I modified my data prep function to allow for more data columns to be produced and hopefully allow for a more accurate prediction.

I started by scaling all data rather then just the price column

```
# transforming the data to the required format
for column in price_column:
    data[column] = scaler.fit_transform(data[column].values.reshape(-1, 1))
```

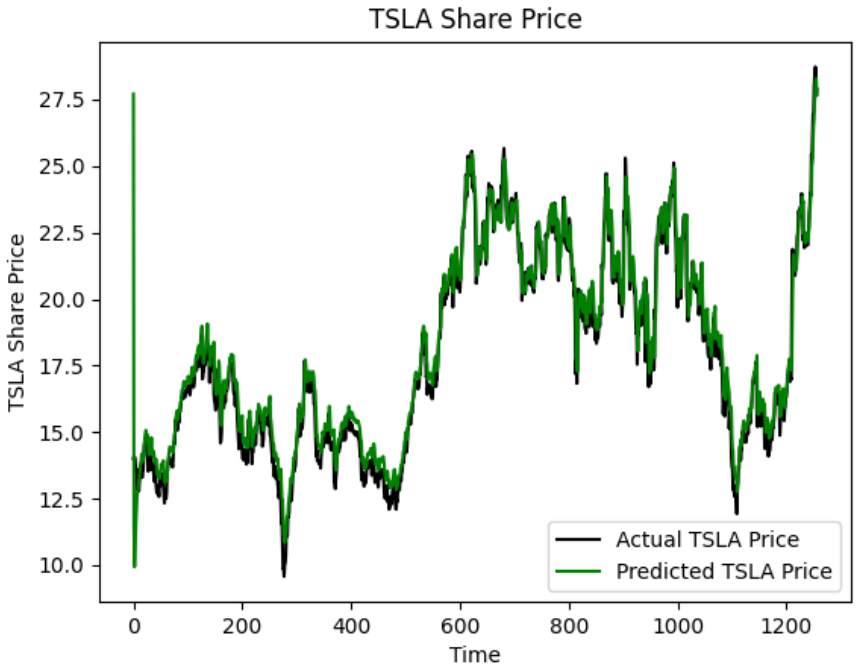
And then when training I added each collum to the training arrays.

```
# training arrays predicts the next point of sequence x and stored in y
x_train = []
y_train = []
# loops through training data
for column in price_column:
    for idx in train_indices:
        # creates training sqeuence
        x_train.append(data[column][idx:idx + prediction_days])
        # target value for training sequence
        y_train.append(data[column][idx + prediction_days])
```

I also added the ability to predict k days into the future at the end of the code simply looping through the number of days predicted.

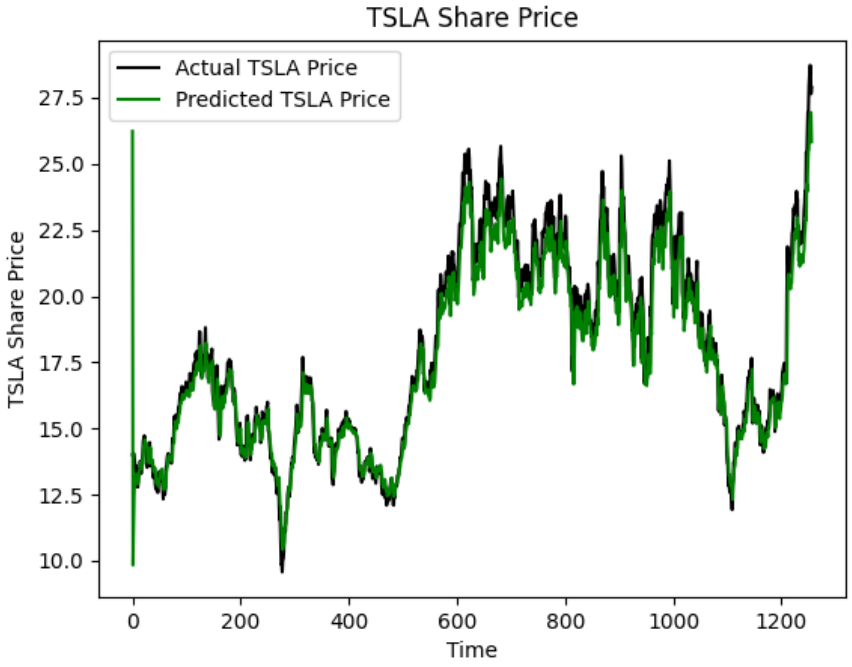
```
k=5

for x in range(k):
    real_data = [model_inputs[(len(model_inputs) - PREDICTION_DAYS+x):, 0]]
    real_data = np.array(real_data)
    real_data = np.reshape(real_data, newshape: (real_data.shape[0], real_data.shape[1], 1))

    prediction = model.predict(real_data)
    prediction = scaler.inverse_transform(prediction)
    print(f"Prediction: {prediction}")
```

I ran similar tests as I did for testing just the closing price column and here are the results, theme is that it performs slightly worse then just the price column overall

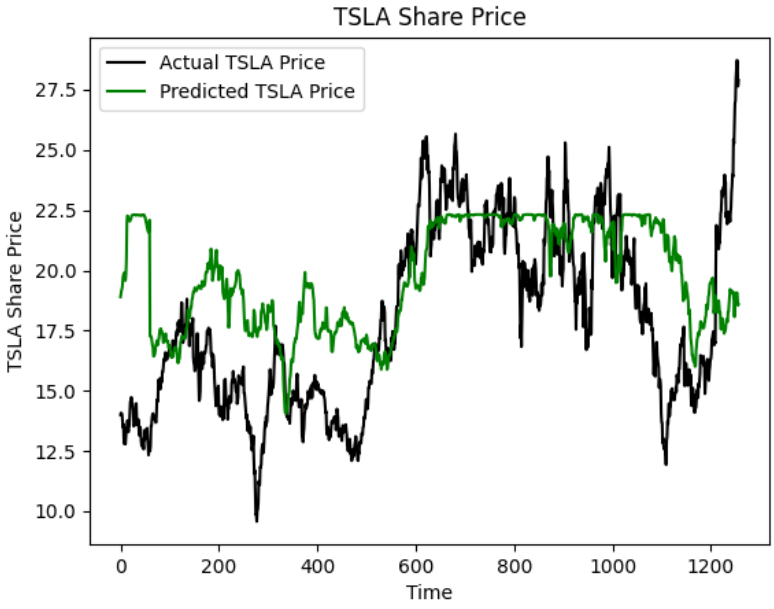


this is GRU with 25 epoch batch of 25, 256 units and 2 layers
is fairly accurate but the first half is noticeably less accurate then the same parameters with only the price column

LSTM shares a similar story with being slightly worse then its close column counter part

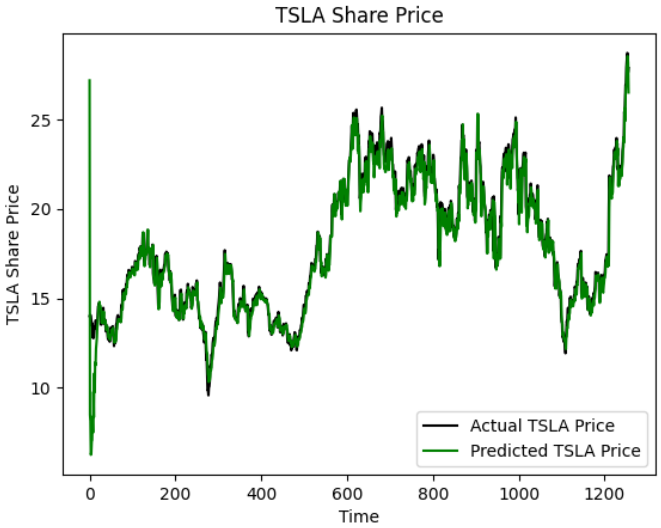


And simplernn is the only model to improve, which is used lightly since it essentaly didn't make a prediction with only the close column.



I then similarly took the best performing cell type (GRU) and tested it with various settings. Each test took considerably longer due to having more data. And still having the similar result of less accuracy

While performing well 50 epoch still didn't quite perform as well as hoped performing worse initaly and still having much more of the actual line visible



And similarly 100 epoch performs ok the start once again being inaccurate but does seem to perform better towards the end, but over predicts the results sightly where as only price column under predicted. Indicating that the model needs some changes to be equipped for the extra data it receives. Meaning the rest of the tests are likely to have the same conclusion