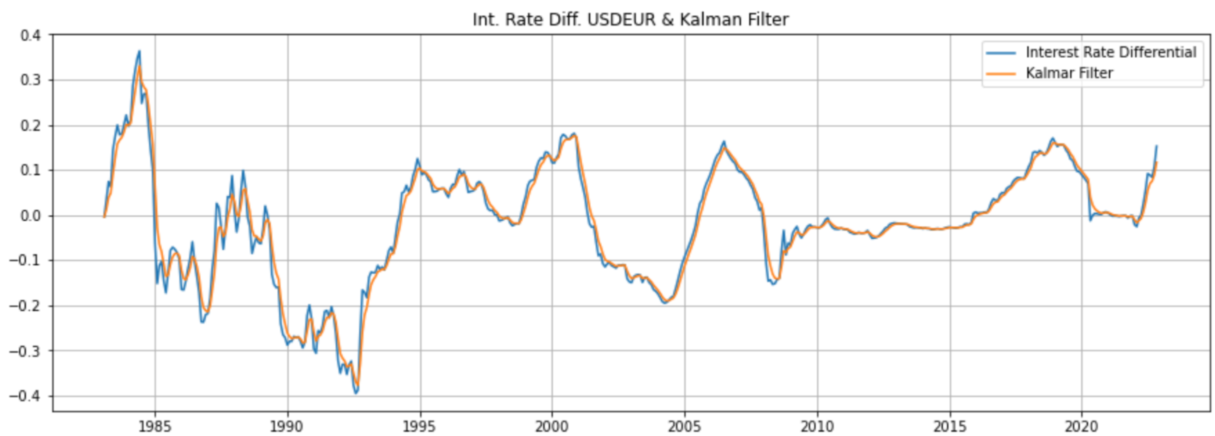
Quentin Bidwell, Cade Foster, Hayden Johnson

Abstract: Our group implemented an ARIMA forecasting model that used the interest rate differential theory with a Kalman filter to predict surprises that would affect the exchange rate between the US Dollar and British Pound. In addition, our group also constructed a Long Short Term Memory (LSTM) neural network machine learning model that used all provided features to forecast the same exchange rate. Comparing the two models, the LSTM performs significantly better than the ARIMA in forecast accuracy and also profit generation from trading strategies derived from the respective forecasts.

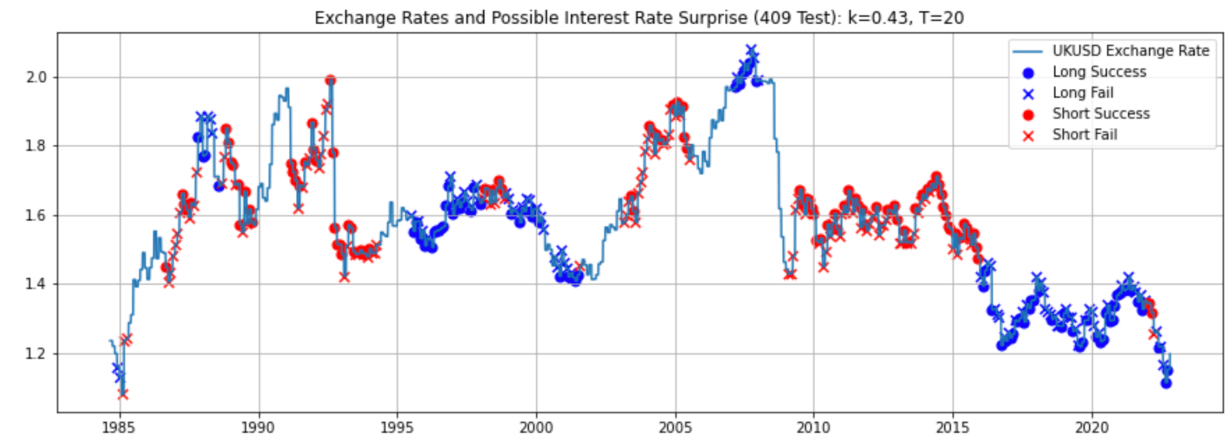
ARIMA Implementation:

For the first trading strategy, we calculated the interest rate differential between the United States and the United Kingdom. Based on economic intuition, we constructed a Kalman filter that fit to the interest rate differential to potentially reduce noise in the signal.



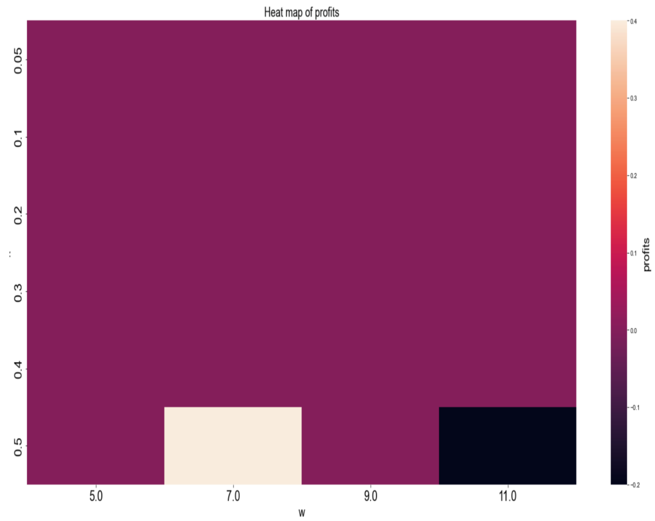
For our initial model,we chose arbitrary values for the Kalman filter, k. Similarly, when building our autocorrelative function we initially chose arbitrary values for the length of the rolling window, T, and the moving average would be bound by as well as an arbitrary value for the statistical significance threshold of the mean predicted by the moving average, z. Our arbitrary values are as follows: k = 0.43, z = 1.645 and T = 20.

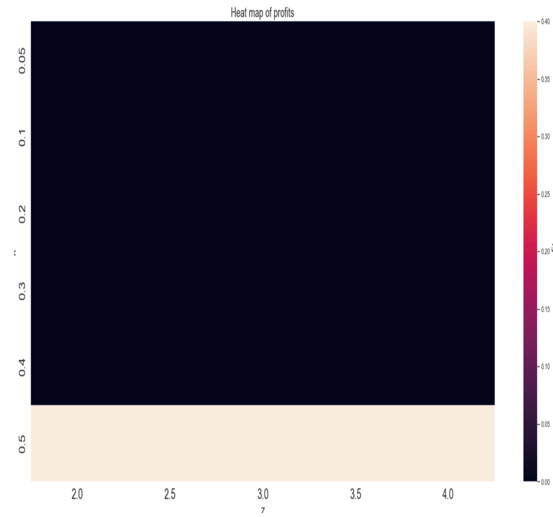
Next we constructed a trading strategy where we entered a long position when a statistically significant forecast projected a positive change for the next period, a short position when the forecast was negative, and no change when there was not a statistically significant forecast. Evaluating this initial model, we could easily see that the strategy was not particularly successful, mostly due to the arbitrary assignment of our hyperparameters.

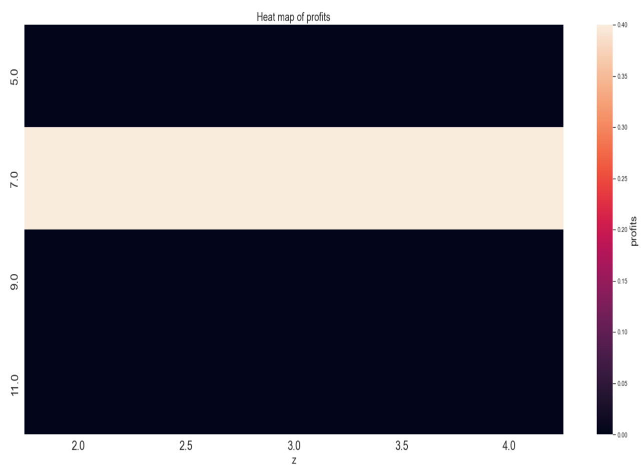


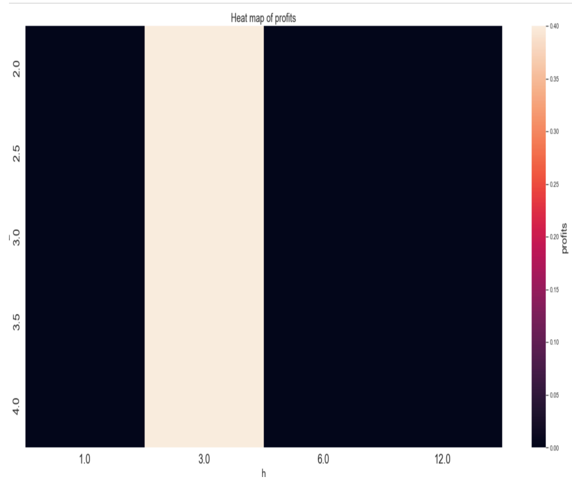
Therefore, our next step was to iterate over the ARIMA model in order to fit our hyperparameters to the time series.

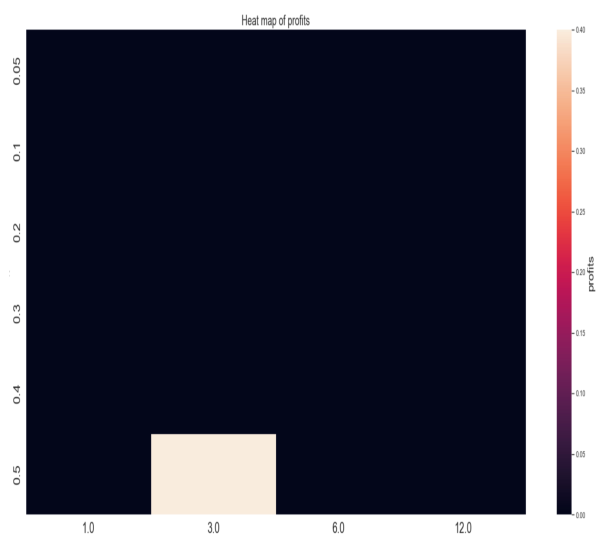
For this interaction, we generated a range of values and updated our hyperparameters to include k,w,z, and h where our target variable for optimization was the profit generated by the trading strategy. The hyperparameters are k, the Kalman filter weight – w, the number of periods over which to evaluate the std – h, the holding period defined for our strategy – and t, the time period over which we evaluate our strategy. Our iteration ran 480 different permutations to estimate the optimal selection. Below we visualized different combinations of our hyperparameters over our tested ranges by generating heatmaps that are keyed to the strategy profits:



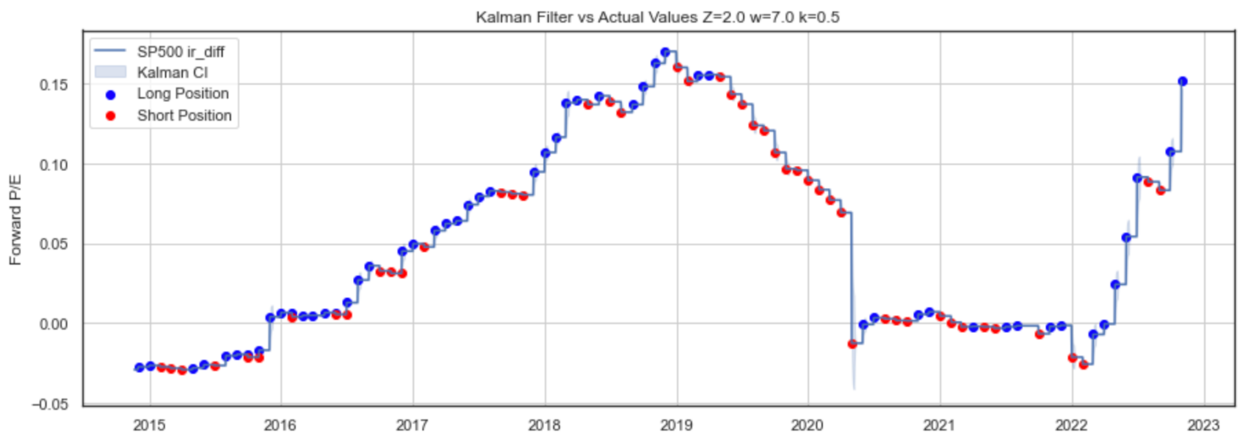








Once realized, we were able to update our forecasting model and trading strategy to include the optimized hyperparameters and generate what would be our final trading strategy and resulting equity curve. The optimal hyperparameters are k = 0.5, z = 2.0, w = 7.0, h = 3.0 and the maximized profits are 0.40.



The trading strategy started off well in 2016, but unfortunately took a deep dive halfway through 2016, never recovering to a positive price earnings ratio. Furthermore, we created a binomial test:

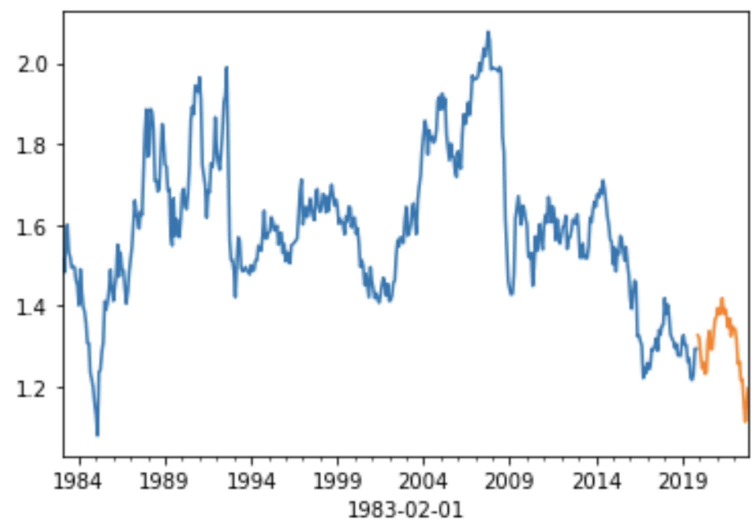
$$H_0 : Cov(D_{t,t+h}, R_{t,t+h}) = 0 \quad H_1 : Cov(D_{t,t+h}, R_{t,t+h}) > 0$$

Test statistic: 21.827819297802534 , 5 % critical value: 1.64
As the test statistic is below the 5% alpha critical value threshold, we can conclude that our model does not pass the binomial test and thus we can conclude that our directional forecasts are uncorrelated with our realized directional changes from our investment strategy. This conclusion shows that our strategy does not significantly capture the realized appreciation or depreciation of U.S/U.K interest rates.

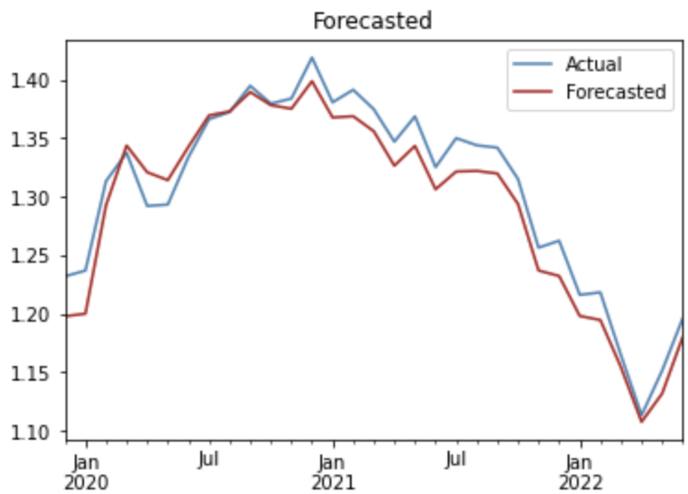
LSTM Implementation:
Our group was disappointed by the performance of our ARIMA model trading strategy so we wanted to follow up with a model that approached the problem from a different forecasting perspective. Neural networks are one of the most popular and powerful machine learning models and have seen significant increases in accessibility in recent years.

For this model, we chose to create a Long Short Term Memory neural network using the Keras machine learning library. When forecasting time series, creating a normal recurrent neural network will result in a vanishing or exploding gradient problem, which will render the forecast unusable. LSTM models solve this problem because the memory cell design allows for partial retention of gradient components, which allows us to generate a forecast for time series with long sequences.

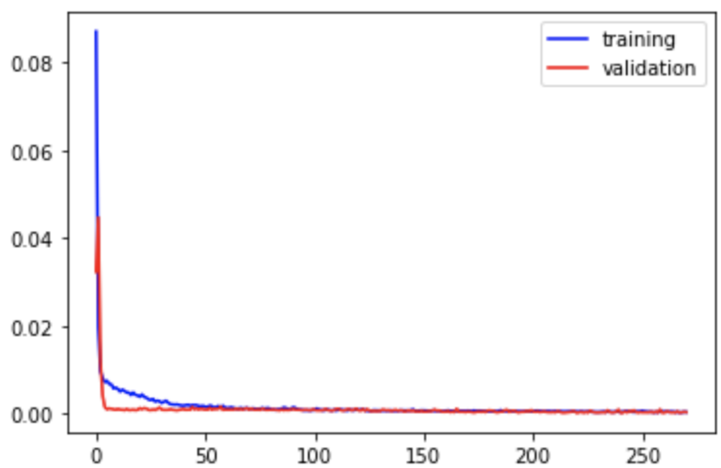
Another strength of this approach is that neural networks are data hungry models, so we could feed it not just one economic component of the data but all 6 indicators: country interest rates, country inflation rates, and country unemployment rates for both the United States and the United Kingdom. However, an equivalent weakness is that we were not able to include some evaluation of which of these indicators were most influential because neural network design does not allow for the typical interpretation of coefficients that is possible with modeling types like regression. Another limitation is the test data length. We wanted to maximize the data available to feed into the neural network to have the best chance of producing a robust forecast while still preserving enough data for the testing set to be dynamic enough for a realistic trading strategy use case. Our end decision was to only implement the forecasting model and trading strategy over the past couple years. Below you can see the training data and then the test data in orange that the forecast and trading strategy was executed over.



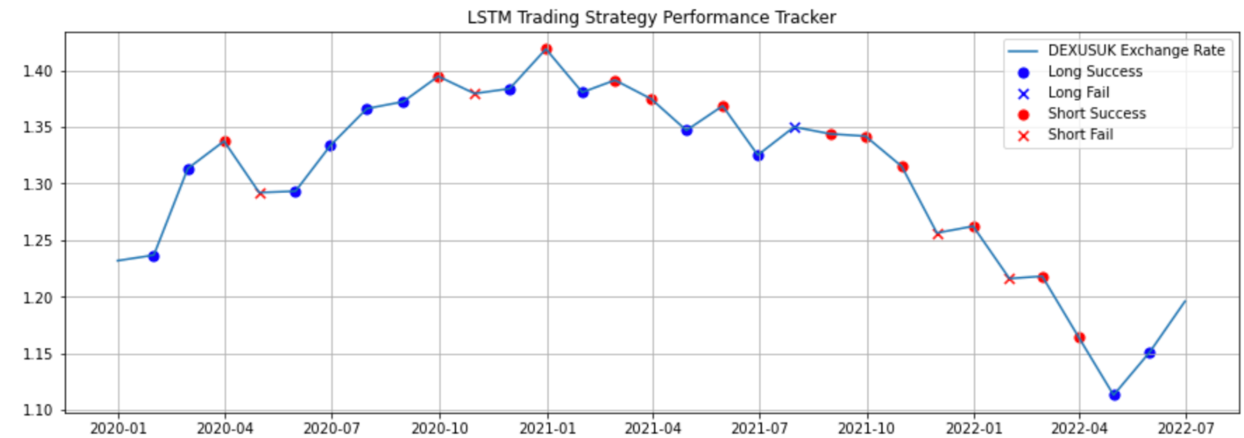
To preserve space, we will avoid going into depth about the neural network architecture but it was creating using tensorflow keras and used 6 month context windows generated from the time series to predict month in the future, resulting in the subsequent forecast history:



We can also see that the model has high out of sample accuracy by observing the loss function convergence and performance on the training and test samples:



We implemented a trading strategy in which when the next period was forecasted to have a positive percent change, we would enter into a long position. If the next period was forecasted to have a negative percent change, we would enter into a short position. This strategy produced the following trading history and equity curve:



Another notable element of the trading history is that typically in the incorrect trades, the magnitude of the mistake was much smaller than the magnitude of profitable trades. We also conducted the binomial test for our LSTM model:

$$H_0 : Cov(D_{t,t+h}, R_{t,t+h}) = 0 \quad H_1 : Cov(D_{t,t+h}, R_{t,t+h}) > 0$$

```
Test statistic :  0.6640804421910136 , 5 % critical value :  1.64
```

As the test statistic is below the 5% alpha critical value threshold, we can conclude that our model does not pass the binomial test and thus we can conclude that our directional forecasts are uncorrelated with our realized directional changes from our investment strategy. This conclusion shows that our strategy does not significantly capture the realized appreciation or depreciation of U.S/U.K interest rates.

HFRI Macro: Currency Index Performance

	B1	B2	B3	HFRIMCUR	**ARIMA**	**LSTM**
Geo. Average	0.31	0.15	0.72	0.13	-0.003	-0.001

Monthly (%)						
Std. Deviation (%)	1.91	1.59	4.71	1.00	0.01	0.02
High Month (%)	5.86	3.91	12.82	3.43	0.06	0.06
Low Month (%)	-9.08	-7.63	-16.80	-2.18	-0.09	-0.005
Annualized Return (%)	3.73	1.82	8.97	1.51	-0.48	26.06
Annualized Std. Deviation	6.63	5.50	16.31	3.46	0.08	0.06
Risk Free Rate (%)	0.68	0.68	0.68	0.68	1.13	0.55
Sharpe Ratio	0.48	0.23	0.57	0.25	-1.02	1.08
% of Winning Months	62.64	62.64	66.48	53.30	44.33	82.76
Max Drawdown (%)	20.05	20.11	46.39	8.39	0.12	0.01
Alpha	0.16	0.14	0.18	-	0.00	1.49
Beta	-0.08	-0.08	-0.06	-	1.01	-0.26
Mtn. R-Squared	0.02	0.02	0.07	-	0.83	0.02
Correlation	-0.15	-0.13	-0.27	-	0.91	-0.13
Up Alpha	0.14	0.19	0.19	-	0.01	1.73
Up Beta	-0.04	-0.08	-0.05	-	1.03	-1.33
Up R-Squared	0.00	0.00	0.02	-	0.79	0.08
Down Alpha	-0.05	-0.10	-0.16	-	-0.01	1.18
Down Beta	-0.17	-0.19	-0.12	-	1.00	-0.39
Down R-Squared	0.07	0.0.7	0.11	-	0.88	0.04

B1 HFRI FUND WEIGHTED COMPOSITE INDEX
B2 HFRI FUND OF FUNDS COMPOSITE INDEX
B3 S&P 500 INDEX TOTAL RETURN

CONCLUSION

One element of our analysis that should be noted is that the LSTM forecast had a much smaller range of test data, whereas we were able to track the interest rate differential ARIMA performance across the full history. As a result, we were unable to see how the LSTM model would have fared in extreme events such as 2008, and some may argue that our execution window prevents us from making a complete evaluation of the LSTM model's performance, much less a fair comparison.

However, despite this limitation, looking at the visualization plots and HFRI metrics, it is clear that the LSTM model was a resounding success. The forecast is extremely robust and tracks the true exchange rate closely. The trading strategy not only produces favorable profit but has low loss on the periods in which the trades failed, has slightly negative correlation with the market which is valuable for diversification, and had consistent and well distributed returns. Our group would fully endorse the LSTM model as an economic trading strategy over the interest rate differential ARIMA strategy, and this project model is potentially not even the most accurate LSTM forecast possible. Our short, sub-2000 row dataset with a limited number of features is small relative to standard machine learning sets, and an LSTM forecast is known to shine when the dataset and feature-set is large. Additionally there are more hyperparameters we could introduce into the LSTM model as we did the ARIMA model such as context window and trading threshold, both of which we saved for future research as neural net generation can be computationally expensive.