DOI:
10.1145/3677052.3698643

Document Version
Peer reviewed version

Detecting Collective Liquidity Taking Distributions

Andrei-Bogdan Balcau
King's College London
London, United Kingdom
andrei-bogdan.1.balcau@kcl.ac.uk

Leandro Sánchez-Betancourt
University of Oxford
Oxford, United Kingdom

Stefan Sarkadi
King's College London
London, United Kingdom

Carmine Ventre
King's College London
London, United Kingdom

Abstract

Tools to identify and characterise the various types of agents in financial markets are essential for both regulators and practitioners. We introduce a methodology that combines agent-based modelling and machine learning to detect collective trading behaviour. Our detection method employs observable market variables to estimate the hidden composition of market participants. More precisely, we use the paths followed by the trend and the volatility of the midprice, and the traded volumes to infer the proportions in which different types of liquidity takers are active in the market (i.e., the market composition). We focus on a market with strategic continuous liquidity provision, populated by three common types of liquidity takers: informed traders, noise traders, and trend followers. We find that the paths of the trend and the volatility carry insufficient information about market composition when employed separately as estimators. However, when these two are non-linearly combined with the volume path, the detector performance increases substantially. Our study contributes to the financial behaviour recognition literature by offering insights into which market factors best describe the collective trading behaviour of liquidity takers.

CCS Concepts

• **Computing methodologies** → **Multi-agent systems**; **Agent / discrete models**; *Machine learning*.

Keywords

agent-based modelling; trading; behaviour detection;

ACM Reference Format:
Andrei-Bogdan Balcau, Leandro Sánchez-Betancourt, Stefan Sarkadi, and Carmine Ventre. 2024. Detecting Collective Liquidity Taking Distributions. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698643

1 Introduction

According to Akerlof and Shiller [1], economic systems are governed by the "animal spirits" that drive their evolution. Financial markets are considered to be extensions of these economic systems [14] in which traders adopt different strategies to make profit [10] and where, as part of these trading activities, "behavioural biases" manifest themselves [33]. To remain competitive, traders engage in an arms race of computational power and effective trading algorithms [21]. However, these algorithms often use models and metrics that are either not reflective of reality or out of sync with it [23]. Whilst collective economic behaviour can trigger crises [3], high-frequency algorithmic trading may exacerbate this risk by incentivising anticompetitive or undesirable practices that potentially jeopardise market stability, thus underscoring the need for rigorous empirical studies on trading interactions [18]. An example of such a study is presented by Byrd [8], who proposed a theory for automatic spoofing detection and a general framework for incorporating such detectors into reinforcement learning (RL). In the same vein, we develop an automatic detector to recognise the market composition of the algorithms guiding the liquidity-taking activity. The liquidity takers in our study belong to one of three categories: (i) *informed traders*, (ii) *noise traders*, and (iii) *trend followers*. These categories are representative of the behaviours found in financial markets. The detector's task is to determine the percentages of market activity attributable to each of these three types.

We employ machine learning (ML) to train a detection system on observable market data showing that behaviour recognition can be solved without the need of identifiable action sequences. More precisely, we study the extent to which volatility of the midprice, the trend of the midprice, and the volumes traded in the market can be used to determine the distribution of behaviour among liquidity takers. Our research question is the following:

(RQ) Do the paths followed by the trend of the midprice, the volatility of the midprice, and the traded volumes carry information about the proportion in which different types of liquidity takers are active in the market?

To address the research question, this paper makes two key contributions:
(1) We present a cost-effective computational method for learning the collective behaviour of active liquidity takers in financial markets. This method uses machine learning and agent-based abstractions of realistic, parameterised algorithmic trading activity. We employ a simple machine learning algorithm to ensure efficiency and to test the feasibility of our approach.
(2) We demonstrate how we employ our approach using observable time-series market data, exponentially weighted moving averages

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11
https://doi.org/10.1145/3677052.3698643

(EWMAs), non-linear feature transformations, and linear discriminant analysis (LDA).

Preliminary results demonstrate that LDA can be successfully applied to detect the proportions of known liquidity-taking types active in the market. This method achieves a Matthews correlation coefficient (MCC) of 0.708 (F1-score = 0.805), using a simple non-linear combination of the paths followed by the midprice trend and volatility, and the traded volumes; that is, the cross-product of 200-step paths followed by the trend, the volatility and the traded volumes. Our findings have practical applications for both regulators and practitioners, enabling them to discern market compositions from observable market data.

2 Background & Related Work

Understanding algorithmic trading behaviour is not a novel research strand. Recent work has employed both supervised and unsupervised ML techniques to categorise algorithmic trading patterns in major exchanges, such as Euronext Amsterdam [10]. For example, Cartea et al. [10] find that in their dataset, around one third of the algorithms with a liquidity provider dealing capacity behave like market makers. To get to this conclusion, the authors use identifiable transactions (transactions with trader-id) to build models that explain behaviour and then cluster the coefficients from their models. In contrast, we take a bottom-up approach when creating our detector, splitting market participants into liquidity providers and instances of typical liquidity takers.

Research in mainstream economics as early as Samuelson's "Foundations of Economics Analysis" [39] highlights economic individuals as profit maximisers. More recently, Lo's "Adaptive Market Hypothesis" [33] suggests seeing economic individuals as adaptive or maladaptive, their ultimate goal being survival and not always profit maximisation. Financial markets possess unique social structures, making it challenging to discern traders' ultimate objectives. In our approach, we define the goals of traders and their methods for achieving them, while leaving the proportion of each liquidity-taking type as an unknown variable to be determined.

Our model is used in conjunction with liquidity providing (market making). We envisage the situation where the liquidity providers learn about the liquidity-taking market composition and decide to adjust their policies accordingly. Therefore, we use sequence modelling on observable market data so our detector can be integrated with ease in learning agentic architectures as the one in [8].

Our approach resembles methods used in the computational opponent modelling literature. In the context of complex markets, most opponent modelling research has focused on agent negotiation strategies, on multi-agent reinforcement learning (MARL) approaches, or on a combination of both. With a few exceptions [35], most of these works have adopted a model-based approach [22], where agents are given a model of their opponent that they learn to optimise – see a detailed account in the survey from Nashed and Zilberstein [37]. Popular approaches include Bayesian optimisation [24], deep learning methods with MARL [41], fictitious play [40] and evolutionary transfer learning [25].

In our study, the liquidity provider uses an optimal strategy that balances profits and inventory control, and models the behaviour of liquidity takers (the 'opponents') who belong to one of three well-studied types in the financial markets literature: trend followers, noise traders, and informed traders.

3 Methodology

To answer our research question, we use a learning architecture combining agent-based models (ABMs) with ML. ABMs are used to generate collective behaviour abstractions. The ML detector uses those abstractions to establish relationships between observable market data and trading behaviour. We approach the problem of detecting collective trading behaviour from market data as a multi-class path classification task. We first create a number of labelled synthetic datasets containing information about the midprice and the volume traded via simulations of agentic markets. Then, we make use of the datasets obtained via simulations to solve our classification problem. We choose LDA as our classifier to analyse the linear separability in the feature space.

Our method analyses the behaviour of traders playing repeated games assuming the market is formed of liquidity providers and liquidity takers. For simplicity, we consider a market scenario with a single liquidity provider and multiple types of liquidity takers. The task is to build a detector that predicts the composition of liquidity takers in the market. This detector can be thought of as part of the liquidity provider's perception of the market to be used in adapting to forthcoming conditions. Alternatively, this can be thought of as a study instrument that any market participant or regulator might use. Our method does not need the member identification in any live market for recognising behaviour.

Calibration & Data Generation. To generate the synthetic datasets we use ABM simulations of trading populations operating in a limit-order book market. We calibrate the simulations to reflect stylised facts in the literature and label the generated data with tags specific to each population type.

Feature Preprocessing. Trend and volatility measurements are smoothed using the EWMA. Volume is computed using the total number of units traded per step. We choose to preserve the information about time. Therefore, all feature-path measurements are spread along the time dimension. Each sample has a size of 200×3 (200 time-steps and 3 feature measurements).

Classification. We first use LDA to find the most discriminant individual feature path for our classification task by considering trend, volatility, and volume. Then, we use sequential data transformations to test various combinations between our basic feature paths. These combinations enable us to highlight relationships in the market data that are useful for classifying sequential collective behaviour.

3.1 Agent-Based Model Overview

We develop our simulator from scratch using the Mesa Python library [27], opting for this approach over existing simulators (e.g., ABIDES [9]) to focus specifically on the interplay between liquidity provision, consumption, and simple limit-order book dynamics. This bespoke design allows us to maintain straightforward interactions whilst tailoring the simulation to our research needs. To generate sequential data in a similar format to [8], we use a trading horizon of 200 steps for our trading interactions. Our market

model uses a limit-order book to record all buy and sell orders, allowing traders to place either immediate-execution market orders or price-specific limit orders. The interactions between traders are computed via sequential model steps. Each model step includes an action from the liquidity provider followed by the actions of the liquidity takers. The liquidity provider lists her quotes using an order size that can fit the incoming demand. Liquidity takers can list market orders of a single unit at the bid or ask price quoted by the liquidity provider.

Our model can be configured to produce realistic trading interactions by changing the parameters of the price dynamics or the market behaviours. It is important to find a good level of abstraction when designing synthetic populations for the detector to fit a range of trading scenarios. Selecting very specific trading algorithms might lead to overfitting, while a broad trading strategy umbrella might be too general. In this study, we employ trading behaviours that are widely recognised as representative in the algorithmic trading literature. Our model permits adding prior information about more specific market cases through the configuration of the algorithmic trading parameters, population sizes or price dynamics.

Price Dynamics. We assume a financial market with a single traded asset. This makes it easier to model the fundamental price evolution without having to deal with cross-asset interactions. To employ the price dynamics in the environment, we use the assumption that financial markets are efficient as defined by the mainstream economics framework [16]. Here, the fundamental price of the traded asset is S_t and is given by $S_t = S_0 + \sigma W_t$ where $\sigma > 0$ is a volatility parameter and W_t is a standard Brownian motion. The liquidity provider quotes around the observed price process $\tilde{S}_t$ which follows $\tilde{S}_t = S_t + \int_0^t \alpha_s \, \mathrm{d}s$ where α_t is a zero mean-reverting (short-lived) signal that only the informed liquidity taker observes. More precisely, and similar to [11, 12, 15, 30] we take $\mathrm{d}\alpha_t = -\kappa \, \alpha_t \, \mathrm{d}t + \sigma^\alpha \, \mathrm{d}W_t^\alpha$ where $\kappa \geq 0$ is the mean-reversion speed, $\sigma^\alpha > 0$ is the volatility of the signal, and W^α is a Brownian motion independent of W.[1]

Market Behaviours. We employ a single market maker that acts as the liquidity provider for the market. Market makers aim to make profits on the spread between their bids and asks. In this study, we use the market making algorithm in [2] that skews quotes based on the inventory held by the market maker. That is, if we denote by Q_t the inventory of the market maker at time t, the bid and ask quotes are $\tilde{S}_t - f^b(t, Q_{t^-})$ and $\tilde{S}_t + f^a(t, Q_{t^-})$ respectively, for deterministic functions $f^{a,b}$ – see Chapter 11 in Guéant [19].[2] The notation Q_{t^-} refers to the inventory of the market maker just before time t (left limit). Given that the liquidity provider follows this strategy, we remark that the path of midprices depends on the activity of the liquidity takers. For liquidity taking, we consider informed traders, trend followers and noise traders. In short, the informed trader observes the alpha signal that influences the observed price updates and sends buy or sell orders based on the value of the signal. The trend follower constructs a measure of the midprice trend using a fast and a slow EWMA, and trades as a function of that measure. The noise trader sends random buy or sell orders. Table 1 summarises the trading algorithms. For completeness, we give the pseudocode of the trading algorithms in Appendix A.

[1] For alternative approaches to modelling informed trading see [4, 7].

[2] See [20, 28, 36] for alternative formulations of the market making problem.

Table 1: Trading algorithms

Trader type	Processes	References
Market maker (MM)	inventory & price	Avellaneda and Stoikov [2]
Informed trader (I)	alpha signal	Kyle [29]
Noise trader (N)	N/A	Kyle [29]
Trend follower (T)	midprice	Levine and Pedersen [31]

4 Experimental Setup

We study the case when there are more liquidity takers of one type than others. Here, we use the Avellaneda-Stoikov market maker [2] to engage in trading interactions of 200 time-steps with different liquidity taking populations. We construct three of those populations, each containing a single market maker; each population has four liquidity takers — the four liquidity takers are two traders of a given type and one for each of the other types. For example, the population 'TTNI' has two trend followers, one noise trader, and one informed trader. The other two populations are 'TNNI' and 'TNII'. We use 10,000 runs for each population type to account for complexity especially generated by the random behaviour of noise traders and fundamental price dynamics. Therefore, each dataset we analyse contains 30,000 labelled runs of 200 steps.

To determine if the historical paths of trend, volatility, and traded volumes provide insights into the composition of liquidity-taking market participants, we first extract these three features from our synthetic dataset. We then analyse whether these feature paths carry information about the market composition by creating different ML pipelines and evaluating them against well-known classification metrics. In this study, we use the accuracy, F1-score and the Matthews Correlation Coefficient (MCC). We are interested in the amount of information the trend, volatility and the traded volumes contain about the market composition. More precisely, we analyse whether these features can be used individually or in combination to find significant differences between the runs generated by the TNII, TNNI and TTNI populations. The training and prediction times of the ML classifiers are also evaluated to offer guarantees about the suitability of our approach within a high-frequency finance setting. Since we believe there is no benchmark for the collective behaviour recognition task in the literature, we use the default random guessing benchmark employed in multiclass classification problems. Here, we use three classes in a balanced fashion for our dataset. Therefore, our benchmark is 0.333 for accuracy (MCC = −0.0). We use LDA for classification, a computationally-efficient learning algorithm with a linear decision boundary.

4.1 Calibration

There are different population templates and parameters that can be selected for creating high-fidelity simulations of real financial markets. A summarised view of the parameters, their meaning, range and selected values after our calibration procedure can be seen in Table 2. The ABM needs to be reflective of the economic reality, but general enough to capture numerous trading interaction typologies. We use market making, trend following, informed trading and noise trading to highlight our view of a realistic algorithmic market representation. This is based on the following financial market literature: (i) Fama [17] separates the common stock price prediction activities

Table 2: Model parameters

Component	Parameter	Description	Range	Value
Price	Signal's volatility	The volatility of the alpha component in the fundamental price.	≥ 0	10
	Signal's mean-reversion rate	The mean-reversion rate of the alpha component in the fundamental price.	≥ 0	1
	Fundamental price initial value	The initial value of the fundamental price.	> 0	100
	Fundamental price volatility	The volatility of the fundamental price.	≥ 0	1
Market maker	Risk aversion (γ)	Risk aversion of the market maker.	(0,1]	0.01
	Generosity sensitivity	How the order flow intensity reacts to distance from observed price.	≥ 0	0.75
Trend follower	Fast decay (f)	The decay of the fast EWMA applied to the midprice.	(0, 1]	0.6
	Slow decay	The decay of the slow EWMA applied to the midprice.	(0, f)*	0.4
	Fast initial	The fast EMWA initial midprice observation.	> 0	100
	Slow initial	The slow EMWA initial midprice observation.	> 0	100
	Threshold	The minimum difference between the fast and slow EWMAs to trigger an order.	> 0	≈ 0.039
Noise trader	Activation probability	The activation rate of the noise trader.	[0, 1]	0.25
	Buy probability	If activated, the noise trader will either buy or sell based on this value.	[0, 1]	0.5
Informed trader	Buy threshold	If the fundamental price signal (alpha) is above the threshold, the trader will buy.	> 0	≈ 5.94
	Sell threshold	If the fundamental price signal (alpha) is below the threshold, the trader will sell.	< 0	≈ -5.94

*The slow EWMA needs to weigh recent midprice observations lower than the fast EWMA as discussed in [31]. Therefore, the slow decay is lower than the fast decay.

in technical and fundamental asset value analysis;[3] (ii) Kyle [29] highlights the liquidity taking activity as either informed of the fundamental asset value evolution or noisy trading;[4] (iii) Cartea et al. [10] characterise market-making behaviour among active algorithms in a real financial exchange. Here, we combine those trading strategies into one model. We continue with a discussion about the empirical calibration of the trading and price dynamics parameters.

Market Making & Price Dynamics. The risk aversion of the market maker is set to 0.01 as in [2]. As a robustness check, we study the sensitivity to different risk aversion levels for the overall performance of the detector in Section 5.1. The market making algorithm is used to quote around the price process $\tilde{S}_t$. This price process can be created in a data-driven way by extracting information from real markets. Here, we use 100 as the initial value of the fundamental price for each run as in [2].

Liquidity Taking. The calibration goal for the liquidity taking population is alignment with the stylised fact discussed in [12]; that is, liquidity providers trade at a loss with informed traders, and profit from noise traders.[5] On average, the profitability of informed traders is higher than that of trend followers, which in turn is higher than that of the noise traders; we show the profitability of the market maker against each of the three populations in Section 5. Liquidity takers are set to be adaptive and strategic, their behaviour being a function of the environment as discussed in [32]. Here we assume the total volume traded per trading horizon (trading intensity ratio) is roughly the same across different trading populations to avoid trivial classification outcomes. More precisely, we set some expected traded volume target and configure the trading algorithms' parameters to keep the value constant across populations and different price dynamics. We believe that a 25% trading intensity ratio is a reasonable assumption and therefore set the traded volume per liquidity taker in any run to be roughly around 50 units. The sensitivity to different trading intensities for the overall performance of the detector is evaluated in Section 5.1.

[3]Here, we use trend following as a representative type of predictive trading activity based on historic data (technical analysis).
[4]This concept is also used in [12] to model informed and uninformed traders.
[5]This is because informed traders are aware of the fundamental price evolution.

4.2 Feature Preprocessing

Our dataset, generated through calibrated simulations, contains aggregated information from multiple trading runs. It comprises 10,000 simulations, each consisting of 200 steps, for three distinct population types: TNII, TNNI, and TTNI. Extracting the trend, volatility and volume paths results in a dataset of size 30,000×200×3.

Trend. We use the EWMA-crossover signal to compute the trend path as in [31]. This method computes a fast EWMA of midprice observations which gives more importance to the most recent ones, and a slow EWMA which values the historic observations more. The signal is computed via the difference between the fast and slow EWMAs. For the fast and slow decays, we use the same values as our trend followers operating in the environment: 0.6 for the fast decay and 0.4 for the slow decay (cf. Table 2).

Volatility. We use EWMA to generate the volatility path using the midprice returns as suggested in the JP Morgan and Reuters RiskMetrics Technical Report [34], choosing a decay of 0.94 and assuming an initial volatility observation of 0.

Volume. We compute volume paths by considering the total volume traded per time-step for each run.

4.3 Classification

We analyse the collective behaviour classification usefulness for the trend, volatility and volume using the LDA classifier from scikit-learn [38]. We employ LDA specifically to evaluate which combinations of features provide better linear separation of the data. We initially generate a labelled dataset using the calibrated ABMs for representative algorithmic trading behaviour. We use a five-fold cross-validation procedure and measure the mean classification accuracy, F1-score and MCC to evaluate the LDA detector.

4.4 Trend, Volatility and Volume Compositions

We analyse different compositions of the preprocessed trend ($\hat{\alpha}$) and volatility ($\hat{\sigma}$) paths and the volume (v) path – our base feature paths – to empirically assess whether our LDA-based detector can classify collective behaviour.

LDA Soft Voting Ensemble. One way of combining the base feature paths is using an ensemble of three LDA classifiers. All sub-models use the LDA algorithm, but each sub-model is trained on a specific type of feature path: one on trend paths, one on volatility paths and one on volume paths. We employ a soft voting classifier that uses the average predicted probabilities of the sub-models to predict the final class labels. This is very similar to a feature bagging ensemble usually used in random forests [6], but without a random feature selection in the underlying models.

Flattening. Another way to combine the base feature paths is stacking them along the time dimension according to Figure 1. This

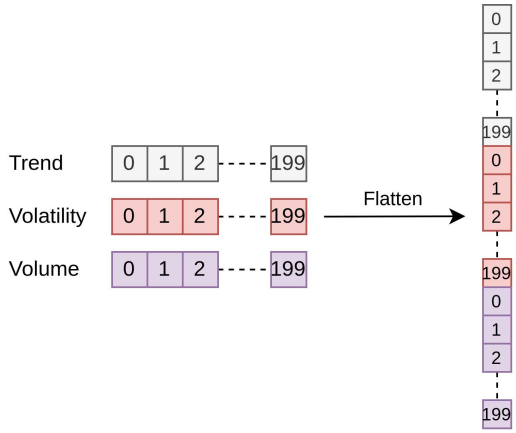

Figure 1: Flattening composition of base features

creates tensors, similarly to how images are flattened in deep learning image recognition tasks. The advantage of this technique is that the information about time is preserved. However, the feature space is large in size, each sample is now represented in 600 dimensions – the new dataset is of 30,000 × 600 points. This might be computationally-inefficient and force the detector into learning feature path relationships that are not useful.

PCA Path Reductions. To address the feature dimensionality issue of flattening, we use PCA on the time dimension for each feature path to reduce its length to a single component. The obtained principal components are stacked in a single vector to represent the sample. Figure 2 highlights the feature composition process via the PCA transformation.

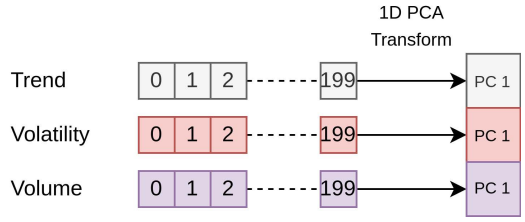

Figure 2: PCA composition of the base features

Covariational Flattening. Finally, we implement a feature transformation to analyse the temporal covariance among trend, volatility, and traded volumes. We introduce six more features obtained using a cross-product combination along the time dimension of the base features. We evaluate their linear separation capabilities individually and in a flattened method as with the base features. The flattened dataset is of size 30,000 × 1,800.

Table 3: LDA detector performance

Feature composition	Accuracy	F1-score	MCC
$\hat{\alpha}$	0.333	0.333	-0.0
$\hat{\sigma}$	0.338	0.338	0.008
v	0.560	0.551	0.342
$\text{VOTING}_{\text{LDA}}(\hat{\alpha}, \hat{\sigma}, v)$	0.493	0.487	0.241
$(\hat{\alpha}, \hat{\sigma}, v)$	0.653	0.645	0.481
$(\text{PCA}_1(\hat{\alpha}), \text{PCA}_1(\hat{\sigma}), \text{PCA}_1(v))$	0.349	0.340	0.024
$(\text{PCA}_5(\hat{\alpha}), \text{PCA}_5(\hat{\sigma}), \text{PCA}_5(v))$	0.410	0.391	0.119
$\hat{\alpha}^2$	0.344	0.344	0.017
$\hat{\sigma}^2$	0.339	0.339	0.009
v^2	0.520	0.518	0.281
$\hat{\alpha}\,\hat{\sigma}$	0.333	0.333	-0.001
$\hat{\alpha}\,v$	0.335	0.335	0.003
$\hat{\sigma}\,v$	0.514	0.506	0.273
$(\hat{\alpha}, \hat{\sigma}, v, \hat{\alpha}^2, \hat{\sigma}^2, v^2, \hat{\alpha}\hat{\sigma}, \hat{\alpha}v, \hat{\sigma}v)$	**0.805**	**0.805**	**0.708**

5 Results & Analysis

After the calibration process, each liquidity taker trades around 50 units per run independently of the population type (parameter values are selected, cf. Table 2). The noise and informed traders do not change their parameters across populations since their behaviour is independent of the midprice evolution. The trend follower uses a 0.03875 signal magnitude activation threshold for TNII and TNNI and 0.039 in TTNI; see Appendix A for how this threshold is used in the trend-following algorithm. This value must be increased when adding trend followers, as they are prone to herding behaviour.

The trend followers use a fast decay of 0.6 and a slow decay of 0.4 to compute their trading signal. We use the same values for computing our EWMA trend model. Figure 3 illustrates the cross-sectional distributions of trend, volatility, and traded volume for individual steps in the paths generated by the three population types comprising a market maker (MM) and four liquidity takers (TNII, TNNI, and TTNI). From this, we observe how rich the volume information is in differentiating between aggregated algorithmic trading behaviours. The population with two trend followers (TTNI) trades the trends created by the high risk inventory position of the market maker at the beginning of the trading horizon. Close to the trading horizon midpoint, it is the population with two informed traders that starts trading more. Simply adding the traded volume units in a sample does not help in classifying it. However, the volume path demonstrates discriminative power. The trend and volatility paths alone seem to offer insufficient information for the classification task.

In Table 3, we quantify the classification usefulness for our base feature paths and their combinations using LDA. Volume alone carries rich information about the distribution of liquidity taking strategies active in the market. LDA's performance using the paths followed by the trend or the volatility is equal to a random guessing system. While soft voting aggregation does not even surpass the volume base feature, flattening offers higher classification power. PCA is not very useful in reducing the time axis for each feature without losing information. Covariational flattening is the suitable

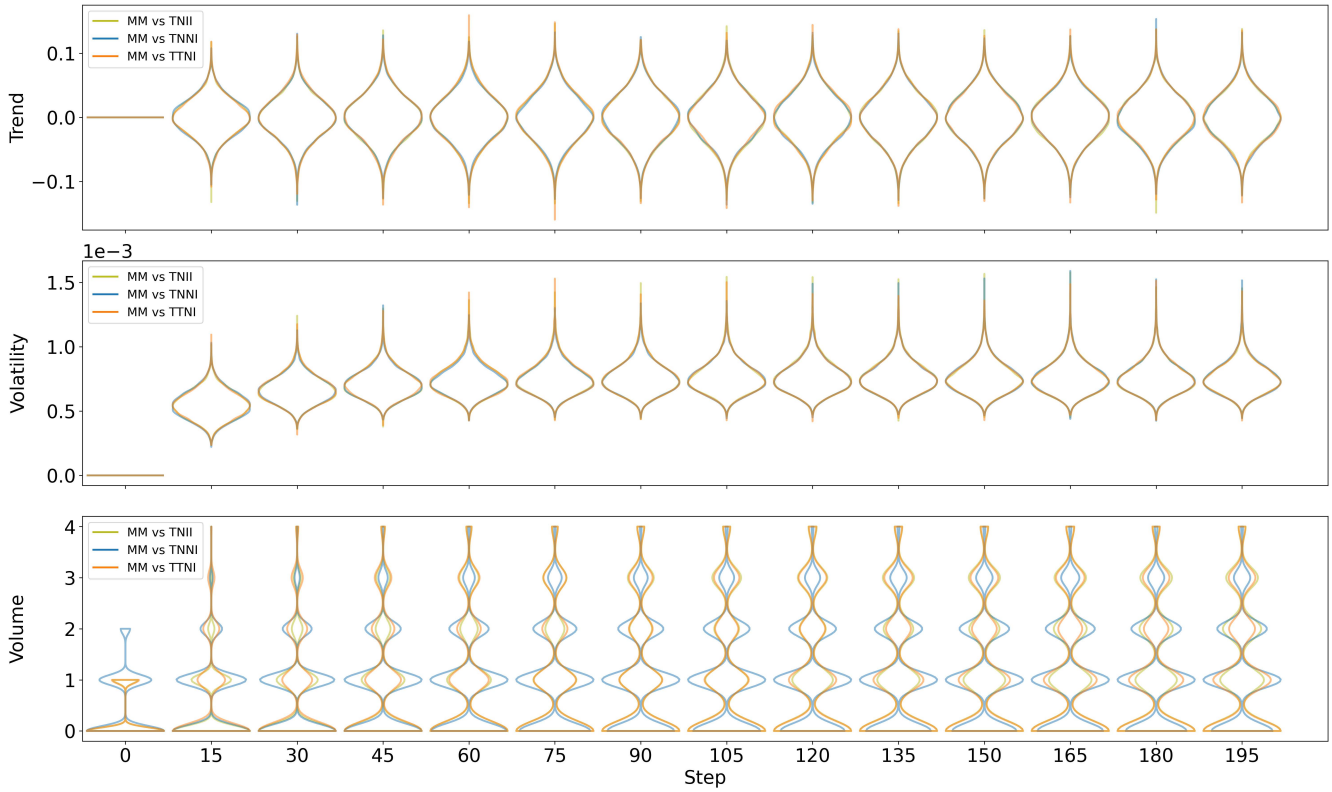

Figure 3: Cross-sectional distributions of the trend, volatility and traded volumes

Table 4: LDA and polynomial SVC comparison

Feature composition	Model	Accuracy	F1-score	MCC	Training time (s)	Prediction time (s)
$(\hat{\alpha}, \hat{\sigma}, v, \hat{\alpha}^2, \hat{\sigma}^2, v^2, \hat{\alpha}\hat{\sigma}, \hat{\alpha}v, \hat{\sigma}v)$	LDA	0.805	0.805	0.708	5.055	0.12
	PCA → LDA	0.749	0.747	0.625	8.22	0.23
$(\hat{\alpha}, \hat{\sigma}, v)$	PolySVC_2	0.842	0.841	0.764	569.25	16.69
	PCA → PolySVC_2	0.817	0.817	0.728	2076.34	7.96

Table 5: Market maker profit

Population	PnL (mean)	PnL (std)
TNII	-37.49	351.34
TNNI	64.63	237.83
TTNI	-10.05	339.32

transformation for the collective behaviour recognition task. This suggests the initial flattened dataset is not linearly separable and we require a higher-dimensional transformation instead.

Covariational flattening is similar to applying a polynomial kernel of degree two to the flattened base feature paths. We test this hypothesis by training a polynomial support vector machine classifier (SVC) of degree two on the flattened base feature paths and confirm the results, as shown in Table 4. We also add the training and prediction times for these algorithms since we envisage the behavioural profile detection system to be used in a high-frequency scenario.[6] The training and prediction times are measured once the base features are extracted and separated into the training and validation sets to account for the composition times. The training and prediction times of the SVC are much higher. Table 4 also highlights our attempt to reduce the feature space to the lowest dimension possible explaining 95% variance in the dataset using PCA. We obtain comparative results with using all the steps in the combined feature paths, but the training and prediction times are larger. We recommend using the covariational flattening transformation of the base feature paths LDA and without PCA if the memory used by the detector is not a concern.

The market maker is integral to our detection method. We use it to gather real-world samples that are later matched with learned trading behaviours. However, the effectiveness of our detection method may be compromised if the market maker sustains significant losses during the initial data collection phase. Table 5 shows the

[6]Results in Table 4 are obtained on a machine with a 13th Gen Intel® Core™ i7-13850HX, 28 cores CPU, and a 32GB memory.

profits of our market maker with the liquidity taking populations used to train the recogniser. As expected, the market maker benefits from a higher proportion of noise traders and loses profitability to the informed traders [13].

Bounding our method to a more specific market case requires calibrating the parameters of the ABM using data from the markets. For example, we can change the parameters of the trend followers to trade more than 50 units if we had information from the studied market. The same can be applied to the market maker or the fundamental price dynamics. Our modelling technique permits the injection of market priors in the population abstractions later used in training the detector.

5.1 Sensitivity Analysis

Next, we empirically evaluate the sensitivity of our trading behaviour detection method. Each experiment is computed independently of the other sensitivity hypotheses. For the trading intensity and risk aversion experiments, the ABM part of the detector needs to be re-calibrated to maintain stylised facts about the trading activity and profitability as in Section 4.1. We observe the following effects on the detector performance.

(1) Performance is improved if a lower value is selected for the decay in the volatility EWMA model. We believe this is because more weight is put on past EWMA predictions that have an influence on the current measurements. Hunter [26] suggests as well that a decay of 0.2 ± 0.1 usually offers better EWMA predictions, but this highly depends on the scale of the econometric data. Figure 4 shows the performance of the detector under different volatility decay values.

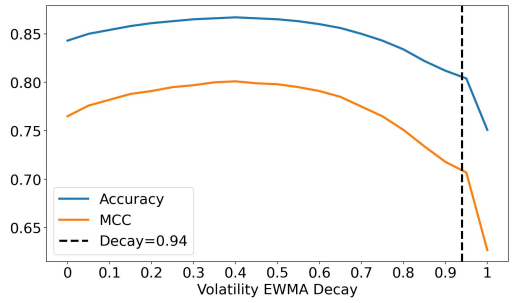

Figure 4: Covariational flattening LDA detector performance per volatility modelling decay

(2) Performance is not sensitive to changes in the fast and slow decays of the trend EWMA preprocessing.

(3) Increasing the trading intensity decreases the performance of the detector, but increases the market maker average profit. However, the standard deviation of the of the profit increases as well. We believe this is due to the activity of the noise traders. Table 6 compares the performance of the detector for different trading intensities after re-calibration.[7]

[7]A trading intensity ratio of 50% is achieved via re-calibration with a market maker generosity sensitivity of 0.875, a trend-following fast decay of 0.9, and a slow decay of 0.1. To increase the trading intensity ratio to 75%, we retain the trend-following parameters but adjust the market maker generosity to 1. The trend-following threshold is also updated for each population, but the values are omitted for brevity.

Table 6: Covariational flattening LDA detector performance per trading intensity ratio

	Detector performance		MM profit		
Ratio	Accuracy	MCC	TNII	TNNI	TTNI
25%	0.805	0.708	-37.49 (±351.34)	64.63 (±237.83)	-10.05 (±339.32)
50%	0.702	0.553	-23.61 (±546.81)	132.1 (±371.38)	13.45 (±544.55)
75%	0.721	0.582	86.59 (±659.69)	227 (±460.26)	121.53 (±654.41)

Table 7: Covariational flattening LDA detector performance per market maker risk aversion

	Detector performance		MM profit		
γ	Accuracy	MCC	TNII	TNNI	TTNI
0.01	0.805	0.708	-37.49 (±351.34)	64.63 (±237.83)	-10.05 (±339.32)
0.1	0.805	0.708	-3.47 (±34.49)	6.45 (±23.96)	-0.84 (±32.9)
1	0.805	0.708	-0.38 (±3.53)	0.62 (±2.45)	-0.12 (±3.35)

(4) Increasing market maker's risk aversion (γ) sequentially to 0.1 and 1 does not change the performance of the detector.[8] The market maker profit changes are nearly proportional to the change in risk aversion as described in Table 7.[9]

6 Conclusion

In this paper we introduced a novel hybrid method by combining agent-based modelling and machine learning to detect collective liquidity taking behaviour in financial markets. An important aspect of our work is the use of observable market data, in contrast to approaches that use identifiable sequences (e.g., [10]).

Our results show that the midprice trend and volatility in the market alone are not helpful in classifying trading behaviour. This is the case even when trend followers are present in the environment. However, we found that the traded volumes carry significant information about the distribution of different types of active liquidity takers in the market. We also showed how accuracy of predictions can be increased through non-linear combinations of the paths followed by the trend, volatility and traded volumes. To summarise, in this paper, we made the following contributions: (i) we described a cost-effective detector for behaviour profiles of existing algorithmic traders in financial markets; and (ii) we calibrated the model of such profiles by learning time-series observable market data.

Our method extends the research agenda of financial behaviour recognition using learned detectors [8]. We envisage an adaptive market maker architecture incorporating the collective behaviour recogniser and following a three-stage process. First, the market maker trains the recogniser. Second, it trades in the market for a predefined period to extract data about market composition. Finally, the market maker uses the trained recogniser to adapt to the discovered market composition. This is also a human-in-the-loop approach because it is the designer of the market-making algorithm that lets the market maker know what population abstractions to learn and what are the adaptation goals. Therefore, the approach is

[8]For $\gamma = 0.1$, we set the alpha signal volatility to 1 and the fundamental price volatility to 0.1 through re-calibration. When $\gamma = 1$, we adjust the alpha signal volatility to 0.1 and the fundamental price volatility to 0.01. The trend-following threshold is also updated for each population, but the values are omitted for brevity.

[9]This largely depends on how trading activity would change once the market maker updates γ. We leave these second-order studies for future research.

sound from a machine safety and trustworthiness perspective, but empirical guarantees need to be provided in future work.

Here, we focused on a market with a single market maker, although real markets might include multiple market makers. Our approach remains applicable in more complex environments. In a multi-market-maker setting, one can deploy our detector by creating the abstraction of how they interact. Incorporating market maker competition [5] is a topic for future research. Another interesting strand for future research is developing an efficient method to adapt our detector to changes in population size and dynamic shifts in activity proportions. Ideally, the detector should recognise the proportions of each liquidity-taking type regardless of the overall population size; for example, identifying the same ratios whether the population is TNII or TTNNIIII (double the traders but maintaining the same ratios). In real-world scenarios, these proportions could even shift during data collection (e.g., from TNII to TNNI within the 200-step sample). These challenges can be addressed through re-training, by providing the detector with more labelled examples of behaviour occurring in real markets. Re-training is computationally-expensive, but LDA is an efficient algorithm.

Acknowledgments

This work was supported by UK Research and Innovation [grant number EP/S023356/1], in the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence (www.safeandtrustedai.org), and was carried out partly while ABB was visiting the Oxford-Man Institute of Quantitative Finance, University of Oxford.

References

[1] George A Akerlof and Robert J Shiller. 2010. *Animal spirits: How human psychology drives the economy, and why it matters for global capitalism.* Princeton university press.
[2] Marco Avellaneda and Sasha Stoikov. 2008. High-frequency trading in a limit order book. *Quantitative Finance* 8, 3 (2008), 217–224.
[3] Jean-Philippe Bouchaud. 2013. Crises and collective socio-economic phenomena: simple models and challenges. *Journal of Statistical Physics* 151 (2013), 567–606.
[4] George Bouzianis, Lane P. Hughston, and Leandro Sánchez-Betancourt. 2024. Information-based Trading. *International Journal of Theoretical and Applied Finance* (2024), 2350030. https://doi.org/10.1142/S0219024923500309
[5] Robert Boyce, Martin Herdegen, and Leandro Sánchez-Betancourt. 2024. Market Making with Exogenous Competition. *Available at SSRN 4904510* (2024).
[6] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.
[7] Dorje C Brody, Mark HA Davis, Robyn L Friedman, and Lane P Hughston. 2009. Informed traders. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 465, 2104 (2009), 1103–1122.
[8] David Byrd. 2022. Learning not to spoof. In *Proceedings of the Third ACM International Conference on AI in Finance*. 139–147.
[9] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2019. Abides: Towards high-fidelity market simulation for ai research. *arXiv preprint arXiv:1904.12066* (2019).
[10] Álvaro Cartea, Samuel N Cohen, Rob Graumans, Saad Labyad, Leandro Sánchez-Betancourt, and Leon van Veldhuijzen. 2023. Statistical Predictions of Trading Strategies in Electronic Markets. *Available at SSRN 4442770* (2023).
[11] Álvaro Cartea and Sebastian Jaimungal. 2016. Incorporating order-flow into optimal execution. *Mathematics and Financial Economics* 10 (2016), 339–364.
[12] Álvaro Cartea and Leandro Sánchez-Betancourt. 2022. Brokers and informed traders: dealing with toxic flow and extracting trading signals. *Available at SSRN 4265814* (2022).
[13] Thomas E Copeland and Dan Galai. 1983. Information effects on the bid-ask spread. *the Journal of Finance* 38, 5 (1983), 1457–1469.
[14] Boudewijn De Bruin, Lisa Herzog, Martin O'Neill, and Joakim Sandberg. 2018. Philosophy of money and finance. (2018).
[15] Fayçal Drissi. 2022. Solvability of differential Riccati equations and applications to algorithmic trading with signals. *Applied Mathematical Finance* 29, 6 (2022), 457–493.
[16] Eugene F Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The journal of Finance* 25, 2 (1970), 383–417.
[17] Eugene F Fama. 1995. Random walks in stock market prices. *Financial analysts journal* 51, 1 (1995), 75–80.
[18] J Doyne Farmer and Spyros Skouras. 2013. An ecological perspective on the future of computer trading. *Quantitative Finance* 13, 3 (2013), 325–346.
[19] Olivier Guéant. 2016. *The Financial Mathematics of Market Liquidity: From optimal execution to market making*. Vol. 33. CRC Press.
[20] Fabien Guilbaud and Huyen Pham. 2013. Optimal high-frequency trading with limit and market orders. *Quantitative Finance* 13, 1 (2013), 79–94.
[21] Campbell R Harvey, Sandy Rattray, Andrew Sinclair, and Otto Van Hemert. 2017. Man vs. machine: Comparing discretionary and systematic hedge fund performance. *Duke I&E Research Paper* 2017-01 (2017).
[22] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. 2016. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*. PMLR, 1804–1813.
[23] Lisa Herzog. 2023. Are Financial Markets Epistemically Efficient? *The Philosophy of Money and Finance* (2023), 91.
[24] Koen Hindriks and Dmytro Tykhonov. 2008. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 326–333.
[25] Yaqing Hou, Yew-Soon Ong, Jing Tang, and Yifeng Zeng. 2019. Evolutionary multiagent transfer learning with model-based opponent behavior prediction. *IEEE transactions on systems, man, and cybernetics: systems* 51, 10 (2019), 5962–5976.
[26] J Stuart Hunter. 1986. The exponentially weighted moving average. *Journal of quality technology* 18, 4 (1986), 203–210.
[27] Jackie Kazil, David Masad, and Andrew Crooks. 2020. Utilizing Python for Agent-Based Modeling: The Mesa Framework. In *Social, Cultural, and Behavioral Modeling*, Robert Thomson, Halil Bisgin, Christopher Dancy, Ayaz Hyder, and Muhammad Hussain (Eds.). Springer International Publishing, Cham, 308–317.
[28] Christoph Kühn and Johannes Muhle-Karbe. 2015. Optimal liquidity provision. *Stochastic Processes and their Applications* 125, 7 (2015), 2493–2515.
[29] Albert S Kyle. 1985. Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society* (1985), 1315–1335.
[30] Charles-Albert Lehalle and Eyal Neuman. 2019. Incorporating signals into optimal trading. *Finance and Stochastics* 23 (2019), 275–311.
[31] Ari Levine and Lasse Heje Pedersen. 2016. Which trend is your friend? *Financial Analysts Journal* 72, 3 (2016), 51–66.
[32] Andrew Lo. 2017. *Adaptive markets: Financial evolution at the speed of thought*. Princeton University Press.
[33] Andrew W Lo. 2004. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management, Forthcoming* (2004).
[34] Jacques Longerstaey and Martin Spencer. 1996. *RiskMetrics Technical Document*. Technical Report. JPMorgan & Reuters, New York. Accessed on 3rd June 2024 at: https://www.msci.com/documents/10199/5915b101-4206-4ba0-aee2-3449d5c7e95a.
[35] Christopher Lu, Timon Willi, Christian A Schroeder De Witt, and Jakob Foerster. 2022. Model-free opponent shaping. In *International Conference on Machine Learning*. PMLR, 14398–14411.
[36] Xiaofei Lu and Frédéric Abergel. 2018. Order-book modeling and market making strategies. *Market Microstructure and Liquidity* 4, 01n02 (2018), 1950003.
[37] Samer Nashed and Shlomo Zilberstein. 2022. A survey of opponent modeling in adversarial domains. *Journal of Artificial Intelligence Research* 73 (2022), 277–327.
[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[39] Paul Anthony Samuelson. 1983. *Foundations of economic analysis*. Vol. 197. Harvard university press Cambridge, MA.
[40] Michalis Smyrnakis and David S Leslie. 2010. Dynamic opponent modelling in fictitious play. *Comput. J.* 53, 9 (2010), 1344–1359.
[41] Xiaopeng Yu, Jiechuan Jiang, Wanpeng Zhang, Haobin Jiang, and Zongqing Lu. 2022. Model-based opponent modeling. *Advances in Neural Information Processing Systems* 35 (2022), 28208–28221.

A Trading Algorithms

Input: Fundamental price $price$, inventory $inventory$, current time-step t
for *each step* **do**
    `// T is the terminal time-step - the trading horizon limit`
    rPrice ← `ReservationPrice(price,inventory,riskAversion,t,T);`
    s ← `Spread(riskAversion,priceVolatility,generositySensitivity,t,T);`

    `// The order size is preset to satisfy the incoming demand`
    `SendBuyOrder(rPrice − s/2, orderSize);`
    `SendSellOrder(rPrice + s/2, orderSize);`
end

Algorithm 1: Avellaneda-Stoikov Market Making [2].

Input: Market $midprice$, $askPrice$, $bidPrice$
for *each step* **do**
    signal ← `FastEWMA(midprice,fDecay) - SlowEWMA(midprice,sDecay);`

    `// The order size is preset to 1`
    if |signal| ≥ threshold **then**
        if signal > 0 **then**
            `SendBuyOrder(askPrice,orderSize);`
        else
            `SendSellOrder(bidPrice,orderSize);`
        end
    else
        `// Do not trade`
    end
end

Algorithm 2: Fast-Slow EWMA Trend Following.

Input: Fundamental alpha $alpha$, Market $askPrice$, $bidPrice$
for *each step* **do**
    `// The order size is preset to 1`
    if alpha ≥ thresholdBuy **then**
        `SendBuyOrder(askPrice,orderSize);`
    else
        if alpha ≤ thresholdSell **then**
            `SendSellOrder(bidPrice,orderSize);`
        end
    end
end

Algorithm 3: Informed Trading.

Input: Market $askPrice$, $bidPrice$
for *each step* **do**
    if activationRate < `Random(0, 1)` **then**
        `// The order size is preset to 1`
        if `Random(`$[true, false]$`)` **then**
            `SendBuyOrder(askPrice,orderSize);`
        else
            `SendSellOrder(bidPrice,orderSize);`
        end
    else
        `// Do not trade`
    end
end

Algorithm 4: Noise Trading.