BT2101 TURNOVER PREDICTION SUMMARY

Lim Jin Ming, Jeremy Denzel (A0172720M)
Cai Shuhang (A0167788A)
Kang Choon Kiat (A0171128L)

NOVEMBER 24, 2018

TEAM 10

Table of Contents

1. Overview

This report aims to summarize the results from backtesting the model on the testing period from November 7th to November 21st and comparing it to the results from the live model.

2. Live Results

1	Deploy	Time	Symbol	Price	Quantity	Type	Status	Value	Tag
2	L-2f0c24b	2018-11-10T09:50:50.000199Z	AUDUSD	0	-49902	Market	7	0	Liquidated
3	L-5264772	2018-11-10T10:31:27.000665Z	AUDUSD	0	-49902	Market	7	0	Liquidated
4	L-d10d374	2018-11-12T01:01:54.000878Z	EURUSD	1.13275	63827	Market	Filled	72300.03	
5	L-f0078c5	2018-11-12T02:31:26.00031Z	EURUSD	1.1327	1	Market	Filled	1.1327	
6	L-c134165	2018-11-12T03:31:33.001756Z	EURUSD	1.13309	19	Market	Filled	21.52871	
7	L-b1094e6	2018-11-12T07:02:32.000609Z	EURUSD	1.12771	-113	Market	Filled	-127.431	
8	L-b1094e6	2018-11-12T08:00:27.000428Z	EURUSD	1.12577	-63734	Market	Filled	-71749.8	
9	L-b1094e6	2018-11-12T08:31:44.000413Z	EURUSD	1.1248	63734	Market	Filled	71688	
10	L-b2af3dd	2018-11-12T15:31:42.000072Z	EURUSD	1.12629	-404	Market	Filled	-455.021	
11	L-5aa5103	2018-11-13T04:31:31.000655Z	EURUSD	1.12464	38	Market	Filled	42.73632	
12	L-6b16237	2018-11-14T01:02:00.000964Z	EURUSD	1.13044	-127285	Market	Filled	-143888	
13	L-6c55857	2018-11-14T11:02:01.000257Z	EURUSD	1.1268	127784	Market	Filled	143987	
14	L-8e455f0	2018-11-15T03:31:48.000712Z	EURUSD	1.13271	411	Market	Filled	465.5438	
15	L-9152b04	2018-11-15T15:31:41.004208Z	EURUSD	1.13157	-220	Market	Filled	-248.945	
16	L-f82798c	2018-11-15T16:31:51.001226Z	EURUSD	1.13124	-57	Market	Filled	-64.4807	
17	L-c12fe53	2018-11-16T03:01:52.000256Z	EURUSD	1.13347	44	Market	Filled	49.87268	
18	L-6c1c3e3	2018-11-16T15:32:09.000709Z	EURUSD	1.14007	190	Market	Filled	216.6133	
19	L-6c1c3e3	2018-11-16T15:40:04.000266Z	EURUSD	1.13949	-64235	Market	Filled	-73195.1	
20	L-6c1c3e3	2018-11-16T16:02:19.000924Z	EURUSD	1.13923	-64224	Market	Filled	-73165.9	

BT2101		
Team	Equity_20Nov	Abs Returns (%)
Group_14	103015	3.02
Group_11	102133	2.13
Group_23	101021	1.02
Group_18	100866	0.87
Group_10	100733	0.73
Group_13	100550	0.55
Group_17	100300	0.30

Figure 1: Live Results

The model managed to earn a net profit of $733 during the period that it was deployed live. Although there were several bugs along the way (as observed in the log), they were minor, and the model managed to make an overall profit.

<u>3. Backtest Results</u>

<u>Figure 2: Backtest Results</u>

Overview | Report | Trades | Logs | Code | Share

Overall Statistics — Download Results

Total Trades	9	Average Win	0.26%
Average Loss	-0.18%	Compounding Annual Return	-1.929%
Drawdown	0.600%	Expectancy	0.216
Net Profit	-0.080%	Sharpe Ratio	-0.345
Loss Rate	50%	Win Rate	50%
Profit-Loss Ratio	1.43	Alpha	0.055
Beta	-4.662	Annual Standard Deviation	0.04
Annual Variance	0.002	Information Ratio	-0.717
Tracking Error	0.04	Treynor Ratio	0.003
Total Fees	$18.00		

Rolling Statistics — Sharpe Ratio

	1 Month	3 Months	6 Months	12 Months
11/22/2018	-0.34489310264016	-0.34489310264016	-0.34489310264016	-0.34489310264016

<u>Figure 3: Overall Statistics</u>

The model made a net profit of $173.76 but it has an unrealized profit of -$248.21, resulting in a total equity of $99,919.99. The win rate and loss rate stand at 50%. Sharpe ratio is -0.345 and compounding annual return is -1.929%. Based on these statistics, the model will make a loss in the long run and it is not viable to go live with this model.

4. Major Decisions by Model in live tracking

Trades Summary — Download Trades

Date Time	Symbol	Type	Price	Quantity	Operation	Status	Tag
2018-11-07 01:00:00	EURUSD	Market	$1.14587 USD	87,000	Buy	Filled	
2018-11-08 15:00:00	EURUSD	Market	$1.14237 USD	-87,000	Sell	Filled	
2018-11-09 01:00:00	EURUSD	Market	$1.13465 USD	-87,000	Sell	Filled	
2018-11-12 02:00:00	EURUSD	Market	$1.13243 USD	87,000	Buy	Filled	
2018-11-13 01:00:00	EURUSD	Market	$1.12475 USD	88,000	Buy	Filled	
2018-11-14 10:00:00	EURUSD	Market	$1.12862 USD	-88,000	Sell	Filled	
2018-11-15 01:00:00	EURUSD	Market	$1.13391 USD	88,000	Buy	Filled	
2018-11-16 13:00:00	EURUSD	Market	$1.13328 USD	-88,000	Sell	Filled	
2018-11-18 19:00:00	EURUSD	Market	$1.14194 USD	87,000	Buy	Filled	

Figure 4: Trades Summary

All in all, the model has made two winning trades and two losing trades, resulting in a 50% overall win rate.

2018-11-09 01:00:00	EURUSD	Market	$1.13465 USD	-87,000	Sell	Filled
2018-11-12 02:00:00	EURUSD	Market	$1.13243 USD	87,000	Buy	Filled

2018-11-13 01:00:00	EURUSD	Market	$1.12475 USD	88,000	Buy	Filled
2018-11-14 10:00:00	EURUSD	Market	$1.12862 USD	-88,000	Sell	Filled

Figure 5: Winning trades

Let us first examine the decisions that the model had made to enter and exit the winning trades.

```
2018-11-10 00:00:00 Predicted price is 1.1420201063156128
2018-11-10 00:00:00 Prev_price is 1.142490029335022
2018-11-10 00:00:00 short
```

Figure 6: 1st Winning Trade

```
2018-11-16 00:00:00 Predicted price is 1.131989121437072
2018-11-16 00:00:00 Prev_price is 1.1287399530410767
2018-11-16 00:00:00 long
```

Figure 7: 2nd Winning Trade

Figure 6 and 7 shows the predicted next day closing price by the model printed in the log.

For the 1st trade, it entered a short position as the predicted price is lower than the current price. For the second trade, it entered a long position as the predicted price is higher than the previous price.

```
# Exit execution Conditions
df1 = self.hist_data  # Calculate and update new atr everyday
df1 = df1[['open', 'high', 'low', 'close']]
self.atr = ta.ATR(df1['high'], df1['low'], df1['close'], timeperiod=14)
self.Debug("atr " + str(self.atr[-1]))
self.Debug("curr_price " + str(curr_price))

if self.currency in self.long_list:
    cost_basis = self.Portfolio[self.currency].AveragePrice  # To calculate the price of our FOREX currency p

    if ((curr_price <= float(cost_basis) - float(self.atr[-1]))  # Stop loss-Take Profit conditions
            or (curr_price >=  # Use of Average True Range to determine price to sell
                float(self.atr[-1] * 1) + float(cost_basis))):
        self.Debug("price is: " + str(curr_price))
        self.SetHoldings(self.currency, 0)
        self.long_list.remove(self.currency)


if self.currency in self.short_list:
    cost_basis = self.Portfolio[self.currency].AveragePrice

    if ((curr_price <= float(cost_basis) - float(self.atr[-1] * 1))  # Stop loss-Take Profit conditions
            or  # Use of Average True Range to determine price to buy back
            (curr_price >= float(self.atr[-1]) + float(cost_basis))):
        self.Debug("price is: " + str(curr_price))
        self.SetHoldings(self.currency, 0)
        self.short_list.remove(self.currency)
```

Figure 8: Exit condition

The exit (Stop Loss-Take Profit) condition for the trades is based on the Average True Range (ATR) indicator. The ATR reflects the volatility of a price in absolute price levels based on the average volatility from past historical data. The number of days used to calculate the ATR can be decided by the trader. A more in-depth explanation for this indicator can be found in the final report and model write up.

For the first trade, the model exited the trade when the price dipped below the cost basis by a ratio of one of the ATR. As for the second trade, the model exited the trade when the price rose above the cost basis by a ratio of one of the ATR, making us some profit.

2018-11-07 01:00:00	EURUSD	Market	$1.14587 USD	87,000	Buy	Filled
2018-11-08 15:00:00	EURUSD	Market	$1.14237 USD	-87,000	Sell	Filled

2018-11-15 01:00:00	EURUSD	Market	$1.13391 USD	88,000	Buy	Filled
2018-11-16 13:00:00	EURUSD	Market	$1.13328 USD	-88,000	Sell	Filled

Figure 9: Losing Trades

```
2018-11-08 00:00:00 Predicted price is 1.148163914680481
2018-11-08 00:00:00 Prev_price is 1.1399849653244019
2018-11-08 00:00:00 long
```

Figure 10: 1st Losing trade

```
2018-11-16 00:00:00 Predicted price is 1.1319891214370728
2018-11-16 00:00:00 Prev_price is 1.1287399530410767
2018-11-16 00:00:00 long
```

Figure 11: 2nd Losing trade

Figure 10 and 11 shows the predicted next day closing price by the model.

For both trades, it entered a long position as the predicted price is higher than the previous closing price.

The exit (Stop Loss-Take Profit) condition is the same as in figure 7. When the cost basis of the currency falls below the current price by a ratio of one of the atr, the model has exited the position and executed the stop loss.

5. Performance Evaluation of Backtested Result

The team identified three main reasons for the poor performance of the model during the backtest: Wrong prediction of price by the LSTM model, Conservative Stop Loss - Take Profit conditions and limitations of QuantConnect.

5.1 Wrong Prediction of Price

The model failed to predict the direction of the price of the currency 50% of the time as shown in the backtest. Although we have tried to optimize the model's hyperparameters, our grid search may not have been rigorous enough to train an accurate model. Perhaps the model could be more robust with more testing of different hyperparameters. However, it can be argued that as the testing period is only two weeks, it is not sufficient to conclude that the model is a poor one. The model may perform better over a longer time period.

5.2 Stop Loss-Take Profit Condition

The Stop Loss-Take Profit condition of our model is too conservative. We set an ATR range that is too low which results in us exiting the trade when there is potential for more profit.

2018-11-09 01:00:00	EURUSD	Market	$1.13465 USD	-87,000	Sell	Filled
2018-11-12 02:00:00	EURUSD	Market	$1.13243 USD	87,000	Buy	Filled
2018-11-13 01:00:00	EURUSD	Market	$1.12475 USD	88,000	Buy	Filled

Figure 12: Trade 2

For example, in figure 2, we enter a short position at 1.13465USD and exited at 1.13243 USD on the 12th. If we were to set a larger ATR margin and exit on the 13th at a price of 1.12475 USD, we could have made even more profit.

Figure 13: Actual Price

From figure 12, we see that price dropped to a low at $1.122USD.

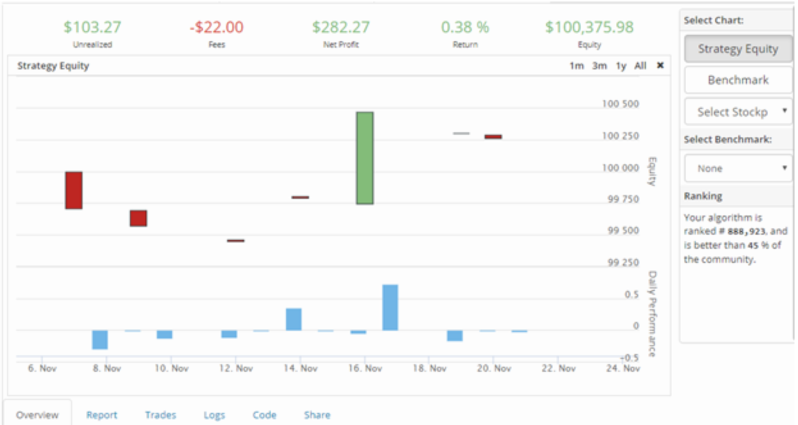

Figure 14: ATR at 1.25

Figure 14 shows the result when we increased our SLTP range to 1.25 * ATR. There is an improvement in the backtest with a gain of 375.98 USD in equity.

However, the reason our SLTP range was 1.00 * ATR is due to the volatility of the FOREX market. We had a conservative risk appetite and did not want to lose a huge amount of capital if the market was not in our favor.

5.3 Limitation of QuantConnect

There is a slight difference in how we envisioned our code to perform in the market and how it actually performed due to shortcomings of QuantConnect.

```
# curr_price = self.Securities["EURUSD"].Price  # Get current market price
curr_price = pd.DataFrame(
    self.History([self.currency], 120,
                 Resolution.Hour))
curr_price = curr_price['close'][-1]
```

Figure 15: Errors

We initially used self.Securities[‘’EURUSD”].Price in the exit condition to get the current price of the currency. However, when we went live with the model, it failed to get the correct current price. This led to us changing the code to get the current price using ‘Resolution.Hour’. Even though the current price could be obtained more accurately using ‘Resolution.Minute’, the team faced many runtime errors during backtesting and subsequently when we went live. After consulting with Mr. Shashank, we realized this is a limitation of QuantConnect and we would need to work around it. Being unable to get the current price may be one of the reasons for the less than stellar results especially in forex where price points change rapidly.

After multiple revisions, we settled with the current code. Although it was not the best solution, it was the most optimal at that point in time. If given more time, we could have solved this bug which might have improved the performance of our model.

6. Live Performance vs Backtested Performance

The difference in the performance between the live and back-tested model could be due to a few reasons. Firstly, the live model was run from the 12th to the 21st while the backtested model ran from 7th to 21st. This resulted in a difference in the trading conditions. The market could have been more favorable during the period from 12th onwards and the model correctly predicted the trend at that timing which explains the profit.

Secondly, as mentioned in section 5.3, the limitation of QuantConnect prevented the model from performing to its full potential. The model performed differently when it went live and when it was backtested. Logs of the live trade indicated multiple runtime errors occurred during the call to get pricing data. This disrupted the live trade and the model had to be redeployed. As a result, the model discarded all the information saved previously such as current positions stored in the list. This caused the model to enter another position when it is already in one. Fortunately, our model managed to make a profit.

The team has realized the importance of considering the bugs that could occur in a live environment. There were some differences in our model's performance during backtesting and live. The difference could have been minimized with proper exception handling in our code for the live environment.

7.Conclusion

This summary describes our model's results in a live environment and a backtest over the same time frame. The team has identified the key trades made by our model. There was a difference in performance between the backtest and live which was mainly due to unforeseen errors during live deployment.