SCHOOL OF
OPERATIONS RESEARCH AND INFORMATION ENGINEERING
COLLEGE OF ENGINEERING
CORNELL FINANCIAL ENGINEERING MANHATTAN
CORNELL UNIVERSITY

Financial Engineering Project

Mind the Gaps: Short-term Crypto Price Prediction

Presented to the Faculty of the Graduate School of Cornell University
in partial fulfillment of the
requirements for the Master of Engineering Degree
and the Financial Engineering Concentration

By:
Payton Martin, William Line Jr., Yuxin Feng, Yunfan Yang, Sharon Zheng, Susan Qi, and
Beiming Zhu

Faculty Advisor: Sasha Stoikov

December 2022

Faculty Advisor's Signature

Date

Abstract

Quant methods for short-term price prediction of tradable assets have been studied by academics and practitioners throughout finance. One such robust predictor of price movements is the micro-price. The micro-price “can be considered to be the ‘fair’ price of an asset, conditional on the information in the order book”, and has been shown to be a better short-term price predictor in equity markets than the mid-price and weighted mid-price (Stoikov 2017). In this study we seek to apply this idea to define a robust estimator of the micro-price for Bitcoin (BTC). Sourcing high-frequency, limit order book (LOB) data from Bitstamp, we construct three mid-price adjustment estimators of the micro-price. We show that the Volume-Adjusted Mid-Price (VAMP) outperforms trade and quote imbalance adjusted mid-prices in both prediction of short-term price direction and larger, one standard deviation price movements.

Introduction

In his paper “The Micro-Price: A High Frequency Estimator of Future Prices,”[1] Stoikov demonstrates the efficacy of using a mid-price adjusted by order book imbalance in making short term price predictions in equities markets. In this paper we explore the idea of micro-price in a cryptocurrency setting, and potential solutions to accurately make short term price predictions for Bitcoin. Starting with the Bitcoin-USD limit order book and trading data sourced from the Bitstamp Crypto Exchange, we developed several measures/adjustments to the classical mid-price in our search for a cryptocurrency micro-price. Unfortunately, crypto currency markets are not as well behaved as equity markets. The unique structure of crypto order books, characterized by frequent and inconsistent gaps in the order book, adds an extra level of complexity in the search for an accurate ‘fair-price’. After some exploratory analysis and critical thinking, we decided to analyze three potential candidates: trade imbalance, volume adjusted mid-price (VAMP), and quote imbalance (similar to Stoikov’s micro-price).

Each of the selected methods for adjusting mid-price to more accurately represent the market’s fair price showed promise in our initial analysis, demonstrating some level of predictive power on short time scales (under 60 seconds). To further delineate between the potential models for the fair price, we analyzed each model’s efficacy as both a binary and multiclass classifier predicting the direction of price movement at time scales ranging from one second to 60 seconds. Lastly, we considered whether combinations of any of these models could produce even better results. At the end of this paper, we present the ultimate short-term predictor of Bitcoin price movements: Volume-Adjusted Mid-Price.

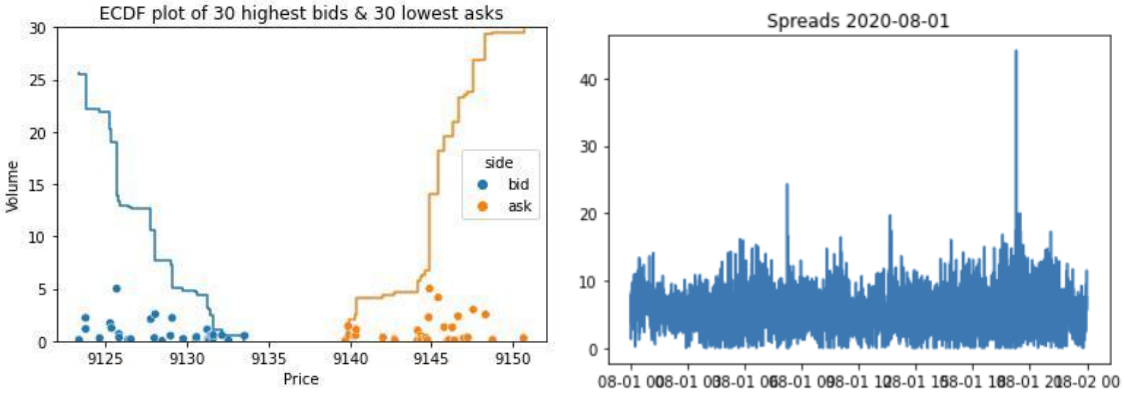
Data

For our research, we relied on two sets of data sourced from the Bitstamp crypto exchange: end of day snapshots of the limit order book and quote by quote updates to the order book throughout each day. We considered three full months of this tick level data, ranging from July 1, 2020 until September 30, 2020. The first step of our exploration was to build a framework to calculate the different features of the order book at every moment throughout each day in our dataset. Starting with the end of day snapshot, we then used each quote update to first update the order book, and then calculated and stored the important features: spread, mid-price, best bid, best ask, as well as volume adjusted versions of all of these. However, this process resulted in a very unwieldy amount of data. In order to condense our full dataset, we decided to only

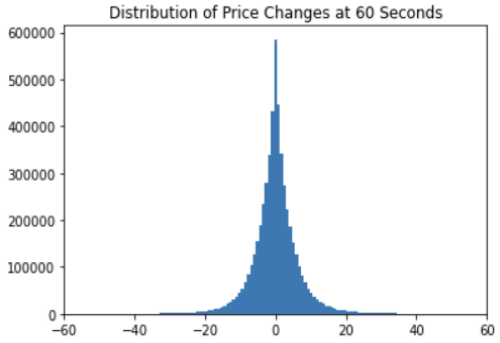
[1] Stoikov, Sasha, The Micro-Price: A High Frequency Estimator of Future Prices (November 26, 2017). Available at SSRN: https://ssrn.com/abstract=2970694 or http://dx.doi.org/10.2139/ssrn.2970694

consider full seconds throughout the day, rather than considering every tick update. This decreased the size of our dataset by multiple orders of magnitude.

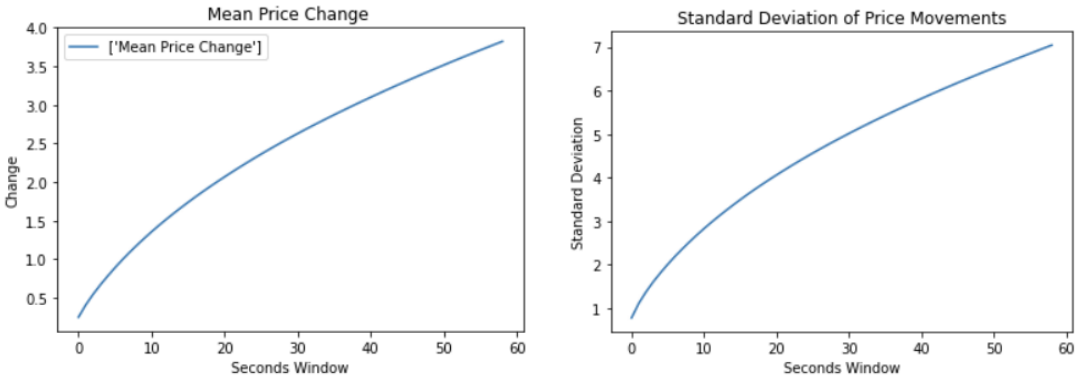
Some initial exploratory data analysis brought to light several critical aspects of the limit order book structure in a market such as the Bitstamp BTC-USD market. The plot below shows the general structure of these order books. We can quickly notice the frequent and inconsistent horizontal lines, which denote gaps in the order book — price levels where no volume is present on the book. Unlike the more well behaved order books of equities markets, the top-of-the-book is quite fragile. Additionally, we can see persistence across our dataset of bid/ask spreads around $7, a relatively narrow spread considering the price levels.



Now that we have an understanding of the high-level structure of the order book, we examined the distribution of price changes. Shown below, we can see that the distribution of price changes looking forward 60 seconds is quite narrow. However, there are very long, skinny tails. The distribution shows that even at our longest time scale, the price of Bitcoin makes very small moves.



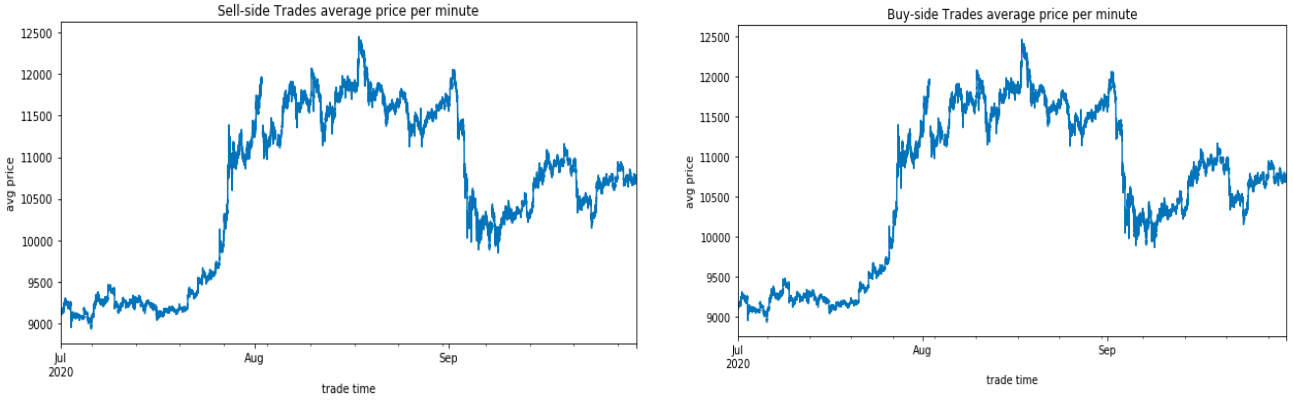
In fact, as shown in the plots below, we can see that the average price change at 60 seconds is still less than four dollars, with a standard deviation of seven (the same size as the bid/ask spread).



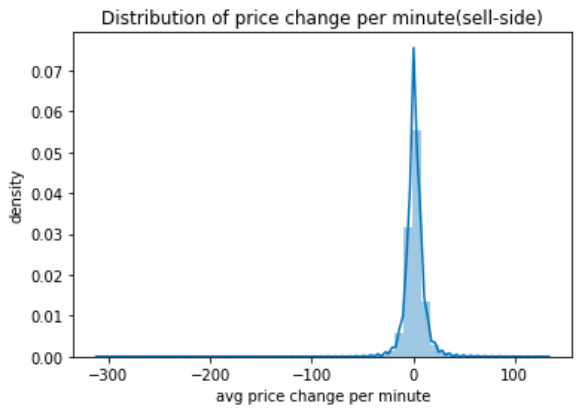
The last piece of information pulled from this initial exploration of the market's microstructure is that both the average price change and the standard deviation of those price movements appear to increase with the square root of our look ahead window.

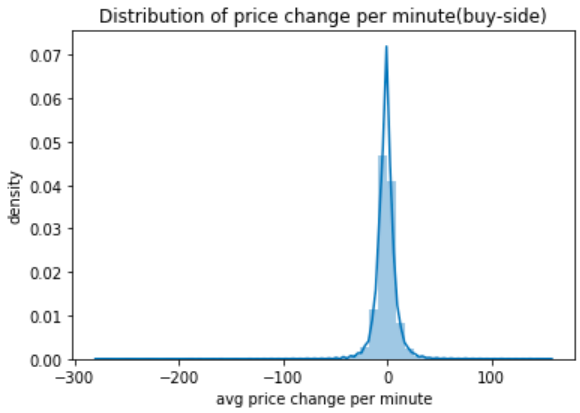
Trade Data

The trading dataset, also sourced from Bitstamp, contains trading price and quantity at the tick level from July 1, 2020 to September 30, 2020. We plotted the average price per minute and noticed that there's a large jump at the end of July and a large downturn around the beginning of September for both the buy and sell sides.

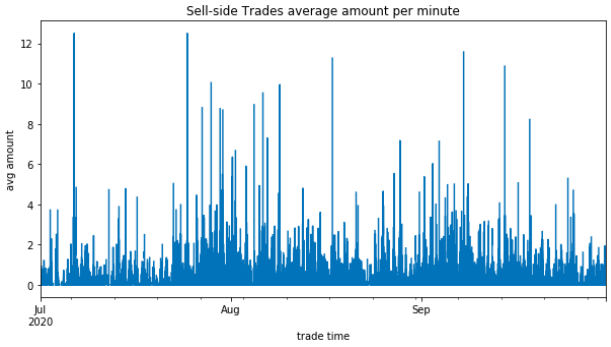


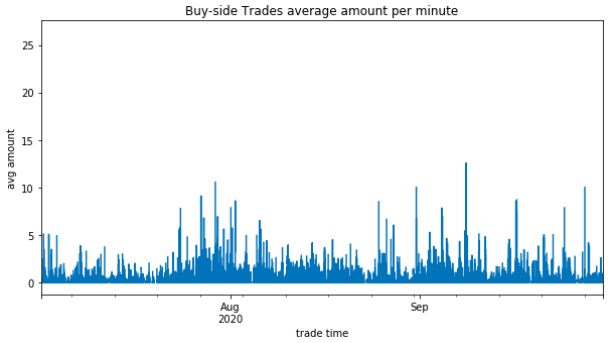
We then plotted the density distribution of price change in each minute since later we will try to predict price changes with factors like trade imbalance. The density distributions for both sides are left-skewed distributions but most of the tradings have a price change around 0.





The following plots are the average trade amount per minute on the sell and buy side. The trade amount is higher during the end of July and the start of August as well as the first half of September.





Feature Engineering

The ability to process and manipulate the large amount of data we have allows us to start generating signals. Through our research we examined many potential signals and will go in-depth on three in particular that have both empirical and theoretical validity as well as the fair price models developed from these signals, each of them hopefully capturing a different trend seen in the LOB data.

Volume Adjusted Mid-Price

Some form of mid-price weighted or adjusted by order book volume has been used for short-term price predictions and trading since the 1980's. Definitions such as the Volume-Weighted Average Price (VWAP) and Volume-Weighted Moving Average

(VWMA) have been useful in a myriad of applications. A feature that we found to be especially interesting for our application is what we call the Volume-Adjusted Mid-Price (VAMP). This can be calculated by the following formula:

$$VAMP_v = \frac{P_b + P_a}{2}$$

$$\text{where: } P_b = \frac{\sum_i^k P_b \times V_b}{v} \text{ and } P_a = \frac{\sum_i^k P_a \times V_a}{v}.$$

In the context of the above equation, 'v' represents volume (in dollars) and P_b and P_a are the volume-weighted bid and ask prices, respectively, where the volume summed to the top of the book is equal to v.

Say at the top of the book we have three best bids of \$91.15, \$91.16, \$91.17 with volumes of \$50k, \$10k, and \$50k, respectively, and three best asks of \$91.21, \$91.22, and \$91.24 with volumes of \$70k, \$60k, and \$40k, respectively. The mid-price of this snapshot would be \$91.19:

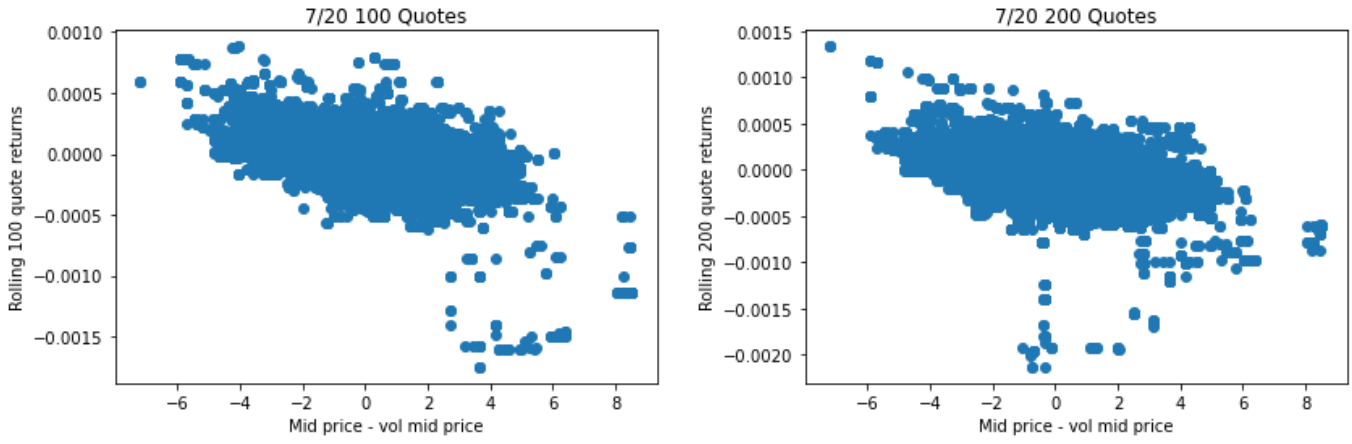
$$\frac{\$91.17 + \$91.21}{2}$$

whereas the VAMP (at \$100k of liquidity) would be \$91.187:

$$\frac{\$91.161 + \$91.213}{2}$$

The idea behind VAMP is to capture some element of liquidity pressure from either side of the order book, similar to order imbalance, but with the ability to capture information from the gaps in the order book and hopefully signaling future price movements more effectively. One key advantage of the VAMP compared to VWAP is that it can be calculated at any time t and as frequently as tick frequency.

We see in the plots below a slight negative correlation between rolling 100 and 200 tick returns and the difference between mid-price and VAMP with a volume cutoff of \$100,000, indicating that when VAMP is larger than the mid-price at time t, the future mid-price increases, and vice versa.



Although the VAMP with a volume cutoff of $100,000 shows some promise, we did also consider cutoffs at $10,000, $50,000, and $200,000 resulting in no visual difference between signal scatter plots. We will use a more refined metric later on to show that the $50,000-$60,000 range is the best for our study of the BTC-USD market on the Bitstamp exchange.

Trade Imbalance Adjusted Mid-Price

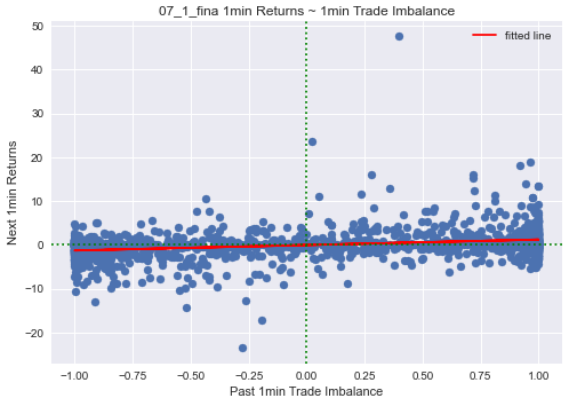
We defined the trade imbalance as follows:

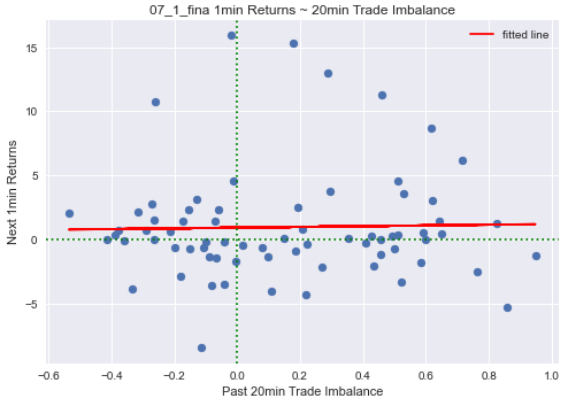
$$TI = \frac{\sum_{i=1}^{60} \gamma_i * (V_{buy} - V_{sell})}{\sum_{i=1}^{60} \gamma_i * (V_{buy} + V_{sell})}$$

where i is the seconds within one minute, V_{buy}, V_{sell} represent the buy and sell volume in the trading dataset and γ_i is the linear weight parameter. In this case, the closer the trade time to the current time, the larger weight it will have in the trade imbalance calculation. Trade imbalance is a number ranging from -1 to 1, demonstrating the main direction of the market trading flow in the past 1 minute.

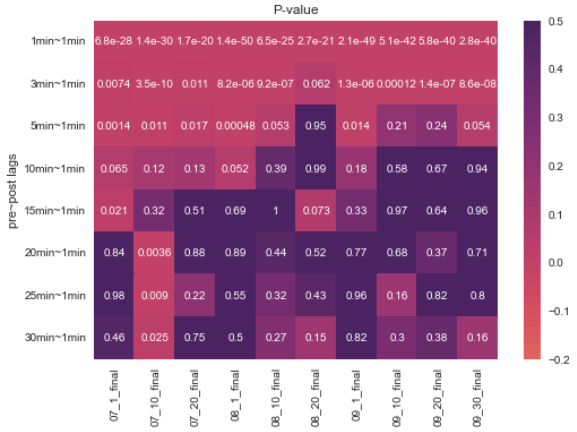
Before we set 1 minute as the window length to gather the trading data, we tested pre-windows of different lengths, from 1 minute, 3 minute, and 5 minute to 30 minute by a 5 minute step. According to the research of Kolm, Turiel and Westray (2021)[2], the deep order flow imbalance shows predicting power on high-frequency returns. In our case, except for the quote imbalance that we get from the order book, we also want to use the trading data as a dynamic signal of the order flow. We assume that there should be a positive correlation between the trade imbalance and the mid-price changes, and intuitively it can be understood as when there is a trend of buying in the market, the demand for this asset is high and thus the price of it should go up. And by plotting these two variables for 9 single days (3 days from each month), we confirmed that this upward trend is demonstrated in the trading environment. And it is also shown that as the time window expands, the upward correlation gets weaker between the trade imbalance and mid-price change (see Appendix for 10 and 20 minute windows).

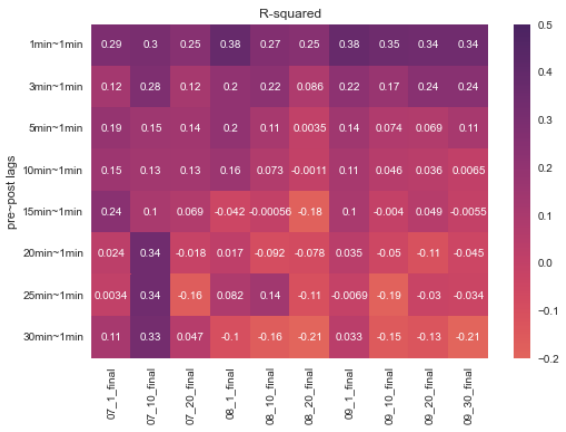
[2] Kolm, Petter N. and Turiel, Jeremy and Westray, Nicholas, Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book (August 5, 2021). Available at SSRN: https://ssrn.com/abstract=3900141 or http://dx.doi.org/10.2139/ssrn.3900141



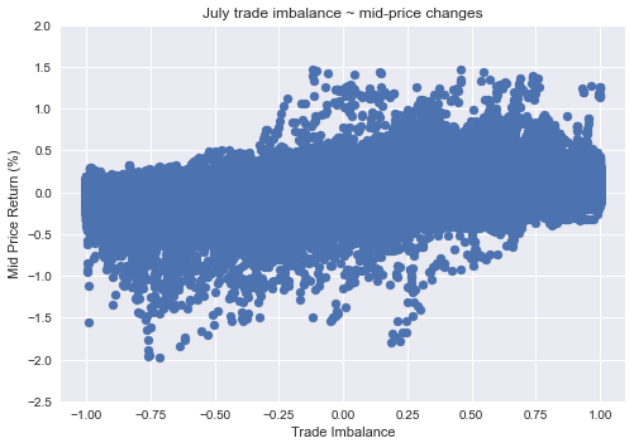


To get more statistically powerful evidence, for different windows, we ran linear regressions on the trade imbalance and the change in mid-prices over 60 seconds. The ideal time window should provide a small p-value (meaning the model is statistically significant) and a high r-squared value (meaning there is a strong linear relationship between the trade imbalance and changes in mid-price). The regression results for these two criteria can be checked in the hearmaps below.





Based on the previous discussion, we picked the 1 minute time window to generate our trade imbalance signal in the fair-price prediction model. And below is the plot between the 1-minute trade imbalance and next 60 second mid-price changes for July, which we use as our training set for the prediction model.



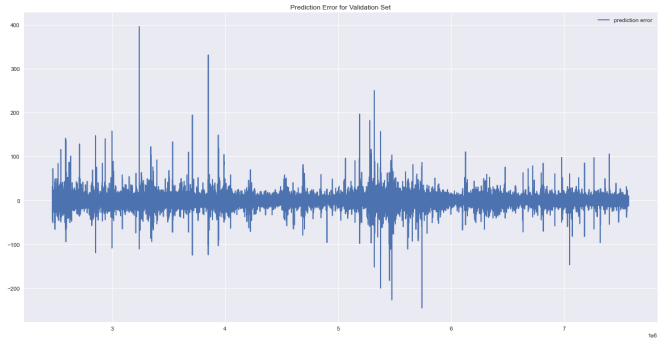
To train the fair price prediction model, we used a linear regression to determine the coefficient of the trailing one-minute trade imbalance, used to predict the difference between mid-price and the fair price.

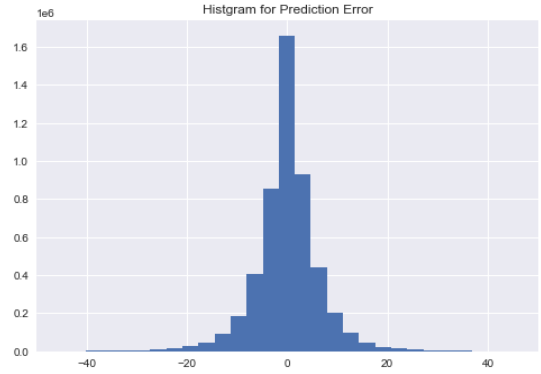
$$min_{\alpha} \sum (F_t - Mid_{t+\Delta t})^2$$

$$s.t. \quad \frac{F_t - Mid_t}{Mid_t} * 100 = \alpha TI_t$$

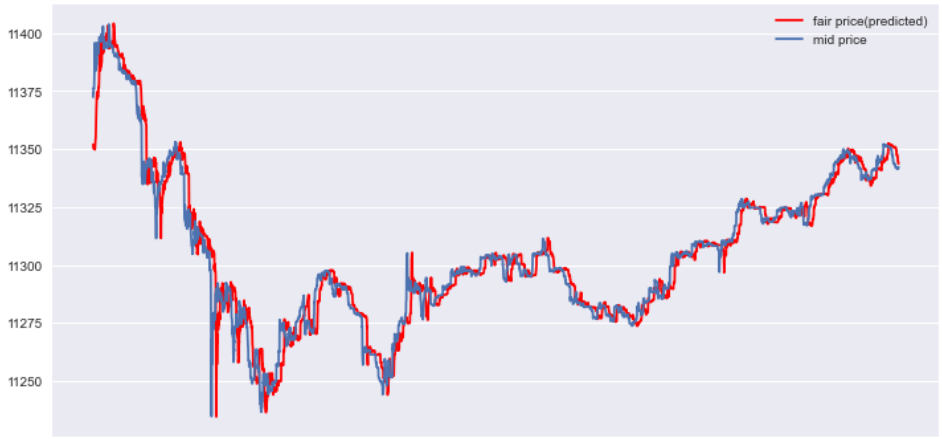
$$\Delta t = 1s, 5s, 15s, 30s, 1min$$

We get a statistically significant (p-value=0) coefficient of trade imbalance equal to 0.0024. And after validating with data from August and September, we got the prediction error (the actual mid-price minus the predicted fair price) as below, with the minimum, 25% quantile, median, 75% quantile and maximum equal to -245.183, -2.830, 0.006,2.995 and 395.603. From the statistics we can see for the prediction error, there are long tails on both ends and generally higher frequency on the positive side, meaning that the trade imbalance tends to overpredict the fair price.





And if we compare the predicted value and the mid-price itself, we can see the predicted value is almost a parallel shift of the mid-price, which means the fair price is very close to the mid-price 60 seconds ago and the trade imbalance signal is not contributing a lot to the fair price prediction.

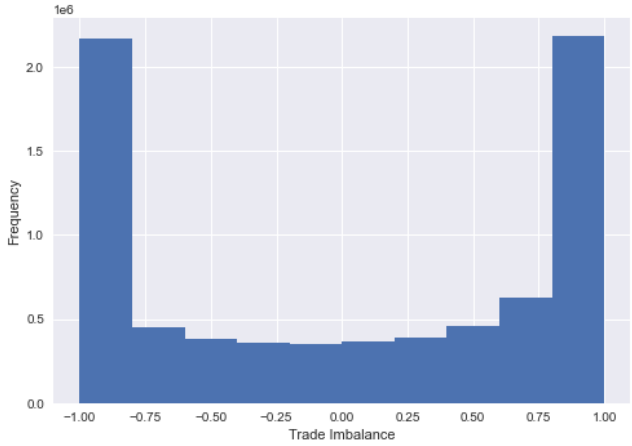


If we take one step back and look at the regression result, such a result should not be surprising because the parameter on trade imbalance is only 0.0024, meaning that the trade imbalance signal can at most contribute +/-0.3 to the mid-price changes, which is much smaller compared to the real changes in 60 seconds.

To figure out a possible explanation to this result, we looked at both the trade imbalance value and the model.

If we look back at the scatter plot, it can be observed that the 1-minute trade imbalance, though has a strong correlation with the mid-price changes at the two ends where trade imbalance equals -1 and 1, at the middle an upward trend is actually not very obvious. What's more, the frequency plot for the trade imbalance in the past 1 minute is shown

below, and here we can see the trade imbalance occurs more often at -1 and 1. In this case, during the regression the two heavy ends might dominate other data and make the model lean towards the steeper slope result, and thus overpredict the price changes.



In terms of the model itself, we think linear regression might not be the best model to capture the dynamic. We can see from the scatter plot between July trade imbalance and mid-price changes that, even though there is an observable monotonic trend between these two variables, given that the percentage price change is a number ranging from -2 to 1.5, a bandwidth of 1 is actually too large for a line to fit. In this case, either the data points should be categorized more detailedly so that thinner lines are available for regression, or other regression methods should be adopted.

Quote Imbalance Adjusted Mid-Price

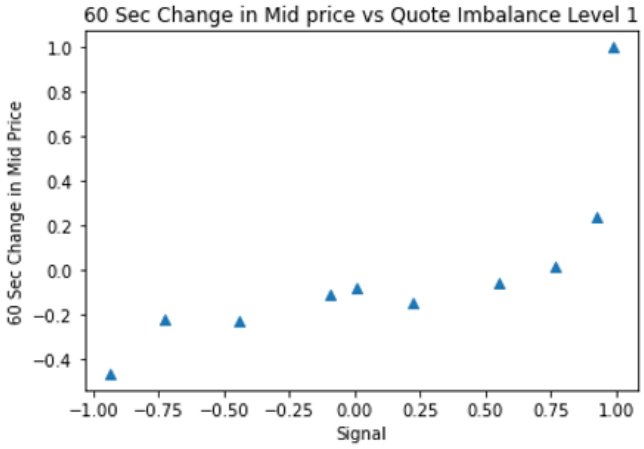
Based on the literature titled 'Algorithmic and High-Frequency Trading' (Cartea et al., 2015), we define quote imbalance as follows:

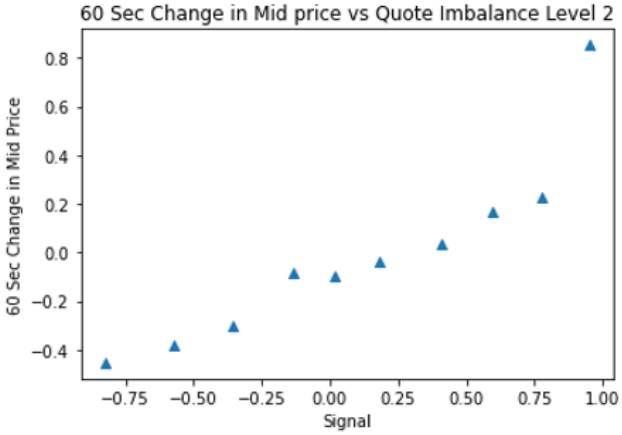
$$QI_{L=k} = \frac{\sum_{i=1}^{k} V_b - \sum_{i=1}^{k} V_a}{\sum_{i=1}^{k} V_b + \sum_{i=1}^{k} V_a}$$

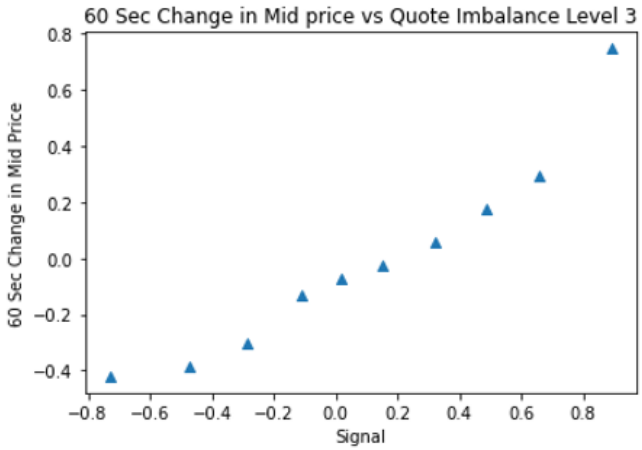
where k is the number of levels of quote data used in the definition and only levels with volumes are included in the quote imbalance, V_b represents the bid volume, and V_a represents the ask volume.

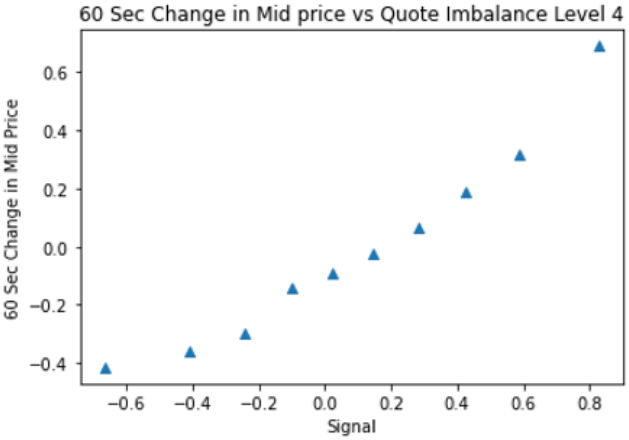
Quote imbalances range from -1 to 1. Generally, when the quote imbalance is close to -1, there is selling pressure in the market, which should result in a decline in the

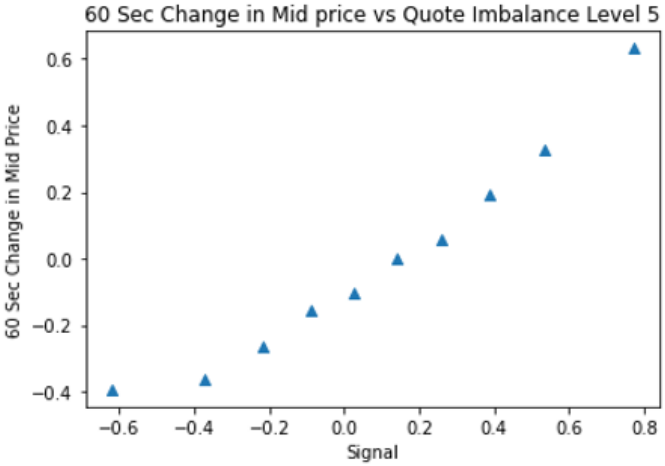
mid-price over the near term, while when the quote imbalance is closer to 1, there is a buying pressure, which may lead to an increase in the mid-price.

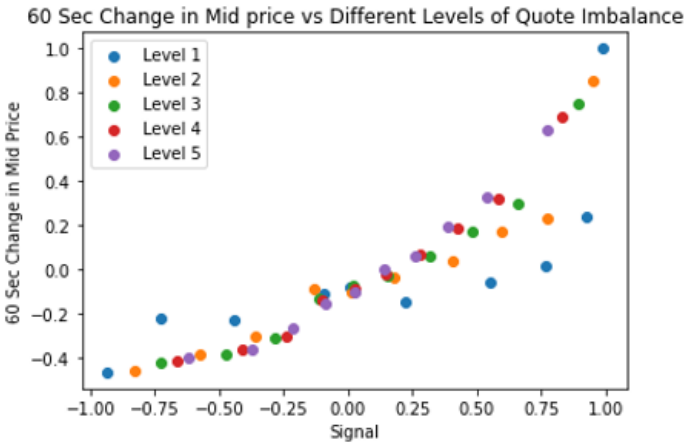










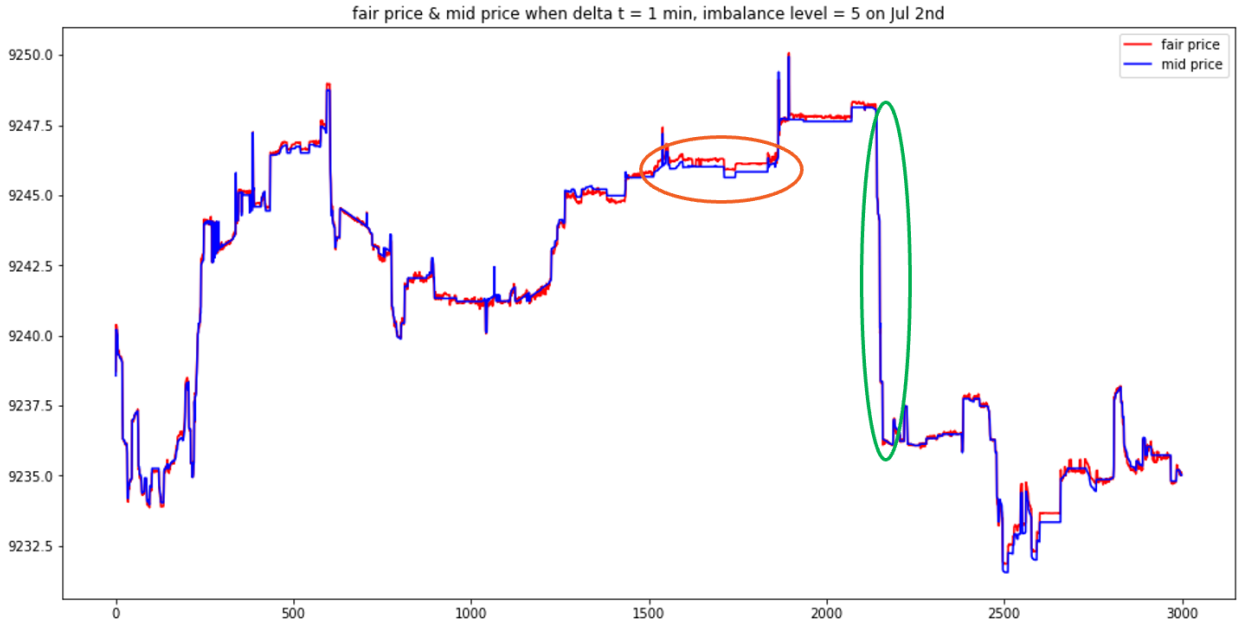


The charts above illustrate the signal plots of quote imbalance, where the y-axis represents 60-second changes in mid-price, while the x-axis represents quote imbalance at different levels from 1 to 5, fit into ten deciles. The plots show that there is an ambiguous relationship between 60-second changes in mid-price and quote imbalance when looking at levels 1 and 2. With the addition of deeper levels of data, the relationship gradually becomes more linear. Using level 5 in the plot is evidently linear enough to move ahead with, so we have decided to model quote imbalance adjusted mid-price up to level 5.

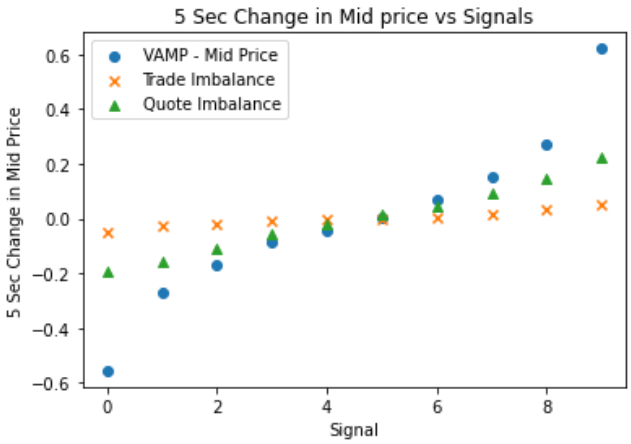
The following is the optimization model for quote imbalance adjusted mid-price:

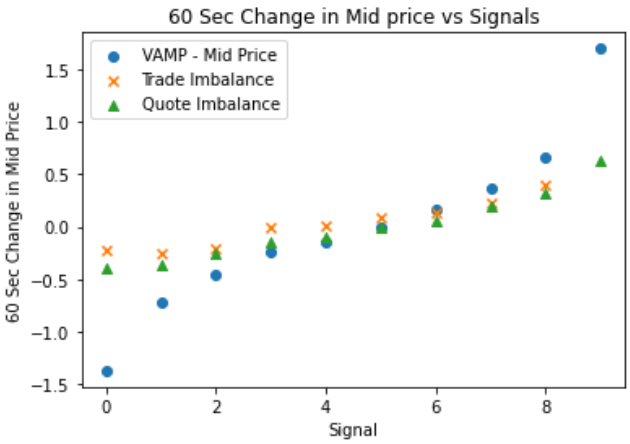
$$\min_{\alpha} \Sigma (F_t - Mid_{t+\Delta t})^2$$
$$\text{s.t.} \quad F_t = Mid_t + \alpha QI_{L=k,\,t} \;;$$
$$k = 1, 2, 3, 4, 5 \;;$$
$$\Delta t = 1s, 5s, 15s, 30s, 1min$$

where F_t is the fair price at time t, Mid_t is the mid-price at time t, k is the number of levels of quote data used, and Δt is the look-ahead window. For the purpose of determining the coefficient $\boldsymbol{\alpha}$ of quote imbalance, we minimize the objective function that represents the Mean-Squared Error (MSE) between the fair price and the mid-price.



Here is a simple visual demonstration of how fair price prediction works using short-term data. The training data is based on July 1st 2020, and the testing data is based on July 2nd 2020. Taking a closer look at the chart, we find that there are small jumps in the fair price prior to jumps in the mid-price in the same direction, which indicates that the fair price has a certain degree of predictive power, for instance, the part circled in orange. However, during other periods, the fair price appears to simply follow the mid-price rather than predicting significant events and does not appear to indicate large drawdowns, for example, the part circled in green.





We show above our three signals compared on decile plots. Here, similar to the earlier quote imbalance signal plots, we split each signal into their respective 10 deciles as the x axis and the corresponding future change in mid-price on the y axis. These plots give us a good indication of a potential hierarchy of signals, showing a weak signal in trade imbalance and a strong signal in both quote imbalance and (VAMP - mid-price). You will notice that the (VAMP - mid-price) signal does not resemble a strictly linear relationship when plotted this way, which is in theory an optimistic indication that the signal changes more proportionally to changes seen at the two tails of the price change distribution.

Preliminary Metrics

So far, we have introduced 3 definitions of our fair prices with different ingredients added in through feature engineering, namely, VAMP, trade imbalance adjusted mid-price, and quote imbalance adjusted mid-price. In order to get better insight and understanding of how good these fair prices are as well as their advantages and shortcomings, we have come up with three preliminary metrics to consider.

In the following subsections, we compare the fair prices' performances on 3 metrics: volatility, mean-squared error and trading PnL. Considering time efficiency and the performances' stability throughout the whole time period, here we have used the quote and trade data from July 1 to July 19, 2020 as the training data, from which we calculate the optimal parameters within each fair price definition. Our testing period is July 20, 25, and 30, 2020, accordingly, and the look ahead windows are set to be 1, 5, 15, 30 and 60 seconds.

Volatility

The volatility is defined as follows:

$$Volatility_{\delta} = \sum_{t}(F_{t} - F_{t+\delta})^{2}$$

where δ is the given look ahead window. This metric calculates the sum of squared differences between fair price at time t and t+δ, which reflects the fluctuation within the fair price time series itself in the testing period. Theoretically, a good definition of fair price should reflect the intrinsic value of the underlying, and thus have lower fluctuation within itself compared to other fair price definitions. Therefore, a lower volatility is more desirable, meaning a more precise approach in capturing the intrinsic value.

The volatility for different fair price estimates with mid-price's volatility as the benchmark are shown in the table below:

Volatility	**Mid Price**	**VAMP**	**TI Adjusted Mid**	**QI Adjusted Mid**
1s window	**0.271**	0.311	0.417	0.273
5s window	**1.483**	1.654	2.074	1.492
15s window	**5.095**	5.435	6.362	5.135
30s window	**10.739**	11.187	12.641	10.833
60s window	**22.057**	22.484	24.614	22.242

First of all, we can see that mid-price itself doesn't change very much within the 60 second time window on average, showing a relatively stable movement pattern in the short term. In addition, the mid-price volatility as a benchmark always has the lowest volatility in all look-ahead windows, while trade imbalance adjusted mid-price performs the worst. This can be explained by the nature of these two prices: mid-price itself as an average price usually changes less, while the trade imbalance also captures the volume

imbalance when certain trades have happened, and these imbalances are usually more volatile.

Mean-Squared Error

The MSE is defined as follows:

$$MSE_{\delta} = \frac{1}{T}\sum_{t}(F_{t} - M_{t+\delta})^{2}$$

This metric calculates the mean of squared difference between fair price at time t and future mid-price at time t+δ. Different from volatility introduced above, MSE focuses more on the predicting accuracy of the fair price. That is, it tells about how precise the fair price is in terms of predicting mid-price at a specified point in the future. This metric reflects the characteristics that are most directly related to whether a fair price definition is a good estimator and can beat mid-price itself, since our final goal is to find the fair price that is most reliable on classification accuracy. Our goal here is to minimize the objective relative to the other estimators.

The MSE for different fair prices are shown in the table below:

MSE	**Mid Price**	**VAMP**	**TI Adjusted Mid**	**QI Adjusted Mid**
1s window	**0.271**	0.944	3.739	0.273
5s window	**1.483**	1.905	5.403	1.492
15s window	**5.095**	5.164	9.627	5.150
30s window	10.739	**10.568**	15.385	10.948
60s window	22.057	**21.737**	26.319	22.892

Here is one thing that is worth emphasizing: for 1s, 5s and 15s time windows, the mid-price itself has the lowest MSE. However, when it comes to 30s and 60s, the VAMP has the lowest MSE. This result shows an early promise in VAMP as a probable better fair price definition compared to mid-price itself at least at the relatively longer look-ahead windows. It demonstrates the trend of being more accurate in future mid-price predicting, which gives us expectation that VAMP may have relatively significant performance in binary classification accuracy and multiclass classification accuracy that are further discussed in later sections.

Trading P&L

We introduce another evaluation metric called Trading PnL. This metric provides an indication of how well our models predict the directional changes in mid-price, and how quickly the predicted fair price is responding to changes in the order book. The metric is structured as follows: we define three ‘positions’, long/short/neutral. At each second, the difference between the predicted fair price and the true mid-price works as the ‘signal’. We then compare every signal to a ‘threshold’ to capture the directional change when the fair price gets far away from the mid-price. Here, the threshold is set to be the standard deviation of the differences between the predicted fair and true mid-price over the whole testing period. Throughout the iterative process, we keep track of the number of trades executed and calculate the return of each trade as (exit point price - enter point price)/enter point price after each trade.

For any testing period, we start with position neutral, cumulative return=1, and number of trades=0. Then, we iterate over the signal at each second:

- If signal > threshold: enter long position, total number of trade+=1;
- If signal < -threshold: enter short position, total number of trade+=1.

If we are already in a long position:

- If signal < - threshold <: exit long position and enter short position, total number of trades += 2, cumulative return *= (exit point price - enter point price)/enter point price;
- If otherwise: position not changed.

If we are already in a short position:

- If signal > threshold: exit short position and enter long position, total number of trades += 2, cumulative return *= (exit point price - enter point price)/enter point price;
- If otherwise: position not changed.

Note that since the use case of the metric is not trading itself, it is unnecessary to look for an optimal value for the threshold. Instead, It could be an arbitrary number, but not too small such that the trade percentage is large enough under this condition to give a clear indication of predicted price’s directional movement.

The test results evaluated by Trading PnL are shown in the table below.

(Trading PnL Cumulative return, Number of Trades)	VAMP	TI Adjusted Mid	QI Adjusted Mid
1s window	**1.2543, 1973**	1.0018, 1502	1.0663, 2287
5s window	**1.2543, 1973**	0.9837, 1659	1.0212, 1368
15s window	**1.2543, 1973**	0.9888, 1122	1.021, 1353
30s window	**1.2543, 1973**	0.9883, 799	1.021, 1339
60s window	**1.2543, 1973**	0.9939, 563	1.0152, 1324

As we can see, VAMP dominates the other fair price estimators over the three trading days, generating the largest return for each second window. This result suggests that the VAMP model may be best at responding to different pressures in the order book and predicting short-term price direction. We will confirm this, now, loosely held hypothesis in the following sections.

Binary Classification Results

Up until this point we have examined our different definitions of fair price against metrics that only give us a partial view of their respective predictive powers. MSE is a good metric for overall accuracy but gives very little indication of how well directions of price moves are predicted. Our trading strategy metric began to answer this question, but in a way that is not as interpretable as we want.

We now want a metric that can evaluate how well our fair prices predict the direction of future price movements from 1 second to 60 seconds in the future. The first of the final two metrics we will use to further evaluate is formed in the context of a binary classification problem. We define our x and y vectors as follows:

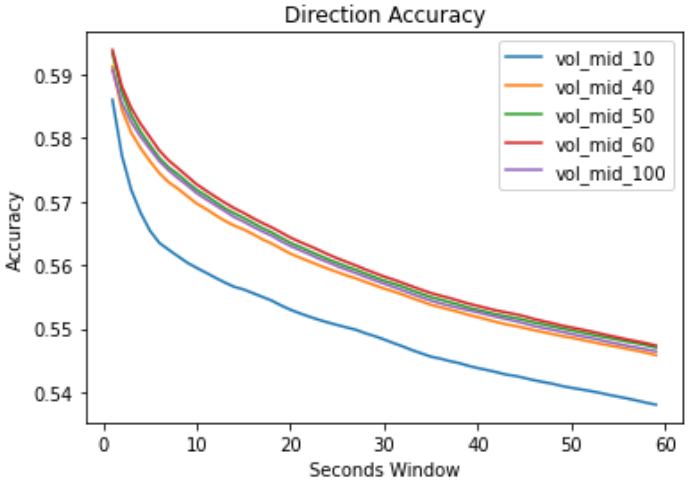
$$x = \begin{cases} -1 \; if \; F_t < Mid_t \\ 1 \; if \; F_t > Mid_t \end{cases}$$

$$y_\delta = \begin{cases} -1 \; if \; Mid_t > Mid_{t+\delta} \\ 1 \; if \; Mid_t < Mid_{t+\delta} \end{cases}$$

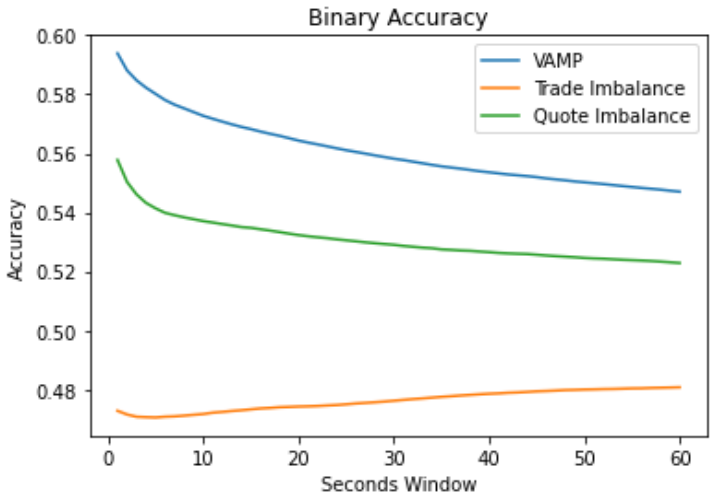
where $F_t \; and \; Mid_t$ are the fair price and mid-price at time t, respectively, and δ is the seconds look ahead window. Note that both conditions in each piecewise are strict inequalities as we only want to look at instances when a change in price is predicted and when it actually happens. Outlining a classification problem this way allows us to evaluate each metric on how well they predict direction of price movements in an interpretable way. We run much of our analysis on individual days randomly taken from July, August and September with very similar results, but to best get a sense of the robustness of our signals we aggregate our three months of data, using the month of July as a coefficient "training" window for both trade and quote imbalances, and use the months of August and September as our "testing" window (which the following results are from).

We evaluated each fair price for $\delta \in (1, 60)$ and plotted accuracy scores (see Appendix for plots of recall, precision, and F1 scores). We determined that accuracy was the most important metric for the binary classification because it best shows how well our fair price predicts price movements when they actually happen.

Before comparing across definitions, we will first use this accuracy metric to narrow down at which volume we use as our final VAMP, first determining $50,000 as the best performer compared to the first four cutoffs and then plotting the accuracy of $10,000, $40,000, $50,000, $60,000, and $100,000 (see Appendix).



As we can see, there is very little difference in prediction accuracy in the $40,000, $50,000, $60,000, and $100,000 levels. It seems $60,000 is slightly better than the rest so we choose this as the cutoff for the VAMP we use for the rest of our analysis.

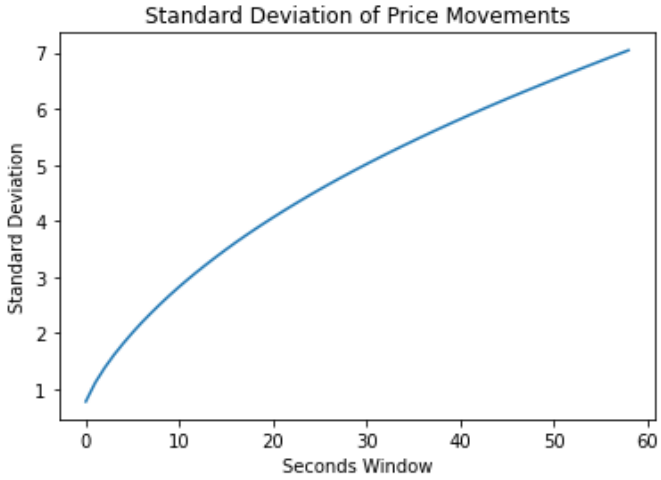


The results here mirror what we saw from the trading strategy PnL in the previous section. Trade imbalance shows behavior no better than a trivial classifier while quote imbalance has accuracies that range from 56% at the 1-second and 52% at the 60-second level. But similar to the previous section, VAMP dominates the quote imbalance definition, ranging from over 59% to just under 56% accuracy, for an average of 4% accuracy improvement at each time window.

The binary classification metric paints a clear picture that VAMP is a strong predictor of future price direction, but still leaves room for things to be desired. We will next look at evaluating our fair prices in the context of a multiclass classification problem to further clarify how well they predict price movements.

Multiclass Classification Results

Although binary classification paints the clearest picture thus far, it does not give us much indication of how well our predictors predict large, infrequent price movements—often very important events in finance. Predicting small movements is captured well by the binary classification as 84% of the price movements are within one standard deviation of the mean (see Appendix for plot of price change distribution). But we want our fair price definition to predict these large price movements as well. To evaluate this, we first define a “large” price movement up or down as one standard deviation in either direction (60 total, one for each seconds window δ).

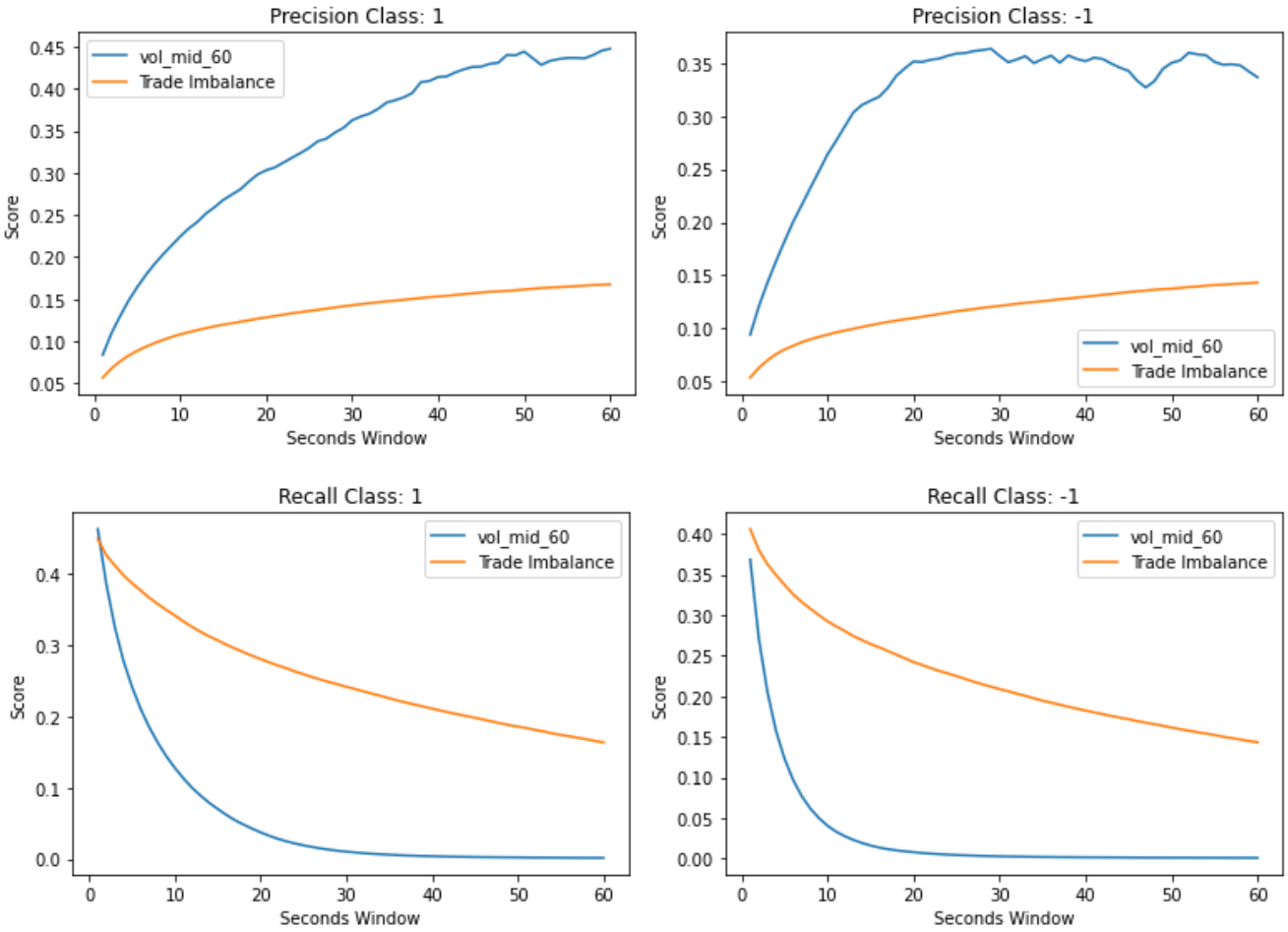


We define a multiclass classification problem with three classes defined as follows:

$$x_\delta = \begin{cases} -1 \; if \; F_t \leq Mid_t - \sigma_\delta \\ 1 \; if \; F_t \geq Mid_t + \sigma_\delta \\ 0 \; otherwise \end{cases}$$

$$y_\delta = \begin{cases} -1 \; if \; Mid_{t+\delta} \leq Mid_t - \sigma_\delta \\ 1 \; if \; Mid_{t+\delta} \geq Mid_t + \sigma_\delta \\ 0 \; otherwise \end{cases}$$

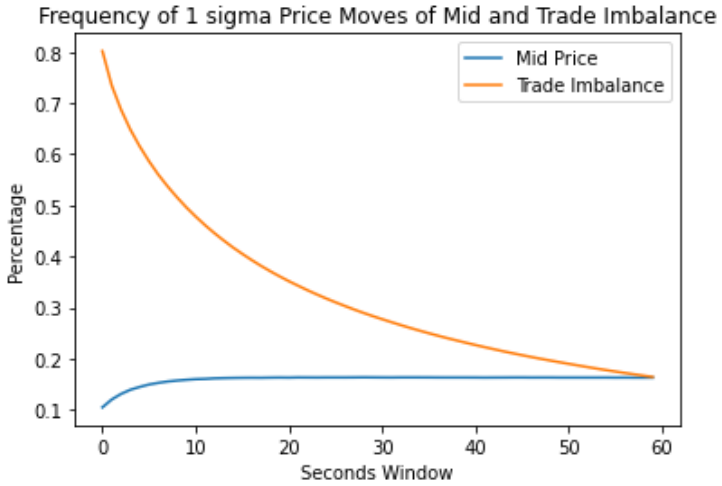
Defining the problem this way allows us to isolate these one-sigma price movement events. Our goal and desired result here is not the same as for the binary classification—we now want to know, when our fair price predicts a one-sigma event, how often is it correct? As we already have a predictor in VAMP that classifies general price direction well, a prime candidate for a combined model would be a predictor that, when signaling a large movement coming, allows our model to react and capture it. To answer this question, we turn to precision scores of the individual classes, but will also take a look at recall scores as they present informative findings (which is also accuracy when evaluating on the per-class level).



One thing easily noticed from these plots is that quote imbalance is missing. This is because for the multiclass problem, the quote imbalance definition becomes a trivial classifier as it is never more than $0.50 away from the mid-price, and never predicts a one-sigma move as the standard deviation at the one-second level is $0.80. We believe this happens because of the way we optimize the beta parameter when engineering the fair price. Optimizing on MSE causes the quote imbalance definition to be close to the mid-price at every time t because price changes are typically small with one-sigma events not happening very often.

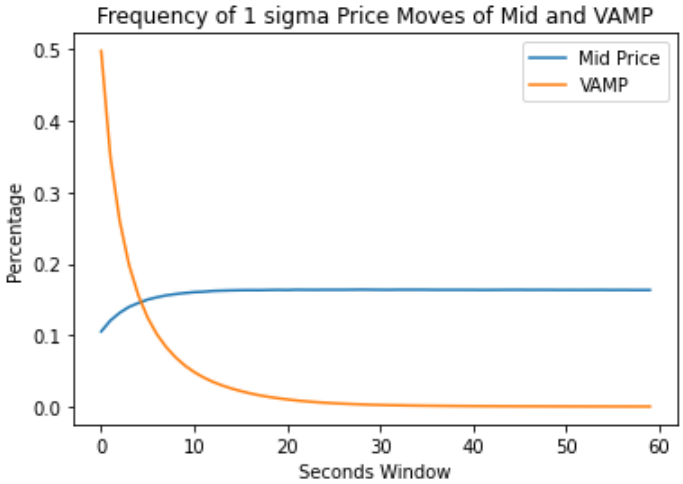
We next move to the performance of the trade imbalance adjusted mid-price. Upon looking at the recall scores, one may come to the incorrect conclusion that trade imbalance is a great indicator of large price movements and presents an obvious candidate to combine with VAMP, but as we already know, precision is the measure that best represents the answer to our main question regarding multiclass: when we predict a large movement, how often are we right? The reason trade imbalance performs so

poorly is because it predicts one-sigma price moves much more often than they actually occur, as seen in the following plot:



Because trade imbalance, especially at the smaller seconds windows, predicts large movements much more often than they actually occur, precision and recall scores heavily disagree. But even at the 60-second window when prediction and occurrence rates are similar, trade imbalance still has a very poor precision score.

VAMP is still the dominant predictor when evaluating on precision, but not quite as triumphant of a winner as in the binary classification case. Larger than a 15-second window, precision on both classes went to around 35% with the max score on upward moves being just under 45%. The precision scores curve can also be explained by a comparison of frequencies:



Here we see the opposite problem past 5-seconds. Similar to the quote imbalance adjusted mid-price, the VAMP does not predict large price movements frequently enough, and when it does, it is only correct 25%-45% of the time.

Conclusion and Future Work

We have seen through multiple metrics that the VAMP is a good indicator of future price movements in this BTC market. Quote and trade imbalance definitions of fair price struggled to compete for significant reasons, made clear by our two classification frameworks. They either struggled in predicting the general direction of price movements or in accurately predicting large, one-sigma events. The VAMP outperformed them in both arenas, showing itself as a strong indicator for price direction, but a definition that needs work as a reliable signal for future large price movements.

Future expansion of the ideas and conclusions presented in this study could go in many directions. One most obvious direction that we would have liked to have taken this analysis (given much more time) is expanding the data in both width and breadth. Performing this analysis on more diverse BTC data could yield interesting results, especially if that data captures events like the recent crypto crash and other highly volatile conditions. Also expanding this analysis to more securities and markets would give a much better sense of the robustness of the VAMP and whether this measure generalizes to not only other crypto assets, but other asset classes as well.

Another way to improve this work is to revisit quote imbalance from a possibly different lens, whether that is changing the way the beta parameter is optimized or weighting larger movements more heavily in the optimization. This was not necessarily a quick fix in our analysis as we did not want to overfit our models to extreme events when this would exclude nearly 90% of the data.

Overall we believe that the VAMP is a strong and robust indicator of price movement direction across the different market microstructure conditions seen from July to September 2020, with potential uses as a parameter in the lowering execution costs, optimal execution algorithms, and proprietary trading strategies. We hope that our analysis will be considered in a practical, financial application at some point in the future.

References

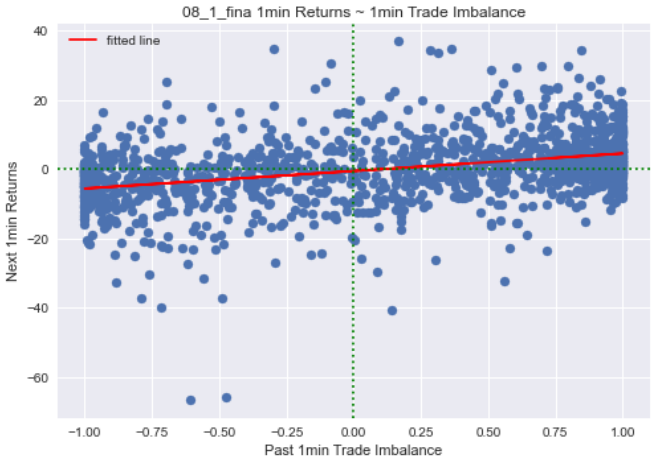
Cartea, Á., Jaimungal, S., & Penalva, J. (2015). Algorithmic and high-frequency trading. *Cambridge University Press.*

Stoikov, S. (2018). The micro-price: a high-frequency estimator of future prices. *Quantitative Finance*, *18*(12), 1959-1966.

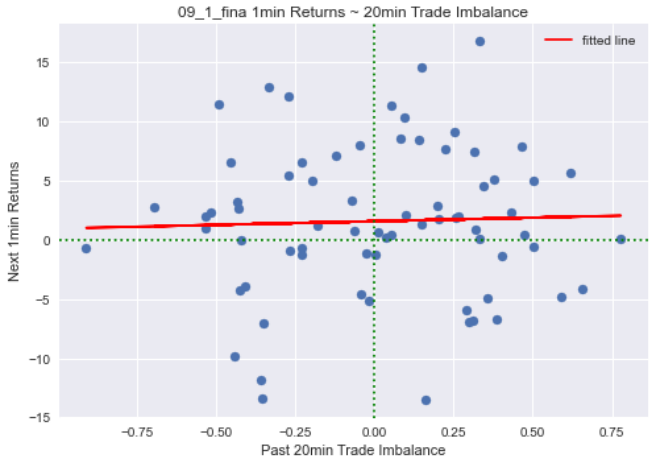
Zheng, B., Moulines, E., & Abergel, F. (2012). Price jump prediction in limit order book. *arXiv preprint arXiv:1204.1381*.

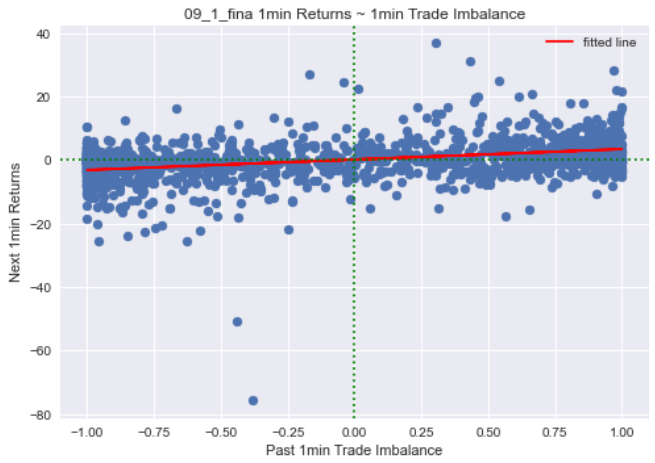
Kolm, Petter N., Turiel, J., & Westray, N. (2021). Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book. *http://dx.doi.org/10.2139/ssrn.3900141*

Appendix


08_1_fina 1min Returns ~ 1min Trade Imbalance
fitted line
Next 1min Returns
Past 1min Trade Imbalance

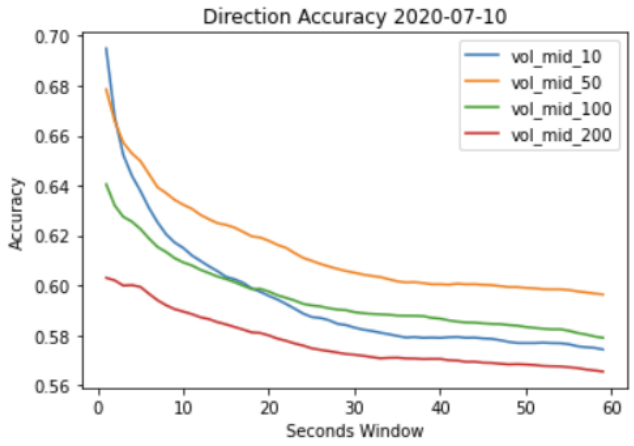

09_1_fina 1min Returns ~ 10min Trade Imbalance
fitted line
Next 1min Returns
Past 10min Trade Imbalance

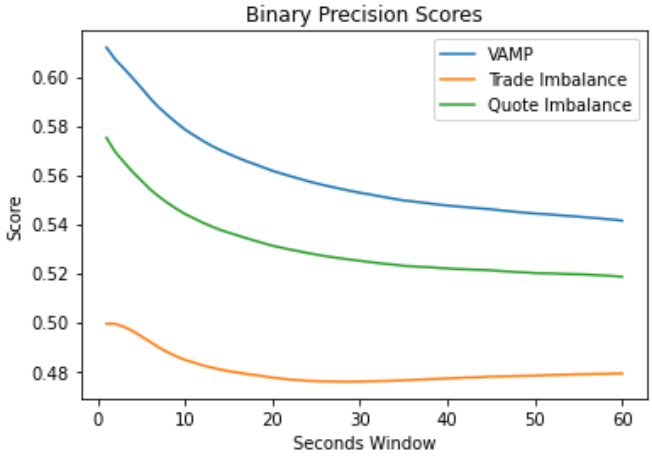

09_1_fina 1min Returns ~ 20min Trade Imbalance
fitted line
Next 1min Returns
Past 20min Trade Imbalance

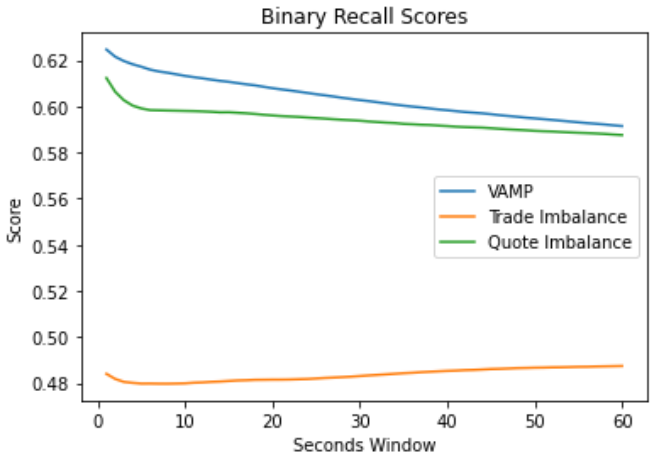

09_1_fina 1min Returns ~ 1min Trade Imbalance
fitted line
Next 1min Returns
Past 1min Trade Imbalance

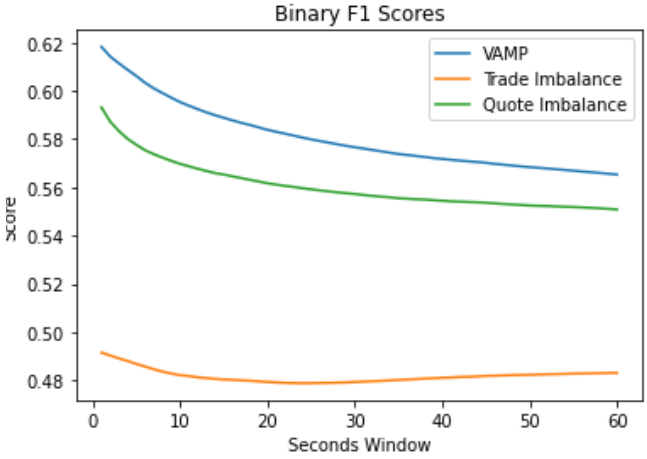

08_1_fina 1min Returns ~ 10min Trade Imbalance
fitted line
Next 1min Returns
Past 10min Trade Imbalance

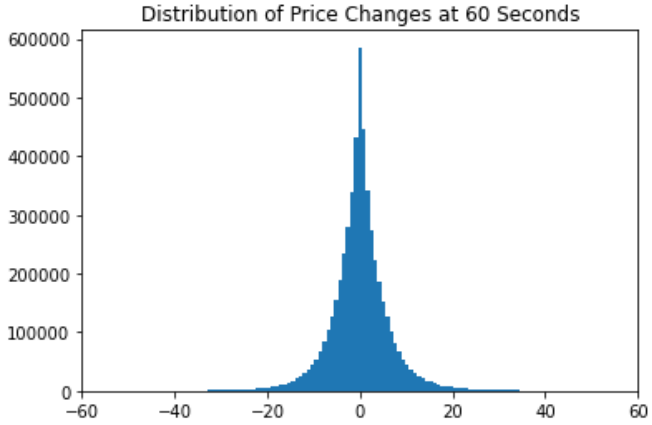

08_1_fina 1min Returns ~ 20min Trade Imbalance
fitted line
Next 1min Returns
Past 20min Trade Imbalance


Direction Accuracy 2020-07-10
vol_mid_10
vol_mid_50
vol_mid_100
vol_mid_200
Accuracy
Seconds Window


Binary Precision Scores
VAMP
Trade Imbalance
Quote Imbalance
Score
Seconds Window


Binary Recall Scores
VAMP
Trade Imbalance
Quote Imbalance
Score
Seconds Window


Binary F1 Scores
VAMP
Trade Imbalance
Quote Imbalance
Score
Seconds Window


Distribution of Price Changes at 60 Seconds