Detection of algorithmic trading

Dimitar Bogoev[a], Arzé Karam[b,*]

[a] Durham University, Mill Hill Lane, Durham DH1 3LB, England.
[b] Durham University, Mill Hill Lane, Durham DH1 3LB, England.

Abstract

We develop a new approach to reflect the behavior of algorithmic traders. Specifically, we provide an analytical and tractable way to infer patterns of quote volatility and price momentum consistent with different types of strategies employed by algorithmic traders, and we propose two ratios to quantify these patterns. Quote volatility ratio is based on the rate of oscillation of the best ask and best bid quotes over an extremely short period of time; whereas price momentum ratio is based on identifying patterns of rapid upward or downward movement in prices. The two ratios are evaluated across several asset classes. We further run a two-stage Artificial Neural Network experiment on the quote volatility ratio; the first stage is used to detect the quote volatility patterns resulting from algorithmic activity, while the second is used to validate the quality of signal detection provided by our measure.

Keywords: `algorithmic trading patterns, quote volatility, price momentum, Artificial Neural Network`

1. Introduction

Over the past decade, technological innovations and changes in financial regulation, e.g. Regulation National Market System in the US, and the MiFiD in Europe, have induced trading to become more automated. This evolution led to changes in the way the information is disseminated to traders. Specifically,

*Corresponding author
Email address: `arze.karam@durham.ac.uk` (Arzé Karam)

automated traders react fast to events and a subset of algorithmic traders, i.e. high-frequency traders (HFTs hereafter), exploit this feature [1].

Concerns have been expressed on the growth of algorithmic traders and their effects on the ability of financial markets to efficiently perform their functions, such as risk sharing. Currently, market regulators explore methods to monitor the activity of these fast traders, and their effects on financial markets see for [2] for a literature review. For instance, the Commodity Futures Trading Commission employ expensive methods to monitor commodities and derivatives trades drawing upon complete data of many levels of order books. We propose a method to identify patterns of algorithmic activity that requires only anonymous and top-of-book information extracted from public data and can thus simplify the process. Further, researchers and practitioners measure algorithmic trading by using data on submitted orders at many levels and the speed at which these orders are submitted. For instance, [3] use the ratio of executions to order submissions, and document that this ratio is lower when algorithmic traders are present in the market. This ratio is widely used by the literature to proxy for algorithmic trading, see for [4] among others. Further,[5] use the fact that the cancellation of a limit order by a trader following by the resubmission of another order by the same trader (a linked message) in less than one second is likely to come from algorithmic traders. Deferring from these measures, our measures use price patterns and can be useful to track the effects of algorithmic activity in the millisecond environment, rather than only the presence of algorithmic traders in the market. To prove the suitability of our measures, we test them on three different assets: the Apple stock, the Bund futures, and the US ETF Oil.

The first contribution is to provide an analytical and tractable way to infer patterns of quote volatility and price momentum. We propose two ratios to quantify these patterns. We discuss how the observed patterns are consistent with different types of strategies employed by algorithmic traders. Our first ratio, namely quote volatility, captures the rapid change of price quotes and expressed by the rate of oscillation of the best ask and the best bid over short period of time. There are many reasons why algorithmic traders might adjust their quotes and stop, thus causing quote volatility. For instance, two or more algorithmic traders may compete by submitting limit orders at the top of the book and engage in several rounds of updates by undercutting each other quote [6]. They might be repeatedly offering the best quote that another trader is frequently filling. Another example is quote stuffing, a strategy that consists in increasing the number of order submissions followed by cancellations. These two examples of behavior, undercutting behavior and quote stuffing, will likely to have the effect of increasing quote volatility and execution costs. We identify episodes of rapid changes in price quotes with specific patterns occurring over short period of time, i.e. over 1-2 seconds. We further consider different specifications, when aggressive quoting occurs at the best ask (in-ask), at the best bid (in-bid), or at both sides of the market (combined).

Our second ratio, namely price momentum, denoted by PM, identifies patterns of price momentum following upward and downward price movements over two minutes on average. Algorithmic traders react fast than humans to the information contained in the limit order book updates, and news announcements [7] or order anticipation [8], and try to exploit it quickly to generate profits. Their activity exacerbates a directional price move by contributing to price volatility. For the two measures, we apply a filtering technique to the data by selecting the observations containing the top percentile of the measures.

Our second contribution is to provide a novel Artificial Neural Network (ANN hereafter) using the quote volatility ratio. The patterns discussed above have a long history in financial markets and they have been extensively discussed in the market microstructure. What is novel is the intensive use of information technologies to implement these strategies and the way they are implemented. On this, very little information is available because algorithmic traders see the implementation of their strategy as the source of their competitive advantage and naturally hide their algorithms. We further demonstrate a useful technique (neural nets) that can accurately identify a defined set of quote volatility patterns consistent with an interesting group of strategies employed by algorithmic traders. Specifically, we run a two-stage ANN experiment using the quote volatility ratio: the first stage is to detect the patterns of quote volatility; and the second stage is to validate the quality of signal detection by the ratio for all the specifications and at different threshold levels. ANN results suggest that quote volatility ratio appears to be a good filter for signals, and an increase of the ratio threshold seems to improve the detection in ANN but only for some levels.

2. Measures of algorithmic trading

In this section, we detail the two measures we use to identify patterns of algorithmic trading. To analyze events, we use the method of rolling time-frames with overlap. Since the data points are unevenly distributed in time, an algorithm is used to collate them into subsamples, referred to as windows hereafter, spanning a specified time length. Therefore, each data point serves as a starting point for a window which includes a number of data points which fall within a pre-specified time from the first one. The time window framework allows for statistics to be estimated for each of the rolling subsample. This simplifies the task of detecting the time intervals containing algorithmic trading activity to designing statistics which capture the similarity of the patterns observed in a given window to that of typical algorithmic trading patterns. The one arbitrary element in this approach is the length of time frames examined. Market observations provide some hints for suitable time frames, see [9].

2.1. Quote volatility

The first measure is based on the rate of oscillation of the best bid and best ask quotes detected over a very short period of time, typically lasting several seconds. During this time, rapid and transient quote updates occur, often following several specific patterns. Certainly, quotes submitted on the limit order book that move faster than human capacity are generated by algorithmic traders. There are many reasons why algorithms might adjust frequently their quotes and then stop, thus causing volatility. For instance, two or more algorithmic traders may compete by submitting their limit orders at the top of the book and engage in several rounds of quote updates by undercutting each other quotes, see [6]. Or, one algorithmic trader might be repeatedly offering a quote either at the ask or at the bid for a small quantity that another algorithmic trader is frequently filling. Rapid small fills on short-lived orders were observed throughout the October 2014 flash crash event on BrokerTec. Alternatively, predatory behavior induces quote volatility. For instance, algorithmic traders would display a large amount of orders then cancel them quickly. This practice is intended to entice institutional traders into trading by creating an illusive liquidity.

The rapid oscillation of quotes can either occur at the bid side, the ask side of the market or simultaneously at both sides of the market. In order to detect these patterns in the sample window, a ratio denoted by QV is estimated. The QV ratio is a geometry based metric that is inspired by the graphical presentation of quote oscillation on a chart. All the episodes share several key characteristics irrespective of the particular pattern: (i) they include small rapid movements in the bid, ask or both levels, which are subsequently rapidly reversed; (ii) this is repeated many times, over a small time frame; (iii) over the span of the entire time frame, the actual direction movement in the quote levels is low, if any.

Let j denote the window index that includes the ask and bid quotes denoted by A and B respectively. The QV ratio has four components: *Carryask*, *Bigask*, *Carrybid* and *Bigbid*. *Carryask* is the sum of absolute incremental (instant-by-instant) changes in the ask price over the period: $Carryask = \sum_{i=2}^{j} |Ai - A_{i-1}|$; *Bigask* is the absolute change in the ask price level between the starting and the ending points of the period examined: $Bigask = |A_j - A_1|$; *Carrybid* is the sum of absolute incremental (instant-by-instant) changes in the bid price over the period: $Carrybid = \sum_{i=2}^{j} |Bi - B_{i-1}|$; *Bigbid* is the absolute change in the bid price level between the ending and the starting point of the period: $Bigbid = |B_j - B_1|$. These variables are used to compute three alternative specifications of the QV ratio:

Ask specification aims at detecting the in-ask quoting activity. This implies rapid quote volatility at the ask side, and a relatively passive bid side. The specification is:

$$QV_{ask} = \frac{\frac{carryask}{bigask}}{\frac{carrybid}{bigbid}} \tag{1}$$

Bid specification aims at detecting quote volatility which occurs at the bid side of the market. This is characterized by a rapid quote volatility at the bid side, while the ask price remains relatively inactive. The specification is given by:

$$QV_{bid} = \frac{\frac{carrybid}{bigbid}}{\frac{carryask}{bigask}} \tag{2}$$

In order to guarantee the function's solutions domain, several special cases are defined: if *bigask*=0, it is instead set at the level of the minimum tick increment at 0.01; if *bigbid*=0, it is instead set at the level of the minimum tick increment at 0.01; if carrybid=0, the entire denominator $\frac{carrybid}{bigbid}$ or $\frac{carryask}{bigask}$ is set to equal to 0.01. This ensures that a corresponding *QV-ratio* can be calculated for any given window.

Combined specification aims at detecting quote volatility activity of the combined type which occurs at both the ask and the bid sides of the market. This is characterized by a period of high quote volatility which occurs over a short period of time, but it is driven by transient movements. The specification is given by:

$$QV_{combined} = \frac{carryask}{bigask} + \frac{carrybid}{bigbid} \tag{3}$$

For the purpose of solutions domain considerations, several specific cases are predefined. If $bigask = 0$, it is instead set at the minimum incremental tick size at 0.01. If $bigbid = 0$, it is instead set at the minimum incremental tick size at 0.01.

After the QV ratio values for each window in the sample have been calculated, the final step of the detection is to determine which ones are indicating a potential period of algorithmic activity. Since a unique window is associated with each data point in the sample, and a QV ratio value is associated with each window; it is possible to use the observed distribution of QV ratio values over the entire sample, and subsequently select a cut-off point for the most promising ones. The Trident tool supports a user specified cutoff point. Once a QV value has been estimated for each window in the sample, the entire array of QV values is ordered in increasing order. A specified percentage is then applied to select the cutoff point. This is done via the below formula:

$$QV_{cutoff} = QV_{arraysize-(rounddown(percentile*arraysize))} \tag{4}$$

The cutoff determined using this technique has the major benefit of coming from the distribution observed within the actual data, rather than an arbitrary level selected. A higher QV ratio should indicate higher likelihood of algorithmic activity. Once the cutoff is determined, it is used to filter out only the windows which have a QV ratio value above the cutoff point.

2.2. Price Momentum

Price momentum arises as a reaction in the market to news events, such as release of an earning report by a company, a macro announcement or changes in market conditions. The pattern of short-term volatility followed by price reversal would be detected. Algorithmic traders can process the new information or the signal faster than humans even if it is already public, and could trigger the pattern of momentum to take advantage of the volatility surrounding the information release in an extremely short period of time. [7] show that algorithmic traders take advantage of a news event in the subsequent few seconds of its public release. They do so by taking a directional bet in one asset in anticipation of an impeding price change related to news events. In addition, their fast access could allow algorithmic traders to detect order splitting strategy by large traders, see [8]. Specifically, the authors show that algorithmic traders anticipate orders submitted by large traders, and mimic these orders. As shown by [10], traders who infer the presence of an aggressive large trader have an incentive to initially trade in the same direction to amplify the downward pressure. Finally, algorithmic traders have also been accused to engage in price manipulation. For instance, they might place buy (sell) market orders in the expectation that other traders would do the same. The buying (selling) pressure might then push prices up (down), allowing them to liquidate their positions at profits. This practice known as momentum ignition might cause similar patterns, to those of directional strategies and order anticipation, i.e. upwards or downwards price momentum.

Therefore, the second measure we propose is based on detecting specific price patterns during upwards and downwards price movements. This usually comprises three main stages: (i) an initial spike in trading volume, which is not accompanied by any significant changes in price; (ii) a subsequent sharp price move (positive or negative), accompanied by a new, even larger increase in volume; (iii) gradual price reversal to levels observed before the event, accompanied by low volume.

This pattern may last for several minutes. It is still prevalent in most traded instruments at least once per day with higher activity in certain sub sectors in the market. The duration of the events, as well as their market impact appear to follow a fat-tailed distribution, with a small fraction of events having major market impact and lasting for a prolonged period of time. This has a directly observable economic impact, which can be measured in relative terms (size of the price move in basis points), or, potentially even in absolute price change multiplied by the estimated position of algorithmic trading.

The characteristic pattern of price momentum includes two dimensions: trade prices as well as trading volume. Let the PM ratio denotes the ratio used for price momentum detection. For the sake of computation efficiency, only trade prices are used as an input in the PM ratio specification used to detect the patterns of this algorithmic activity. Further, it is assumed that, as consistent with previous empirical studies of financial markets, the distribution of asset returns exhibits leptokurtosis see for instance [11]. Therefore, a small fraction of windows will contain the large moves relevant for detecting price momentum. The distribution derived strategy and the specification of the PM ratio used to ensure that the biggest relevant price moves present in the data are examined.

The PM ratio used for price momentum detection is based on 3 key inputs: *StartPrice* is the Trade price in the starting point of the time period; *EndPrice* is the Trade price of the final trade in the time period; and, *PriceSpan* defined as $|EndPrice - StartPrice|$. If this turns out to be 0, then it is set to 0.01 instead for domain purposes. As with quote volatility, the ratio estimated is inspired by the geometry of a graphical representation of the pattern. In the case of price momentum, this involves estimating two distances for each trade (t) in the window: $PM1_t = |P_t\text{-Start Price}|$ and $PM2_t = |P_t\text{-End Price}|$.

These metrics are used to derive a Total Distance: $TPM1_t = PM1_t + PM2_t$. Once this is derived for each trade in the window, the largest TPM is determined across n number of trades which can then be used to derive the value of the PM ratio for the window:

$$PM = \frac{TPM_{max}}{PriceSpan} \tag{5}$$

Once a PM ratio value is estimated for each window in the sample, the array of PM ratio values is ordered and a cutoff point is determined. This is then used to filter out the top values encountered in the sample. The focus on price data only means that this approach will select the windows with the biggest price moves which have subsequently reversed back to their starting point. Once these are determined, one can use the built in functionality of the Trident Tool to look for the characteristic pattern in volume, finally yielding a confirmed finding.

3. Data and descriptive statistics

We use data from the Thomson Reuters Tick History (TRTH) supplied by the Securities Industry Centre of Asia-Pacific (SIRCA hereafter). TRTH provides millisecond-time stamped tick data, sourced from the Reuters Integrated Data Network (IDN) which obtains feeds directly from the exchanges. We select a diverse but limited variety of assets that appear to be favorite to algorithmic traders: Apple stock (ticker APPL) traded on US National Market System (NMS) markets, the US Oil ETF (ticker USO) traded on NYMEX, the Bund futures contracts maturing in September 2015 traded on Eurex. Apple is the most actively traded stock in the world with an average daily volume of over 63 million shares in the last 50 days [1]. This implies that Apple is likely to attract high levels of activity from a large amount of diverse market participants, including the HFTs. The Bund futures contract is extremely popular with traditional proprietary trading firms and market makers, and it is considered one of the most accurate indicators of the prevailing interest rates in the Eurozone. Finally, the use of US Oil ETF is particularly relevant, as it is reported as one of the top holdings of major HFT firms such as the Virtu Financial [2].

For this study, we use the level 1 quote and trade data for each asset. The level 1 data displays top-of-book data that includes the best bid and the best ask, i.e. highest bid and lowest ask, with corresponding quantity across multiple market participants or market centers. The Level 1 quote data for Apple stock is supplemented with the National Best Bid and Offer (NBBO) that provides the best quotes consolidated across all the National Market System (NMS) markets [3]. The Level 1 quote and trade data for USO are supplemented by the Chicago Mercantile Exchange (CME) and as reported on the electronic GLOBEX market

[1] It is known that during the events of the 2010 Flash Crash, Apple stock was briefly driven up in value to as high as $ 100 000 within a few instantaneous trades by malfunctioning algorithms, while the majority of the other assets were collapsing [12]

[2] http://www.bloomberg.com/news/articles/2015-02-19/berkshire-hathaway-exotic-etfs-among-flash-boy-holdings.

[3] Thirteen market centers submit quotations to the NMS for US stocks including BATS, BATS Y, CBOE, Chicago Stock Exchange, EDGA, EDGX, NASDAQ, NASDAQ OMX BX, NASDAQ OMX PSX, National Stock Exchange, NYSE, NYSE AMEX, and NYSE Arca.

[4]. The level 1 quote data and trade for the Bund futures are supplemented by Eurex exchange.

The sample period selected for Apple is the week spanning from 26-30 January 2015, around the earning report release. The sample period for USO is 13-14 of July 2015, days of significant volatility in the Oil markets after the lifting of international sanctions on Iran. As for the Bund, the asset is heavily influenced by the monetary policy of the European Central Bank (ECB). Therefore, the week selected for this study spans from June 1st till June 5th, as this week has been marked by significantly high volatility in European Fixed Income markets referred to "'bloodbath"'. During this week, the monthly monetary policy decisions and press conference were hosted by the ECB on the 3rd of June. While news' events are periods of heightened volatility, these news only constitute a small fraction of all "news" in our sample in a given day[5]. Algorithmic traders react to a myriad of signals that in principle could move market prices in an extremely high-frequency data, i.e. millisecond data. For instance, quote updates, trades and order submissions is another way to anticipate price movements in the short run. Examining data on non-news days of our sample and/or during periods of relatively lower intraday volatility (lower trading activity) is another way to anticipate price movements in the millisecond environment.

Table 1 reports sample statistics for the three assets. In total, our sample contains 2.63 million of trades with 6.94 million of Level 1 quote updates. On average, Bund futures are traded with 43 297 contracts per day, each contract has a notion value of 100 000; while USO-ETF are traded with 26 276 contracts daily. For Apple stock, on average, 1.23 million shares are submitted daily at the Level 1 of the market, resulting in 473 546 daily trades on average.

We compute market performance metrics such as the bid-ask spreads, total

[4]The CME data does not include floor trades or negotiated block trades.

[5][7] show that algorithmic traders quickly place market orders in the subsequent short period of time, i.e. ten seconds of the macro news release. Further, [13] show that there is a little change in the behavior of algorithmic traders by examining volatile and less volatile days.

Table 1: Descriptive statistics.

	Bund	Apple	USO-ETF
Number of trades	216,487	2,367,728	52,552 [t]
Level 1 quote updates	238,921	6,117,449	584,807
Daily average number of trades	43,297	473,546	26,276
Daily average of Level 1 quote updates	47,784	1,223,490	292,404
Average quoted spread	0.0119	0.0109	0.0105
Average trade size	10.58	197.08	14.62

This table reports descriptive statistics for the three assets used in the sample: Bund Futures, Apple stock and USO-ETF Oil. For each asset, we report the total number of trades and the number of quote updates at the level 1; We report also the daily averages of number of trades, Level 1 quote updates, Level 1 quoted spread, and trade size.

market depth at the best ask and bid quotes, trading volume and implementation shortfall (IS hereafter). The first two measures are mostly used in the market microstructure literature to evaluate market liquidity at any point of time. We use these two metrics as indicators of the level of market liquidity during quote volatility episodes. The trading volume is crucial for the correct identification of price momentum practices. We also compute the IS as it is widely used by practitioners. IS measures the execution performance of traders by benchmarking it against a hypothetical paper portfolio executed at the midpoint (the average of ask and bid quote prices) once the order is received. The result is a variable following the price movements during the period, but it is adjusted for the initial midpoint. It is calculated assuming a buyer point of view, therefore a positive value indicates that a buyer would have been better off executing immediately at the midpoint at the start of the time period examined (the window), rather than delay execution partially or fully. Similarly, negative values indicate that the price moves lower through the window so from a seller point of view, it is ideal to execute immediately.

To begin processing the data, we shed light on several important properties of algorithmic trading. For instance, these occur over a specific time interval marked by a starting point, a time span, and an ending point. As these are driven by algorithms sensitive to market conditions, the period immediately preceding an outburst of algorithmic trading activity might be of particular interest to the analysis. The time spans over which events last can be quite variable, and might follow a fat tailed distribution as discussed by [9]. This means that a one-size fits all approach could be wrong, and a certain level of flexibility is needed. Further, different types of events might occur over drastically different time horizons. While quote volatility episode may only last for several seconds in most cases, price momentum episode typically spans over several minutes. Therefore, a robust strategy for algorithmic trading patterns' detection would necessitate a sufficient built in scalability to cope with this without any fundamental alteration. It is also important to note the institutional features, such as the difference between pre-market, regular trading hours, and after hours trading, which will have a profound impact on the level of activity during times of the day [6]. All these considerations need to be built into the analytical strategy to ensure that it is appropriate for the current analysis.

4. Patterns of algorithmic trading

We identify 372 episodes of quote volatility and 112 episodes of price momentum. Some of the patterns observed seem to closely match patterns identified in the literature. This may indicate that the detection techniques utilized are appropriate. We first present results on the quote volatility ratio followed by the results of the price momentum ratio.

4.1. Quote volatility

The majority of events occur within the bigger time scale examined of 10 seconds. A breakdown within the group of quote volatility events shows that the distribution by specifications, between in-ask, in-bid and combined are similar in terms of occurrence. For instance, the trade price tends to move in the direction of the algorithmic activity, i.e. increases when rapid quote updates occur at the ask side or declines if rapid quote update occurs at the bid side. These effects should be observable in the data, and are therefore tested for. Further results

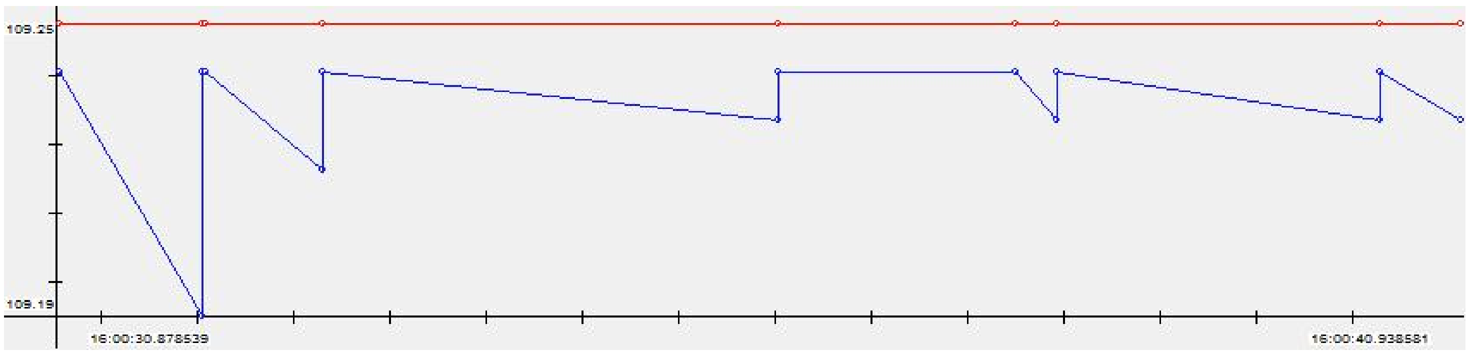
[6] Regular trading hours in local exchange time for Apple are between 9:30 and 4 pm, for USO between 10:00 and 2:30 pm and for Bund between 7:30 am and 5:30 pm.

indicate that a majority of in-ask and in-bid events are not accompanied by trading activity. The characteristic pattern is confirmed by IS results: when IS drifts lower to negative values during the time window examined, this is an indication of declining prices. Similarly, as it increases and remains positive, this is an indication of prices rising. It seems that the majority of quote volatility events exhibits the characteristic pattern. These results are especially important as they shed light on the impact of algorithmic presence on prices.

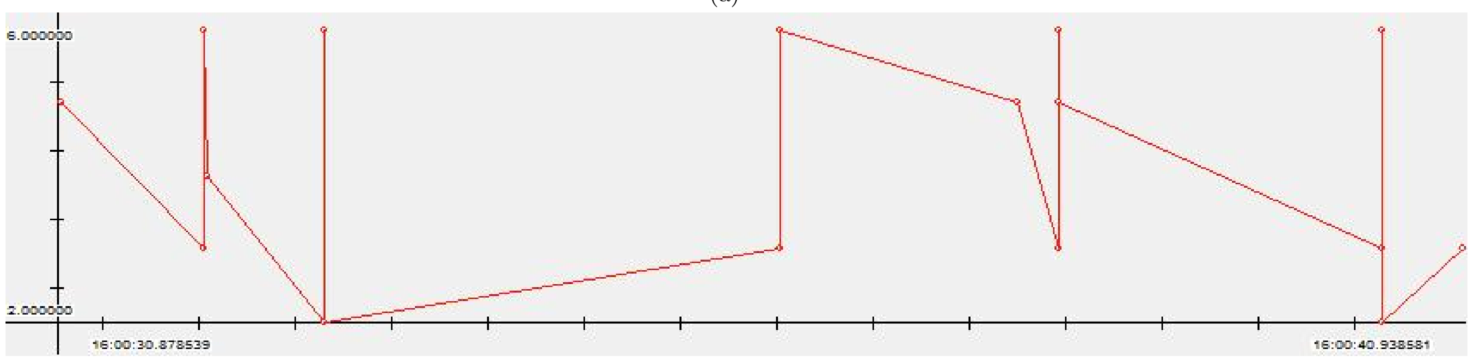
Another characteristic pattern which links quote volatility with the level 1 quoted depth is observed, as shown in Figures 1 and 2. A frequent observation during quote volatility episodes is that quote updates which narrow the quoted spread, appear to be associated with a significant decrease in level 1 quoted depth. This pattern is very pronounced and may have important implications for correctly interpreting the impact of algorithmic activity on the market.

While these results could potentially be caused by trading activity depleting quoted depth in the order book, the characteristic pattern is also similarly observed during episodes which involve no trades at all. This suggests that the change in depth levels could be due to new quotes being posted rather than old ones being depleted. Additionally, posted orders are characterized by very low quantities offered, which is another evidence of algorithmic activity. One of the most significant impact of the increasing presence of algorithmic trading in financial markets is a steady decline in the average trade size. The observation of small orders being posted and disappeared rapidly over a very short period of time fits the expected patterns. Moreover, this also lends support to the argument that liquidity provision by algorithmic activity may be transient in nature. Finally, this pattern could also be consistent with the technique of pinging, since the orders posted narrow the spread and may be intended to entice institutional traders into trading.

A final pattern is observed at the event level which confirms the intuition that the quoted spread also experiences volatility particularly during one-sided (in-ask or in-bid only) quote volatility episodes, as shown in Figure 3. This finding seems to suggest that while algorithmic traders seem to provide liq-

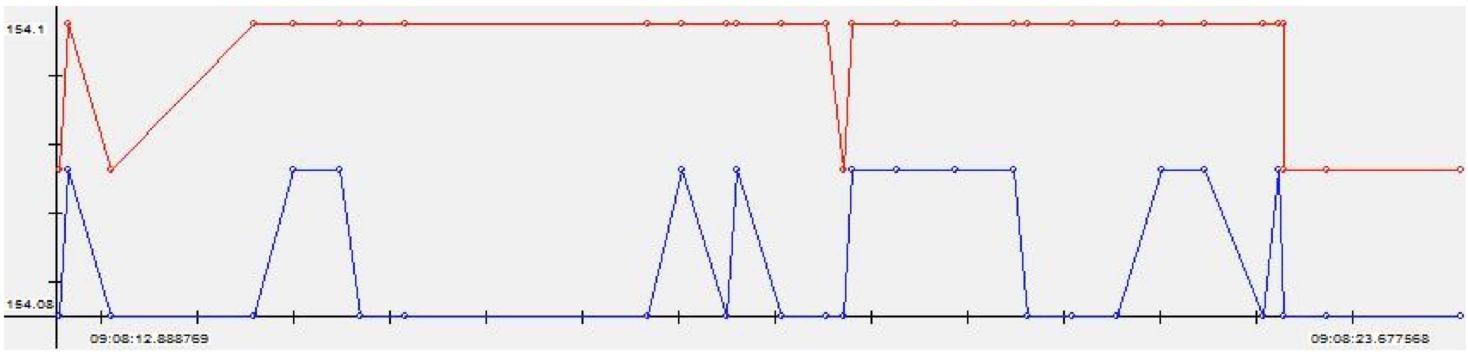


(a)

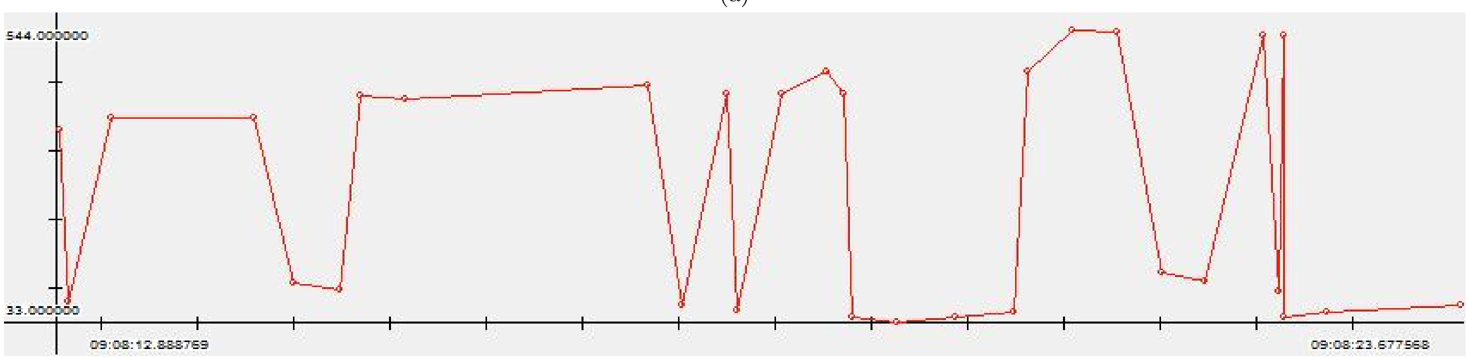


(b)

Figure 1: [a] Apple in-bid quote volatility and [b] The level 1 quoted depth during the same interval. These Figures depict [a] an episode of in-bid quote volatility and [b] the corresponding variation of the Level 1 quoted depth within the same interval for Apple stock.



(a)



(b)

Figure 2: [a] Bund combined quote volatility and [b] The level 1 quoted depth during the same interval. These Figures depict [a] an episode of combined quote volatility and [b] the corresponding variation of the Level 1 quoted depth within the same interval for Bund futures.

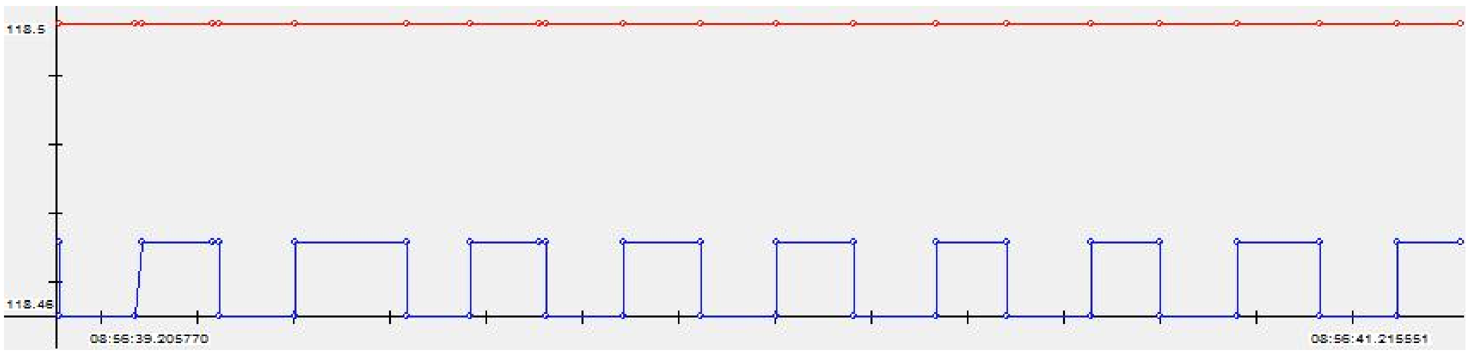
uidity through posting small but competitive orders which initially narrow the bid-ask spread, the rapid disappearance of these orders increases quote volatility, and may actually increase trading costs over the long run or even introduce an additional risk-premium for traders. This could potentially offset some or all of the benefits of added liquidity by algorithmic traders. A thorough investigation of this hypothesis is beyond the scope of this paper, and may serve as a suggestion for future research in the area of asset pricing in the spirit of [14].

An examination of the intraday patterns of the QV values reveals characteristic peaks around the beginning and the end of the regular trading hours from 9:00 a.m. till 16:00. This was observable across all the three assets as shown in the Figure 4. Further, an examination of the distribution of QV ratio values seems to strongly suggest that these follow a Chi-squared distribution, characterized by a fat tail, see Figure 5. This is a finding which warrants further investigation and could potentially lead to a more formalized quantitative method of detecting algorithmic activity.

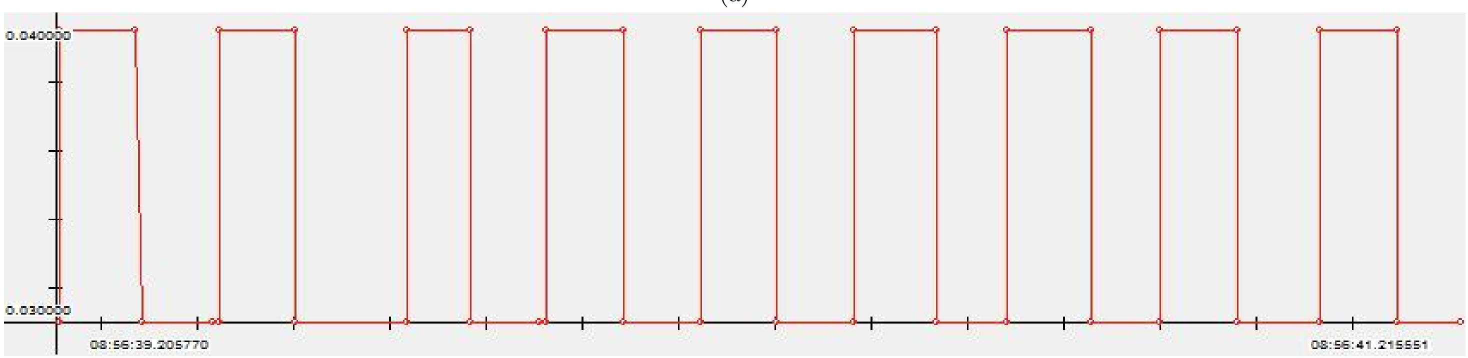
4.2. Price Momentum

The majority of events are observed on time frames of 30, 60 and 90s. The total size of price changes for each event is recorded in basis points, as well as the potential volume traded by algorithmic traders. The method used for this is an approximation based on the volume observed during the initial volume peak and the second volume peak, as shown in Figure 6. The aim of the analysis is to provide an estimate of the direct economic result derived by algorithmic traders.

As shown in Figure 7, an examination of the intraday PM ratio values chart reveals a similar pattern to the one observed for quote volatility ratio. The pattern observed for Bund contracts is different than the one for the exchange traded assets in the US. This may be due to the longer regular hours trading session on EUREX, as opposed to the market hours observed on US exchanges, where both Apple and USO are listed. A start of trading, end of trading and mid-day peaks are observable for the three asset classes.

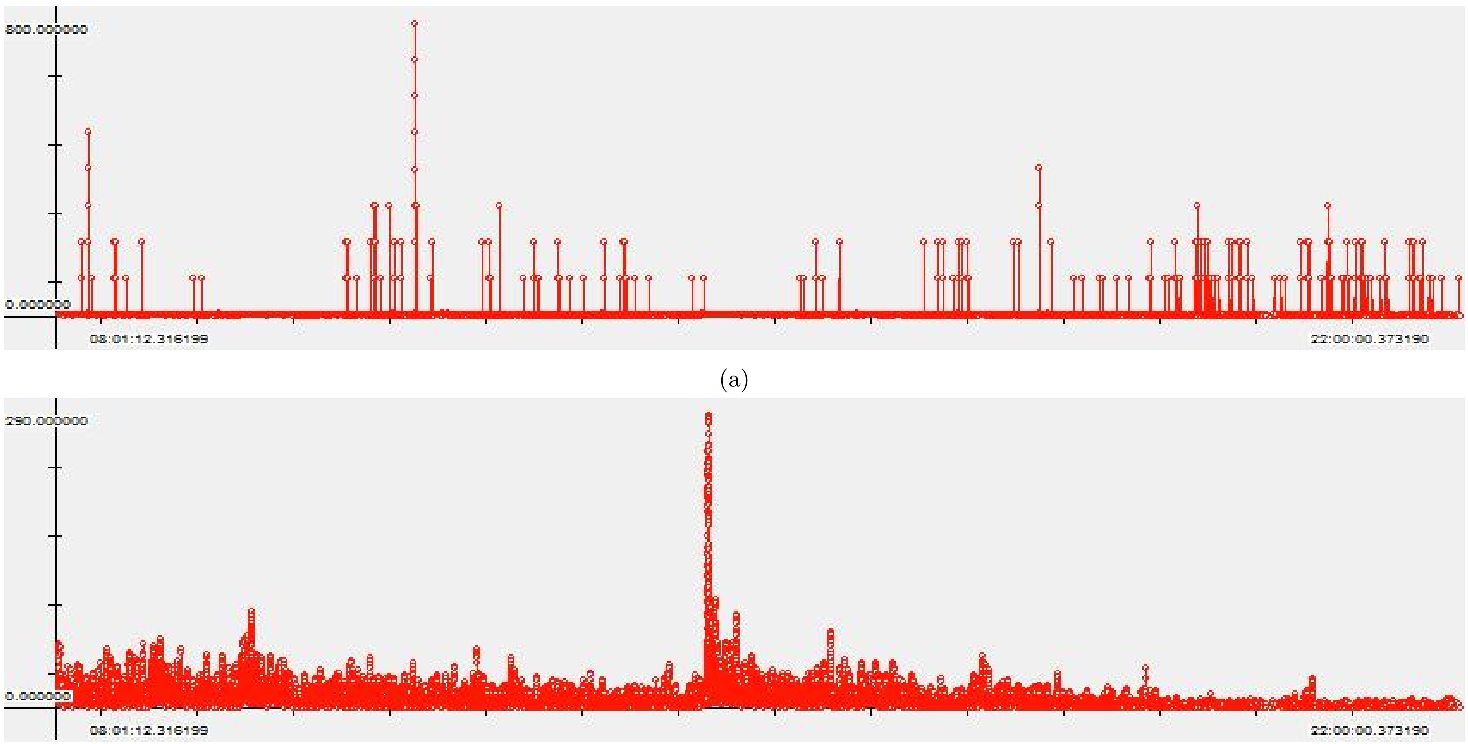


(a)



(b)

Figure 3: [a] Apple In-bid quote volatility and [b] quoted spread. These Figures depict [a] an episode of in-bid quote volatility and [b] the corresponding variation of the Level 1 quoted spread within the same interval for Apple stock.



(a)

(b)

Figure 4: Intraday patterns of QV for [a] Bund futures and for [b] Apple stock. This Figure depicts the intraday patterns of quote volatility ratio for Apple stock.

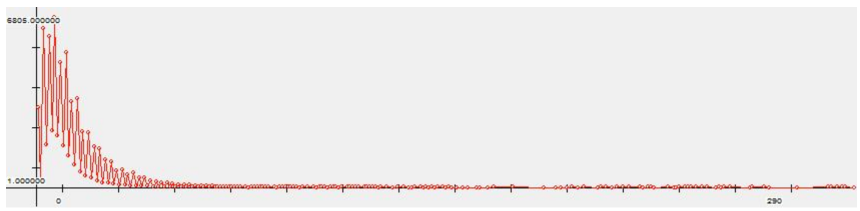


Figure 5: QV histogram. This Figure depicts the distribution of quote volatility ratio for the Bund futures.

While the average return per event observed in the sample is 23.09 basis points, this number is significant when considered within the context of the extremely short time frames of its occurrence between 0.5 and 1.5 minutes. The largest observed relative return in the sample is 106.7 basis points, during an event on the Bund futures market. Using Equation 6, the potential gross profit generated during this is EUR 82,692.5.

$$GrossProfit = Quantitytraded * IndicativePrice * \frac{Returninbp}{10000} * TickSize \quad (6)$$

Indicative price is a rough indication of the relevant assets price. For Bund futures, this is assumed at a constant EUR 155 for Apple shares at \$113 and for the USO-ETF is at \$17.36. The tick size of the Bund Futures contract is for a nominal value of EUR 10 per 0.01 change in price. The total profits generated over the sample studied amounts to almost \$ 5.25 million. The breakdown of event count by day of the week reveals no particular pattern, although it seems to suggest that midweek days may tend to contain higher algorithmic activity.

The distribution of the PM values appears to follow the Chi-square distribution pattern observed for quote volatility, but with even longer fat tails and greater skewness, as shown in Figure 8. This suggests granularity in the data, and the presence of extreme outliers during the episodes of algorithmic activity. This also provides additional basis for a further future quantitative research on this distribution.

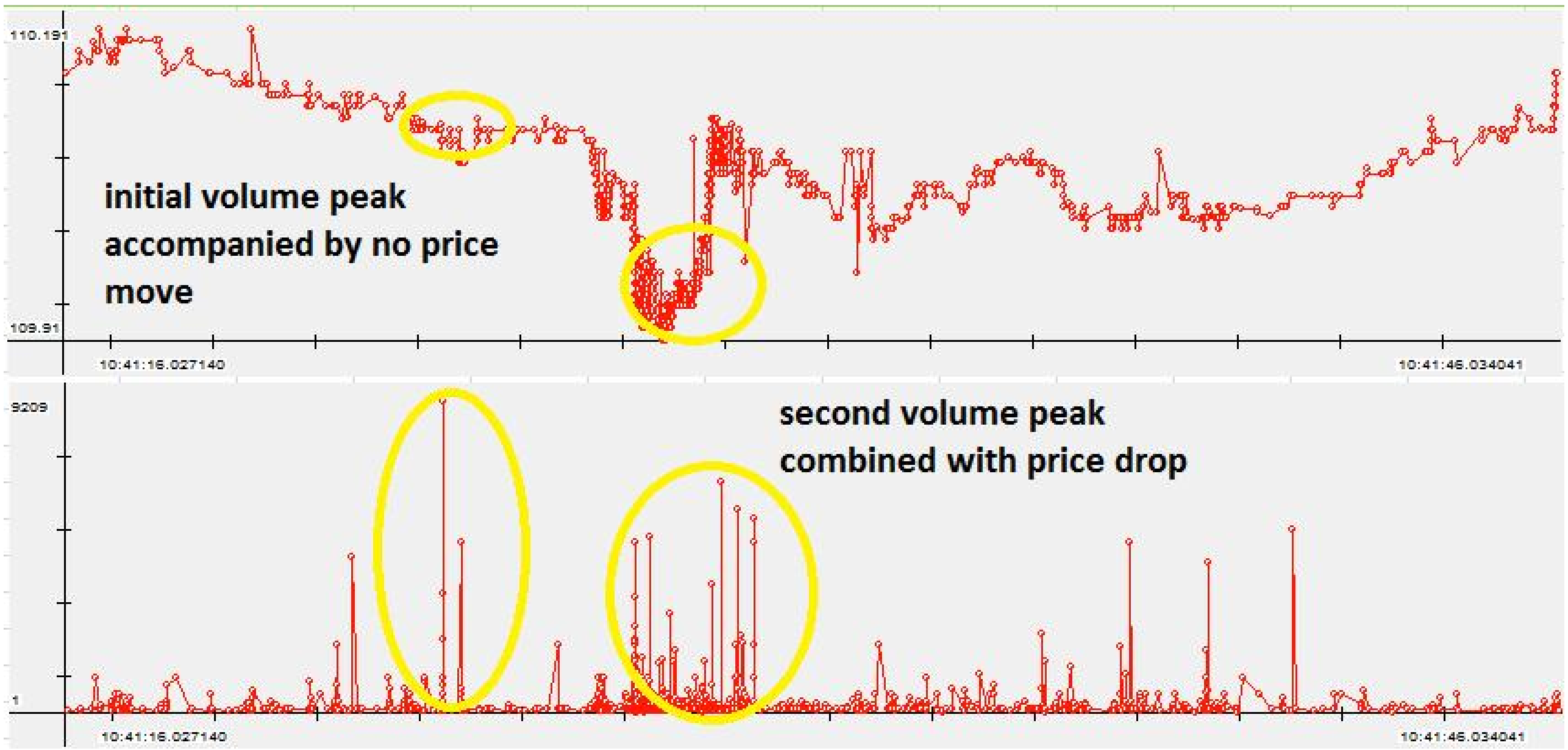
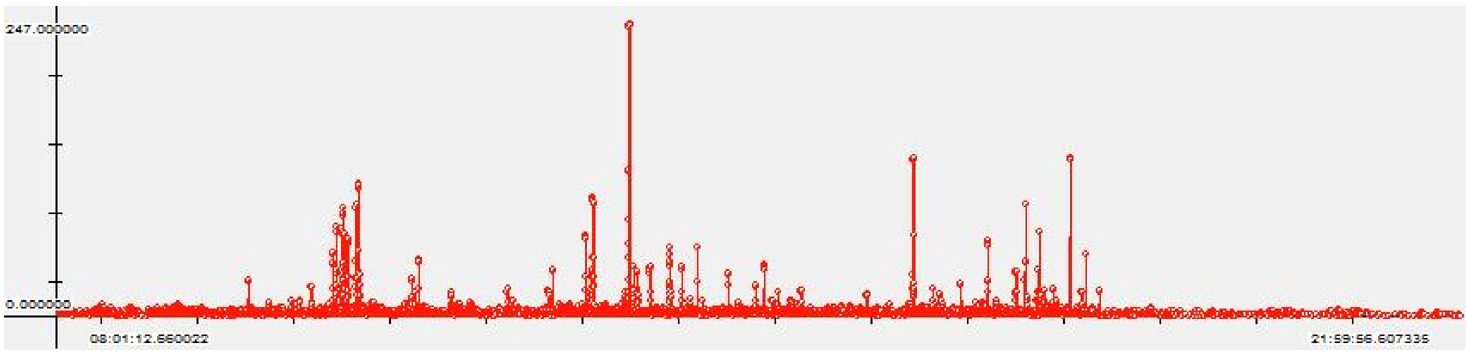
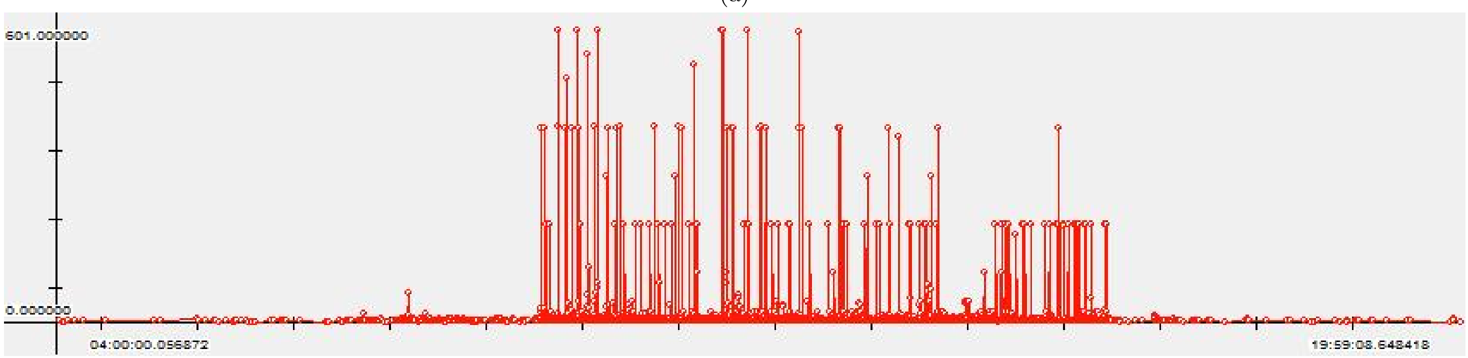


Figure 6: Apple price momentum patterns. This Figure depicts an episode of price momentum for Apple stock.

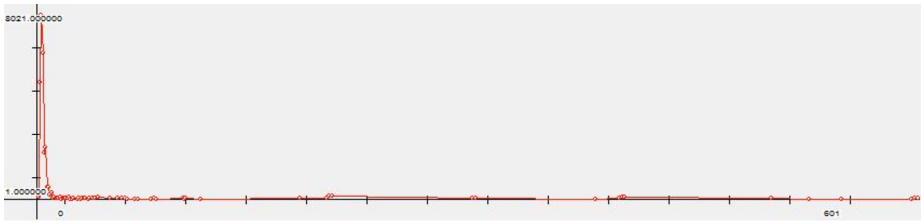


(a)

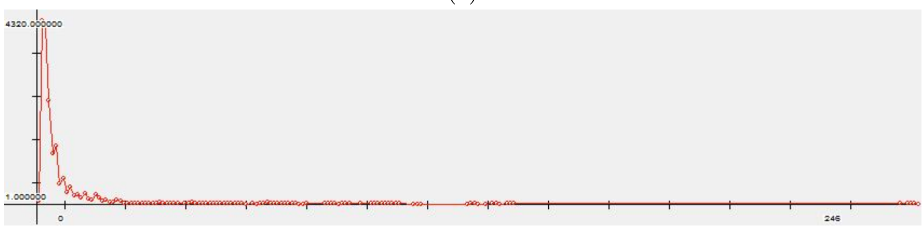


(b)

Figure 7: Intraday patterns of PM ratio for [a] Bund futures and [b] for USO ETF. These Figures depict the intraday patterns of price momentum ratio for [a]Bund futures and [b] USO ETF.



(a)



(b)

Figure 8: *PM* histogram. These Figures depict the distribution of *PM* values for [a] USO-ETF and [b] Bund futures.

5. ANN experiments

A two-stage ANN experiment is carried out on the quote volatility ratio. The first stage is used to validate the efficiency of the proposed ratio, and the second stage is used to detect the quote volatility patterns consistent with the group of strategies as detailed in section 2.1.

The initial stage of the neural networks is set as follows: the data is scanned using the QV ratio. The cutoff threshold, defined in Equation 4, is used to obtain the positive sample denoted as +1. An equal sample size outside the threshold is selected at random, and denoted as the negative sample -1. These two samples are combined and shuffled at random. The data is converted into machine readable format. The granularity used is 10 units in width and 10 units in height, resulting in 100 identical rectangular zones on the chart for each window in the sample. This data is then processed 150 times as a training sample through the ANN algorithm. A final sample of 100 randomly scrambled observations is used to measure and verify the performance.

The second stage of the neural networks experiments is to detect potential commonalities which may signal that an episode of algorithmic activity may be ongoing or is imminent. There may be reasons to believe that at least part of such activity may be predictable to some extent. Fundamentally, algorithms are triggered by market conditions. If these conditions were known, it would be possible then to forecast when algorithmic activity is imminent. However, this information would constitute a very closely guarded company secret, and is almost certain to be protected as intellectual property. Therefore, an alternative method is to detect commonalities in market conditions immediately preceding an episode of algorithmic activity by running ANN experiment.

ANNs are types of statistical learning models which are designed in a way that mimics the logical structure of a biological brain. These models are particularly useful for pattern, speech and image recognitions, and have been applied as well for analyzing patterns of consumer behavior in financial markets. ANN models require at least two basic characteristics: (i) a topology and (ii) a transfer function [15]. ANNs are constructed out of nodes called neurons which act as simple I-O transformers. Data is fed into neurons as a signal input, and this is processed via a transfer function which generates an output signal. There are multiple transfer functions available, which have different characteristics and may be appropriate for analyzing specific problems. Some of the most widely used ones include logistic function, linear function, and a hyperbolic function, and a threshold function [7]. For the current study, we use the following hyperbolic function: $O = \tanh(I)$. The derivative of the hyperbolic function is approximated by: $1\text{-}I^2$. This ensures that outputs can take on values between 1 and -1, as shown in Figure 9. Additionally, a large central region of the function is characterized by a relatively constant slope, allowing for strong learning performance in a wider region of input values:

The neurons of an ANN are structured in functional groups called layers. Most topologies will consist of 3 layers, an input layer, a hidden layer, and

[7] For instance, a logistic transfer function implies that the value of the potential outputs may range between 0 and 1. The derivative of the transfer function has important implications for the performance of the module during learning on training data sets.

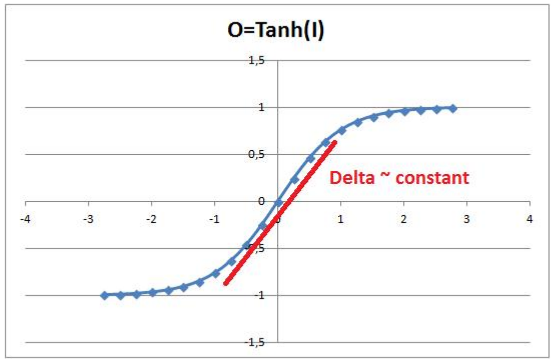


Figure 9: Plot of the hyperbolic transfer function

output layer. Each neuron in a layer is connected to all the neurons on the layer immediately preceding it, and to an additional bias neuron, which has a constant output. These connections are assigned a specific weight each, and the weighted sum of the signal coming from all connections forms the total input. The input layer neurons are used as input nodes, where raw data feeds into the network directly. This is then processed via the transfer function of the neurons, and fed via connections to the hidden layer, which then processes the signal and transmits it to the output layer. The output layer generates the final output of the network.

Training is a key stage of using ANN. Features or relationships which are influencing the data are inferred by ANN through a process of iterative learning. During the learning, ANN models process a data set designated for training and utilize an algorithm to adjust their connection weights so that their outputs converge closer to the desired values. While there are many strategies documented in the literature, the most popular algorithm is back-propagation, see [16]. Back-propagation is a strategy which adjusts network connection weights using the derivative of the transfer function. The information during learning flows in the opposite direction to the flow observed during processing. This begins at the output layer with a comparison between ANN current output and the target output known ex ante. This is used to calculate the deviation be-

Figure 10: Information flows within Artificial Neural Network

tween the two also known as error. The derivative of the transfer function is then used to make adjustments to connection weights further down the network, until all connections are updated. The new information learned is incorporated into the connection weights. The back-propagation algorithm is used for our experiment.

There are many reasons why ANN may be a suitable technique for carrying out the present experiment. The evaluation of market activity over a short period of time can be seen as a pattern recognition exercise. Further, the commonalities preceding an episode of algorithmic activity, if present, are not known exante. However, ANN does not require such information, as long as all the necessary data is fed into the model. Finally, the question of whether a certain window is immediately preceding an algorithmic episode can be restated as a Boolean problem, with 1 denoting a period preceding algorithmic trading, and -1 otherwise. The narrower focus of the present experiment is on quote volatility by looking specifically for a graphical pattern in quote updates immediately prior to the episode of algorithmic activity. One significant challenge when analyzing two-dimensional data points using ANN models is posed by what is known as the curse of dimensionality. This is a catch all phrase for many diverse issues arising from the problem of representing two dimensional features in a format suitable for ANN processing. There is a significant body of literature detailing alternative strategies for dealing with this set of issues. For the purposes of the present research, a simplistic approach is adopted, based on 2-D image processing strategies [17], as shown in Figure 11.

Each window of quote updates examined is seen as a two-dimensional area in time and price. This is further segmented into a number of sections of equal area. The exact granularity of the division along the X and Y axes is determined and can be set within the ANN suite of the Trident tool. A granularity of 4 in

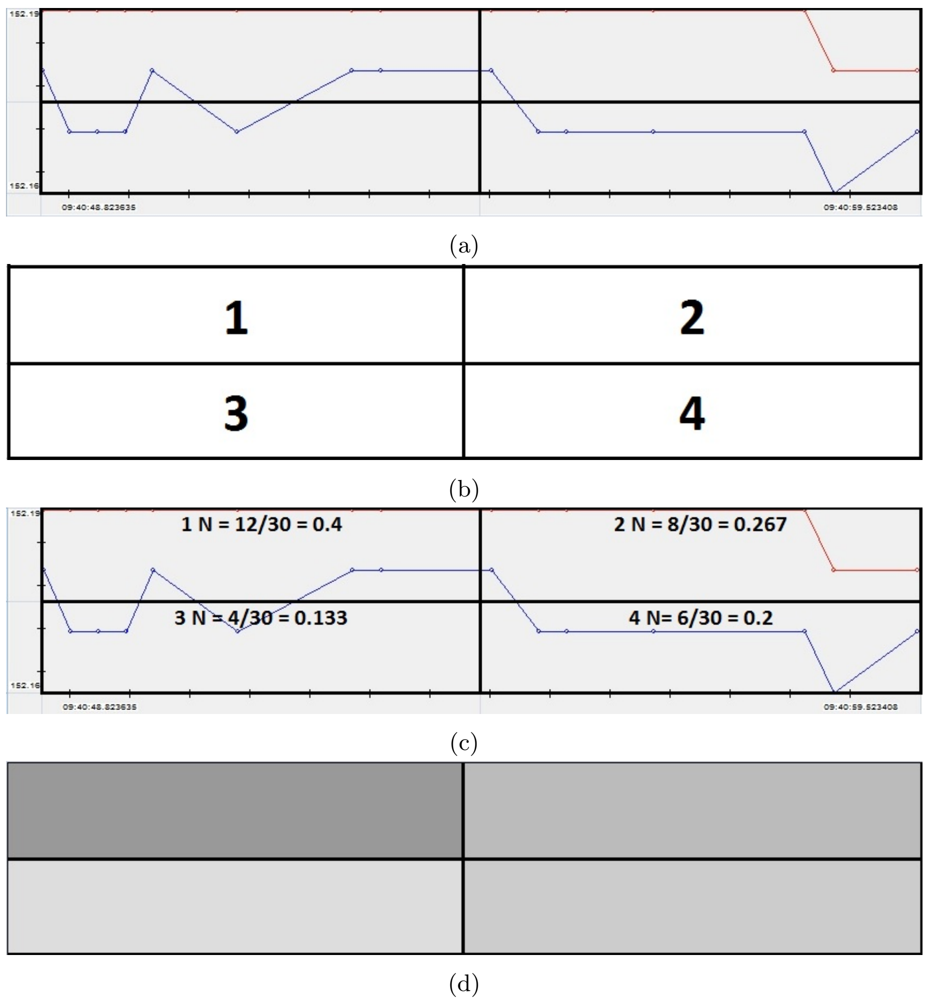


Figure 11: These Figures depict [a] the initial segmentation of a chart sample. For demonstration purposes granularity is set to 4 in both dimensions, [b] the representation of the 4 input nodes corresponding to the 4 chart segments; [c] the event Density within each region. There are a total of 30 events (quote updates) over the sample period, [d] and the representation of the result by filling each region with a % of black color in accordance with the event density calculated.

Price and 5 in Time is selected, yielding 20 segments of equal area. Once these regions are determined, the number of quote update events falling within each segment is estimated, and calculated as a fraction of the total number of quote updates in the time window examined. The end result is an array consisting of 20 fractions denoting relative event density, which sums up to 1. This approach is very similar to the one used in image processing, where images are segmented into areas and pixel counts are performed in each segment to transform the shape of the image into digital form.

The resulting set of inputs is readily processed by an ANN model. A training sample of 638 observations is used, with 319 windows immediately preceding a previously detected quote volatility episode, which are assigned a target value of 1, and 319 randomly selected alternative samples which are assigned a desired output value of -1. The ANN models are used to process 600 iterations of the training dataset, and once this is accomplished, a final holdout sample consisting of 50 periods with a target value of +1 and 50 periods with a target value of -1, is used for evaluation purposes.

In table 2, we present the first stage results of the ANN experiment for each asset within each quote volatility specification, i.e. in-ask, in-bid and combined. A success rate greater than 50% indicates detection of signals. Results suggest detection rates ranging between 50% and 60% for in-bid and in-ask quote volatility specifications. Further results suggest that increasing the QV ratio threshold (the third column) improves the detection in ANN at some levels. ANN does not seem to detect the signal for events with a very high data points, e.g. for Apple stock within the combined specification. A plausible explanation is that as the details in the data are too fine, the 10x10 resolution seems not to capture all the relevant features. Another explanation is related to the distribution of the QV values and the existence of outliers which might decrease modeling accuracy in ANN, as suggested by [18]. We investigate this further by running an additional experiment 110 times. We consider $QV\%$ of 10, and move lower in increments of 1% at a time, observing the corresponding changes in ANN accuracy. Realizing that there is uncertainty in the actual ANN experiment itself, we run it 10 times for each $QV\%$, and record averages and standard deviations of the results. Table A in the appendix shows the results for the Bund futures within the in-ask specification. Interesting pattern is seen - accuracy increases rapidly with reduction of the QV from 10 to about 7%. It then declines noticeably from 6% to about 2%, before increasing again. As we decrease the $QV\%$ initially, accuracy rises as we reduce the noise. However, when we reach the transition around 6% the sample begins to change as it contains a mix of heterogeneous data, therefore the ANN model struggles to detect it correctly. When QV is lowered further, the sample fully transitions to an homogenous state again, and the model is able to pick it up. Also, as we decrease $QV\%$ naturally, the number of relatively high $QV\%$ data points that may end up as part of the negative sample rises. It seems that ANN hardly distinguishes between a positive sample data point, and a just below threshold negative sample data point.

The second stage of ANN experiment is used as a proof of concept for potential forecasting techniques of our quote volatility ratio. It is set up and carried out as previously described. A simple rule of thumb is to check whether the forecasts add any incremental value to a naive forecast of 50%. Table 3 summarizes the results for several alternative specifications with the basic parameters and topology used, and seem to suggest that models examined here have forecasting power.

6. Conclusion

We propose two measures of algorithmic activity based on patterns of quote volatility and price momentum. We also run a two-stage ANN experiment using the quote volatility measure. Results documented here have several important implications such as the patterns of quoted spread and trading volume during quote volatility episodes, the economic performance of price momentum and the underlying distribution followed by the proposed measures. Further, we provide a novel ANN framework using the quote volatility measure. ANN results

Table 2: ANN experiment results

Observations	Asset	Strategy	QV ratio %	Success rate %
1	Bund	Ask	5	59
2	Bund	Ask	1	53
3	Bund	Ask	0.5	50
4	Bund	Bid	5	60
5	Bund	Bid	1	49
6	Bund	Bid	0.5	55
7	Bund	Combined	5	52
8	Bund	Combined	1	48
9	Bund	Combined	0.5	55
10	Apple	Ask	0.1	54
11	Apple	Ask	0.05	53
12	Apple	Ask	0.025	60
13	Apple	Bid	0.1	54
14	Apple	Bid	0.05	58
15	Apple	Bid	0.025	54
16	Apple	Combined	0.1	40
17	Apple	Combined	0.05	48
18	Apple	Combined	0.025	48
19	USO	Ask	0.12	55
20	USO	Ask	0.06	48
21	USO	Ask	0.03	51
22	USO	Bid	0.12	48
23	USO	Bid	0.06	56
24	USO	Bid	0.03	52
25	USO	Combined	0.12	56
26	USO	Combined	0.06	56
27	USO	Combined	0.03	49

This table reports the first stage results of ANN experiment for each asset within each specification. [1] Note: I=100 (Input layer); H-1=15, H-2=15, H-3=1 (Hidden layers); O=1 (Output layer); $\eta = 0.45$ (momentum coefficient) and $\alpha = 0.5$ (learning rate).

Table 3: ANN forecasting experiment results summary

	1	2	3	4	
Input layer I	20	20	20	20	neurons
Hidden layer H-1	3	4	4	3	neurons
Hidden layer H-2	3	4	4	3	neurons
Hidden layer H-3	1	0	0	1	neurons
Output layer O	1	1	1	1	neurons
Success rate %	53	50	50	51	
Momentum coefficient η	0.45	0.45	0.75	0.75	
Learning rate α	0.50	0.50	0.40	0.40	

This table reports the second stage results of ANN for several alternative specifications with the basic parameters and topology used.

suggest a detection rate that ranges from 50% to 60%, in particular during one-sided quote volatility episodes. By increasing the QV ratio threshold levels, we document an improvement in ANN detection at some levels.

Appendix

We run 110 times the ANN experiment within the in-ask specification for Bund futures.

Table A - Additional ANN experiment results for the Bund

Asset	QV ratio %	Mean Success rate %	Standard deviation
Bund	10	52.5	2.6
Bund	9	52.7	4.3
Bund	8	54.5	2.3
Bund	7	55.6	3.0
Bund	6	48.5	3.6
Bund	5	47.6	5.6
Bund	2	50.9	3.0

This table reports additional ANN experimental results by running the experiment 110 times within the in-ask specification for Bund futures. We consider QV% of 10, and move lower in increments of 1% at a time, observing the corresponding changes in ANN accuracy (success rate). We run 10 times for each QV%, and report the averages and standard deviations of the results.

Acknowledgment

We are grateful to the editor H. Eugene Stanley and the anonymous referees. We thank participants at Midwest Finance Association Meetings in Atlanta, and at the 6th International Conference of the Financial Engineering and Banking Society in Malaga for their suggestions.

References

[1] M. OHara, High frequency market microstructure, Journal of Financial Economics 116 (2) (2015) 257–270.

[2] A. J. Menkveld, The economics of high-frequency trading: Taking stock, Annual Review of Financial Economics 8 (2016) 1–24.

[3] T. Hendershott, C. M. Jones, A. J. Menkveld, Does algorithmic trading improve liquidity?, The Journal of Finance 66 (1) (2011) 1–33.

[4] J. Upson, R. A. Van Ness, Multiple markets, algorithmic trading, and market liquidity, Journal of Financial Markets 32 (2017) 49–68.

[5] J. Hasbrouck, G. Saar, Low-latency trading, Journal of Financial Markets 16 (4) (2013) 646–679.

[6] J. Hasbrouck, High frequency quoting: Short-term volatility in bids and offers, Available at SSRN 2237499.

[7] J. Brogaard, T. Hendershott, R. Riordan, High-frequency trading and price discovery, Review of Financial Studies 27 (8) (2014) 2267–2306.

[8] V. van Kervel, A. J. Menkveld, High-frequency trading around large institutional orders.

[9] J. Tse, X. Lin, D. Vincent, High frequency trading–measurement, detection and response, Credit Suisse, Zürich, Switzerland, Tech. Rep.

[10] M. K. Brunnermeier, L. H. Pedersen, Predatory trading, The Journal of Finance 60 (4) (2005) 1825–1863.

[11] R. Cont, Empirical properties of asset returns: stylized facts and statistical issues.

[12] S. Patterson, Dark pools: The rise of the machine traders and the rigging of the US stock market, Crown Business, 2013.

[13] B. Hagströmer, L. Norden, The diversity of high-frequency traders, Journal of Financial Markets 16 (4) (2013) 741–770.

[14] L. Pastor, R. F. Stambaugh, Liquidity risk and expected stock returns, Tech. rep., National Bureau of Economic Research (2001).

[15] C. M. Bishop, Neural networks for pattern recognition, Oxford university press, 1995.

[16] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, Tech. rep., DTIC Document (1985).

[17] M. Egmont-Petersen, D. de Ridder, H. Handels, Image processing with neural networksa review, Pattern recognition 35 (10) (2002) 2279–2301.

[18] A. Khamis, Z. Ismail, K. Haron, A. Tarmizi Mohammed, The effects of outliers data on neural network performance, Journal of Applied Sciences 5 (2005) 1394–1398.