the different strategies over time for 500/100 train/test split

Similarly to Step 2, when we compare the results of the 500 train/test and 500/100 train/test within this step itself, the 500/100 results are superior to the 500 train/test. The reasons are similar to the ones given in Step 2.

However, when we compare the results in Step 3 against those in Step 2, we can see that those in Step 3 are not as good as those in Step 2 (other than the buy and hold strategy). This is likely because the leakage has been purged, thus reducing the overfitting in the backtests.

Group Number: 7634

LSTM model

a)

Strategy	Return (%)
Long	0.0
Long/Short	-1.16
Buy and Hold	1.18

b)

Strategy	Return (%)
Long	0.0
Long/Short	-1.16
Buy and Hold	1.18

CNN model

a) Results using non-anchored walk forward method, with train/test split of 500 observations in each set.

Strategy	Returns (%)
Long only	63.26
Long/Short	27.69
Buy and hold	93.73

b) Results using non-anchored walk forward method, with a train/test split of 500 observations in the training set, and 100 observations in the test set.

Strategy	Returns (%)
Long only	47.65
Long/Short	-5.01

Buy and hold	93.73

Again we have same result as of step 2 as compared to MLP model. As we introduced the leakage in our data we expected a better performance from CNN model as compared to MLP model but here they did not perform as expected with the same reason of overfitting and imbalance data.

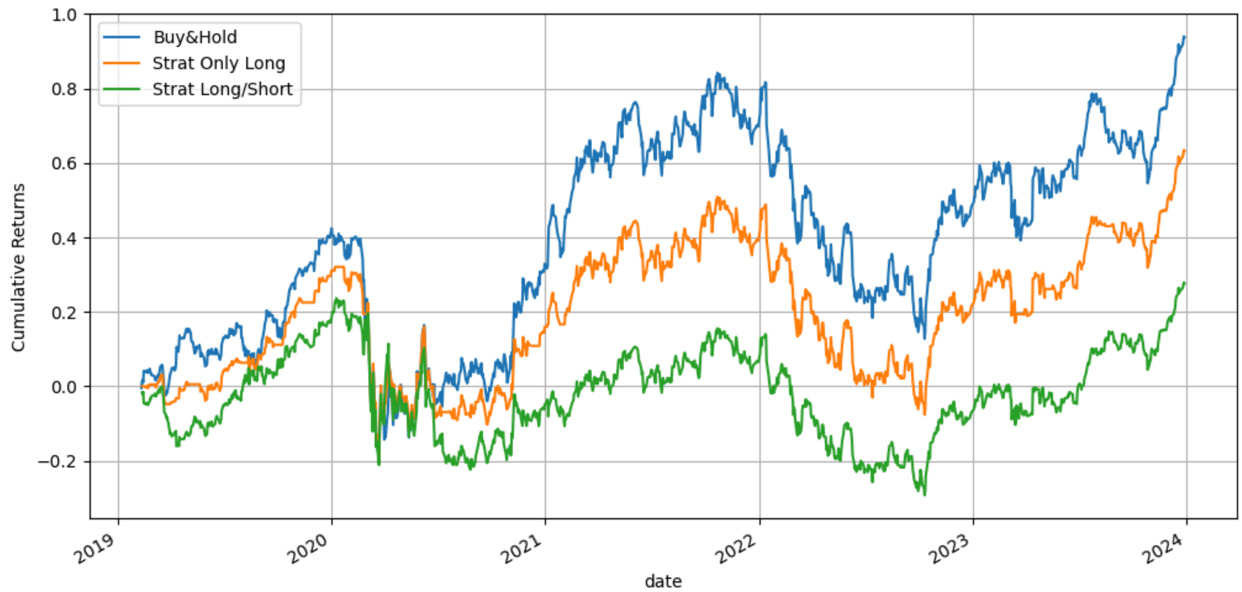

Fig : Plot of the cumulative returns of the different strategies over time for 500/500 train/test split for CNN model with more leakage as compared to step 2 CNN model part.

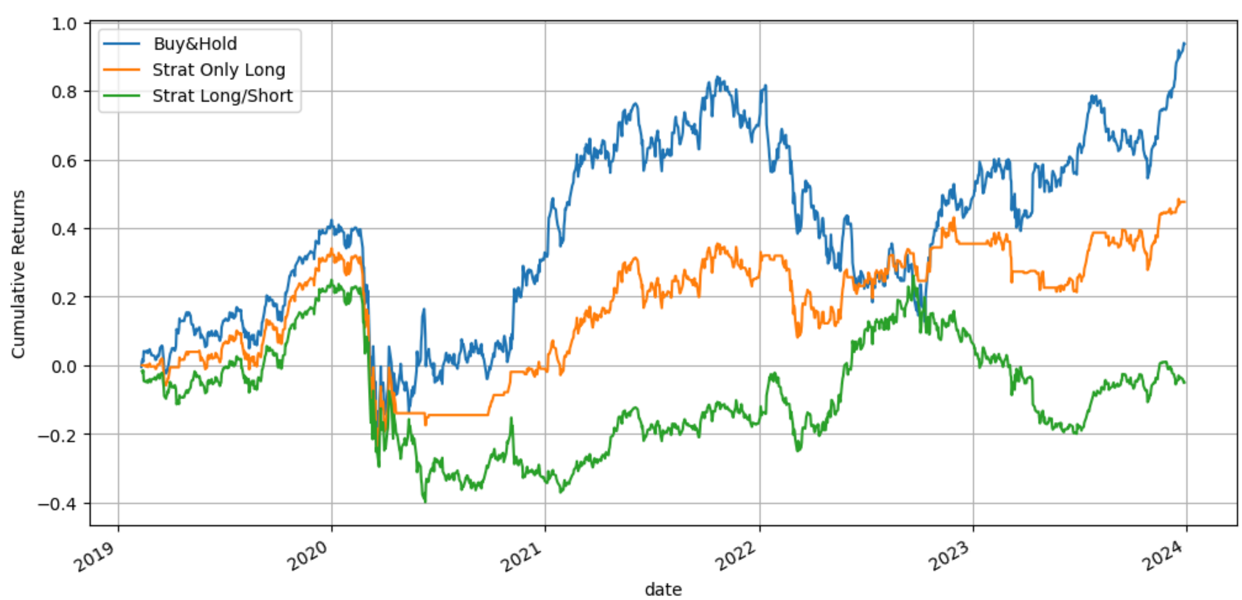

Fig : Plot of the cumulative returns of the different strategies over time for 500/500 train/test split for CNN model with more leakage as compared to step 2 CNN model part

Group Number: 7634

References

1. Gray, Chad. "Stock Prediction with ML: Walk-forward Modeling." *The Alpha Scientist*, 18 July 2018, https://alphascientist.com/walk_forward_model_building.html.