FX Trading with a Distributed Quote Book

Jingbo Yang, Carolyn Kao, Xiaoye Yuan, Jiachen Ge, Jon Braatz, Sunny Shah

Spring 2020

Abstract

Foreign exchange (FX) trading consists of trading pairs of national currencies in a large, decentralized over-the-counter (OTC) market. The sheer volume traded on a daily basis calls justifies the rise of algorithmic trading systems. Despite the fact that large volumes of FX trades are already conducted via algorithms, trading opportunities still exist in abundance thanks to the fragmented nature of FX market. Unlike centralized exchange-based markets like the equities market, liquidity providers for the foreign exchange market act independently and quote different exchange rates to the counterparties who have established a credit relationship with them, leading to opportunities associated with latency and patterns that are unique to individual institutions. This paper investigates the extent to which latency arbitrage opportunities between different liquidity providers and machine learning techniques that can forecast the future movements of exchange rates can be profitably used. For each of these two aspects, we consolidated model outputs into actionable trading signals. We found that a latency arbitrageur with zero execution risk could attain profits in the high single-digit millions of dollars for the currency pairs in our dataset in the time window we investigated, and on the order of hundreds of thousands of dollars for reasonable execution latencies. Nevertheless, our machine learning models suffered from the noisy nature of financial time series and have a limited effect on achieving significant returns.

1 Introduction

Unlike centralized exchanges that handle stock transactions, foreign currency trading is conducted in a decentralized way, often directly between trading partners without involvement of a central exchange. In this setting, liquidity providers trade with each other for profit, hedging and satisfying demand of their customers. Despite the decentralized nature, its digitization has allowed algorithmic trading to grow, in contrast to resistance experienced in bond and debt market. Methods used for algorithmic trading largely carries over from market to market, ranging from statistical techniques to correlating movements with news and even Twitter feeds. All of these algorithmic methods rely on reasonably accurate quotes/orders from which patterns can be discovered and acted upon. Depending on the size and time span of these opportunities, different types of analysis and testing need to be conducted to ensure that valid trading signals can be generated and time window available for trading is realistic.

In this paper, we tackled foreign exchange trading using a distributed quote book in two directions. The first direction is to identify latency arbitrage opportunities and estimate potential profit due to time disparity among quotes provided by different liquidity providers. The second is to use machine learning to make predictions for prices of the next time interval. With these two direction, we could capture trading opportunities in both high frequency and intra-day trading domain. For latency arbitrage, we found that a large number of opportunities exist, yet only $\$215k$ profit for EURUSD can be realistically achieved given reasonable assumption on network and computation speed. For machine learning models, we were able to use a regression based model to achieve accumulated 1.7% profit. These profits do need further verification but indeed serve as solid foundation for further investigation toward both directions of research.

2 Background and Related Works

2.1 FX Market Structure

The foreign exchange market, the most liquid market in the world with an average daily trading volume of \$5.1 trillion[1], differs significantly in several respects from markets that retail investors are typically more familiar with like the equities market. For example, in the equities market, trading takes place on centralized exchanges with order information readily available to participants in the form of a limit order book, which contains prices and quantities at which participants have advertised that they are willing to trade. Unless an order submitted to the exchange is cancelled by the entity who submitted it, it is guaranteed to be available for a counterparty

[1] According to the Bank for International Settlements, in 2016.

to trade against. While there are multiple exchanges that deal in each equity, with each having their own limit order book, there is nonetheless a notion of a single bid and ask price for each equity due to a regulation (Regulation NMS) by the US Securities and Exchange Commission mandating that brokers execute customers at the lowest available ask price when buying securities and the highest available bid price when selling securities.

In contrast, the spot FX market is a decentralized over-the-counter (OTC) market, with the analogue of exchanges being market makers called Liquidity Providers (LPs) operating at major commercial and investment banks. Entities desiring to trade directly with an LP do so by establishing a credit relationship with them. The LP can then advertise a stream of quotes to the trading entity. In general, the LP can quote different bid and ask prices to different counterparties depending on each counterparty's credit, past trading activity, and other factors. In this sense, there is no single exchange rate for a currency pair, as each participant in the market is quoted rates that are specific to that entity's credit relationship with each LP. Furthermore, an LP is not beholden to the quotes that it advertises to each counterparty, which is to say that if a counterparty wishes to trade at the most recent exchange rate quotes it received from the LP, the LP may decline to trade at that rate. We largely ignored this execution risk in this project. Another unique aspect of the FX market (at least as it pertains to our dataset) is each quote from an LP is for a fixed quantity of 1,000,000 units of the base currency, meaning that any strategy that we implemented does not need to consider the volume of currency to exchange with an LP in any given trade as we didn't have the freedom to choose this.

2.2 Latency arbitrage

Latency arbitrage is a practice wherein the trading entity exploits a time disparity between quote updates by its liquidity providers to earn profits. Concretely, a latency arbitrage opportunity arises when one liquidity provider is willing to buy a currency in question for a higher price than another liquidity provider is willing to sell for at the same time. As a consequence, firms are willing to undertake steeper costs to obtain premium data feeds and suitable locations so as to better be able to detect and exploit scenarios like these.
Despite recent improvements in technology, firms are not satisfied with the transmission times, offered by fiber optic cables. These cables are about 30% slower than microwave technology [11] and hence, select firms have invested in microwave technology, allowing them to engage in competition to obtain the fastest transmission speeds possible. This is advantageous because the profitability of executing a detected arbitrage scenario depends on the extent to which the prices in question drift by the time the order reach the counter-parties involved.
Key limitations for latency arbitrage is that submitted orders can affect market pricing. Latency arbitrage opportunities are also limited in size because such orders depend on availability of open orders and the possibility of order cancellations [14]. A proper latency arbitrage system must also account for slippage, regulations and tax implications to properly account for associated costs [4]. In a deeper level, market structure and clearling rules are difficult to model yet their importance cannot be ignored by high frequency trading firms [13]. In fact, focus of academic research is on modeling of said trading limitations and "frictions" because the capability of executing high frequency trading depends largely on available infrastructure in terms of communication network and computing hardware and access to high frequency data streams.

2.3 Machine learning

Machine learning models are common tools researchers use to make predictions on the financial markets. Inputs to these models range from simple time series to those including sentiments from Twitter and Reddit. Goals of these models vary, but most aim to build correlation between input data with market return or volatility. Direct prediction on daily market returns has been shown to have positive result [10], with less than 46.3% binary classification error rate using a k-nearest neighbour model. This work has also demonstrated that building a joint classifier further reduces error rate to 40.99%, though no confidence interval has been reported, nor was time-dependent cross validation used during training. An important pattern found from previous research is that price movement prediction benefit from additional information. For example, researchers at HSBC has found that order book, order flow and pre-generated techncials combine to achieve the best returns compared to predictions made using independent set of features [2]. Moreover, it has been found that non-traditional methods, such as genetic programming, could also lead to positive results [3]. With such a setup, price movements are binarized and the "genetic code" converts to a set of boolean operators that process a series of binarized price movements. Most importantly, this system used a rolling training period, which respects the time-dependent nature of financial time series.
In this study, we analyzed the performance of many machine learning models, including linear regression, kernel ridge regression, K-nearest neighbours and LightGBM. Other models such as Naive Bayes and support vector machines have also been studied but through an initial set of experiments we have determined that they are of limited utility for our purpose. In this section we expand on usage of relevant models and hyper-parameters tuned to improve results.

2.3.1 Linear regression

Linear regression is a common tool for predictive analytics that attempts to explain a target variable in terms of linear combinations of the input variables. We used linear regression as one of our machine learning technique, as it is easy to implement, interpret and efficient to train. Although linear regression might be over-fitting to the training data, but it can be easily avoided by dimension reduction, cross validation and regularization.

2.3.2 Kernel ridge regression

Kernel ridge regression (KRR) combines ridge regression with kernel methods. The main purpose behind KRR is to solve the least square problem with an additional L_2 regularization term so to mitigate over-fitting. This further forces the regression coefficients down to nearly zero, preventing the large and mutually cancelling coefficients in regular linear (non-regularized) regression.

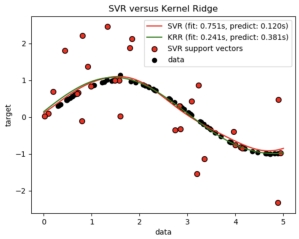


Figure 1: KRR versus Support Vector Regression (SVR) [1]

.

2.3.3 K-nearest neighbors

The k-nearest neighbors classification (kNN) is a non-parametric method where the class membership of a target variable is determined by those of the k nearest neighbours according to some distance metric on the input space. The kNN algorithm is a simple one, requiring no training before making a prediction. Its simplicity and ease of implementation made it a desirable model for us to use, in this project.

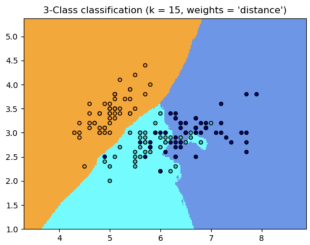


Figure 2: kNN with an L_2 distance metric classifies an input data point based on a majority vote of its 15 nearest neighbors in the training set (taken from [8])

.

2.3.4 Light gradient boosting machine

Light gradient-boosting machine (LightGBM) is a gradient-boosting framework, using tree-based learning algorithms. It grows a decision tree vertically by choosing the leaf with the maximum delta loss. It is designed to be distributed and efficient with faster training speed, better accuracy and lower memory usage. These characteristics made the LightGBM a suitable model for both the classification and regression tasks.

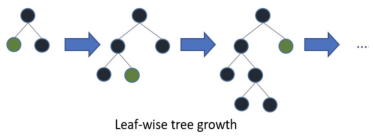


Figure 3: Light gradient-boosting machine generates the tree by choosing the leaf with the maximum delta loss (taken from [5])

.

In addition to the four models introduced above, we've also implemented several additional models, such as Support Vector Machines, Gamma Support Vector Machines, Gaussian Process Classifier, Random Forest Classifier, Multi-layer Perceptron classifier, AdaBoost classifier and finally, the Gaussian Naive Bayes algorithm for classification. However, these models were not able to achieve sufficient performance to warrant discussion (for example, having AUROC scores less than 0.5). This could be either a consequence of wrong labeling of our targets or a wrongly implemented model.

2.3.5 Performance measurements

In measuring the performance of our machine learning models, we utilized standard evaluation metrics like classification accuracy, area under Receiver Operating Characteristic AUROC) curve, F1 score, R^2, etc. to evaluate our model performance. Model performances are evaluated in Section 5.

Kendall Rank Correlation Coefficient For the purpose of generating trading signals, regression models that directly predict return or amount of price movement do not align with the eventual goal of earning consistent profit. Common regression metrics like MAE and MSE have limited utility in this context because direction of movement is more important than difference between predicted and ground truth value. Kendall rank correlation coefficient is an ideal emulation for AUROC score computed for binary classification, computed as:

$$\tau = \frac{\#\text{ of concordant pairs} - \#\text{ of discordant pairs}}{\binom{n}{2}} \tag{1}$$

where concordant pairs are pairs of inputs where the input with the greater label has a greater regression output than that of the input with the lesser label (in other words, the model orders the two inputs correctly), and vice versa for discordant pairs. For this to work, we apply an affine transformation from the range $[-1, 1]$ to $[0, 1]$ to be in the same scale as AUROC scores. With Kendall rank correlation coefficient, a regression model with high score is more likely to be correct in terms of direction of price movement than a model with low score.

3 Data

The data provided by Integral, a price aggregator and liquidity pool provider, consisted of tick-level bid and ask prices provided by five major liquidity providers (LPs), with millisecond-level resolution. Furthermore, the prices were provided for eight different currency pairs - USDCAD, USDCHF, USDJPY, USDSEK, AUDUSD, EURUSD, GBPUSD, and NZDUSD, with USD always being one of the currencies for each pair. The raw data contained the bid and ask prices in the quote currency at which the LP was willing to buy/sell 1,000,000 units of the base currency over a 1 month window (February 1, 2019 to March 1, 2019). The bid and ask volumes were also provided, but they were always fixed at 1,000,000 units. A trading week starts during Sunday afternoon and ends at on Friday night. For the purpose of this project, we truncated trades prior to 6PM Sunday and those after 10PM Friday to remove large gaps of data during those inactive hours. In short, our data derives from a total of 25 trading days, accounting for a total of 400 active trading hours. A short sample of the data is presented in Table 1

LP	Currency pair	Time	Bid price	Bid volume	Ask price	Ask volume
LP-1	EURUSD	02.25.2019 00:00:00.819	1.13417	1000000	1.13424	1000000
LP-1	EURUSD	02.25.2019 00:00:00.819	1.13417	1000000	1.13423	1000000

Table 1: Samples of the raw data provided by Integral.

3.1 Data processing

Although containing a rich set of information, data provided by Integral were overlapping and contained quotes with 0 volume. Re-organizing and removal of these quotes were obvious first steps. After doing so, we computed open, close, high, low prices of 10, 30, 60, 300 and 600 seconds time intervals in the traditional candlestick manner. As shown in Figure 4, we plotted prices of EURUSD on 2019-02-05 after data cleaning and aggregation. This indeed agrees with historic data we could obtain from the Internet.

3.2 Data Splitting

For each currency pair, we used the first 20 days to train the models and then test the models on the data from the last 5 trading days of the month. This division is illustrated in Table 2. Since we used time dependent cross

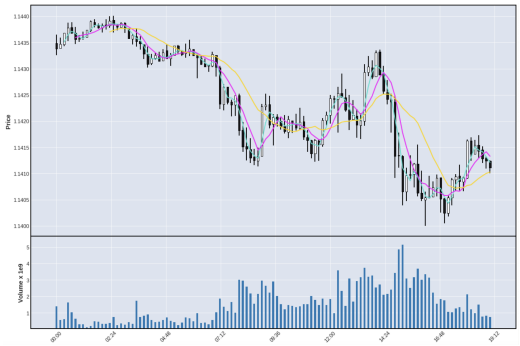


Figure 4: Prices of EURUSD on 2019-02-05, plotted using 1 minute candles. The top panel indicates the open, close, high and low prices for the time window being used. The bottom panel displays the volume traded within the time window.

validation as presented in Section 3.3, evaluating trading performance on the select test days do not violate time dependency, nor raise concerns such as tuning model parameters on test data.

Sun	Mon	Tues	Wed	Thu	Fri
					2/1
2/3	2/4	2/5	2/6	2/7	2/8
2/10	2/11	2/12	2/13	2/14	2/15
2/17	2/18	2/19	2/20	2/21	2/22
2/24	2/25	2/26	2/27	2/28	3/1

Table 2: The first 20 days were used as training data and the last 5 days as test data.

3.3 Time Dependent Cross Validation

Cross validation is an important step for developing machine learning models. Machine learning for time series requires special care accidentally including future data in the inputs or training set can skew model performance. The general setup for our time dependent cross validation is to simply always use validation samples that take place after training samples. An illustration of our time dependent cross validation is presented in Figure 5. Notice that even though the later folds can train on all previous data, we randomly sampled from previous training samples to keep training size comparable to earlier folds to maintain fairness and reduce memory usage. After cross validation, all samples are re-trained on the best set of parameters for later usage.

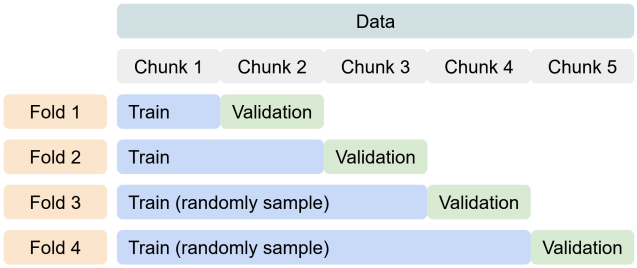


Figure 5: Illustration of data splitting for time dependent cross validation.

4 Latency Arbitrage Observations

To investigate the prevalence of latency arbitrage opportunities in the dataset for a given currency pair, we created lists of the highest bid and lowest ask prices across LPs for each time step (constituting the top of the "quote book" in analogy to a limit order book). In this paper, we started our trading window as soon as the highest bid is greater than the lowest ask (resulting in a negative spread). This window ends when the best quote prices change such that the spread is no longer negative. We also kept track of the duration of these trading windows, as they wouldn't be able to be taken advantage of if they lasted for less time than it would take to execute both legs of the arbitrage. We were particularly interested in windows that are longer than

a round trip to and from New York City to Chicago as a reasonable approximation for the time required to execute the trades. Figure 6 illustrates a timeline showing how a trading window is defined:

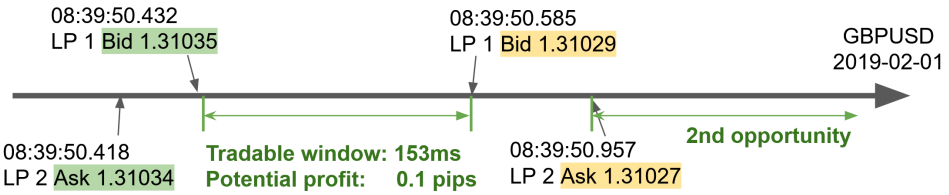


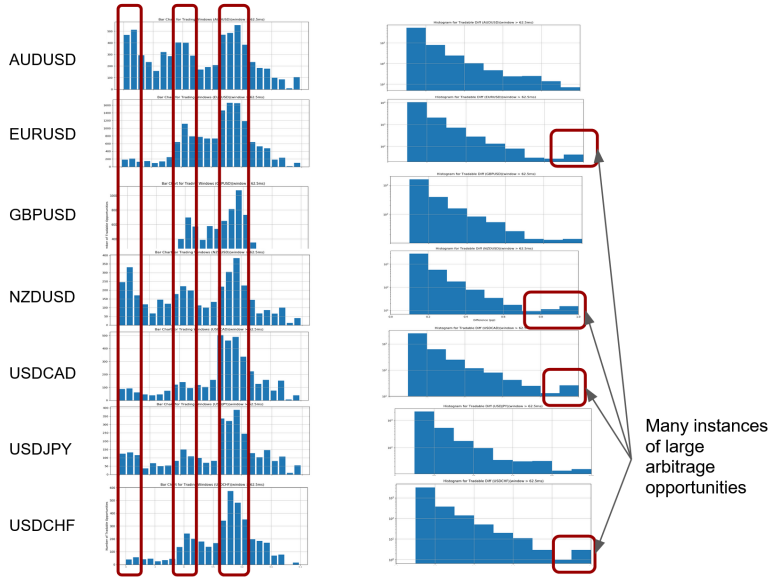
Figure 6: Trading window starts when the highest bid is greater than the lowest ask (the spread is 0.1 pip). Since LP-1 changes their order, our trading window ends since the spread is no longer negative.

Latency arbitrage is sensitive to network communication time. We referenced ping time between Google servers [9], presented in Table 3 for reference. We used latency from Chicago to New York because it is the most representative of North America based trading opportunities. Since Googles data centers communicate via dedicated fibre optic cables [6], its transmission time is representative of pre-microwave-based technologies used by high frequency trading operations. Notice that microwave technologies employed by many HFT firms is much faster than speed of traditional fibre-optic internet, transmitting at almost speed of light.
We arbitrarily picked 62.5 milliseconds, which is 50×1.25 milliseconds, accounting for signal transmission time and compute overhead. Note that cross-Atlantic trading windows are much larger. We also ignored the fact that the book at hand are quotes from liquidity providers rather than orders. Therefore liquidity providers can choose to "regret" on their quote, invalidating a proposed trade.

Destination	Network Ping Time (ms)	Light Transmission Time (ms)
Chicago	46	3.81
San Francisco	140	8.55
London	143	11.54
Hong Kong	495	26.84

Table 3: Network ping time and light transmission time among Google servers from New York City to the destinations mentioned above.

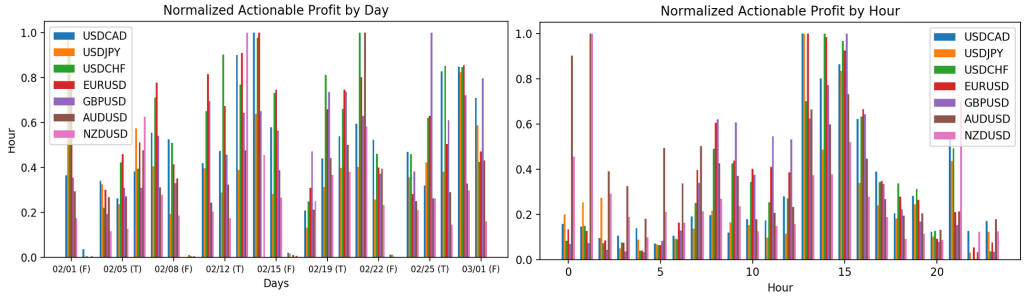
Comparing results from Figure 7 and Figure 8, we noted that the day is most profitable in the afternoon (EST), especially on Thursdays. This falls in line with general trends [12], such as a slight decrease in volatility on Wednesdays, with an increase in volatility on Thursdays, until the market closes for the week.



(a) Trading time over 1 day **(b) Latency arbitrage opportunities**

Figure 7: Tradable opportunities present in a single trading day, as noted in our data.

Even though this trend is not perfectly consistent from week to week, Figure 8 Thursday is still one of the best days for risk-averse traders operating with high volume to trade in the FX market, due to its high volatility. Furthermore, the pair EURUSD is the pair with the most actionable profit, potentially being due to that pair having the highest volume of any currency pair.



(a) Profits made over 1 trading day **(b) Profits made over the entire month**

Figure 8: From the data, we can note that the most profitable time of the day is in the afternoon (EST) and the most profitable day of the week is Thursday. EURUSD continues to be the most actionable profitable pair so far.

Aggregating opportunities with negative spread and thresholding for actionable trading windows, we found that EURUSD is the most profitable pair, with total actionable profit of 2152.5 pips, or $\$215k$ US dollars. This is much lower than the $85,000$ pips of total potential arbitrage profits that could be achieved without network and compute latency. This result supports previous researchers' effort on rigorous modeling of market structure and casts doubt on works that downplay the effect of latency.

Pair	**Potential Profit** (0ms) (pips)	**Actionable Profit** (62.5ms) (pips)
EURUSD	85000.0	2152.5
GBPUSD	50100.0	1431.6
AUDUSD	73700.0	999.5
NZDUSD	92300.0	656.6
USDJPY	10380000	61820.0
USDCAD	44300.0	685.7
USDCHF	38000.0	513.4

Table 4: Potential and actionable profits that can be achieved by executing on latency arbitrage opportunities.

5 Machine Learning Signal Generation

Machine learning models are best suited to serve as trading signal generator. A successful trading agent requires a reliable model that makes a prediction and a sizing strategy that decides the size of capital to commit to a particular trade. For this project we tackled the modeling component first as a classification problem then as a regression problem. Sizing of the trade are done using a *tanh* activation that takes prediction probability or rate of return as input. As explained in Section 3.3, our time-dependent cross validation setup allows testing directly on selected test days without concern on tuning hyper-parameters on the test set.

5.1 Modeling as Classification

For classification models, we aimed at determining whether the mean of the closing price in the next interval will be lower or higher than the mean of the closing price of the current interval. In order to ascertain this, we utilized classification models such as the k-Nearest Neighbours and the LightGBM to make such classifications. We ran the kNN model initially with the following parameters: leaf size = 30, $p = 2$, and number of neighbours = 5. In addition to the kNN, we also used the LightGBM classifier to generate the same signal. Hyper-parameters used for our LightGBM classifier are presented in Table 5.
Results produced on AUDUSD and EURUSD for classification models are presented in Figure 9. Additional results for performance of these models on all currency pairs are reported in Appendix Section 8.1 as part of Tables 7 , Table 8, Figure 19, Figure 17 and Figure 18.

Parameters	Model 1
Minimum weight in a child	0
Class weights	"balanced"
Maximum tree depth	4
Number of leaves	63, 127, 511
Minimum split gain	0
L_1 regularization constant	0.001, 0.01
L_2 regularization constant	0.01, 0.1
Number of estimators	256, 512

Table 5: For each of the LightGBM models used, we used the above values for each parameter in the GridSearch with cross-validation to find the best optimal results, in terms of metrics.

Scores achieved by classification models are surprisingly good, with F1 scores reaching above 0.6 and lower bound of confidence interval above 0.55. Accuracy scores achieved by the models concur with F1 scores, agreeing with the fact that proportion of up/down movement is mostly balanced. However, the trend is not shown in AUROC score, indicating that the cutoff threshold is very important, because F1 score is only a "snapshot" of the AUROC curve. This observation indicates that our models rely heavily on the threshold set to determine up/down, hence its prediction probabilities are of questionable utility.

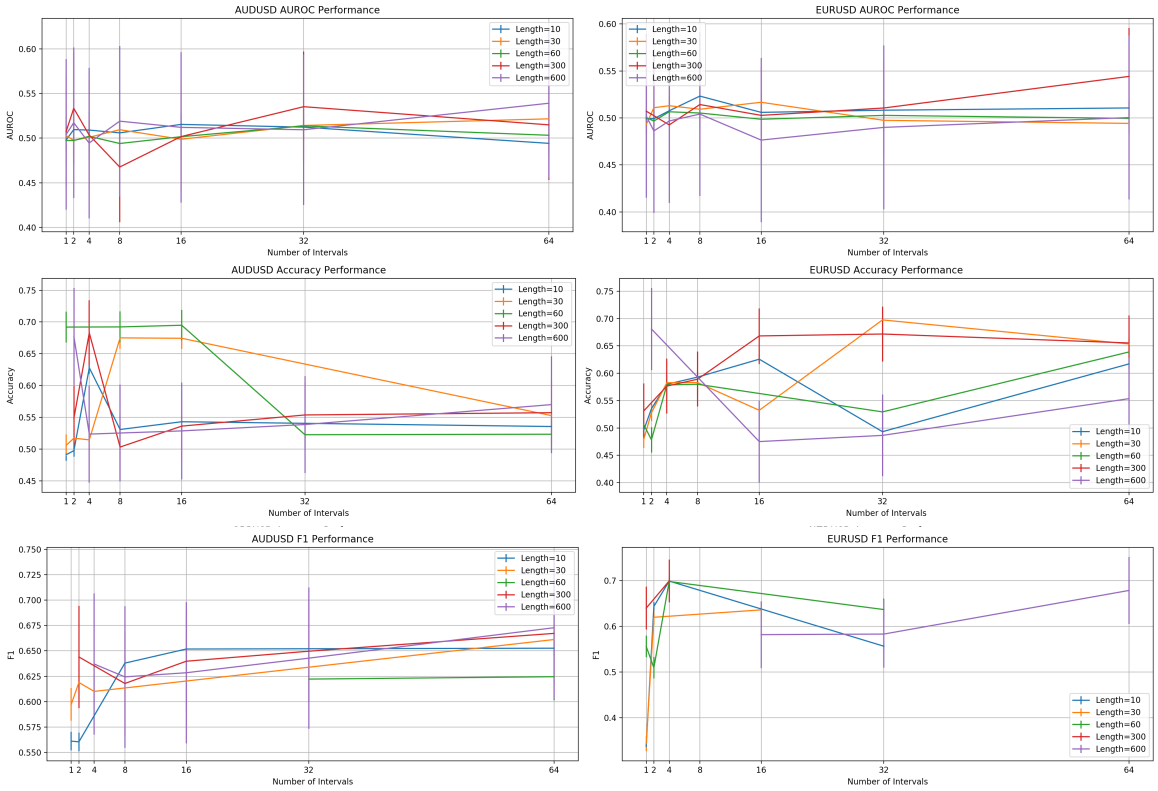
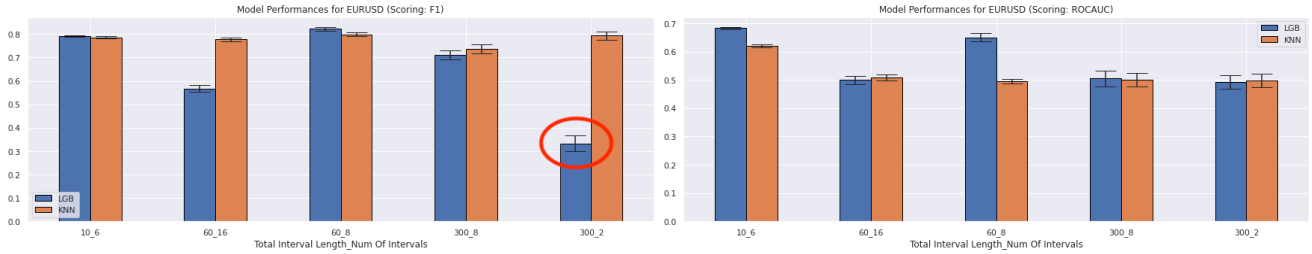


Figure 9: Performance of classification models for AUDUSD and EURUSD. Length are length of time intervals used to generate a "candlestick" in number of seconds. Top: AUROC evaluated on test data. Middle: Accuracy score evaluated on test data. Bottom: F1 score evaluated on test data.

Overall, classification scores benefit from longer history and using medium-sized "candlestick" intervals. Much longer time intervals (300 and 600 seconds) lead to worse performance and much wider confidence interval and using short time intervals and long history. It is possible that the models are capable of capturing trend that manifest in approximately 30 minute range.
Furthermore, we compared F1 and AUROC for LightGBM and kNN models in Figure 10. These models achieved similar performance in many scenarios, but LightGBM performed poorly on looking back 2 periods on 300 time interval. This anomaly could be attributed to LightGBM being sensitive to choice of hyper-parameters and this particular setup is not within range of our hyper-parameter search.



(a) F1 scores **(b) AUROC scores**

Figure 10: The currency pair used here is EURUSD. Note the significant difference in the F1 score for the last set of data (candle interval = 300 s, number of candles = 8).

5.1.1 Trading Profit using Classification Models

We selected the model that has the highest performance for each setting to act as trade signal generator for simple simulated trading. Daily returns achieved by these models on EURUSD, GBPUSD and USDCAD are presented in Figure 11. Poor performance is observed for all except EURUSD pair.

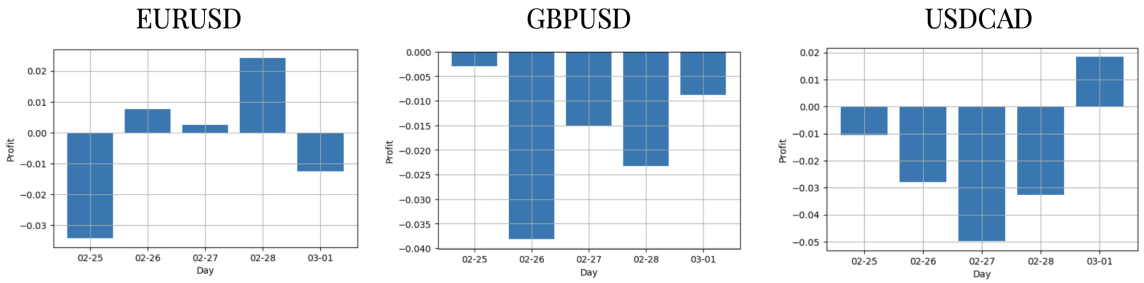


Figure 11: Daily trading profit achieved on testing days using classifier for signal generation.

Similarly, we computed hourly returns achieved by the model for the 3 chosen currency pairs as below. The model has the best performance at around 10 : 00 and performs poorly past 20 : 00. In Section 5.2.1 we presented Figure 16 for underlying price movement of the EURUSD pair. We observe that the price movements do not necessarily correlate to performance of the model.

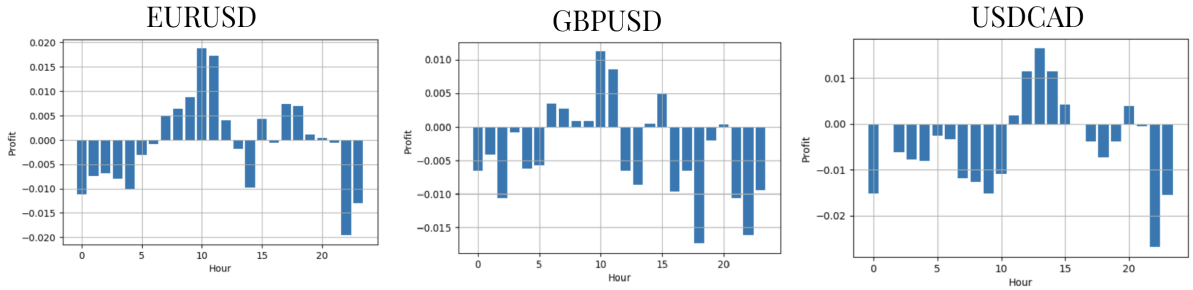


Figure 12: Hourly trading profit achieved during testing days using classifier for signal generation.

5.1.2 Discussion

On the one hand, we noted that the performance of the LightGBM model across all candle intervals and quantity tend to differ, even for only a single currency pair. On the other hand, the statistics for the kNN model tend to be consistently steady and hence, we believed that the kNN model is more reliable in determining the nature of the mean of the closing price in the next time window. Such analyses congruently apply for all seven currency pairs. In terms of error analysis, we've also marked the 95 percent confidence interval for each observation because the observed statistics can vary significantly across setups. Admittedly, some of the CI lower bounds are lower than 0.5, indicating the results from our models might not do well in performance scoring.

5.2 Modeling as Regression

With our regression models, we aimed at trying to predict the returns at the end of the *next* time window. For this purpose, we scaled all price-related inputs using mean of the closing prices of the last time interval. In order to achieve this objective, we used linear regression (no tunable parameters), kernel ridge regression (KRR) ($\alpha = 0.01, 0.1, 1$ and linear kernel), and LightGBM regressor. For our LightGBM regression model, we used different combinations of the following parameters as shown in Table 6.

Parameters	Values/Type
Minimum weight in a child	0
Class weights	"balanced"
Maximum tree depth	2, 4, 8
Number of leaves	15, 31, 127
Minimum split gain	0
L_1 regularization constant	0.001, 0.01
L_2 regularization constant	0.01, 0.1
Number of estimators	32, 64

Table 6: Hyper-parameters used for grid search to find optimal LightGBM model.

Results produced on AUDUSD and EURUSD for regression models are presented in Figure 13. Additional results are presented in Appendix 8. For these experiments, R^2 and Kendall Tau ranking score serve a similar purpose of evaluating whether the predicted return correlates with the actual return. R^2 results are mostly negative, while Kendall ranking scores are around 0.5 (recall from section 2.3.5 that we have scaled Kendall ranking score to $0 - 1$), reflecting poor predictive power of our models. The general trend seems to be that including additional time intervals does not improve performance, which means regressing on the return of the next interval has little correlation with distance history. In addition, it is worth pointing out that shorter time intervals in general lead to slightly worst average performance. However, size of confidence interval is not affected by number of intervals used nor length of time interval. This could reflect the inherent randomness in the underlying data.

Observing the results, we noticed that although R2 scores are hardly positive, lower bound of Kendall ranking scores are occasionally above 0.5, suggesting that there exists weak patterns to be discovered. We would use models whose lower bound Kendall scores are above 0.5 for simulated trading.

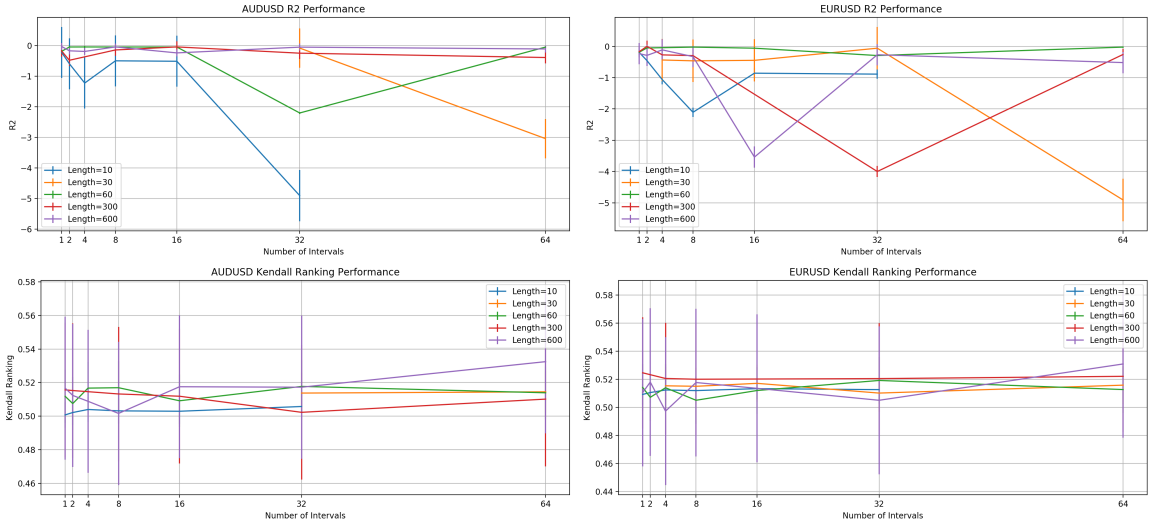


Figure 13: Performance of regression models for AUDUSD and EURUSD. Length are length of time intervals used to generate a "candlestick" in number of seconds. Top: R2 evaluated on test data. Bottom: Kendall ranking score evaluated on test data.

5.2.1 Trading Profit using Regression Models

We used the highest performing models for EURUSD, GBPUSD and USDCAD as trading signal generator for a simple simulated trading. Daily returns achieved by these models are illustrated in Figure 14. Achieved profits are higher than those achieved in Section 5.1.1, especially with high consistency for EURUSD pair.

Unfortunately the profit achieved cannot be replicated for GBPUSD nor USDCAD pair, even though the models are trained in a similar fashion with similar R2 and Kendall ranking scores.

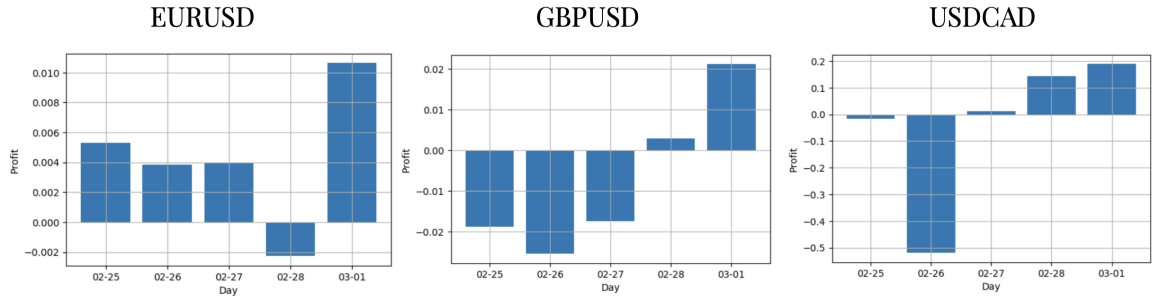


Figure 14: Daily trading profit achieved on testing days using regressor for signal generation

Accumulated hourly profits have high variance. As illustrated in Figure 15, the model is able to achieve profit at $1pm$ (Hour 13 in Figure 15) and $10pm$ (Hour 22 in Figure 15). Numerous low and consistent low profit earning hours reflect the fact that our model is many many mistakes, yielding continuous small losses. We have compared with hourly profits for the underlying EURUSD pair, shown in Figure 16. There does not seem to be a direct correlation between hourly movement and return achieved by our model, suggesting that trading profit for EURUSD pair is due to model prediction alone, not dependent on overall price movement.

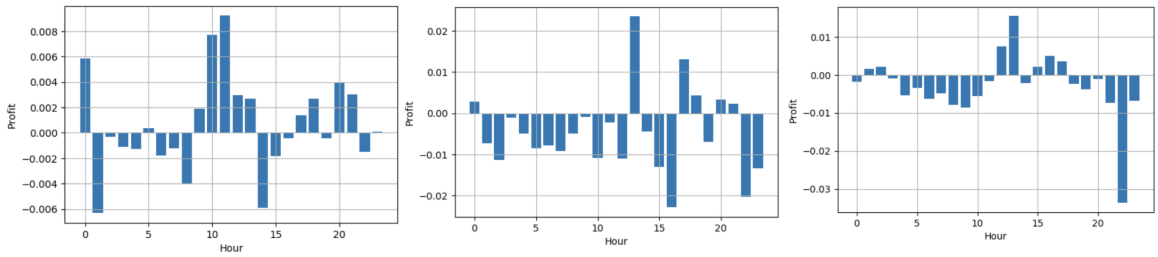


Figure 15: Hourly trading profit achieved during testing days using regressor for signal generation.

Figure 16: Underlying hourly price movement for EURUSD from 02/25 to 03/01/. Brown lines indicate $13:00$ **and blue lines indicate** $22:00$**.**

5.2.2 Discussion

Regression metrics cannot be used to directly compare against classification metrics computed for Section 5.1. However, our scaled Kendall ranking score can be interpreted as AUROC evaluated for continuous output. Although our LightGBM classifiers were able to achieve high AUROC in some circumstances, majority of the values for regressor and classifiers agree. The improvement in trading profit for regression, though equally fluctuating, could be attributed to better sizing, meaning that sizing according to predicted return is more reliable than sizing on probability of price movement. Just like our classification models, a major draw back of our approach is that action is taken at every time interval, which can incur numerous small losses, hence more intelligence sizing functions is required.

6 Conclusion

FX trading consists of trading on pairs of currencies in a large, decentralized market. The sheer volume traded on a daily basis ($5.1 trillion[1]) calls for an automated, algorithmic trading system. Such a trading system should be built on statistical and machine learning techniques for high frequency and accurate execution to capture profit during price movements.
The data used for this work is provided by Integral FX. Our data consisted of bid/ask orders put in by 5 major liquidity providers, over a monotonically-increasing time period, over 1 trading month. After cleaning up the data, we retained a total of 400 hours' worth of trading data from 25 trading days.

6.1 Achievements of This Work

For this project, we performed detailed analysis of behaviors of liquidity providers and successfully identified latency arbitrage opportunities. While we identified that at least tens of millions of dollars of profit could be attained by an infinitely-fast latency arbitrageur with no execution risk, more work would be needed to develop a profitable trading strategy that takes into account price movement between the time an order is submitted and the time the LPs processes it, as well as the risk that the LPs decline to accept the trade.
In parallel, we investigated machine learning techniques for identifying price movement. We experimented with a variety of classification and regression models to predict the binarized movement and the rate of return of at close of the next time interval. Briefly speaking, we noticed remarkable consistency of the scores reported by the kNN model, as opposed to that reported by the LightGBM model. This observation aligns with a belief that kNN is a well-defined classifier and is compatible with arbitrary prior distribution. Unfortunately trading profits achieved by kNN model is not consistent and accumulates to negative total return on the the trading days. Furthermore, we identified that the kernel ridge regression (KRR) model is the most capable regressor for our purpose. Using KRR as signal generator, we were able to achieve higher and more consistent return than those achieved using classification models.
We believed that with more compute power for parameter tuning, models that can incorporate cross-sectional and multi-time-horizon features, and better position sizing strategies, a machine learning model for price movement prediction that lead to consistent profits can be found.

6.2 Next Steps

By this point, we've learned enough to outline in broad strokes the future work needed to build a profitable latency arbitrage strategy.

First, the ability to process quote streams in real time, continuously keeping track of the best bids and asks across LPs, would be required. Since the mode of the tick interval distribution for most LPs is below a millisecond, we would need to parse each incoming quote and update the best bids and offers in less than a millisecond, likely requiring implementation of the algorithm in a higher-performance language than Python like C++. Once the spread between best bids and asks can be calculated in real time, the presence of a negative spread would initiate a decision to submit orders to the relevant LPs or not. This decision would need to take into account several risk factors.

One of the risk factors is whether each LP would execute the trades or not, as both LPs would need to execute the trade for the arbitrage to have a hope of being successful. The second risk factor is whether the spread will still be negative by the time each LPs execute the trades. Machine learning models like the ones explored in this paper could be used to model this second source of risk, but modelling the risk of trades not being executed by each LP would require the creation of training datasets using ground truth labels for whether given trades were executed or not. It would be difficult to obtain such a dataset without creating it via live interaction with the LPs in question, which would require a significant up-front investment to create those exploratory trades and see which ones are executed.

In the front for hyper-parameters, we suspected that the model is overfitting due to the amount of noise in the underlying time series data. Using non-uniform grid with more automated hyper-parameter optimization strategies could lead to more consistent results.

Sizing of orders on top of signals generated by machine learning models can be achieved through an additional model or an intermediate agent between modeling and execution. Such agent can be a cross-sectional portfolio management strategy or simply more sophisticated betting based on the Kelly criterion or its variants. In short, work toward this direction invokes the need for reinforcement learning, justifying researchers' focus on

multi-bandit strategies and Q-learning.

Multi-horizon deep learning model have also gained traction [7]. Adaptation of this work to intra-day level would be desirable for volatility and price movement predictions.

7 References

[1] *1.3. Kernel ridge regression.* URL: https://scikit-learn.org/stable/modules/kernel_ridge.html.

[2] Mark P Austin et al. "Adaptive systems for foreign exchange trading". In: *Quantitative Finance* 4.4 (2004), pp. 37–45.

[3] Siddhartha Bhattacharyya, Olivier V Pictet, and Gilles Zumbach. "Knowledge-intensive genetic discovery in foreign exchange markets". In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 169–181.

[4] Samuel N Cohen and Lukasz Szpruch. "A limit order book model for latency arbitrage". In: *Mathematics and Financial Economics* 6.3 (2012), pp. 211–227.

[5] Pranjal Khandelwal, Pranjal, and Analytics Vidhya. *Light GBM vs XGBOOST: Which algorithm takes the crown.* Mar. 2020. URL: https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/.

[6] Dmitri Kurjanov. *The Best Days of the Week to Trade Forex.* Oct. 2018. URL: https://admiralmarkets.com/education/articles/forex-strategy/best-days-of-the-week-to-trade-forex.

[7] Bryan Lim et al. "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting". In: *arXiv preprint arXiv:1912.09363* (2019).

[8] *Nearest Neighbors Classification.* URL: https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py.

[9] Wonder Network. *Global Ping Statistics New York.* URL: https://wondernetwork.com/pings/New%5C%20York.

[10] Bo Qian and Khaled Rasheed. "Foreign exchange market prediction with multiple classifiers". In: *Journal of Forecasting* 29.3 (2010), pp. 271–284.

[11] Shkilko et al. *Every Cloud Has a Silver Lining: Fast Trading, Microwave Connectivity and Trading Costs.* Oct. 2016. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2848562.

[12] Ben Treynor Sloss. *Expanding our global infrastructure with new regions and subsea cables.* Jan. 2018. URL: https://www.blog.google/products/google-cloud/expanding-our-global-infrastructure-new-regions-and-subsea-cables/.

[13] Elaine Wah and Michael P Wellman. "Latency arbitrage in fragmented markets: A strategic agent-based analysis". In: *Algorithmic Finance* 5.3-4 (2016), pp. 69–93.

[14] Elaine Wah and Michael P Wellman. "Latency arbitrage, market fragmentation, and efficiency: a two-market model". In: *Proceedings of the fourteenth ACM conference on Electronic commerce.* 2013, pp. 855–872.

8 Appendix

8.1 Classification Performance

AUROC and the F1 metrics for one currency pair are presented in Table 7.

Model	Candle interval	No. of candles	AUROC		F1	
			Train	Test	Train	Test
kNN	10	6	0.669	0.506	0.846	0.760
	10	6	0.822	0.620	0.877	0.786
	60	8	0.743	0.490	0.886	0.758
	60	8	0.618	0.495	0.853	0.798
	60	16	0.749	0.519	0.891	0.784
	60	16	0.705	0.508	0.877	0.777
	300	2	0.702	0.519	0.885	0.797
	300	2	0.633	0.498	0.858	0.793
	300	8	0.704	0.482	0.886	0.781
	300	8	0.654	0.500	0.858	0.737
LightGBM	10	6	0.820	0.684	0.803	0.791
	10	6	0.820	0.683	0.804	0.791
	60	8	0.820	0.639	0.814	0.763
	60	8	0.826	0.651	0.822	0.822
	60	16	0.841	0.687	0.829	0.824
	60	16	0.814	0.500	0.513	0.568
	300	2	0.940	0.500	0.947	0.696
	300	2	0.672	0.492	0.736	0.333
	300	8	0.985	0.491	0.986	0.727
	300	8	0.948	0.505	0.954	0.710

Table 7: The AUROC and F1 metrics for all the models ran on EURUSD (the results for other currency pairs are noted in Table 8). The parameters used in the kNN models (in blue) were: leaf size = 30, p = 2, number of neighbours = 5, while those in the kNN models (in green) were the best scores reported by the GridSearchCV on the following combination of parameters: leaf size = [15, 30, 60], p = [1, 2], number of neighbours = [3, 5, 6]. The parameters for the LightGBM models (in red) can be found in column 2 of Table 5, while those for the LightGBM models (in orange) are reported in column 3.

Additional values, with the table alternative between scores achieved by kNN model are shown in Table 8.

Currency pair	Candle interval	No. of candles	AUROC		F1	
			Train	Test	Train	Test
GBPUSD	10	6	0.667	0.500	0.843	0.755
	10	6	0.721	0.527	0.836	0.847
	60	8	0.742	0.504	0.881	0.767
	60	8	0.780	0.525	0.878	0.846
	60	16	0.750	0.523	0.887	0.844
	60	16	0.807	0.525	0.887	0.843
	300	2	0.694	0.509	0.882	0.247
	300	2	0.690	0.503	0.879	0.836
	300	8	0.692	0.500	0.886	0.843
	300	8	0.692	0.506	0.881	0.262
AUDUSD	10	6	0.829	0.589	0.911	0.787
	10	6	0.667	0.500	0.788	0.693
	60	8	0.699	0.498	0.882	0.765
	60	8	0.694	0.495	0.880	0.880
	60	16	0.753	0.499	0.879	0.754
	60	16	0.715	0.514	0.881	0.772
	300	2	0.702	0.493	0.884	0.767
	300	2	0.697	0.494	0.882	0.786

	300	8	0.697	0.513	0.880	0.795
	300	8	0.651	0.520	0.856	0.749
NZDUSD	10	6	0.669	0.510	0.846	0.748
	10	6	0.665	0.514	0.846	0.749
	60	8	0.738	0.599	0.883	0.817
	60	8	0.784	0.516	0.888	0.728
	60	16	0.750	0.493	0.882	0.764
	60	16	0.803	0.503	0.886	0.757
	300	2	0.698	0.496	0.883	0.793
	300	2	0.698	0.496	0.883	0.793
	300	8	0.711	0.514	0.873	0.778
	300	8	0.711	0.514	0.874	0.779
USDCAD	10	6	0.663	0.598	0.834	0.777
	10	6	0.718	0.629	0.824	0.770
	60	8	0.741	0.500	0.873	0.760
	60	8	0.782	0.518	0.877	0.752
	60	16	0.751	0.593	0.872	0.768
	60	16	0.802	0.618	0.869	0.757
	300	2	0.688	0.518	0.884	0.805
	300	2	0.614	0.515	0.871	0.822
	300	8	0.673	0.523	0.880	0.812
	300	8	0.596	0.506	0.865	0.816
USDCHF	10	6	0.662	0.490	0.838	0.752
	10	6	0.735	0.479	0.810	0.661
	60	8	0.733	0.502	0.876	0.756
	60	8	0.682	0.508	0.882	0.792
	60	16	0.758	0.510	0.881	0.762
	60	16	0.700	0.493	0.886	0.799
	300	2	0.688	0.506	0.882	0.782
	300	2	0.879	0.488	0.862	0.625
	300	8	0.711	0.523	0.888	0.791
	300	8	0.737	0.521	0.857	0.737
USDJPY	10	6	0.666	0.501	0.839	0.760
	10	6	0.734	0.505	0.809	0.669
	60	8	0.734	0.497	0.877	0.762
	60	8	0.692	0.515	0.883	0.811
	60	16	0.747	0.489	0.880	0.755
	60	16	0.798	0.509	0.842	0.679
	300	2	0.666	0.512	0.877	0.801
	300	2	0.568	0.504	0.860	0.831
	300	8	0.716	0.490	0.884	0.768
	300	8	0.881	0.510	0.865	0.619

Table 8: The AUROC and F1 metrics for some of the models ran on the 7 currency pairs. The parameters used in the odd rows were: leaf size = 30, $p = 2$, number of neighbours = 5, while the parameters used in the even rows were: leaf size = [15, 30, 60], p = [1, 2], number of neighbours = [3, 5, 6].

In Figure 17, Figure 18 and Figure 19 we presented the full set of experiment conducted using kNN classifier on all currency pairs. Analysis of these experiments are in Section 5.1.

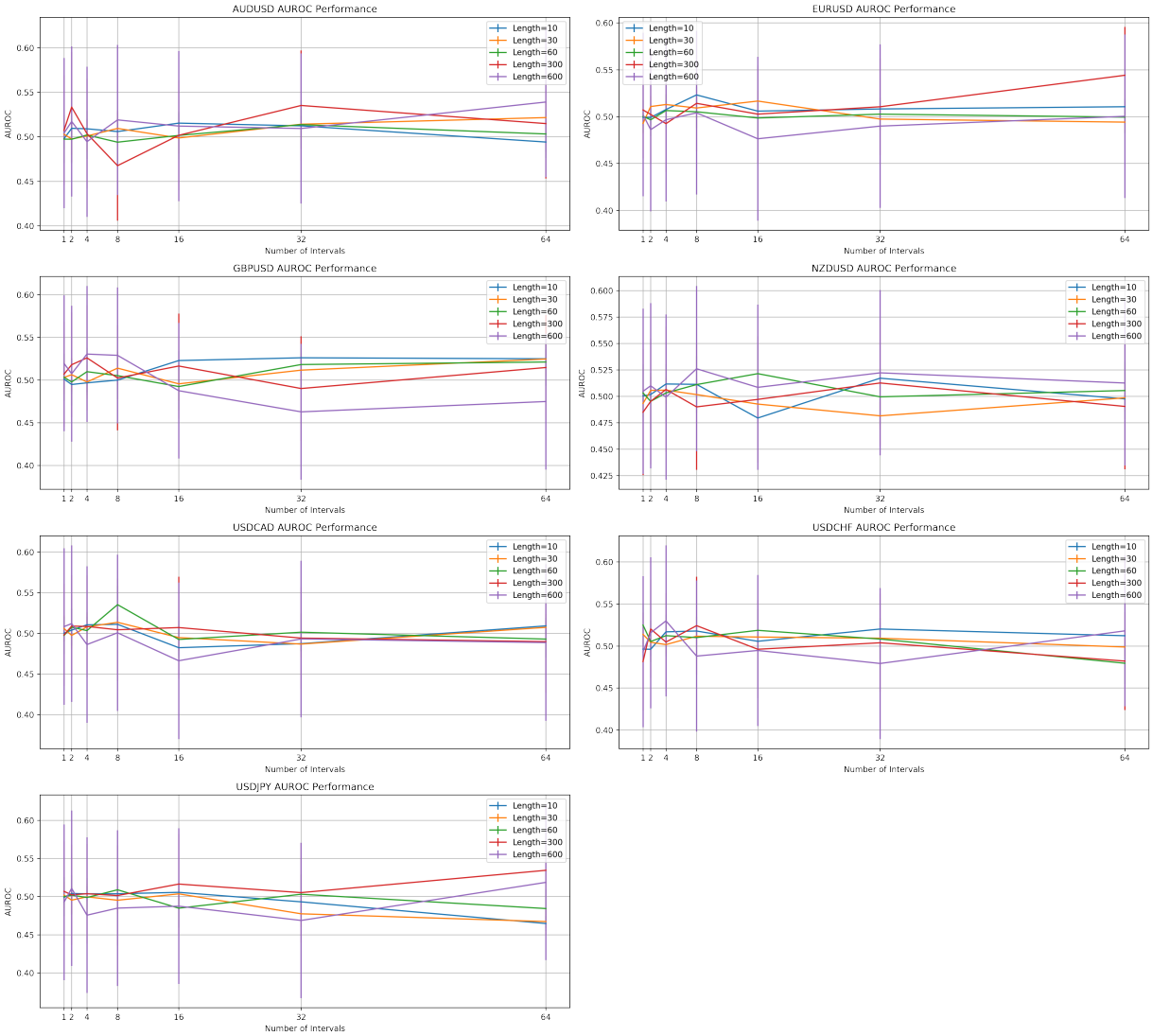


Figure 17: AUROC score of classification using kNN classifier

8.2 Regression Performance

In Figure 21 and Figure 22 we presented the full set of experiment conducted using Kernel Ridge regressor on all currency pairs. Analysis of the experiments are in Section 5.2.

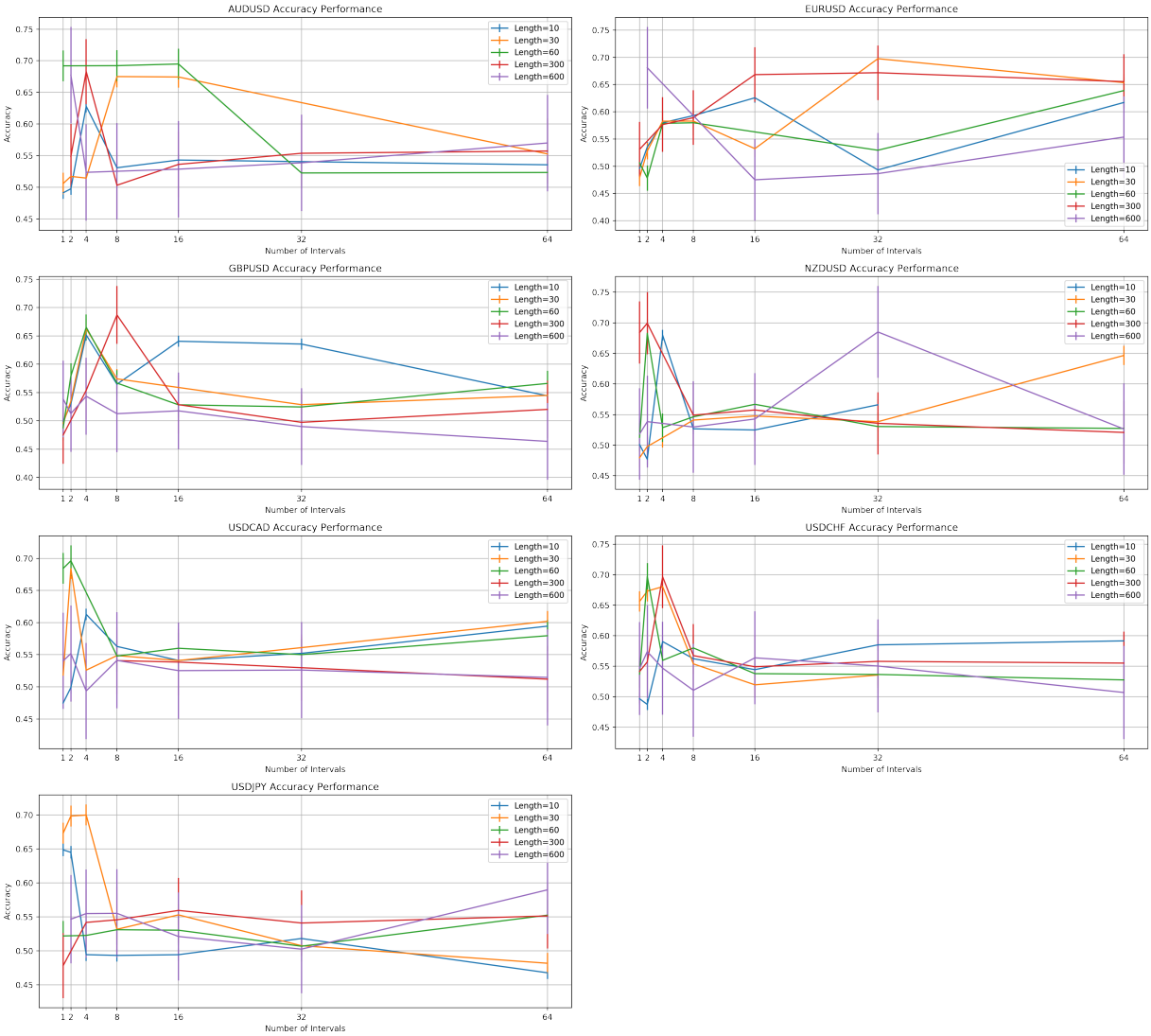


Figure 18: Accuracy score of classification using kNN classifier

9 Additional Materials

The following contents are <u>not referenced</u> in the main text nor in the Appendix. They are included to present results from additional experiments but do not form part of the overall story.

Pair	Candle interval	No. of candles	R^2 (train)	R^2 (test)
AUDUSD	300	2	0.992	0.00174
	60	8	0.997	0.994
	300	8	0.97	0.00255

Table 9: Some of the R^2 values of the training and test data obtained, using our linear regression model.

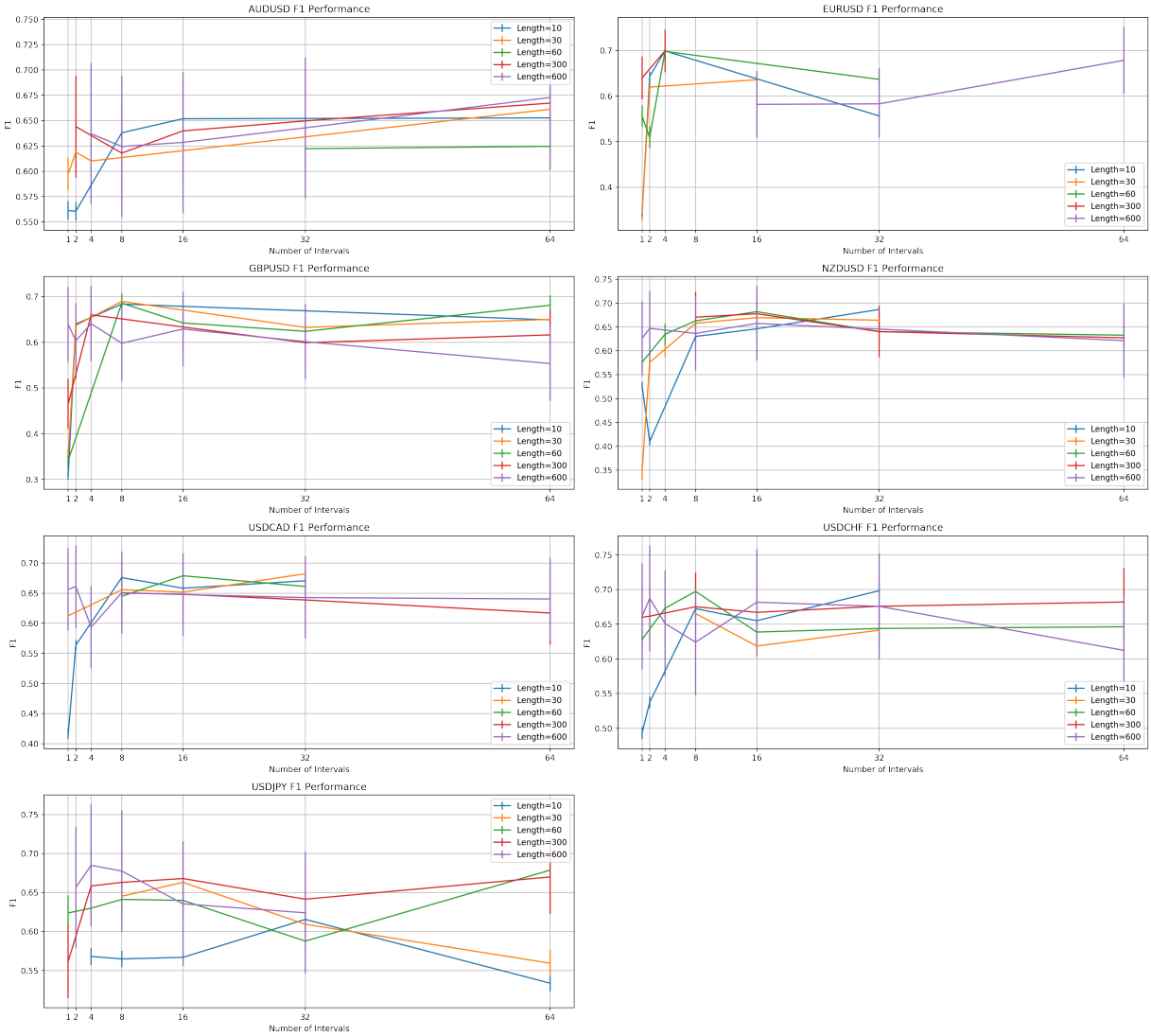


Figure 19: F1 score of classification using kNN classifier

Pair	Candle interval	No. of candles	R^2 (train)	R^2 (test)
AUDUSD	300	2	0.995	0.992
	60	8	0.996	0.994
	300	8	0.98	0.971

Table 10: Some of the R^2 values of the training and test data obtained, using our KRR model.

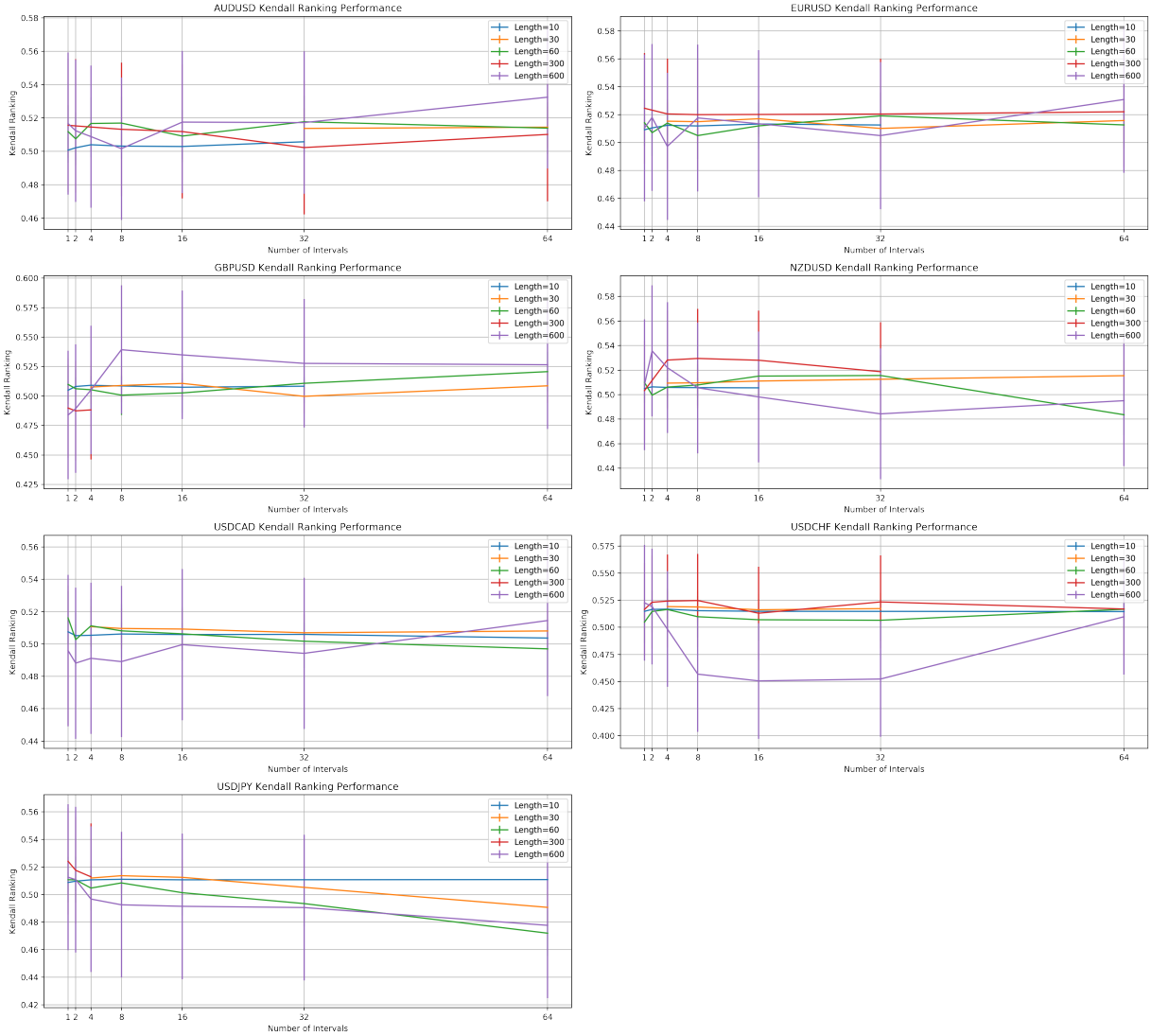


Figure 20: Kendall Ranking score of regression using Kernel Ridge regressor

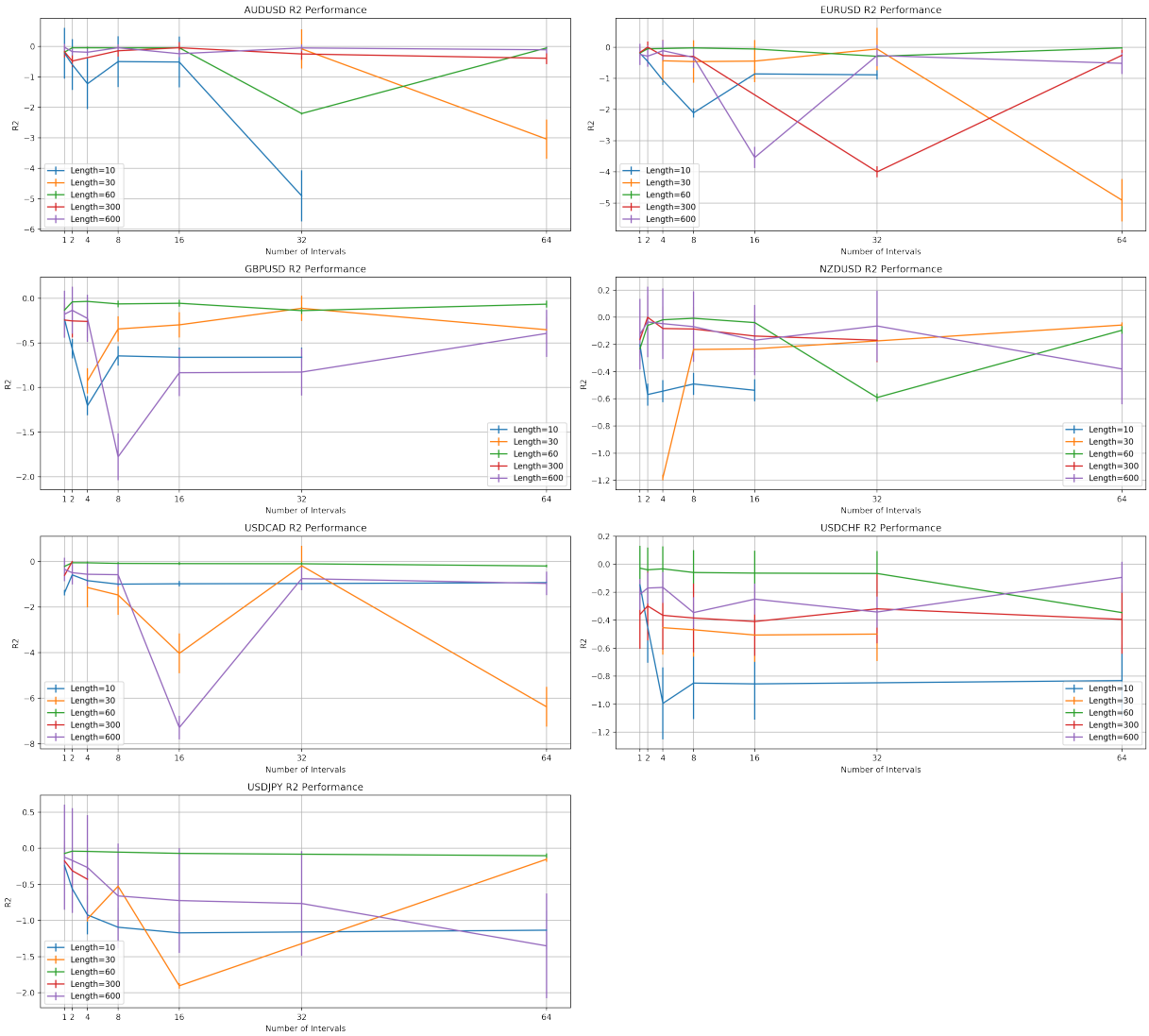


Figure 21: R^2 score of regression using Kernel Ridge regressor

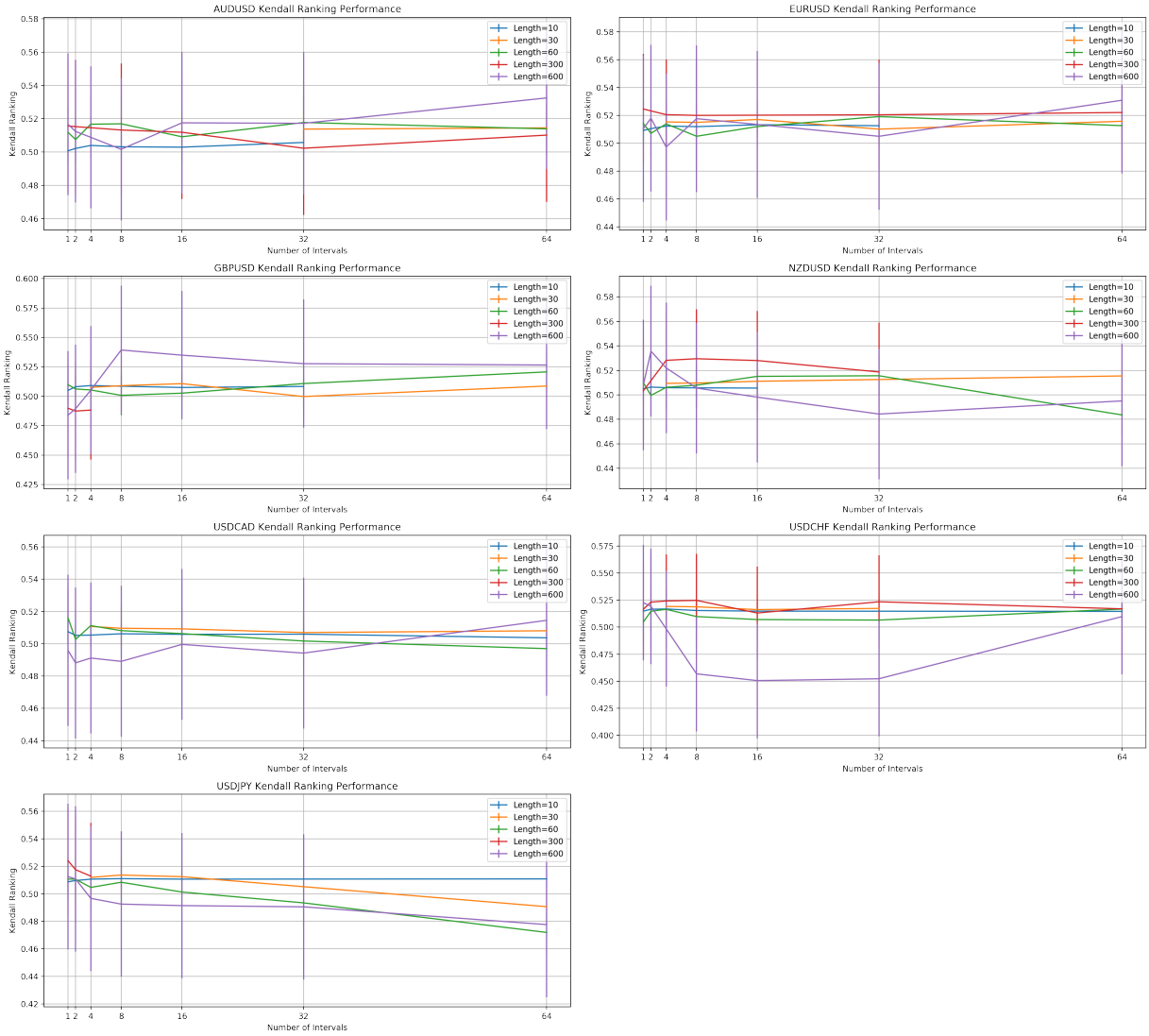
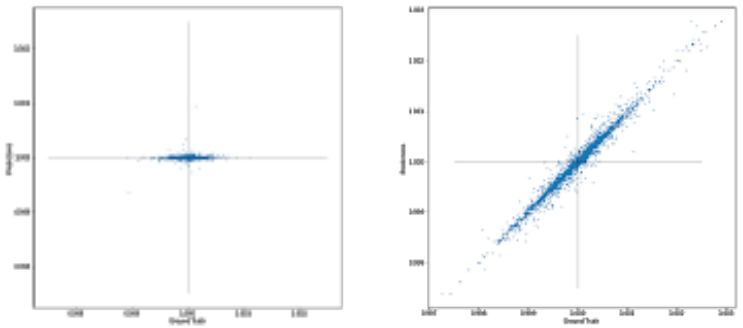
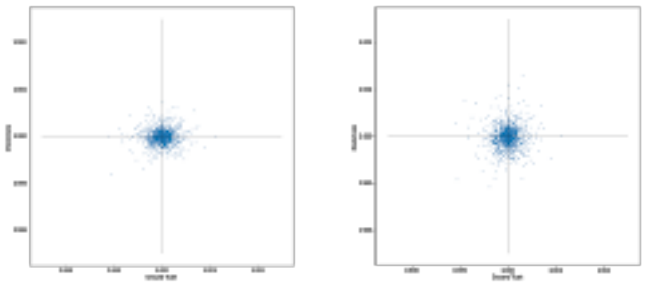


Figure 22: Kendall Ranking score of regression using Kernel Ridge regressor



(a) interval = 300 s, 8 candles **(b) interval = 60 s, 16 candles**

Figure 23: Scatterplots of the true labels versus the predicted labels from the EURUSD pair as in the above table, using the linear regression model.



(a) interval = 300 s, 8 candles

(b) interval = 60 s, 16 candles

Figure 24: Scatter plots of the true labels versus the predicted labels from the EURUSD pair, using the KRR model.