



UNIVERSITY OF  
CAMBRIDGE

Department of Computer  
Science and Technology

# Real-Time Detection of Potential Insider Trading in Blockchain Prediction Markets

Yasith Disanayaka Medagama Disanayakage

Lucy Cavendish College

June 2026

Submitted in partial fulfillment of the requirements for the  
Master of Philosophy in Advanced Computer Science

Total page count: 65

Main chapters (excluding front-matter, references and appendix): 48 pages (pp 8–55)

Main chapters word count: 14945

Methodology used to generate that word count:

Overleaf's word count tool

## Declaration

I, Yasith Disanayaka Medagama Disanayakage of Lucy Cavendish College, being a candidate for the Master of Philosophy in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose. In preparation of this report, I adhered to the Department of Computer Science and Technology AI Policy. I am content for my report to be made available to the students and staff of the University.

**Signed: YASITH DISANAYAKA MEDAGAMA DISANAYAKAGE**

**Date: 08/06/2026**

# Abstract

Prediction markets are among the most effective mechanisms for aggregating distributed information, but their pseudonymous and permissionless structure creates conditions under which informed trading on material non-public information is difficult to detect and impossible to police through conventional identity-based surveillance. Polymarket, the dominant decentralised prediction market operating on the Polygon blockchain, processed over \$3 billion in monthly trading volume by late 2025 [8], yet its fully public on-chain ledger has remained, until now, largely unexamined as a forensic resource for detecting informed trading.

In this work we design, implement, and evaluate the first open and live-deployable insider trading detection system for decentralised prediction markets, organised into three analytical layers. The first applies five trade-level behavioural detectors, each testing a single trade for statistical anomaly against its market’s recent history, fused into one per-trade suspicion score. The second adds context no individual trade reveals, finding wallets that trade in coordinated synchrony and wallets that habitually trade just ahead of large price moves. The third corroborates this with on-chain forensics, tracing USDC transfers between flagged wallets on the Polygon blockchain to surface common ownership independent of trading behaviour.

Evaluation confronts the absence of any authoritative insider list, which a pseudonymous, unregulated market cannot provide. Following the convention standard in empirical finance since Meulbroek [27], abnormal post-resolution returns serve as a statistical proxy for informed trading, and the central claims are corroborated against a curated set of ten markets in which informed trading was reported independently of any return analysis. On these the system recovers 16 of 22 reported insider entries, against just 1 for the strongest comparable causal baseline, the only method to recover more raises roughly ten times as many false alerts. Across four consecutive, held-out windows, it attains the highest precision and  $F_{0.5}$  of any causal method tested and remains competitive with retrospective methods that exploit post-resolution information it cannot use, while flagging an order of magnitude fewer wallets than the high-recall causal baselines. As retrospective methods cannot be acted on as trades arrive, the system produces the most actionable worklist of any considered, and it sustains this at a throughput exceeding 130,000 trades per second, satisfying requirements for live deployment.

# Acknowledgements

To Prof. Stajano and Dr. Samaria, thank you for believing in one of my strangest ventures yet and for making the degree a unique experience, to say the least.

To family, thank you for your unconditional support, as always.

# Contents

|          |                                                          |           |
|----------|----------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                      | <b>8</b>  |
| <b>2</b> | <b>Related Work</b>                                      | <b>11</b> |
| 2.1      | Prediction Markets and Information Aggregation . . . . . | 11        |
| 2.2      | Insider Trading Detection in Financial Markets . . . . . | 12        |
| 2.3      | Blockchain Forensics and Graph-Based Analysis . . . . .  | 14        |
| 2.4      | Prediction Market Integrity and Polymarket . . . . .     | 16        |
| 2.5      | Research Gap . . . . .                                   | 17        |
| <b>3</b> | <b>Design and Implementation</b>                         | <b>18</b> |
| 3.1      | Polymarket: Platform Mechanics . . . . .                 | 18        |
| 3.2      | System Architecture . . . . .                            | 19        |
| 3.3      | Trade Representation . . . . .                           | 21        |
| 3.3.1    | The Trade object . . . . .                               | 21        |
| 3.3.2    | Columnar batch representation . . . . .                  | 21        |
| 3.3.3    | Notional pre-filter . . . . .                            | 21        |
| 3.4      | Position Tracking and Derived Quantities . . . . .       | 22        |
| 3.5      | Trade-Level Detectors . . . . .                          | 23        |
| 3.5.1    | Volume Anomaly Detector . . . . .                        | 23        |
| 3.5.2    | Probability Impact Detector . . . . .                    | 24        |
| 3.5.3    | Accumulation Detector . . . . .                          | 24        |
| 3.5.4    | Extreme Position Detector . . . . .                      | 24        |
| 3.5.5    | Contra-Outcome Silence Detector . . . . .                | 25        |
| 3.6      | Signal Fusion and Flagging . . . . .                     | 26        |
| 3.6.1    | Noisy-OR fusion . . . . .                                | 26        |
| 3.6.2    | Boost application and alert threshold . . . . .          | 26        |
| 3.6.3    | Flag rate and wallet prediction . . . . .                | 27        |
| 3.7      | Temporal Co-Activity Clustering . . . . .                | 27        |
| 3.7.1    | Graph Construction: Bucket Projection . . . . .          | 27        |
| 3.7.2    | Pruning and Community Detection . . . . .                | 29        |
| 3.7.3    | Cluster Multiplier . . . . .                             | 29        |
| 3.7.4    | Amplification, Not Independent Detection . . . . .       | 30        |

|          |                                                                            |           |
|----------|----------------------------------------------------------------------------|-----------|
| 3.8      | Outcome Awareness . . . . .                                                | 30        |
| 3.9      | Jump Anticipation . . . . .                                                | 31        |
| 3.9.1    | Jump Detection . . . . .                                                   | 31        |
| 3.9.2    | Wallet-Level Alignment Scoring . . . . .                                   | 31        |
| 3.9.3    | Jump-Anticipation Multiplier . . . . .                                     | 32        |
| 3.10     | On-Chain Attribution . . . . .                                             | 32        |
| 3.11     | Engineering Infrastructure . . . . .                                       | 33        |
| 3.11.1   | Caching . . . . .                                                          | 33        |
| 3.11.2   | Causal boost replay . . . . .                                              | 34        |
| 3.12     | Live Deployment . . . . .                                                  | 34        |
| <b>4</b> | <b>Evaluation Methodology</b>                                              | <b>36</b> |
| 4.1      | Ground Truth . . . . .                                                     | 36        |
| 4.1.1    | Return-based wallet labelling . . . . .                                    | 36        |
| 4.1.2    | Full-history labels . . . . .                                              | 37        |
| 4.1.3    | Resolution inference . . . . .                                             | 37        |
| 4.2      | Circularity and Proxy-Label Limitations . . . . .                          | 38        |
| 4.3      | Metric Families . . . . .                                                  | 38        |
| 4.4      | Optimisation Objectives . . . . .                                          | 39        |
| 4.4.1    | Wallet-oriented objective: $F_{0.5}$ . . . . .                             | 39        |
| 4.4.2    | Trade-oriented objective: lower-confidence-bound weighted return . . . . . | 40        |
| 4.5      | Optimisation Procedure . . . . .                                           | 40        |
| 4.6      | Temporal Validation Protocol . . . . .                                     | 41        |
| 4.7      | Strict Rolling-Window Replay with Full-History Labels . . . . .            | 41        |
| 4.8      | External Validation and Comparison Protocols . . . . .                     | 42        |
| 4.9      | Baseline Methods . . . . .                                                 | 43        |
| 4.10     | Implementation Correctness . . . . .                                       | 44        |
| <b>5</b> | <b>Results</b>                                                             | <b>45</b> |
| 5.1      | Runtime Performance . . . . .                                              | 45        |
| 5.2      | Wallet-Level Surveillance Configuration . . . . .                          | 46        |
| 5.3      | Trade-Level Configuration . . . . .                                        | 46        |
| 5.4      | Comparison Against Prior Methods . . . . .                                 | 47        |
| 5.5      | Insider Trading Case Studies . . . . .                                     | 48        |
| 5.6      | Detector Ablations . . . . .                                               | 50        |
| 5.7      | Clustering Effectiveness . . . . .                                         | 51        |
| <b>6</b> | <b>Summary and Conclusions</b>                                             | <b>53</b> |
| 6.1      | Summary . . . . .                                                          | 53        |
| 6.2      | Limitations . . . . .                                                      | 53        |
| 6.3      | Future Work . . . . .                                                      | 54        |

|          |                                                  |           |
|----------|--------------------------------------------------|-----------|
| <b>A</b> | <b>Technical details</b>                         | <b>61</b> |
| A.1      | Glossary of Terms . . . . .                      | 61        |
| A.2      | Trade-Level Detector Details . . . . .           | 61        |
| A.3      | Parameter Search Grids . . . . .                 | 62        |
| A.3.1    | Trade-Level Detector Grids . . . . .             | 63        |
| A.3.2    | Clustering and Jump-Anticipation Grids . . . . . | 63        |
| A.4      | Trade-Oriented Objective: Derivation . . . . .   | 64        |

# Chapter 1

## Introduction

“It’s a tax on stupidity.” That was Warren Buffett’s verdict on the boom in legalised sports betting, delivered shortly after he stepped down from Berkshire Hathaway [14], and for a casino or a bookmaker it is hard to argue with. There the house sets odds carrying a negative expected return, and over enough wagers that built-in edge transfers wealth from players to the operator with arithmetic certainty: the house always wins.

A prediction market looks at first like the same proposition, people staking money on uncertain future events, but it differs in one decisive respect. There is no operator setting the odds to guarantee its own profit. Participants trade contracts directly with one another, each paying a fixed sum if a specified event occurs and nothing if it does not, and because the price of a contract is the market’s own estimate of how likely the event is, a participant bets not against rigged odds but against the aggregate judgement of everyone else trading. Anyone who knows more than the rest of the market can expect to profit at their expense. That property is what makes prediction markets valuable as forecasting instruments, placing them in theory among the most efficient information-aggregation mechanisms yet devised [46], and it is exactly what this dissertation sets out to police.

Polymarket, a decentralised prediction market, has brought this mechanism to an unprecedented scale: by late 2025 the platform was processing over \$3 billion in monthly trading volume across hundreds of active markets, with more than 450,000 participating wallets [8]. Yet the same properties that make Polymarket compelling as a forecasting platform (open access, pseudonymity, and permissionless trading) also make it attractive to a different kind of participant: one trading not on carefully considered public information, but on material non-public information unavailable to everyone else. In October 2025, Norwegian Nobel Committee officials opened an investigation after an unusual surge in prediction market bets correctly anticipated the Nobel Peace Prize winner hours before the public announcement [43]. The incident is illustrative rather than isolated. In traditional financial markets, insider trading has been documented, prosecuted, and studied

for decades. In decentralised prediction markets, it remains almost entirely unexamined.

This gap reflects a deeper structural puzzle. Polymarket’s underlying transaction ledger is fully public and permanently archived on the Polygon blockchain, every trade, wallet address, and timestamp is available to anyone with a network connection. And yet coordinated informed trading, if it occurs systematically, goes largely undetected. Not because the data is hidden, but because transparency alone is not surveillance. A public ledger records what happened; it does not identify who knew what, or when, or why. Extracting an actionable signal from millions of pseudonymous transactions requires deliberate methodology: statistical models of normal trading behaviour, graph-theoretic tools for detecting coordination, and a way to establish whether the resulting system works at all when there is no authoritative list of known insiders to test it against.

This project develops exactly that methodology, organised into three analytical layers. The word *layer* here denotes a functional partition of the system, not a stage in a pipeline: the layers run concurrently, and the later ones supply evidence that adjusts the first layer’s output as trades arrive rather than waiting for one stage to finish before the next begins.

The first and primary layer is a set of five trade-level behavioural detectors. Each inspects a single trade against the recent history of its market for statistical anomalies. Four of the five adapt signals established in the market-microstructure and insider-trading literature [20, 10, 27] to the binary-token structure of a prediction market; the fifth, contra-outcome silence, is new. Their individual confidences are then fused into a single per-trade suspicion score.

The second layer adds contextual evidence and runs as periodic background jobs that never block the per-trade path. A temporal co-activity component builds a graph over wallet addresses and looks to find clusters of wallets trading in suspiciously synchronised patterns; a jump-anticipation component scores each wallet’s tendency to trade just ahead of large price moves. Neither raises an alert on its own, only boosting future alert scores. The third layer is an on-chain attribution step that traces USDC transfers between cluster members on the Polygon blockchain, yielding forensic evidence of common ownership that is structurally independent of the behavioural signal.

The complete system runs continuously in a containerised deployment. Its parameters are tuned by multi-start coordinate descent: an optimiser that repeatedly proposes a *candidate configuration*, that is one complete setting of every tunable parameter, such as each detector’s thresholds and the alerting cut-off, then scores it, and retains the best. A cached-signal infrastructure (Chapter 3) reduces the cost of evaluating each candidate significantly, which is what makes this search feasible, even on a single laptop.

Validation confronts the central weakness of the problem head-on. Without a confirmed list of insider traders on Polymarket (there is no regulator with subpoena power over a pseudonymous, permissionless market) the project cannot ask whether it recovers a

known set of offenders. It instead adopts the convention standard in empirical finance since Meulbroek [27]: persistent abnormal post-resolution returns are the best available proxy for informed trading, with wallets whose returns are statistical outliers within their own market labelled *statistical insiders*. This proxy is unavoidably circular, the system is tuned to find abnormally profitable wallets, and an insider is defined as one, so Chapter 5 judges the system against several independent lines of evidence rather than any single score, and is explicit throughout about what each can and cannot establish.

The thesis this dissertation defends can be stated plainly: informed trading in a decentralised prediction market leaves a behavioural signal detectable *in real time*, from information available at the moment each trade arrives, rather than only in the retrospective post-mortem prior work has relied on. Building on this, the dissertation aims to demonstrate a system built on that signal that is competitive with the retrospective state of the art while operating under the strictly harder causal constraint.

As a concrete illustration of what this buys in practice: run against a curated set of independently reported Polymarket insider-trading cases, the system flags 16 of the 22 reported insider entries at least once, against just 1 for the strongest comparable causal baseline. The only method to recover more, an Isolation Forest, does so while flagging roughly an order of magnitude more wallets, so the system produces the most selective and actionable worklist of any method considered.

# Chapter 2

## Related Work

### 2.1 Prediction Markets and Information Aggregation

Prediction markets let people buy and sell contracts whose payoff depends on a future event, so that the price of a contract can be read as the market’s collective estimate of how likely that event is. This mechanism predates blockchain. The standard survey, Wolfers and Zitzewitz [46], describes the centralised, small-scale “event” or “information” markets of 2004, none were permissionless, pseudonymous, or on-chain. The relevant contract type they distinguish is the *winner-take-all* contract, paying a fixed sum if an event occurs and nothing otherwise, the type Polymarket’s binary YES/NO markets implement. Most such markets cleared trades through a continuous double auction, a standard exchange order book.

The reason such markets aggregate information well is one of incentives. Attaching money to a belief forces participants to back it with capital, and the poorly informed lose that capital to the better informed. That the informed earn this return is structurally necessary, not incidental: Grossman and Stiglitz [12] show prices cannot fully reflect private information, since if they did no one would be paid to gather it. As trades absorb new information, the contract’s price, which is just its implied probability, moves toward the event’s true probability.

Wolfers and Zitzewitz [46] document that across elections, economic indicators, and other domains these implied probabilities matched or beat expert forecasts and polling aggregates, meanwhile Arrow et al. [2], in *Science*, argue the same incentive alignment makes prediction markets a valuable complement to surveys and expert panels that regulators should accommodate, again for centralised, experimental venues rather than public blockchains. The on-chain markets studied here inherit this price-as-belief logic but, as the rest of the chapter argues, change the market-integrity problem entirely.

Polymarket implements the aforementioned winner-take-all mechanism on a public blockchain. Each market is a yes/no question with a fixed deadline (“Will the Federal Reserve cut

rates in March?”), curated and resolved by Polymarket, with two tokens whose prices lie in  $[\$0, \$1]$  and sum to roughly \$1.00, read directly as probabilities. The full order-matching, settlement, and resolution mechanics are outlined in Section 3.1; what matters here is that every completed trade is recorded publicly on-chain. The blockchain chosen was Polygon [18], an Ethereum-compatible [47] chain built to scale Ethereum at far higher throughput and fees of only a couple cents while inheriting much of its security [49], making a market of many small bets viable where Ethereum’s per-transaction cost and speed would not. For this work, Polygon’s decisive property is not throughput but its status as a public blockchain. Like Ethereum and similar systems, it records transactions openly and persistently, exposing the wallet, token, price, size, and timestamp information that later chapters use as a forensic resource.

These same properties create the integrity problem this dissertation addresses. Polymarket’s trading is intermediated, as orders are matched off-chain on its own servers and only the resulting trade settles on Polygon, so the platform could in principle identify its users and does apply registration and geofencing controls in some jurisdictions, though IP-based geofencing can be circumvented by standard means such as a VPN. The surveillance gap is therefore not an absence of sign-up checks but structural: on-chain, participants are only pseudonymous wallet addresses anyone can create and fund; there is no obligation to disclose holdings as on a regulated exchange; and no regulator holds subpoena power over them. What does exist is the complete public record of every settled trade. Whether prediction market prices can be trusted therefore depends on how much trading on material non-public information that record conceals, and whether it can be recovered, the question this dissertation takes up.

## 2.2 Insider Trading Detection in Financial Markets

The theoretical basis for treating order flow as a signal of private information comes from Kyle [20] and Glosten and Milgrom [10]. Kyle considers a trader who knows something the rest of the market does not and wants to profit from it before it becomes public. Buying aggressively would push the price up and give the game away, so the trader’s best strategy is to spread purchases out over time and hide them among the random orders of uninformed “noise” traders. The market maker filling these orders cannot tell informed from uninformed flow and so sets prices that merely break even on average. The prediction that matters here is that informed trading reveals itself as a gradual drift in the price toward the value the informed trader is targeting, and that this drift is more visible the larger the informed share of total volume. Glosten and Milgrom reach a complementary result from a different model: because any incoming order might carry private information, the market maker defends itself by widening the gap between its buy and sell quotes, the bid–ask spread, in effect charging every trader a premium for the risk of trading against an insider [10]. The more likely informed trading is, the wider that premium.

This work reuses the central idea of both models, that informed flow leaves a measurable footprint in prices and volume, but applies it under the opposite constraint. Where these models describe how a market maker *should price*, the detectors of Chapter 3 look directly for the patterns the models predict: the gradual, one-sided accumulation of Kyle’s informed trader (the Volume Anomaly and Accumulation detectors), and the market maker’s defensive withdrawal of Glosten and Milgrom (the Contra-Outcome Silence detector).

Empirically, the canonical study is Meulbroek [27]. Using SEC prosecution records from 1980–1989 as ground truth, she shows that days on which insiders are known to have traded carry abnormal returns of around 3%, and that nearly half of the pre-announcement price run-up in takeover targets occurs on those days, which is direct evidence that informed trading is statistically visible even when insiders try to conceal it. Two features matter for this dissertation. First, the ground truth is regulatory: the cases were caught by the SEC through its own enforcement machinery, and Meulbroek’s contribution is to show that they left a return footprint, not to catch them. Such enforcement-confirmed labels cannot exist for a pseudonymous, unregulated market. Second, and as a consequence, this work borrows Meulbroek’s core premise, that abnormal returns are the best available proxy for informed trading, but is forced to promote it from a validation device to the *only* available ground-truth signal, a dependence whose circularity is confronted directly in Chapter 4.

Later work qualifies how reliable that footprint is. Ahern [1], on a more recent sample of prosecuted cases, finds that standard liquidity-based measures of informed trading become unreliable when insiders hold long-lived information and can time their trades patiently; the order-flow signal is dependable only when information is short-lived and insiders must act quickly. Kacperczyk and Pagnotta [17] reach a similar verdict: prosecuted insiders tend to trade precisely when the market is liquid and adverse-selection measures are low, that is, when they are hardest to detect. Both findings bear directly on prediction markets, where information lifetimes range from hours (a leaked decision) to weeks (a slow shift in polling): no single microstructure signal can be trusted across both regimes, which is the empirical case for combining several detectors rather than relying on one.

More recent work brings computation to the problem. Mazzarisi et al. [25], in a collaboration with the Italian securities regulator CONSOB, build networks of investors whose trading is statistically synchronised around known price-sensitive events and then apply community detection to surface clusters of apparently coordinated traders. Their guiding premise is the one this dissertation also adopts: coordination is more reliably caught at the level of the network than of the individual trade, and this work reuses the same community-detection machinery (Section 3.7). Two differences are worth stating. First, their setting is a regulated market with a known calendar of price-sensitive events and resolved investor identities, structural advantages absent on Polymarket, where there is

no event calendar and traders are pseudonymous wallets; this work recovers the missing identity evidence through on-chain attribution (Section 3.10) instead. Second, the authors are explicit that their method was developed as a proof of concept and is “in no way” a tool used in CONSOB’s actual investigations [25]; CONSOB has separately described related AI systems as prototypes tested to support preliminary insider-trading analyses, with regular operational integration only a future step [34]. A separate strand flags accounts whose order-flow patterns around events are statistically extreme, reporting strong results on detailed broker-level data [16], but it relies on broker account identifiers that pseudonymous blockchains do not provide.

Concurrent work by Jaeger et al. [15] is methodologically the closest to the clustering layer of this dissertation: working from a decade of SEC-reported insider trades, they link insiders by the temporal similarity of their trading and flag anomalous subgraphs, the same premise as the temporal co-activity graph built here (Section 3.7). The differences are the recurring ones: their insiders are named, regulator-collected filings rather than pseudonymous wallets, and their analysis is retrospective over a fixed ten-year window rather than causal and real-time.

The common thread is that individual trade-level signals are necessary but rarely sufficient on their own: volume and price-impact measures detect informed trading only under favourable conditions, and network- and timing-based analysis supplies the discriminating power they lack. The system developed here combines both, which is the design the next chapter sets out.

## 2.3 Blockchain Forensics and Graph-Based Analysis

Blockchain transaction ledgers, while pseudonymous, are structurally transparent: every transfer of value between addresses is publicly recorded with timestamps and amounts. Ron and Shamir [35] were among the first to exploit this systematically, reconstructing the full Bitcoin transaction graph and showing that structural regularities, major entities, large flows, could be recovered despite the absence of identity information. Meiklejohn et al. [26] turned this into practical de-anonymisation: they clustered addresses likely under common control using heuristics based on evidence of shared authority, then attached real-world labels through their own test purchases, and argued that an agency with subpoena power could follow funds from a pseudonymous service to an exchange that could be compelled to name the owner.

The relevance to this work is twofold. First, on method: the shared-authority clustering principle is exactly what the on-chain attribution layer of the present system reuses (Section 3.10), substituting USDC transfers between Polymarket wallets for the Bitcoin idioms Meiklejohn exploited. Second, on impact: unlike the equity-market detection methods of the previous section, this line of work has had concrete operational influence.

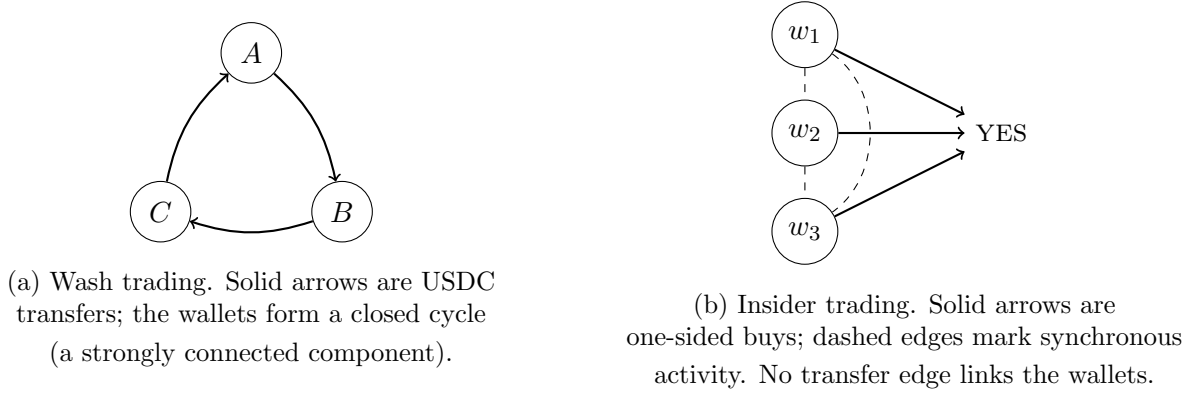


Figure 2.1: The transfer-graph signatures of wash trading and insider trading are nearly orthogonal. In wash trading (a) capital circulates and returns, forming a detectable cycle. In insider trading (b) the coordinated wallets place synchronised one-sided bets on the same outcome but need never transfer funds to each other, so the cycle a wash-trade detector looks for does not exist; the signal lives instead in the dashed co-activity structure.

Address-clustering heuristics are now standard in blockchain-intelligence tools used for enforcement and compliance [6], its Silk Road case study assisting later law-enforcement blockchain tracing in the Ulbricht prosecution [41], and the paper was subsequently recognised with the IMC 2024 Test-of-Time Award [39]. Where the trade-detection literature produces triage, blockchain forensics has produced *convictions*, because the public ledger supplies the one thing a regulated market hides: a permanent, common record of who moved money to whom.

In decentralised exchange settings, graph-based analysis has been applied most extensively to wash trading. Victor and Weintraud [42] build a directed transfer graph over Uniswap trades, with addresses as nodes and token transfers as edges. Wash trading, inflating reported volume by trading with oneself across multiple addresses, shows up in this graph as strongly connected components: closed cycles in which capital circulates among a set of colluding addresses and returns, approximately, to its origin, and they find that a substantial share of apparent Uniswap volume is generated this way. This work reuses their basic object, a transfer graph over addresses, but reads it for the opposite signature, which is the distinction the next paragraph draws out.

The structural distinction between wash trading and insider trading is more fundamental than a difference of degree, and is illustrated in Figure 2.1. A cycle-based detector finds nothing in the insider case because its target, the closed loop, is structurally absent; conversely, the temporal co-activity graph built in this work (Section 3.7) finds nothing in the wash-trading case because it looks for synchronised directional bets, which wash traders by design do not place. The two signatures are nearly orthogonal: detecting one phenomenon says almost nothing about the other.

## 2.4 Prediction Market Integrity and Polymarket

Wash trading has been documented at substantial scale on decentralised platforms. Sirolly et al. [36] give an iterative, network-based algorithm to detect it on Polymarket specifically, estimating that roughly 25% of all trading volume between 2022 and 2025 was artificial. Their method scores each wallet first on its propensity to open and close positions repeatedly on the same outcome, a signature of volume inflation rather than conviction, and then propagates suspicion through the trade graph, raising a wallet’s score when it trades predominantly against others already deemed suspicious, iterating to convergence. The relationship to this work is one of precondition rather than method: Sirolly et al. target a different phenomenon (volume inflation, not informed trading) with a different graph signature (the position-cycling rings of Figure 2.1a), so none of their machinery is reused directly. What their result obliges this work to do is account for the fact that a large fraction of recorded Polymarket trades are not information-motivated at all; the notional pre-filter and outcome-aware bucketing of Chapter 3 are in part a response to exactly this contamination.

Ng et al. [30] study price discovery across Polymarket and Kalshi, a US, CFTC-regulated prediction market that runs on a conventional order book rather than a blockchain, during the 2024 US presidential election. Their central finding is that net order imbalance from large trades predicts subsequent returns, and that whichever platform sees the greater directional flow from large trades tends to lead price discovery. This indirectly supports the detector design of Chapter 3: if large directional trades carry price-relevant information, then unusually large or unusually well-timed directional trades may carry *private* information.

The closest existing work to this dissertation is Mitts and Ofir [28], the first systematic empirical study of informed trading on Polymarket. Across more than 93,000 markets and nearly 50,000 wallets from February 2024 to February 2026, they build a per-wallet-per-market suspicion score by standardising and summing five signals, cross-sectional and within-trader bet size, historical profitability, pre-event timing, and directional concentration, and flag the top decile. Flagged traders win 69.9% of the time, exceeding the null by more than 60 standard deviations under a permutation test, with estimated suspicious profits of roughly \$143 million; the paper also presents market-level case studies, notably the February 2026 US–Israeli strike on Iran, where six newly created wallets collectively earned about \$1.2 million.

Despite its scale, the Mitts and Ofir study differs from this work in three ways that together define the gap this dissertation fills. First, their analysis is entirely retrospective: the suspicion score uses post-resolution profitability as one of its five signals and so cannot be computed until a market has resolved. This work instead produces alerts in real time, as trades arrive, with no access to resolution outcomes, it is causal by construction. Second, their score treats each wallet–market pair independently, with no mechanism for

detecting coordination across wallets; the temporal co-activity graph and community-detection layer developed here is aimed precisely at that gap. Third, they do not use on-chain forensics, USDC transfer tracing, proxy-wallet attribution, common-ownership detection, whereas the present system adds blockchain evidence as a third analytical layer. The two projects are therefore complementary: Mitts and Ofir establish, retrospectively and at scale, that informed trading is prevalent on Polymarket, which is the empirical motivation for building the real-time detection system this dissertation contributes.

## 2.5 Research Gap

The literature reviewed above converges on three strands that remain largely disconnected. Market microstructure theory [20, 10] explains how private information enters prices through order-flow dynamics and predicts the statistical signatures informed trading should leave. Blockchain forensics [35, 26] shows that pseudonymous transaction ledgers can nevertheless be analysed with graph-based methods to recover structural relationships between addresses. And recent empirical work on Polymarket confirms that both wash trading [36] and informed trading [28] are present on the platform at non-trivial rates.

What is missing is a system that joins these strands into a unified, real-time detection pipeline. Insider-trading detection methods from traditional finance assume regulatory infrastructure, broker identities, enforcement records, centralised order books, that does not exist in decentralised markets. Blockchain forensic methods target wash trading or de-anonymisation rather than information-motivated trading. And the most directly comparable work on Polymarket [28] operates retrospectively, folding resolution outcomes into its score, and exploits neither network structure nor on-chain attribution.

This dissertation addresses the gap with a three-layer detection framework that combines trade-level behavioural detectors with temporal graph clustering and blockchain forensics, operates in real time on live Polymarket data, and validates its output using post-resolution economic performance without feeding that performance back into the detection logic. Polymarket itself operates an internal monitoring system, credited with the first insider-trading arrest in decentralised prediction-market history [33, 38], but provides no public detail on how it works, and as the platform operator it surveils with privileged access to off-chain order flow and registration data that no outside party holds. The framework presented here is, to the author’s knowledge, the first *open, reproducible* methodology to detect informed trading on a decentralised prediction market in real time, from the public on-chain record alone, the vantage point of an external investigator with no privileged access, and is benchmarked against existing detection methods adapted to the same causal, real-time regime.

# Chapter 3

## Design and Implementation

This chapter details the detection system and the engineering infrastructure that makes the whole system work, but also fast and robust enough to be useful. Parameter search ranges for each detector are given in Appendix A.3.

### 3.1 Polymarket: Platform Mechanics

Each outcome of a binary YES/NO market in Polymarket is an ERC-1155 token on Polygon, issued under the Gnosis Conditional Token Framework (CTF) [32]. The CTF enforces the invariant behind that probability reading: one YES and one NO token merge to redeem exactly one USDC (a dollar-pegged stablecoin), so the two prices sum to one.

Markets are resolved by the UMA Optimistic Oracle [40], a decentralised resolution mechanism (Appendix A.1). Once the outcome is fixed, holders of the winning token can redeem each token for \$1.00 of USDC, meanwhile losing tokens expire worthless.

**Trading mechanics.** Order matching is handled off-chain by a central limit order book (CLOB), a standard financial exchange mechanism in which buyers and sellers post limit orders at specific prices and the exchange matches them when a counterparty’s price crosses the spread. Polymarket operates its own CLOB infrastructure; orders are matched on Polymarket’s servers, not on the blockchain. Settlement, however, is on-chain, with every matched trade producing a public transaction on Polygon.

**Data access.** The system retrieves trade data via Polymarket’s REST Data API, polling at a configurable interval. Each API response returns up to a configured number of the most recent fills across one or more `condition_ids`. The response is a JSON array of fill records, where each record corresponds to one counterparty in a matched fill: a trade that matched buyer A against seller B produces two rows, one for each party. Market metadata is fetched separately from Polymarket’s Gamma API. The live system reads from these APIs directly; the historical windows used for backtesting and optimisation (Chapter 4)

use a local store of all past fills, built with the open-source collection tooling of warproxxx’s `poly_data` [44], which was then converted to an sqlite database (over 50GB) for use in the system. All access uses publicly documented APIs and requires no privileged credentials. This project received ethical approval by the university departmental ethics committee. The data is public and pseudonymous it raises no human-subjects concern, and the system attributes wallets only to one another, never to real-world identities. No flagged-wallet worklist or other identifying output is published, and the case studies (Section 4.8) cite only suspicion already made public by journalists, investigators, or regulators.

**Participant taxonomy.** The detectors are designed to separate a small number of qualitatively different actors. *Noise traders* trade in small sizes on intuition or entertainment and they should not trigger alerts. *Market makers* quote on both sides of the order book, profiting from the bid-ask spread, and carry approximately zero net directional exposure; their individual trades can be large but lack directional conviction. *Arbitrageurs* exploit temporary mispricings between outcomes or between Polymarket and other prediction platforms and as such, their positions are hedged rather than concentrated. *Informed fundamental traders* trade directionally on public information and can earn abnormal returns. They are not the target of this system and are, in effect, what makes the market accurate. *Insiders* trade on material non-public information and produce the same behavioural signature as informed fundamental traders.

No system can distinguish the two classes from behavioural data alone, as it would require timestamped data relating to news events as an input, a limitation discussed in Chapter 6. Its job is to identify *abnormally informationally advantaged* trading; whether the information advantage was lawful is a question for downstream human investigation.

## 3.2 System Architecture

The detection pipeline is organised around three design principles that together make backtested parameters directly deployable in the live system.

**Shared mutable state, stateless analysers.** All state that changes as trades arrive (current prices, wallet position histories, rolling trade-size statistics, cluster assignments) lives in a single `DetectionContext` object. Detectors, however, are stateless analysers: each `Detector` subclass implements an `analyze(trade, context)` method that reads from the context and returns either a `DetectionSignal` carrying a confidence score and a reason string, or `None`. Detectors do not write to the context. The separation is enforced by code structure: the `Detector` abstract base class exposes only read access to context, and an orchestrating `process_trade` method in `InsiderTradingDetector` calls `context.add_trade()` only after all detectors have run, so no detector sees a partially-updated state. This makes detectors independently testable and allows the parameter

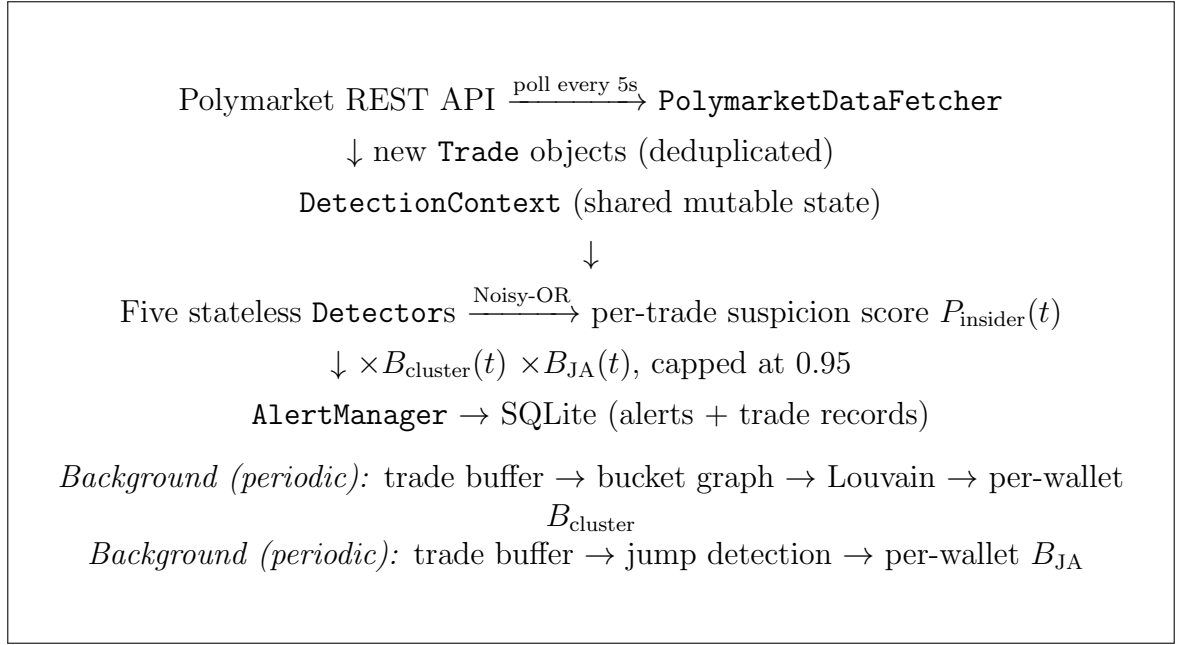


Figure 3.1: System data flow. Per-trade detection runs synchronously in the main polling loop; the clustering and jump-anticipation components run periodically as background jobs and publish multipliers that the per-trade path reads without blocking.

optimiser to swap detector configurations without rebuilding the shared state.

**Three processing stages.** The per-trade path executes synchronously in the main polling loop (Figure 3.1): the five detectors analyse the trade, Noisy-OR fuses their confidences into  $P_{\text{insider}}(t)$ , the two background multipliers are applied, and the result is thresholded against  $\alpha$  to decide an alert (Section 3.6 gives the formulae). The context is updated only *after* all detectors have run, so none sees a partially updated state. The background clustering and jump-anticipation components run periodically on accumulated trade buffers and publish the per-wallet multipliers that subsequent trades read, without blocking the per-trade path.

**Live-backtest parity.** The same code path executes in live deployment and historical replay: only the data source differs, with `PolymarketDataFetcher` replaced by `HistoricalDataLoader` reading a SQLite store of historical fills, while every downstream component, `DetectionContext`, detectors, Noisy-OR fusion, boost application, alert thresholding, runs identically. The background layers preserve this parity too, by replaying their periodic schedule from event timestamps rather than wall-clock time (Section 3.11.2). This invariant is what makes parameters optimised in backtesting safe to transfer to the live system without re-tuning.

## 3.3 Trade Representation

### 3.3.1 The Trade object

A *trade* in this system corresponds to one side of one matched fill on the Polymarket order book. When buyer A’s order matches against seller B’s order, the Polygon blockchain records a single on-chain transaction, but Polymarket’s Data API returns two rows for it, one from A’s perspective and one from B’s, each with the `proxyWallet` field set to the relevant party. The system creates exactly one **Trade** object per API row.

Table 3.1: Fields of the normalised **Trade** object.

| Field                      | Description                                                        |
|----------------------------|--------------------------------------------------------------------|
| <code>wallet</code>        | Polygon address of the trader (proxy wallet)                       |
| <code>condition_id</code>  | CTF condition identifier (market key)                              |
| <code>outcome_index</code> | Outcome token index (0 or 1 for binary markets)                    |
| <code>side</code>          | Trade direction: BUY or SELL                                       |
| <code>price</code>         | Execution price in USDC, $p \in (0, 1)$                            |
| <code>size_tokens</code>   | Number of outcome tokens traded                                    |
| <code>notional_usdc</code> | Dollar value: <code>size_tokens</code> $\times$ <code>price</code> |
| <code>timestamp_ms</code>  | Execution timestamp in milliseconds since epoch                    |
| <code>tx_hash</code>       | On-chain transaction hash                                          |
| <code>asset</code>         | ERC-1155 token identifier ( <code>positionId</code> )              |

### 3.3.2 Columnar batch representation

For the live system, individual **Trade** objects are sufficient. The backtester, however, must process tens of thousands of trades per market per optimisation candidate. Allocating a Python object per trade at that scale is prohibitive. The **TradeBatch** class represents a per-market trade collection in a columnar layout: numeric fields (`price`, `size_tokens`, `notional_usdc`, `timestamp_ms`, `outcome_index`, `side`) are stored as NumPy arrays, while high-cardinality string fields (wallet addresses, transaction hashes) are compressed via integer pooling where each unique string is stored once in a list and all rows reference it by index. A **TradeView** proxy object presents the same attribute interface as **Trade** to downstream code, so detectors and evaluators require no modification.

### 3.3.3 Notional pre-filter

Trades below a configurable minimum notional (price) threshold  $v_{\min}$  are removed before they enter the detection pipeline. The prior is that actors trading on genuine private information will deploy meaningful capital; a wallet placing a \$2 bet on an obscure market adds noise to rolling statistics and inflates the flag-rate denominator without adding informational content. For the experiments ran, a default of \$500 is used.

### 3.4 Position Tracking and Derived Quantities

The `WalletProfile` class tracks each wallet's state within each market, updated incrementally as trades arrive. Three derived quantities are read by multiple detectors and introduced here to avoid repetition.

Let  $w$  denote a wallet,  $m$  a market, and  $o \in \{0, 1\}$  an outcome index. Let  $s_{w,m,o}$  be the wallet's net position in shares on outcome  $o$  (buys increment, sells decrement),  $p_o$  the most recent execution price for that outcome, and  $V_{w,m,o}$  the wallet's cumulative buy-plus-sell USDC volume on outcome  $o$ . Write  $V_{w,m} = \sum_o V_{w,m,o}$ .

**Directional ratio**  $r_{\text{dir}}(w, m)$  measures whether the wallet's activity is building net exposure or cycling it (buying then selling to earn spread, the difference between best bid and ask prices):

$$r_{\text{dir}}(w, m) = \min\left(1, \frac{\sum_o |s_{w,m,o} \cdot p_o|}{V_{w,m}}\right). \quad (3.1)$$

A value near 0 means buys and sells roughly cancel (market maker or short-horizon arbitrageur); a value near 1 means the wallet is a pure accumulator.

**Outcome concentration**  $c_{\text{out}}(w, m, o)$  measures how focused the wallet's activity is on a single outcome:

$$c_{\text{out}}(w, m, o) = \frac{V_{w,m,o}}{V_{w,m}}. \quad (3.2)$$

A value of 1.0 means all volume is on outcome  $o$ ; 0.5 in a binary market indicates roughly equal trading on both sides, consistent with hedging. The two ratios are independent and answer different questions:  $r_{\text{dir}}$  asks whether the wallet's dollars became lasting exposure or were traded away, while  $c_{\text{out}}$  asks *where* those dollars were placed, on one outcome or split across both. A high-conviction hedger scores high on the first and near 0.5 on the second.

**Outcome directional ratio**  $r_{\text{out}}(w, m, o)$  restricts the directional ratio to a single outcome:

$$r_{\text{out}}(w, m, o) = \min\left(1, \frac{|s_{w,m,o} \cdot p_o|}{V_{w,m,o}}\right). \quad (3.3)$$

The distinction from  $r_{\text{dir}}$  matters: a wallet can accumulate YES tokens while simultaneously cycling NO tokens. The market-level ratio would not detect this; the outcome-level ratio does. Both directional ratios are capped at 1, since a wallet that buys early and sees the price rise holds more than it ever committed, pushing the raw ratio above 1. Outcome concentration needs no such cap: both its terms are execution-price volumes and an outcome's volume cannot exceed the market total, so  $c_{\text{out}} \in [0, 1]$  by construction.

**Outcome-level tracking.** All position and volume tracking is keyed by (`condition_id`, `outcome_index`) rather than by market alone. A wallet buying \$1,000 of YES and \$1,000 of

NO is hedging; market-level tracking would report a directional ratio near 1.0 and misclassify it as an accumulator. Outcome-level tracking correctly identifies zero net directional exposure. This granularity ripples through the clustering layer too, where it prevents same-market simultaneous trading from being mis-scored as coordination (Section 3.8).

## 3.5 Trade-Level Detectors

Five detectors form the first analytical layer. Each receives a `Trade` and the current `DetectionContext`, and returns either a `DetectionSignal` carrying a confidence  $c \in [0, 1]$  alongside a human-readable reason string, or `None`. The reason string is also operationally significant: a human analyst acting on a flagged wallet needs to understand *which* trades triggered the flag and why, for example, “volume outlier:  $z = 3.7\sigma$  (\$42,000 vs  $\mu = \$8,500$ ,  $\sigma = \$9,100$ )”, rather than an opaque score.

The first four detectors apply microstructure signals identified in the insider trading literature [20, 10, 27], adapted to the binary token structure of prediction markets. The fifth, Contra-Outcome Silence, is a contribution of this work. Scaling constants, preconditions, and parameter ranges are deferred to Appendix A.2 and A.3.

All detectors share a configurable maximum confidence cap  $c_{\max}$  that bounds their individual contribution to the fused score. Two of them, Volume Anomaly and Extreme Position, additionally consult a shared *non-directional-trader filter*, which suppresses a wallet judged likely to be a market maker or arbitrageur in market  $m$  (directional ratio  $r_{\text{dir}}(w, m) < r_{\text{nd}}^*$  once it has accrued enough volume,  $V_{w,m} \geq V_{\text{nd}}^*$ ; Appendix A.2).

### 3.5.1 Volume Anomaly Detector

**Intuition.** Informed traders must at some point commit capital. This detector flags trades whose notional value is a statistical outlier against the recent size distribution for the same market, following Meulbroek [27], who finds insider-trading days carry significantly elevated volume.

**Definition.** Let  $\bar{v}$  and  $\sigma$  be the rolling mean and standard deviation of trade notional over a configurable lookback window, maintained incrementally via Welford’s online algorithm [45]. The detector fires when the z-score  $z_t = (v_t - \bar{v})/\sigma$  meets a threshold  $z^*$ , with confidence

$$c_{\text{vol}}(t) = \frac{c_{\max}}{1 + \exp(-(z_t - z^*)/1.5)} \quad (3.4)$$

And the sigmoid for smooth saturation for very large outliers. The non-directional-trader filter is applied: market makers routinely place large trades without directional conviction and should not be flagged for size alone.

### 3.5.2 Probability Impact Detector

**Intuition.** A trade moves the price by consuming the resting orders on the opposite side of the book: a large or aggressive order walks up (or down) successive price levels, dragging the last-traded price (the market’s implied probability) with it. A trade that substantially moves the price is therefore providing new information, public or private; Glosten–Milgrom [10] predicts informed trades cause larger price updates than uninformed ones of the same size.

**Definition.** Let  $p_{t-1}$  and  $p_t$  be the execution prices immediately before and after the trade. The detector computes a raw-probability sub-score  $s_p$  from  $\Delta p_t = |p_t - p_{t-1}|$  and a log-odds sub-score  $s_\ell$  from  $\Delta \ell_t = |\text{logit}(p_t) - \text{logit}(p_{t-1})|$ , each scaled to  $[0, 1]$ . It fires when *either* clears its threshold; when both do, they combine as

$$c_{\text{imp}}(t) = (0.4 s_p + 0.6 s_\ell) \cdot c_{\text{max}}. \quad (3.5)$$

The log-odds term carries the higher weight because it captures informational surprise more naturally: a move from 0.90 to 0.95 spans the same five percentage points as one from 0.50 to 0.55 but represents a far larger revision of a near-certain belief. The non-directional-trader filter is *not* applied: a single market-moving trade carries informational content regardless of the trader’s broader pattern.

### 3.5.3 Accumulation Detector

**Intuition.** An actor with strong private information continuously builds a large directional position on one outcome without hedging. The detector requires three conditions at once, outcome focus, directional persistence, and position magnitude, because each alone also describes ordinary retail momentum or liquidity-seeking flow; the conjunction isolates the pattern of interest.

**Definition.** The detector fires on a BUY by wallet  $w$  on outcome  $o$  of market  $m$  iff

$$c_{\text{out}}(w, m, o) \geq c_{\text{out}}^*, \quad r_{\text{out}}(w, m, o) \geq r_{\text{dir}}^*, \quad |s_{w,m,o} \cdot p_o| \geq A^*, \quad (3.6)$$

with confidence proportional to the outcome directional ratio,  $c_{\text{acc}}(t) = r_{\text{out}}(w, m, o) \cdot c_{\text{max}}$ , where  $c_{\text{out}}^*$ ,  $r_{\text{dir}}^*$  and  $A^*$  are minimum thresholds set during optimisation. Only BUY trades are checked, since sells are exits rather than new evidence. The detector does not use the shared filter: its outcome-level requirement  $r_{\text{out}} \geq r_{\text{dir}}^*$  is strictly stronger.

### 3.5.4 Extreme Position Detector

**Intuition.** A trader who knows an unlikely event will occur has strong incentive to buy the corresponding token cheaply. A large bet at a tail price, say \$20,000 on a YES token

at four cents, pays off only if the buyer holds information the market has not yet priced.

**Definition.** With tail threshold  $\tau$ , a trade is in the low tail if it is a BUY below  $\tau$ , or the high tail if it is a SELL above  $1 - \tau$ . Letting  $d_t$  be the distance from the tail boundary, confidence weights tail depth and trade size equally,

$$c_{\text{ext}}(t) = (0.5 \min(1, d_t/\tau) + 0.5 \min(1, v_t/5000)) \cdot c_{\text{max}}, \quad (3.7)$$

reflecting the joint hypothesis that both an extreme price and real capital commitment are needed for suspicion. The non-directional-trader filter is applied: market makers routinely quote at tail prices.

### 3.5.5 Contra-Outcome Silence Detector

**Intuition.** This detector operationalises an observation specific to binary prediction markets: informed buying pressure on one outcome should cause market makers to reduce activity on the *contra* outcome. The intuition follows Glosten–Milgrom [10]: when the probability of facing an informed counterparty rises, adverse-selection risk increases, so competitive liquidity providers must quote wider bid-ask spreads to break even. In a two-token market linked by the CTF invariant, this is observable as an unusual gap in trading on the opposite side.

**Definition.** Let  $\Delta t_{\text{contra}}$  be the time since the most recent contra-outcome trade, and  $g_{\text{med}}$  the median of the contra outcome’s recent inter-trade gaps. The *silence ratio* is

$$\rho_t = \Delta t_{\text{contra}}/g_{\text{med}}, \quad (3.8)$$

and the detector fires when  $\rho_t \geq \rho^*$  with confidence

$$c_{\text{contra}}(t) = \min(1, \rho_t/(2\rho^*)) \cdot c_{\text{max}}, \quad (3.9)$$

reaching the cap when silence persists at twice the threshold multiple of the normal gap. The trigger is deliberately asymmetric, the signal is a focal-outcome trade arriving *during* a contra-outcome silence, with no condition on the focal side’s own activity. A symmetric “both sides quiet” trigger would conflate informed-flow-driven withdrawal with ordinary lumpiness in a low-activity market. The non-directional-trader filter is not applied, since the triggering signal is a property of the market’s microstructure, not of the trading wallet.

## 3.6 Signal Fusion and Flagging

### 3.6.1 Noisy-OR fusion

The signals from whichever detectors fired on a trade are combined using the Noisy-OR model from Bayesian network theory [31]. Given  $k$  firing signals with confidences  $c_1, \dots, c_k$ , the fused score is

$$P_{\text{insider}}(t) = 1 - \prod_{i=1}^k (1 - c_i). \quad (3.10)$$

The model treats each detector as an independent potential cause of the trade being suspicious: the probability that the trade is *not* suspicious is the product of the per-detector complements, under the independence assumption, and the fused score is the complement of that product.

Two properties make Noisy-OR preferable to a weighted sum here. First, the output stays in  $[0, 1]$  regardless of how many detectors fire. Second, a detector returning **None** contributes a factor of one to the product and leaves the score unchanged, which is the correct behaviour when a detector has no opinion rather than a confident negative opinion.

The main limitation is the independence assumption: Volume Anomaly and Accumulation are positively correlated (a very large buy also tends to be an accumulation event), so Noisy-OR treats correlated evidence as independent and can over-state the combined score. The per-detector maximum confidence cap  $c_{\text{max}}$  partially mitigates this in practice: the optimiser settles on low caps for the frequently co-firing detectors (Volume Anomaly and Accumulation), bounding the per-trade contribution of the correlated pair.

### 3.6.2 Boost application and alert threshold

Before thresholding,  $P_{\text{insider}}(t)$  is multiplied by two trade-level boost factors from the second analytical layer and the result is capped at  $S_{\text{max}} = 0.95$ :

$$\tilde{P}(t) = \min(S_{\text{max}}, P_{\text{insider}}(t) \cdot B_{\text{cluster}}(t) \cdot B_{\text{JA}}(t)). \quad (3.11)$$

The two multipliers and the construction of the amplification layer are described in Section 3.7; both default to 1 when not applicable (wallet not in any cluster, or insufficient pre-jump history). Trade  $t$  is flagged when the boosted score meets or exceeds the alert threshold  $\alpha$ :

$$\text{flagged}(t) = \mathbb{1}[\tilde{P}(t) \geq \alpha]. \quad (3.12)$$

### 3.6.3 Flag rate and wallet prediction

A single flagged trade however, is weak evidence. A wallet that generates one borderline alert among hundreds of normal trades carries less suspicion than one whose trades are consistently flagged. The system therefore computes a *flag rate*:

$$\text{FR}(w, m) = \frac{|\{t \in T_{w,m} : \text{flagged}(t)\}|}{|T_{w,m}|}, \quad (3.13)$$

where  $T_{w,m}$  is the set of wallet  $w$ 's trades in market  $m$  that passed the notional pre-filter. A wallet is classified as suspicious if its flag rate exceeds a threshold  $\theta$ :

$$\hat{y}(w, m) = \mathbb{1}[\text{FR}(w, m) \geq \theta]. \quad (3.14)$$

There is no further wallet-level adjustment. Because the cluster and jump-anticipation multipliers are applied to  $P_{\text{insider}}(t)$  *before* thresholding, their effect is already embedded in which trades entered the numerator of Equation 3.13. A wallet whose trades never crossed the alert threshold, even after boosting, cannot receive a positive prediction. This preserves a clear separation between primary evidence (trade-level detector signals) and contextual amplification (multipliers that can push a borderline score over the threshold, but cannot manufacture a signal from nothing).

## 3.7 Temporal Co-Activity Clustering

Insider trading rings, groups of wallets controlled by the same individual, or coordinating on shared information, are typically invisible to trade-level detectors that analyse each wallet independently. A coordinating entity that splits a \$100,000 position across ten proxy wallets at \$10,000 each may produce no volume anomaly on any individual account, yet the coordinated timing of the ten positions is strong evidence of common control or common information. The clustering component addresses this by constructing a temporal co-activity graph over wallets and applying community detection, specifically the Louvain method [4], a standard modularity-maximising algorithm detailed in Section 3.7.2, to identify groups of wallets that trade in suspiciously coordinated patterns. The output is a per-wallet cluster multiplier  $B_{\text{cluster}}^{(w)}$  that takes the value 1 for wallets outside any cluster and a value greater than 1 for wallets in a cluster, scaled by the cluster's size, density, and on-chain ownership evidence (Section 3.10).

### 3.7.1 Graph Construction: Bucket Projection

The co-activity graph is constructed using a bipartite projection through discrete time buckets, a standard technique in network science for building co-occurrence networks [29, 21], and replaces an earlier pairwise incremental approach that was  $O(T \cdot W \cdot E)$  in the number of trades, active wallets, and existing edges, reducing the cost of running

significantly. Each trade is assigned to a directional bucket keyed by

$$\mathbf{b}(t) = \left( m, \lfloor \tau_t/B \rfloor, o_t, d_t \right), \quad (3.15)$$

where  $m$  is the market identifier,  $\tau_t$  is the trade timestamp in seconds,  $B$  is the bucket size in seconds,  $o_t$  is the outcome index, and  $d_t \in \{\text{BUY}, \text{SELL}\}$  is the trade direction. The bucket key encodes both the outcome and the direction, so that only wallets trading the same outcome in the same direction within the same time window can share a bucket.

For each bucket containing two or more distinct wallets, all pairs within the bucket contribute an edge. If a wallet contributes multiple trades within a single bucket, its trade sizes are summed before edge-weight computation. A single bucket co-occurrence contributes

$$w_{\text{bucket}}(i, j) = \min(\sqrt{V_i \cdot V_j}/S, M), \quad (3.16)$$

where  $V_i, V_j$  are the aggregated trade sizes in that bucket,  $S$  is a size normaliser, and  $M$  is a maximum size multiplier cap. The geometric mean rewards co-occurring trades of similar magnitude, a \$50,000 pair produces a stronger signal than a \$50,000 / \$100 pair, while  $M$  prevents a single pair of extremely large trades from dominating the graph. This models splitting a large trade into roughly equally sized smaller trades. Weights accumulate across buckets:

$$w(i, j) = \sum_{\mathbf{b}: i, j \in \mathbf{b}} w_{\text{bucket}}(i, j), \quad (3.17)$$

so that wallets co-occurring in many time windows accumulate stronger edges, capturing persistent coordination rather than one-off coincidences.

Only trades above a clustering-specific minimum notional (default \$5,000, substantially higher than the detector pre-filter) are used for graph construction. The rationale is that coordination patterns of interest involve large capital commitments; including small retail trades would create dense but uninformative edges between wallets that happen to trade the same market around the same time. The threshold also materially reduces graph size. Each trade is assigned to exactly one bucket in  $O(1)$ , and the projection step is  $O(\sum_b \binom{n_b}{2})$  where  $n_b$  is the number of distinct wallets in bucket  $b$ ; in practice most buckets contain one to three wallets, so the total cost is near-linear in the number of trades.

**Robustness.** A surveillance system should not collapse once its parameters are known; relying on secret thresholds would be security by obscurity. Because the detectors are built on the *economic* structure of informed trading rather than incidental artefacts, evading any one of them is not free, as each evasive move degrades the edge the insider is trying to capture or exposes them elsewhere. Spacing a coordinated entry out to avoid a shared co-activity bucket sacrifices the synchronisation that made it profitable, since later fills execute at prices the earlier ones have already moved; and slicing a position under the

volume threshold either stretches the window over which the market can move against it, or spreads the same capital across more wallets, which lengthens the on-chain funding trail the attribution layer (Section 3.10) follows. The intent is therefore not undetectability but *cost*: the conjunctive, multi-signal structure (Section 3.7.4) forces an adversary to defeat behavioural, temporal, and on-chain signals at once.

### 3.7.2 Pruning and Community Detection

Before community detection the raw graph is pruned in two steps. Edges with weight below a minimum threshold  $w_{\min}^*$  are removed, eliminating weak or incidental co-occurrences; a  $k$ -core decomposition then retains only wallets embedded in a denser pattern of mutual co-activity, removing isolated and leaf nodes. A pair of coordinated wallets whose edge weight is just below the threshold at a given point in time will not form a cluster immediately, but the live clustering manager recomputes clusters periodically as new trades arrive (Section 3.12), so weights that eventually accumulate past the threshold cross it on a subsequent run.

The Louvain method [4] is then applied to the pruned  $k$ -core to partition wallets into communities. Louvain iteratively reassigns nodes to communities to maximise the modularity

$$Q = \frac{1}{2W} \sum_{ij} \left[ w_{ij} - \frac{k_i k_j}{2W} \right] \delta(c_i, c_j), \quad (3.18)$$

where  $W = \sum_{ij} w_{ij}$ ,  $k_i = \sum_j w_{ij}$  is the weighted degree of node  $i$ ,  $c_i$  is node  $i$ 's community assignment, and  $\delta$  is the Kronecker delta. Louvain was chosen because it handles weighted graphs natively, scales well to graphs with thousands of nodes, and does not require the number of communities to be specified in advance [4]. This last property is important, since the number of coordination groups in a market is unknown a priori and varies substantially across markets. The algorithm is however seeded with a fixed random state for reproducibility.

Each detected community becomes a *cluster*. For each cluster  $C$ , three scalar summaries are computed from the induced subgraph: the number of member wallets  $|C|$ , the edge density  $\rho_C = 2|E_C|/(|C| \cdot (|C| - 1))$ , and the total intra-cluster edge weight. These summaries drive the cluster-multiplier formula in the next subsection.

### 3.7.3 Cluster Multiplier

A wallet's cluster multiplier answers a single question: *how much more suspicious should this wallet's trades be, given the company it keeps?* It starts from a neutral value of 1 (no amplification) and adds three independent, bounded contributions, each capturing a distinct reason a cluster is suspicious:

- **Size**: larger coordinated groups are less likely to arise by chance. The contribution

grows with the number of member wallets  $|C|$  but is capped, since beyond a handful of wallets additional members add little extra evidence of coordination.

- **Density:** a cluster whose members nearly all co-trade with one another (high edge density  $\rho_C$ ) is stronger evidence than a loose, stringy group of the same size.
- **Ownership:** if on-chain attribution (Section 3.10) finds a direct USDC transfer between cluster members, that is qualitatively harder evidence than behavioural co-activity alone, and earns a fixed bonus.

Concretely, for a wallet  $w$  in cluster  $C$ ,

$$B_{\text{cluster}}^{(w)} = \min\left(B_{\text{max}}, \underbrace{1}_{\text{neutral}} + \underbrace{\min(|C|/N_s, \lambda_s)}_{\text{size, capped}} + \underbrace{\rho_C \cdot \lambda_d}_{\text{density}} + \underbrace{\mathbb{1}[\text{ownership}] \cdot \lambda_o}_{\text{ownership}}\right), \quad (3.19)$$

and wallets in no cluster take  $B_{\text{cluster}}^{(w)} = 1$ . The overall cap  $B_{\text{max}}$  (default 2.0) bounds the total amplification so that a single trade’s boosted score cannot be dominated by cluster membership alone.

The additive-with-cap form is deliberate, keeping each contribution independently interpretable, which a multiplicative or learned-nonlinear combination would obscure. Bounding every term, and the sum, guarantees the multiplier stays in  $[1, B_{\text{max}}]$  regardless of cluster pathologies: a single giant or fully-dense cluster cannot send the score to infinity. Each of the weights are parameters that are tuned by the multi-start coordinate descent optimisation loop.

### 3.7.4 Amplification, Not Independent Detection

An alternative design would let cluster membership flag wallets on its own, regardless of trade-level signals. The present system rejects this for two reasons. First, co-activity has many innocent explanations, traders reacting to the same news, bots running similar strategies, market makers quoting simultaneously, so without trade-level anomalies to separate informed coordination from coincidence, membership alone would produce too many false positives. Second, amplification establishes a hierarchy of evidence: a wallet must trade anomalously *and* sit within a coordinated network to be flagged strongly, a conjunction that suppresses false positives more than either condition alone, at the cost of missing rings whose members stay below the trade-level detectors.

## 3.8 Outcome Awareness

The binary structure of a prediction market admits trading with no close analogue in equity markets: a single wallet can hold both YES and NO at once, often rationally (a market maker quoting both sides, an arbitrageur hedging another venue). Untreated,

such two-sided activity reads as directional accumulation at the trade level and as spurious coordination at the clustering level, and both failures concentrate in the market categories where precision matters most. The system therefore has made outcome and direction first-class at every stage where wallets are compared: outcome concentration  $c_{\text{out}}$  (Eq. 3.2) and per-outcome directional tracking  $r_{\text{out}}$  (Eq. 3.3) keep a hedger from reading as an accumulator, while the co-activity graph keys its buckets on outcome and direction (Eq. 3.15), so cross-outcome or cross-direction wallets can never share an edge, a hard constraint rather than a tunable penalty. Enforcing outcome, and direction, awareness as a hard structural constraint wherever wallets are compared is a design contribution of this work.

## 3.9 Jump Anticipation

The cluster multiplier captures *whom* a wallet trades alongside; the jump-anticipation multiplier captures *when* a wallet trades relative to large price moves. The two are complementary, a lone wallet with no partners can still anticipate jumps, and a synchronised cluster need not trade before them. The component operationalises a simple null model: a random directional trade has a 50% chance of aligning with any subsequent large move. Wallets whose alignment rate exceeds this null *and* whose activity concentrates in pre-jump windows receive a multiplier  $B_{\text{JA}}^{(w)} > 1$ ; all others receive 1.

### 3.9.1 Jump Detection

For each outcome, the chronological price series is swept by a two-pointer window: from each trade  $i$ , a second pointer advances to the last trade within  $T_{\text{jump}}$  seconds, and a jump is recorded when the price move over that span meets a threshold,

$$\mathcal{J} = \left\{ (\tau_i, \tau_j, p_j - p_i, o) : \tau_j - \tau_i \leq T_{\text{jump}}, |p_j - p_i| \geq \Delta p_{\text{jump}}^* \right\}, \quad (3.20)$$

with the sign of  $p_j - p_i$  giving the direction. The sweep produces overlapping events, one price move qualifies from many starting trades, but the scoring stage counts each trade at most once, so overlap inflates neither per-wallet counts nor the scoring.

### 3.9.2 Wallet-Level Alignment Scoring

For each wallet  $w$ , two counters are computed over its trades:  $N_{\text{pre}}(w)$ , the trades falling within  $T_{\text{lookback}}$  seconds *before* a jump on the same outcome; and  $N_{\text{aligned}}(w)$ , those pre-jump trades whose direction matched the jump (BUY before an up-jump, SELL before a down-jump). A trade preceding several jumps is assessed against the largest; one preceding none is not counted. Jump lookups use binary search on the per-outcome

index, making scoring  $O(N_{\text{trades}} \log |\mathcal{J}|)$ . Two rates follow,

$$\text{align}(w) = \frac{N_{\text{aligned}}(w)}{N_{\text{pre}}(w)}, \quad \text{conc}(w) = \frac{N_{\text{pre}}(w)}{N_{\text{total}}(w)}, \quad (3.21)$$

where alignment has null value 0.5 and concentration measures the fraction of the wallet’s activity in pre-jump windows. Wallets with fewer than  $N_{\text{pre}}^*$  pre-jump trades are left at  $B_{\text{JA}}^{(w)} = 1$  to avoid noisy single-observation multipliers.

### 3.9.3 Jump-Anticipation Multiplier

Alignment excess rescales the rate so the null maps to zero and perfect alignment to one, with below-null performance clamped:

$$\text{excess}(w) = \max(0, 2(\text{align}(w) - 0.5)). \quad (3.22)$$

The raw score is then the product  $\text{raw}(w) = \text{excess}(w) \cdot \text{conc}(w)$ , so a meaningful score requires the wallet to trade both in the right direction *and* at the right times. The multiplier interpolates linearly from 1 to a configured maximum,

$$B_{\text{JA}}^{(w)} = 1 + (B_{\text{max}}^{\text{JA}} - 1) \cdot \text{raw}(w), \quad (3.23)$$

and is read into  $B_{\text{JA}}(t)$  (Equation 3.11) from the most recent scoring pass available at event time.

## 3.10 On-Chain Attribution

The behavioural detectors and co-activity graph operate on trading patterns; they produce no evidence about whether two wallets are physically related. The attribution layer adds a third, structurally independent evidence type: direct on-chain transfers of USDC between wallets in a detected cluster. If two wallets have both traded in coordinated patterns *and* transferred funds between themselves on the Polygon blockchain, the joint probability of this occurring under an independence hypothesis is very low. The architecture is cache-first: USDC transfer histories are fetched from the Polygonscan API once per wallet and stored in a local SQLite database, so the live system, the backtester, and the optimiser all draw from the same accumulated cache without re-querying the API.

**USDC transfer tracing.** For each wallet in a detected cluster, the system queries the Polygon USDC contract (address `0x2791Bca1f2de4661ED88A30C99A7a9449Aa84174`) for all historical ERC-20 transfer events involving that address. Transfer events are cached keyed on transaction hash, sender, and recipient, with indexes on both address fields and timestamp. Given a cluster, the forensic pass queries for *intra-cluster* transfers, transfers where both sender and recipient are cluster members, and aggregates them into

a directed transfer graph over the cluster’s wallets. The transfer graph and the co-activity graph share no edges by construction: the co-activity graph records when wallets traded similarly, while the transfer graph records whether they moved capital between themselves. Their intersection is the forensic signal.

**Common ownership classification.** A cluster is flagged as exhibiting common ownership if it contains at least one intra-cluster transfer. The threshold is deliberately low: the prior probability of any two wallets in a co-activity cluster having a direct USDC transfer between them is already very small, so the co-activity edge provides the first filter and the transfer edge provides a second independent one. When common ownership is confirmed, the ownership bonus  $\lambda_o$  is added to the cluster multiplier (Equation 3.19) for every wallet in the cluster.

**Lookahead avoidance in backtesting.** In the backtesting path, all transfer queries include a timestamp parameter restricting results to transfers that occurred *before* the earliest trade in the market under analysis. Transfers that occurred after the market resolved would use future information to label present activity; the filter keeps the backtest causally consistent with the live system.

The Polygonscan API is rate-limited at five requests per second on a free key. The `PolygonscanClient` implements a sliding-window rate limiter and exponential-backoff retry logic. On cache miss, the optimiser is restricted to a single worker to avoid simultaneous queries to the same wallet and once a market’s wallets are cached, full parallelism is restored.

## 3.11 Engineering Infrastructure

The detection pipeline and backtesting/optimisation infrastructure is implemented in Python, released open source at <https://github.com/cam-insiders/polymarket-surveillance>. This section describes the engineering decisions that make the system fast enough to optimise meaningfully and correct enough to trust.

### 3.11.1 Caching

Coordinate descent sweeps one block at a time with all others held fixed, so the non-target detectors produce identical signals on every candidate, recomputing them is pure waste. The `FrozenSignalCache` exploits this. Before each sweep, a single pass with the current best configuration records, for every detector, which trades it fired on and with what `DetectionSignal`. This mapping is sparse, since most detectors fire on only a small fraction of trades. During the sweep, `CachedBacktestRunner` serves the frozen detectors from it in  $O(1)$  per trade and calls `analyze()` fresh only for the target detector.

The alert-threshold sweep is a sharper special case: the Noisy-OR combined score is threshold-independent, so the cache holds a dense array of per-trade combined scores and each threshold candidate is evaluated by re-thresholding that array with zero detector computation. Together these give an order-of-magnitude reduction in wall-clock time per run, making multi-start coordinate descent with restarts feasible even on a laptop.

### 3.11.2 Causal boost replay

The clustering and jump-anticipation boosts cannot be frozen like detector signals, because a wallet’s boost is not a fixed property of the wallet: it depends on the entire history seen *so far*. The tempting shortcut, clustering once over the completed market and stamping every trade with its wallet’s final boost, is non-causal: an early trade would carry a boost reflecting a cluster that only formed from later trades, leaking future information the live system could never have at decision time and overstating the boost’s contribution.

`build_live_parity_boost_schedule()` avoids this by reproducing the live system rather than short-cutting it. It walks the trade stream in strict time order, holding exactly the state the live `ClusteringManager` would hold, and re-runs clustering only when the live triggers would fire (enough new eligible trades, a minimum inter-cluster interval, or a maximum interval reached). Driving those triggers from event timestamps rather than wall-clock time makes the replay complete in seconds while producing an identical sequence of updates. At each trade  $t_i$  it records the multipliers *in force at that moment*, so causality is structural: entry  $i$  is written by a forward walk that has seen only trades before  $t_i$ . Applying it during the backtest is then a single  $O(1)$  lookup per trade, keeping boosted evaluation as cheap as non-boosted.

A second saving follows from *what* the schedule depends on: only the trade stream and the second-layer parameters, not the detector parameters. Because optimisation alternates between coordinate-descent stages (Section 4.5), the schedule is reused unchanged across every detector sweep.

## 3.12 Live Deployment

The system runs continuously on a virtual machine, containerised via Docker with four SQLite databases mounted as volumes for persistence across container rebuilds. The *alert database* records each flagged trade with its detector attributions, fused score, and full trade metadata. The *clustering database* persists accumulated trade entries per market, allowing the co-activity graph to be reconstructed at restart without re-fetching historical data from the API. The *wallet cache* stores per-wallet platform metadata queried on first encounter. And finally, the *USDC transfer cache* stores per-wallet transfer histories from Polygonscan.

Markets can be added or removed at runtime through a lightweight HTTP control API, so an operator can begin monitoring a market created in response to a breaking-news event without restarting the container. The clustering and jump-anticipation managers run as periodic background jobs on the adaptive schedule of Section 3.11.2 (clustering triggered by accumulated trade count or elapsed time, whichever fires first; jump-anticipation rescored on a fixed interval), neither blocking the per-trade polling loop.

# Chapter 4

## Evaluation Methodology

Insider trading on a regulated exchange can be checked against enforcement records identifying who traded on material non-public information. No such record exists for Polymarket and none can: it is pseudonymous, permissionless, and beyond any regulator’s reach, so there is no authoritative insider list against which a true detection rate could be measured. The evaluation instead triangulates across independent lines of evidence: recovery of a return-based statistical proxy, the economics of the flagged set, stability across consecutive out-of-sample windows, and recovery of wallets independently reported as suspicious in the public record. None is decisive alone; together they argue the system detects informationally advantaged trading, not noise.

### 4.1 Ground Truth

#### 4.1.1 Return-based wallet labelling

In the absence of enforcement data, the evaluation adopts the convention standard in empirical finance since Meulbroek [27]: persistent abnormal profitability is the strongest available indicator of an information advantage. Empirical studies of informed trading on Polymarket, including Mitts and Ofir [28], often use a return-based proxy for the same reason. This project does likewise, and Section 4.2 addresses directly the circularity the choice entails.

For each resolved market  $m$  and each wallet  $w$  active in it, the system computes a realised profit-and-loss figure from first principles. Let  $o_m^*$  be the winning outcome and  $S_{w,m,o^*}$  the number of winning-outcome tokens the wallet holds at resolution. Each such token redeems for exactly \$1.00 under the CTF invariant, so the wallet’s terminal payout is  $S_{w,m,o^*}$  USDC. Let  $C_{w,m}$  be the wallet’s net cost basis, the USDC spent on buys minus the USDC returned by sells, accumulated across all outcomes traded. Realised PnL is payout minus cost basis:

$$\text{PnL}(w, m) = S_{w,m,o^*} - C_{w,m}. \quad (4.1)$$

Return scales PnL by the gross capital the wallet committed to buy trades,  $V_{w,m}^{\text{buy}}$ :

$$R(w, m) = \frac{\text{PnL}(w, m)}{V_{w,m}^{\text{buy}}}. \quad (4.2)$$

Return rather than absolute PnL is the basis for labelling because it isolates information advantage from capital scale: a wallet earning a 40% return on \$5,000 and one earning 40% on \$500,000 reflect the same per-dollar edge, and only the per-dollar edge is evidence of an information advantage rather than simply a large bankroll.

To identify wallets whose returns are anomalous, each wallet’s return is converted to a *z-score*: the number of standard deviations by which its return exceeds the mean return of all eligible wallets in the same market,

$$z(w, m) = \frac{R(w, m) - \mu_m}{\sigma_m}, \quad (4.3)$$

where  $\mu_m$  and  $\sigma_m$  are the mean and standard deviation of returns across all eligible wallets in market  $m$ . Eligibility requires a minimum gross notional of \$500, excluding wallets whose tiny positions produce high-variance returns by chance. A wallet with  $z(w, m) > 2.0$ , roughly the top two to three per cent of the within-market return distribution, is labelled a *statistical insider*. The z-score is used purely as an order statistic to partition the distribution; no normality assumption is made. The threshold of 2.0 is chosen to retain enough positive labels for classification metrics to be meaningful while remaining strict enough that ordinary trading variation does not routinely cross it.

### 4.1.2 Full-history labels

A wallet’s realised PnL, return, and z-score are computed from its *complete* trade history in the market, up to resolution, not from a truncated subset. This matters because the detector’s input is later restricted to a narrow time window (Section 4.7). If labels were computed only from in-window trades, a wallet that opened a large position before the window would appear to start from zero inventory, corrupting its cost basis, its realised PnL, its return, and ultimately its label.

### 4.1.3 Resolution inference

Realised PnL requires a known winning outcome. The UMA Optimistic Oracle that formally resolves Polymarket markets often lags market close by several days, and some markets are disputed or cancelled and never resolve formally; restricting evaluation to oracle-confirmed markets would discard a large fraction of the dataset. The system therefore supplements oracle resolutions with a price-based inference: if the final trade price of some outcome  $o$  exceeds  $\tau_{\text{resolve}} = 0.99$ , the CTF invariant that outcome prices sum to one dollar implies the market has effectively resolved to  $o$ . The threshold is deliberately

conservative; prices in the 0.90–0.98 range can reflect genuine residual uncertainty in a market’s final hours and are not treated as resolved.

## 4.2 Circularity and Proxy-Label Limitations

Return-based labelling is circular: the system is optimised to find wallets with abnormal returns, and a statistical insider is *defined* as a wallet with abnormal returns, so a wallet-level classification score partly measures whether the system recovers the quantity that defines its own ground truth. This is not a flaw in the present design but an irreducible feature of evaluating informed trading where no enforcement-labelled data exists; it is shared by every empirical study of the platform.

A distinct limitation compounds it: even setting circularity aside, an abnormal return need not reflect information at all. The skill-versus-luck literature in empirical finance shows that part of any high-return cohort is the right tail of chance rather than genuine edge [9, 3], a point a recent sign-randomization test makes for Polymarket directly [11]. The circularity is therefore acknowledged, not solved. Wallet-level classification is read throughout as *proxy-label recovery*, evidence that the system reproduces a defensible statistical definition of informed trading, never as proof of legally defined insider trading. The dissertation does not rest on it: it rests on combining proxy-label recovery with three lines of evidence that do not share the same circularity (Section 4.8): external case-study validation, a comparison against prior methods judged on worklist precision rather than raw recall, and trade-level economic diagnostics judged against realised market resolution rather than the z-score labels.

**What the z-score proxy misses.** The z-score label captures unusually high *percentage* returns and is blind to a distinct, plausible signature of informed trading: a relatively modest percentage return earned on very large notional size. A wallet that deploys a large sum and earns a few per cent with near-certainty may be highly suspicious and highly profitable in absolute terms yet never cross the threshold. This is one reason the evaluation reports economic metrics alongside classification, and the regime is revisited directly in the clustering analysis (Chapter 5).

## 4.3 Metric Families

Three metric families are reported, in increasing order of independence from the z-score labels and so decreasing exposure to the circularity of Section 4.2.

*Wallet-level classification* scores the flag-rate prediction  $\hat{y}(w, m)$  (Section 3.6.3) against

the z-score labels by precision, recall, and

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}, \quad (4.4)$$

with  $F_{0.5}$  the headline figure (Section 4.4). This family is the most exposed to circularity.

*Flagged-wallet economics* reports mean and median realised return and net PnL for each cohort (all flagged, true positives, false positives), confirming the surfaced wallets are profitable in absolute terms; the true-positive cohort’s strength however is partly mechanical as the labels are return-derived, and hence is not read as fully independent confirmation.

*Trade-level economics* takes no input from the z-score labels, only the per-trade flag and the market’s resolution, making it the most circularity-free diagnostic. For each BUY trade in a resolved market the resolution return is

$$r(t) = \begin{cases} (1 - p_t)/p_t & \text{if } o_t = o_m^*, \\ -1 & \text{otherwise,} \end{cases} \quad (4.5)$$

so a winning BUY at  $p = 0.20$  returns +4.0. Flagged and unflagged BUYs are compared by their mean difference in resolution return, and by a copytrade simulation, one notional-matched (exact copy) and one fixed-size at \$100 per trade, the latter giving equal weight per alert and so a more honest test of whether the detector consistently picks winners.

## 4.4 Optimisation Objectives

The parameter space is organised into *blocks*, coherent groups of parameters tuned together. There are eight: one per detector (each holding that detector’s full parameter set, e.g. the Volume Anomaly Detector’s z-score threshold, lookback window, minimum baseline count, minimum notional, and confidence cap), the alert threshold, and the clustering and jump-anticipation blocks, whose parameters, edge weights, bucket size, boost caps, alignment thresholds, are optimised on the same footing as the detectors rather than fixed by hand.

Two objectives are used, producing two distinct configurations with two distinct purposes. The *wallet-oriented configuration* is optimised for wallet-level  $F_{0.5}$  and is the system configuration used for all surveillance evaluation. The *trade-oriented configuration* is optimised for a notional-weighted return objective defined below and is used to test whether the architecture can isolate economically profitable trades.

### 4.4.1 Wallet-oriented objective: $F_{0.5}$

The wallet-oriented objective is  $F_{0.5}$  against the z-score labels (Equation 4.4,  $\beta = 0.5$ ). This is a surveillance and triage tool: flagged wallets are a worklist for downstream

human investigation, and investigative attention is a scarce resource. A configuration that maximises recall by flagging a large share of all wallets is operationally useless, however many true insiders it captures, because the resulting worklist is too large to act on.  $F_{0.5}$  weights precision twice as heavily as recall in the harmonic mean, formalising the preference for a compact, high-quality worklist while still requiring nontrivial coverage.  $F_1$ , which weights the two equally, was considered but rejected: empirically it pushes the optimiser toward larger flagged populations that require further post-hoc filtering before they are usable.

#### 4.4.2 Trade-oriented objective: lower-confidence-bound weighted return

The trade-oriented objective must reward configurations whose flagged BUY trades are genuinely profitable per dollar deployed, while resisting the failure mode in which apparent profitability rests on a handful of extreme trades that may not recur month-to-month.

The chosen objective is therefore the one-sided 95% lower confidence bound on the *notional-weighted* mean return of flagged BUY trades (Equation 4.5). Its point estimate rewards genuine per-dollar profitability rather than mere statistical separation, and does not depend on the volatile unflagged comparison group, unlike, for example, Cohen’s  $d$  [7]. The lower-bound penalty then discounts that estimate by its statistical uncertainty, so a configuration whose edge rests on too few trades, or is concentrated in too few large ones, is punished. Because the trades are weighted by notional rather than counted equally, the sample size entering this discount is not the raw trade count but Kish’s effective  $n$  [19], which falls toward one when a single large trade dominates, so an edge resting on one big bet is penalised most heavily of all. The full derivation is given in Appendix A.4.

### 4.5 Optimisation Procedure

Given an objective, the system searches the block-structured space of Section 4.4 for the configuration that maximises it. Exhaustive enumeration is infeasible, and even scoring one candidate is expensive, since it means replaying the whole training window through the detection pipeline. Two pieces of infrastructure described in Chapter 3 make this affordable, the `FrozenSignalCache` (Section 3.11) for detector sweeps and the causal boost replay (Section 3.11.2) for the second-layer blocks.

**Coordinate descent over blocks.** The search is coordinate descent over those blocks: one block is optimised at a time, its full parameter grid enumerated exhaustively while every other block is held at its current best, and the best-scoring candidate replaces that block’s values. The procedure cycles through the blocks until a full pass improves none of them or a given pass limit is reached. This turns an intractable joint search into a

sequence of tractable per-block sweeps, at the cost of possibly missing improvements that require two blocks to move together, a limitation the multi-start strategy below partially offsets.

**Multi-start and validation-based selection.** Coordinate descent converges to a local optimum that depends on its starting point, so the procedure runs three independent trajectories from random starting configurations. To choose between them without touching the held-out test window, the training window is split 80/20 by trade time: each trajectory descends on the first 80%, and the one scoring highest on the held-back 20% is selected. Model selection therefore stays strictly inside the training window as the test window of Section 4.6 plays no part in it. The two splits serve distinct roles: the validation slice arbitrates between trajectories during optimisation, while the test window gives the out-of-sample estimate of the selected configuration.

## 4.6 Temporal Validation Protocol

A single train/test split on one period would leave open the possibility that the reported scores are specific to that period. The protocol instead runs four consecutive monthly experiments, January, February, March, and April 2025. Within each month, the first two weeks form the training window and the following two weeks form the held-out test window. No test-window data influences any configuration evaluated against it.

This design is a *generalisation control*, in the precise sense of Section 4.2. Held-out testing demonstrates that the reported objective values are not artefacts of fitting a noisy objective to a particular training sample, and that the method’s performance persists across four consecutive, non-overlapping out-of-sample windows rather than being tuned to a single favourable period.

## 4.7 Strict Rolling-Window Replay with Full-History Labels

A trade enters a window’s detector input only if its market resolves inside the window *and* the trade itself executed inside it. The first condition makes labelled evaluation possible (only resolved markets yield returns), and the second prevents pre-window trades from leaking across the train/test boundary. The z-score labels, however, are still computed from each wallet’s complete history up to resolution (Section 4.1.2). This deliberately denies the detector the pre-window evidence its labels were built from, making the test strictly harder, but fits it to the recent market climate it will actually run in rather than to shifting historical dynamics. The information flow is strictly one-directional: in-window trades go in, resolution is applied only afterwards to score them, so no future outcome

reaches the detector while it is alerting.

## 4.8 External Validation and Comparison Protocols

The protocols below carry the weight wallet-level classification cannot, because each references evidence that does not originate from the z-score labelling pipeline.

**Case-study validation.** The strongest such evidence is a set of Polymarket markets in which informed trading was alleged or reported independently of any return analysis, with the suspicious wallets identified by journalists, on-chain investigators, or regulators rather than by this system. Ten such markets, listed in Table 4.1, form a curated corpus spanning four episodes: coordinated trading ahead of Israeli military action against Iran [13]; a single trader, later identified as a Google employee, who anticipated both the Gemini 3.0 release and the year’s most-searched person [37]; early positioning on the 2025 Nobel Peace Prize [43]; and three geopolitical markets surfaced by on-chain analysts [23, 5, 24, 48], one of which led to a CFTC enforcement action against the trader [38]. The evidence quality is deliberately heterogeneous, ranging from a regulatory prosecution and major-outlet reporting down to investigator posts on X; for a curated set we accept media-reported suspicion as the bar, since no stronger public record exists. Because the suspicion is externally motivated, recovering these wallets is genuine external validation: it tests whether the system’s alerts coincide with independently established suspicion, breaking the circularity of proxy-label recovery. The Results chapter reports which of these wallets each method recovered and how the system compares against the baselines on this set.

Table 4.1: Curated case-study markets and the independent sources reporting suspicion.

| Market (Polymarket slug)                | Episode       | Source   |
|-----------------------------------------|---------------|----------|
| israel-announces-end-of-military-...    | Israel–Iran   | [13]     |
| israel-military-action-against-iran-... | Israel–Iran   | [13]     |
| israel-strike-on-iran-on-june-24        | Israel–Iran   | [13]     |
| israel-military-action-against-iran-... | Israel–Iran   | [13]     |
| will-gemini-3pt0-be-released-by-...     | Gemini 3.0    | [37]     |
| will-d4vd-be-the-1-searched-person-...  | Google search | [37]     |
| will-mar(i)a-corina-machado-win-the-... | Nobel Peace   | [43]     |
| maduro-out-by-january-31-2026-318       | Maduro        | [23, 38] |
| us-strikes-iran-by-february-28-2026     | US–Iran       | [5]      |
| will-axiom-be-accused-of-insider-...    | Axiom         | [24, 48] |

**Comparison against prior methods on worklist precision.** Any method can raise recall by flagging more wallets, so prior methods are judged on the precision of the worklist they produce rather than on raw flag counts or recall alone: the operative question is the density of true suspicion in that worklist, reported alongside the flag count each method

incurs, and in particular whether the system recovers the externally identified case-study wallets that baselines miss.

**Ablations and clustering effectiveness.** Two further experiments, reported in Chapter 5, isolate where the result comes from. An ablation removes each detector and each second-layer component in turn and recomputes the held-out metrics, testing whether detection rests on the combination of signals rather than one dominant heuristic. A separate clustering analysis evaluates the layer’s ability to surface relational structure the per-trade detectors cannot see; it is also where the return-z-score proxy’s blind spot, coordinated and large-notional, low-percentage positions (Section 4.2), is examined most directly.

## 4.9 Baseline Methods

The surveillance configuration is compared against six baselines, two of them non-causal. A baseline is *causal* if it uses only information available at trade time and *non-causal* if it folds in post-resolution information; the two non-causal methods enjoy an unfair advantage and are included only to assess how much retrospective access can assist a method. Each baseline produces a per-wallet flag on the same in-window trades and is scored against the same z-score labels as the full system.

**Mitts–Ofir (causal and correct-side).** Following Mitts and Ofir [28] (Section 3.1), each wallet receives a score that standardises and sums their five signals, and the top decile is flagged. The method is run in two variants. The *correct-side* (retrospective) variant uses the full signal set, including the post-resolution profitability and directional-correctness signals, and is therefore non-causal. The *causal* variant drops every signal that requires knowing the outcome, scoring only the resolution-independent components, and is the variant a live system could actually run.

**Isolation Forest.** An unsupervised anomaly detector [22]. Each wallet is a feature vector of its in-window trade behaviour, and an ensemble of random trees repeatedly splits the wallet population on random features. An outlier is separated from the rest after few splits, giving a short average path length; this is its anomaly score, and wallets above a cutoff are flagged. It is causal: the features use in-window behaviour alone, with no resolution information.

**CONSOB-PCA.** An unsupervised anomaly detector in the spirit of the regulator-facing direction of Mazzarisi et al. [25]. Principal component analysis first learns the few dominant patterns that describe how most wallets trade, then tries to reconstruct each wallet from those patterns alone. A wallet that the typical patterns reconstruct poorly is trading unlike the bulk of the population, and is flagged as anomalous. It is treated as non-causal because its feature set includes post-resolution performance.

**Timing heuristic.** A deliberately naive causal baseline that flags wallets trading shortly before a large price move. It tests whether crude pre-event timing alone carries usable signal, and is a common signal shared on social media as indicative of insider trading.

**Random flagging.** A control that flags a random subset of wallets, establishing the precision a method achieves by chance at a given flag budget.

## 4.10 Implementation Correctness

Several components, incremental rolling statistics, the causal boost replay, the label/input separation, fail silently rather than obviously when wrong, so correctness is therefore established by unit and parity tests against scenarios with known correct output. Each detector is tested against hand-constructed inputs (for instance, the Accumulation detector is given textbook hedging, equal buy volume on both outcomes, and asserted to stay silent), and Noisy-OR fusion against a manually computed score. Two parity tests guard the causal claim directly: the boost replay (Section 3.11.2) is asserted to be a bit-for-bit no-op when clustering and jump anticipation are disabled and never to apply a boost retroactively, and an evaluator assertion prevents the in-window detector list from leaking into the full-history label computation, a contamination bug caught during development and now regression-guarded.

# Chapter 5

## Results

This chapter reports the empirical evaluation set out in Chapter 4, following the protocol fixed there: each month from January to April 2025 is an independent trial, trained on its first fortnight (itself split 80/20 by trade time for trajectory selection) and tested out-of-sample on its second. Trades are replayed strictly in time order (Section 4.7), so the detector never sees a market’s outcome while deciding whether to flag a trade in it.

The results are organised as follows. It first reports the two trained configurations, the wallet-oriented configuration that is focused on surveillance, and the trade-oriented configuration that demonstrates a separate and more economic property of the same architecture, across the four months. It then compares the wallet-oriented configuration against prior methods across all four months. Followed by evaluating the system on a curated set of Polymarket markets where insider trading was reported independently in the public record, which is where the practical case for the system is made most directly. Ablations and an evaluation of clustering complete the chapter.

### 5.1 Runtime Performance

The live-deployability claim of Chapter 3 requires the system to keep pace with live trade flow; the figures below are from the February out-of-sample window (39,036 trades, 871 markets), with the other months equivalent.

Aggregate throughput is roughly 133,000 trades per second, a conservative floor depressed by the many tiny markets where per-market setup dominates with the highest-volume markets sustaining 140,000 trades per second. Per-trade detection latency averages  $6.4\mu\text{s}$ , rising to  $7.3\mu\text{s}$  including alert construction and persistence. Against a live trade rate on the order of tens per second, this leaves throughput headroom of several orders of magnitude, so per-trade detection is never the live bottleneck.

## 5.2 Wallet-Level Surveillance Configuration

The wallet-oriented configuration is trained to maximise  $F_{0.5}$  against the return-based insider labels of Section 4.1.1. Table 5.1 reports its held-out classification performance in each month, and Table 5.2 the realised economics of its flagged set.

Table 5.1: Held-out classification of the surveillance ( $F_{0.5}$ ) configuration. “Insiders” is the number of ground-truth statistical insiders in the window.

| Month    | Wallets | Insiders | Prec   | Rec    | $F_1$  | $F_{0.5}$ |
|----------|---------|----------|--------|--------|--------|-----------|
| January  | 159,840 | 1,259    | 0.0457 | 0.0365 | 0.0406 | 0.0435    |
| February | 79,119  | 504      | 0.0839 | 0.0536 | 0.0654 | 0.0753    |
| March    | 35,598  | 535      | 0.0783 | 0.0673 | 0.0724 | 0.0758    |
| April    | 49,575  | 533      | 0.0369 | 0.0413 | 0.0390 | 0.0377    |

Table 5.2: Realised economics of the flagged wallet set, by month.

| Month    | Mean net PnL | Median net PnL | Mean return |
|----------|--------------|----------------|-------------|
| January  | −\$74        | +\$16          | +7.2%       |
| February | +\$48        | +\$82          | +37.3%      |
| March    | +\$745       | +\$258         | +62.0%      |
| April    | +\$1,037     | +\$363         | +51.0%      |

The data shows performance is consistent across four separately trained, non-overlapping windows:  $F_{0.5}$  hovers around 0.038–0.076 and precision 3.7–8.4%, which against a base rate of under 1% insiders, is roughly an order of magnitude over chance. The flagged set is also economically meaningful: the median flagged wallet is profitable in every month (+\$16 to +\$363), and mean flagged return runs from +7% to +62%. The only negative figure, January’s mean net PnL (−\$74), sits against a positive median, indicating a few large losing wallets rather than a broadly unprofitable set.

## 5.3 Trade-Level Configuration

The same architecture, trained instead on the lower-confidence-bound return objective of Section 4.4.2, targets individual profitable trades rather than suspicious wallets. Table 5.3 reports the held-out result.

Between 84% and 96% of the flagged trades resolved on the winning side in every month, and flagged trades beat unflagged trades by one to four percentage points of return throughout. Read on its own, the return figure looks modest, but the combination is economically significant: a win rate above 90% on a per-trade edge of around one per cent, sustained across hundreds to thousands of trades in a two-week window, compounds into a substantial return for a trader recycling capital efficiently across those trades.

The contrast with the surveillance configuration is one of operating point, not quality: one flags suspicious wallets, one flags reliably winning trades and which is useful depends

Table 5.3: Trade-oriented configuration, held-out across four months. Win rate is the fraction of flagged buy trades that resolved profitably; mean return diff is the gap between flagged and unflagged trade returns.

| Month    | Flagged trades | Win rate | Per Trade ROI | Mean return diff |
|----------|----------------|----------|---------------|------------------|
| January  | 8,071          | 94.0%    | +0.65%        | +0.0123          |
| February | 744            | 93.2%    | +1.65%        | +0.0212          |
| March    | 159            | 83.7%    | +4.21%        | +0.0345          |
| April    | 852            | 96.0%    | +1.52%        | +0.0414          |

on the downstream purpose. The remainder of the chapter concerns the surveillance configuration only.

## 5.4 Comparison Against Prior Methods

The surveillance configuration is compared across all four months against the six prior methods of Section 4.9, each scored on the same full set of resolved markets in each window. Tables 5.4–5.7 report the results; the **Causal** column marks the methods using only trade-time information, the retrospective ones folding in post-resolution data and so included only as an unfair upper bound.

Table 5.4: Method comparison, January 2025. Best causal value per column in bold.

| Method                    | Causal | Flagged | Prec          | Rec           | $F_{0.5}$     | Flagged avg ret | Mean net PnL |
|---------------------------|--------|---------|---------------|---------------|---------------|-----------------|--------------|
| <b>Full system</b>        | Yes    | 1,007   | <b>0.0457</b> | 0.0365        | <b>0.0435</b> | +7.2%           | −\$74        |
| Mitts–Ofir (causal)       | Yes    | 132     | 0.0000        | 0.0000        | 0.0000        | +0.1%           | +\$4,683     |
| Isolation Forest          | Yes    | 8,698   | 0.0123        | <b>0.0855</b> | 0.0148        | +3.8%           | +\$446       |
| Random flagging           | Yes    | 2,513   | 0.0060        | 0.0120        | 0.0066        | −1.6%           | +\$786       |
| Timing heuristic          | Yes    | 858     | 0.0023        | 0.0016        | 0.0021        | −2.7%           | +\$462       |
| Mitts–Ofir (correct-side) | No     | 40      | 0.1500        | 0.0048        | 0.0212        | +155.3%         | +\$32,310    |
| CONSOB-PCA                | No     | 578     | 0.0311        | 0.0144        | 0.0253        | +43.1%          | +\$2,043     |

Table 5.5: Method comparison, February 2025.

| Method                    | Causal | Flagged | Prec          | Rec           | $F_{0.5}$     | Flagged avg ret | Mean net PnL |
|---------------------------|--------|---------|---------------|---------------|---------------|-----------------|--------------|
| <b>Full system</b>        | Yes    | 322     | <b>0.0839</b> | 0.0536        | <b>0.0753</b> | +37.3%          | +\$48        |
| Mitts–Ofir (causal)       | Yes    | 125     | 0.0080        | 0.0020        | 0.0050        | −5.5%           | +\$23        |
| Isolation Forest          | Yes    | 3,936   | 0.0152        | <b>0.1190</b> | 0.0185        | +4.9%           | +\$287       |
| Random flagging           | Yes    | 1,259   | 0.0071        | 0.0179        | 0.0081        | −0.6%           | +\$516       |
| Timing heuristic          | Yes    | 624     | 0.0064        | 0.0079        | 0.0067        | +0.2%           | −\$116       |
| Mitts–Ofir (correct-side) | No     | 122     | 0.1721        | 0.0417        | 0.1058        | +168.8%         | +\$11,390    |
| CONSOB-PCA                | No     | 400     | 0.0200        | 0.0159        | 0.0190        | +35.3%          | +\$1,723     |

The full system attains the highest precision and  $F_{0.5}$  of any causal method in every month, occasionally beating the non-causal ones, while flagging five to twelve-fold fewer wallets than the high-recall methods. The two extremes frame it: Isolation Forest reaches the highest causal recall (7.5–11.9%) but buys it with 2,900–8,700 flags per month versus the full system’s 4–7% from only 322–1,007 flags, while causal Mitts–Ofir recovers under 1% of insiders at any budget.

Table 5.6: Method comparison, March 2025.

| Method                    | Causal | Flagged | Prec          | Rec           | $F_{0.5}$     | Flagged avg ret | Mean net PnL |
|---------------------------|--------|---------|---------------|---------------|---------------|-----------------|--------------|
| <b>Full system</b>        | Yes    | 460     | <b>0.0783</b> | 0.0673        | <b>0.0758</b> | +62.0%          | +\$745       |
| Mitts–Ofir (causal)       | Yes    | 103     | 0.0097        | 0.0019        | 0.0053        | −5.9%           | −\$1,055     |
| Isolation Forest          | Yes    | 3,033   | 0.0145        | <b>0.0822</b> | 0.0174        | +5.3%           | +\$441       |
| Random flagging           | Yes    | 651     | 0.0092        | 0.0112        | 0.0096        | −0.2%           | +\$348       |
| Timing heuristic          | Yes    | 599     | 0.0083        | 0.0093        | 0.0085        | +3.4%           | +\$944       |
| Mitts–Ofir (correct-side) | No     | 32      | 0.2188        | 0.0131        | 0.0528        | +247.8%         | +\$11,091    |
| CONSOB-PCA                | No     | 308     | 0.0292        | 0.0168        | 0.0255        | +43.3%          | +\$1,571     |

Table 5.7: Method comparison, April 2025.

| Method                    | Causal | Flagged | Prec          | Rec           | $F_{0.5}$     | Flagged avg ret | Mean net PnL |
|---------------------------|--------|---------|---------------|---------------|---------------|-----------------|--------------|
| <b>Full system</b>        | Yes    | 596     | <b>0.0369</b> | 0.0413        | <b>0.0377</b> | +51.0%          | +\$1,037     |
| Mitts–Ofir (causal)       | Yes    | 159     | 0.0189        | 0.0056        | 0.0129        | −6.6%           | +\$2,329     |
| Isolation Forest          | Yes    | 2,906   | 0.0138        | <b>0.0753</b> | 0.0165        | +7.8%           | +\$708       |
| Random flagging           | Yes    | 796     | 0.0088        | 0.0132        | 0.0094        | +0.7%           | +\$1,046     |
| Timing heuristic          | Yes    | 611     | 0.0033        | 0.0038        | 0.0034        | −4.9%           | −\$654       |
| Mitts–Ofir (correct-side) | No     | 59      | 0.0508        | 0.0056        | 0.0196        | +79.7%          | +\$22,107    |
| CONSOB-PCA                | No     | 451     | 0.0044        | 0.0038        | 0.0043        | +51.8%          | +\$2,303     |

For a triage tool, where recall is useful only until it is bought with proportionally more false positives, precision is the operative criterion and  $F_{0.5}$  weights it accordingly. CONSOB-PCA’s high flagged returns come from a tiny set at near-zero recall using post-resolution information and the timing heuristic performs at or below random on every metric, so crude pre-event timing alone carries no usable signal. The full system’s flagged set is also the most economically sound among causal methods, positive (+7% to +62%) in every month against causal Mitts–Ofir’s near-zero or negative figures.

## 5.5 Insider Trading Case Studies

The curated evaluation tests the system against suspicion established independently of the return-based labels. It uses ten Polymarket markets in which informed trading was reported by journalists, investigators, or platform observers, containing 22 reported insider entries across 18 distinct wallets. All methods were run through one harness using the April-trained configuration, and the wallet-level classification is scored against an identical 272-wallet positive set for every method. Figure 5.1 summarises both reported-insider recovery (left) and z-score label quality (right) across methods, and Table 5.8 gives the full classification figures against the shared labels, including recall and raw true and false-positive counts.

Isolation Forest achieves a higher reported-insider recovery than the full system (95.5% vs 72.7% ever-flagged), but at 3,448 false positives against the full system’s 383 on the shared label set. That reduction in recall is immediately justified by the difference in false positives and total alerts, and is reflected in the z-score precision (0.199 vs 0.065) and  $F_{0.5}$  (0.217 vs 0.080), where the proposed system leads every causal method by a wide

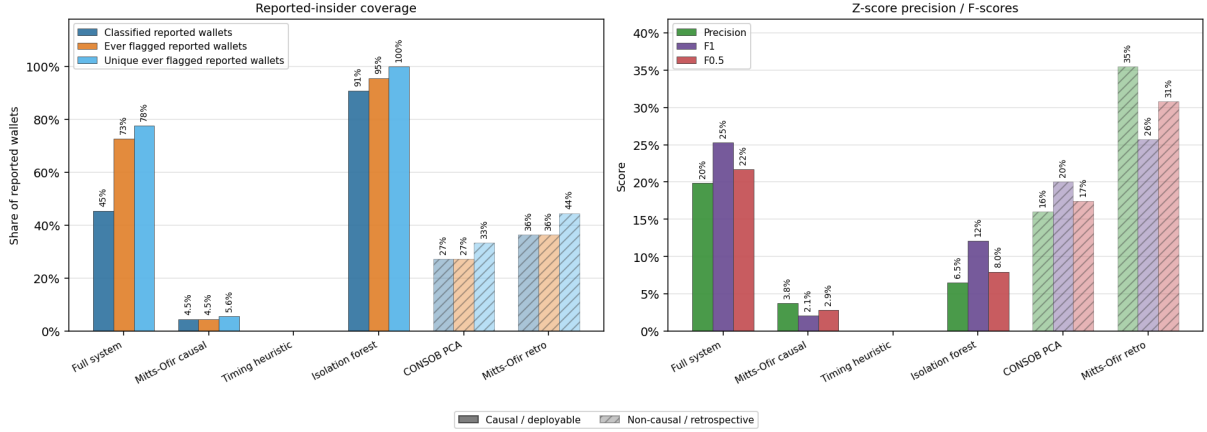


Figure 5.1: Curated evaluation across the six methods. **Left:** recovery of independently reported insiders - *classified* (wallet receives a final positive verdict) and *ever flagged* (entry flagged at any point, including a cluster boost), each as a share of the 22 reported entries, and *unique ever flagged* as a share of the 18 distinct reported wallets. **Right:** classification quality against the shared 272-wallet z-score positive set ( $F_{0.5}$ ,  $F_1$ , precision). Solid bars are causal/deployable methods; hatched bars are non-causal/retrospective. The timing heuristic flags none of the reported insiders.

Table 5.8: Classification on the ten curated markets against a shared 272-wallet z-score positive set. Best causal value per column in bold.

| Method                     | Causal | Prec         | Rec          | $F_1$        | $F_{0.5}$    | TP  | FP    |
|----------------------------|--------|--------------|--------------|--------------|--------------|-----|-------|
| <b>Full system</b>         | Yes    | <b>0.199</b> | 0.349        | <b>0.253</b> | <b>0.217</b> | 95  | 383   |
| Isolation Forest           | Yes    | 0.065        | <b>0.879</b> | 0.121        | 0.080        | 239 | 3,448 |
| Mitts–Ofir (causal)        | Yes    | 0.038        | 0.015        | 0.021        | 0.029        | 4   | 102   |
| Timing heuristic           | Yes    | 0.000        | 0.000        | 0.000        | 0.000        | 0   | 9     |
| Mitts–Ofir (retrospective) | No     | 0.355        | 0.202        | 0.258        | 0.308        | 55  | 100   |
| CONSOB-PCA                 | No     | 0.160        | 0.268        | 0.201        | 0.174        | 73  | 383   |

margin. Against the causal field the gap is decisive: causal Mitts–Ofir recovers 1 of 22 reported entries and posts  $F_1 = 0.021$ , against the full system’s 16 and  $F_1 = 0.253$ . The retrospective methods are the only ones competitive on the shared labels, retrospective Mitts–Ofir edges the full system on  $F_{0.5}$  (0.308 vs 0.217), but they use post-resolution information the deployed system cannot, and even so recover only 8 of 22 reported entries. The full system of this dissertation is therefore the only method combining high external recovery with leading precision among causal methods, therefore, the most actionable upon.

## 5.6 Detector Ablations

To establish that detection rests on the combination of detectors rather than a single component, each detector and stack component was removed in turn from the surveillance configuration and the held-out metrics recomputed, in all four months. Table 5.9 reports the change in  $F_1$  from removing each component, alongside its multi-detector co-firing share, the fraction of the trades it flags that are also flagged by at least one other detector, which is needed to read the table correctly.

Table 5.9: Component ablation.  $\Delta F_1$  is the change in held-out  $F_1$  from removing the component (negative means removal hurts). The full-system  $F_1$  for each month is given for reference. “Multi%” is the detector’s co-firing share, given as the range across the four months.

| Component                           | $\Delta F_1$ on removal |               |               |               | Multi%  |
|-------------------------------------|-------------------------|---------------|---------------|---------------|---------|
|                                     | Jan                     | Feb           | Mar           | Apr           |         |
| <i>Full-system <math>F_1</math></i> | <i>0.0406</i>           | <i>0.0654</i> | <i>0.0724</i> | <i>0.0390</i> |         |
| Extreme Position                    | −0.0406                 | −0.0654       | −0.0724       | −0.0390       | 75–99%  |
| Probability Impact                  | −0.0390                 | −0.0615       | +0.0024       | −0.0240       | 81–100% |
| Contra-Outcome Silence              | +0.0000                 | −0.0008       | −0.0724       | −0.0029       | 88–100% |
| Volume Anomaly                      | +0.0000                 | +0.0000       | +0.0000       | +0.0001       | 100%    |
| Accumulation                        | +0.0000                 | +0.0000       | +0.0000       | +0.0000       | 100%    |
| Clustering                          | +0.0000                 | +0.0002       | +0.0000       | +0.0000       | –       |
| Jump anticipation                   | +0.0002                 | +0.0002       | +0.0010       | +0.0003       | –       |

The table should be read on  $F_1$ , not just PnL: removing a key detector *raises* the flagged set’s mean PnL (dropping Extreme Position lifts April’s from \$1,037 to \$3,613) only because the flag count collapses with it, in January, removing Probability Impact cuts the flagged set from 1,007 wallets to 30. Precision over the residual holds; coverage dies, leaving PnL an average over a tiny, high-variance remainder. PnL is therefore informative only where  $F_1$  is preserved.

So read, the components fall into three roles.

**Extreme Position** is the load-bearing flagger: removing it zeroes  $F_1$  in all four months, because almost no wallet is flagged without it. This follows from its selected confidence cap, the highest of any detector, which lets a single qualifying trade cross the threshold alone. The dependence is by design but a genuine concentration risk, noted in the limitations.

**Probability Impact and Contra-Outcome Silence** form a secondary pillar whose carrier shifts by month: removing Probability Impact craters  $F_1$  in January, February, and April yet is neutral in March, where Contra-Outcome Silence instead becomes decisive. That the pillar moves across windows is the ablation evidence for the design premise of Chapter 2, that no single microstructure signal holds across the range of information lifetimes in the data, and the combination is what gives coverage across regimes.

Table 5.10: Per-detector economic profile of flagged activity, by month, with the four-month average. “Trade return” is the mean resolution return of the detector’s flagged buy trades; “wallet return” is the mean return of the wallets it flags. These figures describe the activity each detector responds to, independent of how often it is the decisive flagger.

| Detector                  | Jan    | Feb     | Mar    | Apr    | Avg    |
|---------------------------|--------|---------|--------|--------|--------|
| <i>Mean trade return</i>  |        |         |        |        |        |
| Volume Anomaly            | −6.0%  | +116.8% | −17.0% | +9.1%  | +25.7% |
| Probability Impact        | −6.1%  | +10.6%  | +2.9%  | −13.0% | −1.4%  |
| Accumulation              | +10.5% | +8.3%   | +2.9%  | +14.3% | +9.0%  |
| Extreme Position          | −14.8% | +15.8%  | −27.4% | −26.0% | −13.1% |
| Contra-Outcome Silence    | +3.3%  | +2.1%   | +1.1%  | −2.2%  | +1.1%  |
| <i>Mean wallet return</i> |        |         |        |        |        |
| Volume Anomaly            | +34.7% | +64.4%  | +28.2% | +54.6% | +45.5% |
| Probability Impact        | +16.0% | +23.7%  | +11.4% | +22.3% | +18.4% |
| Accumulation              | +9.7%  | +3.5%   | +2.8%  | +10.9% | +6.7%  |
| Extreme Position          | +16.8% | +25.4%  | +52.0% | +32.1% | +31.6% |
| Contra-Outcome Silence    | +14.3% | +27.9%  | +46.5% | +44.2% | +33.2% |

**Volume Anomaly and Accumulation** seem to have no impact on  $F_1$ , until their co-firing rate is read, both have a 100% co-firing share in every month, so neither originates a unique alert and removing either leaves  $F_1$  unchanged. They are therefore corroborators, not waste: Table 5.10 shows Accumulation is the only detector with positive mean flagged-*trade* return in every month (+9.0% average), and Volume Anomaly flags the highest-return *wallets* of any detector (+45.5% average).

Parameters are held fixed across all ablations rather than re-optimised after each removal, as with three random restarts already producing configurations that differ non-trivially in  $F_1$ , a re-optimised ablation could show a drop owing more to an unlucky restart than to the disabled component. This slightly overstates each component’s importance, to which a re-tuned system might partly compensate, in exchange for a lower-variance attribution.

The two stack components barely move  $F_1$ . Clustering’s contribution is evaluated separately in Section 5.7. Jump anticipation originates no alerts of its own, so its co-firing share is not comparable to a detector’s, but the wallets it boosts are economically distinctive, carrying among the highest mean PnL of any component in three of four months (\$8,220 at +14.3% in January, and +29.1% in April). This is the expected signature of what it targets, wallets that trade just ahead of large price moves, since acting on that kind of conviction tends to come in larger position.

## 5.7 Clustering Effectiveness

The clustering layer rarely changes the wallet-level verdict, and this is by design rather than a shortfall. Its trigger, synchronised, same-direction co-activity across wallets, is

a specific behaviour that should be uncommon, so the layer is expected to engage on a thin slice of activity: across the four months it was decisive for 0, 3, 0, and 0 wallets, and aggregate  $F_1$  is essentially unchanged in every window. Its value is therefore not a system-wide lift but the quality of the slice it does touch, which this section examines.

The clearest effect is at the trade level, and it is robust regardless of whether the boost changed any final verdict. Table 5.11 shows that trades receiving any cluster boost won 99.5%, 99.0%, 100%, and 100% of the time across the four months, against 84%, 77%, 75%, and 73% for trades the layer never touched, with a higher mean PnL in every month. The effect holds even in March and April, where clustering was decisive for no wallet at all: the boost still selected reliably winning trades.

Table 5.11: Trades receiving any cluster boost versus trades the clustering layer never touched. The win-rate and mean-PnL gaps hold in all four months, independent of whether the boost was decisive for any wallet.

| Month    | Cluster-boosted |          | Not touched |          |
|----------|-----------------|----------|-------------|----------|
|          | Win rate        | Mean PnL | Win rate    | Mean PnL |
| January  | 99.5%           | +\$183   | 83.9%       | +\$22    |
| February | 99.0%           | +\$87    | 77.4%       | −\$16    |
| March    | 100.0%          | +\$76    | 75.4%       | +\$4     |
| April    | 100.0%          | +\$49    | 72.9%       | +\$31    |

These boosted trades come with a qualification that ties back to the rest of the evaluation: they win with incredible reliability and carry higher dollar PnL than untouched trades, but their *percentage* returns are modest (roughly 1–2%) and none of the wallets behind them are z-score insiders. They are the same proxy-invisible profile the z-score labels miss (Section 4.2), a dependable edge earning small percentage returns, now visible at trade granularity rather than overlooked.

Finally, the on-chain attribution layer (Section 3.10) found direct common-ownership evidence in one window: in February, three flagged in-cluster wallets were linked by USDC transfers between cluster members, the strongest forensic corroboration the system can produce, and they carried a higher mean PnL than the typical flagged wallet. That this surfaced in only one of four months, on three wallets, is the expected result, it requires both synchronised co-activity *and* an observable money trail between the same wallets, all while the evaluation sees only markets resolving inside a two-week window.

# Chapter 6

## Summary and Conclusions

### 6.1 Summary

This dissertation defended two claims: that informed trading in a decentralised prediction market leaves a behavioural signal detectable *in real time*, from the public on-chain record alone; and that a system built on that signal can match the retrospective state of the art under the strictly harder causal constraint. It developed such a system for Polymarket, running causally at over 130,000 trades per second (Section 5.1) on a code path shared with its backtester so optimised parameters transfer directly to deployment.

Both claims hold. The first is borne out by the curated case studies: on ten markets where informed trading was reported independently of any return analysis, the system recovered 16 of 22 reported insider entries from trade-time information alone, against 1 for the strongest causal baseline. The second is borne out by the held-out comparison: across all four months it led every causal method on precision and  $F_{0.5}$  and stayed competitive with the retrospective baselines despite their post-resolution advantage, while flagging an order of magnitude fewer wallets. The only method recovering more reported insiders did so at roughly ten times the false-positive volume. The clustering and attribution layers rarely changed the verdict, but the trades they touched won with near-perfect reliability, and the one window where on-chain transfers linked cluster members surfaced exactly the coordinated, modest-return activity the return proxy overlooks. The result is a tool for an external investigator, journalist, or regulator with no privileged access: a compact, high-precision worklist drawn from the public ledger alone.

### 6.2 Limitations

Four limitations of the work warrant explicit acknowledgement.

**Return-based ground truth.** The labelling is circular: a statistical insider is *defined* by abnormal returns, the quantity the system is tuned to find (Section 4.2). This is irreducible without a large enforcement dataset, and the evaluation mitigates it by anchoring its central claims in the externally identified case-study wallets and every wallet-level figure should be read with the dependence in mind.

**Untested rolling recalibration.** The intended deployment model refits the system to each recent two-week window and runs it on the next. The evaluation is almost consistent with this, four independently trained windows, each tested out-of-sample on the following fortnight, but does not test the recalibration loop as a continuous process: each window is fitted in isolation, not as a live system re-optimising against its own accumulating history. Whether performance holds, drifts, or degrades under genuine continuous recalibration over a longer horizon remains open.

**Optimisation procedure.** Multi-start coordinate descent suits the block-structured space, but remains a local search with no guarantee of approaching the global optimum. The reported scores should therefore be read as a lower bound on what the architecture can achieve, not the performance of an optimally tuned system.

**Differentiating information.** The system separates economically anomalous trading from background activity, but cannot, from behavioural data alone, separate abnormal returns earned on private information from those earned on legitimate skill or luck; that distinction requires case-level investigation using evidence the system does not hold.

## 6.3 Future Work

Listed here are some potential avenues for future work on this matter.

**Order-book microstructure.** The detectors operate on settled fills, inferring liquidity dynamics only indirectly. Polymarket’s central limit order book exposes the underlying quote and cancellation stream directly, which would let the system observe the adverse-selection response Glosten and Milgrom [10] predict as it happens, rather than reading it from a gap in executed trades. The cost however is operational as order-book data is higher-volume and not retained in any public source.

**Cross-market behavioural linking.** Each market is analysed in isolation, but an insider is a person, not a market participant. Clustering markets by topic and propagating per-wallet suspicion across the cluster would catch insiders who spread a position across linked markets to stay below per-market thresholds.

**Market selection and prioritisation.** A realistic deployment cannot monitor every market equally, and a detectable edge generally requires a market whose resolution turns on information knowable early. An LLM classifier could therefore select monitoring targets up front rather than only stratifying results after the fact. One caveat must be considered. A trader with information on potential conflict escalations might deliberately avoid the obvious market and take a position in a nominally unrelated proxy, the 24-hour Gold price (considered a proxy for risk and uncertainty), to evade scrutiny, so any deployment-time restriction must prioritise attention and not hard-exclude markets.

Beyond these, several extensions follow naturally from the existing architecture: benchmarking global or population-based optimisers against coordinate descent to test whether they escape its local optima; richer on-chain evidence such as cross-platform transfer and bridge-flow analysis from Ethereum mainnet; and extension from binary to multi-outcome markets. The lasting contribution of this work is not any single number reported here, but the working system it leaves behind: a foundation others can build on to keep investigating a question that, on the public evidence, is far from settled, how much insider trading goes on in these markets, and how much can be caught.

# Bibliography

- [1] Kenneth R. Ahern: Do Proxies for Informed Trading Measure Informed Trading? Evidence from Illegal Insider Trades. *The Review of Asset Pricing Studies*, 10(3):397–440, 2020.
- [2] Kenneth J. Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O. Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D. Nelson, George R. Neumann, Marco Ottaviani, Thomas C. Schelling, Robert J. Shiller, Vernon L. Smith, Erik Snowberg, Cass R. Sunstein, Paul C. Tetlock, Philip E. Tetlock, Hal R. Varian, Justin Wolfers, and Eric Zitzewitz: The Promise of Prediction Markets. *Science*, 320(5878):877–878, 2008.
- [3] Laurent Barras, Olivier Scaillet, and Russ Wermers: False Discoveries in Mutual Fund Performance: Measuring Luck in Estimated Alphas. *The Journal of Finance*, 65(1):179–216, 2010.
- [4] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre: Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [5] Bubblemaps: On-chain analysis of the *US strikes Iran by February 28* market. Post on X, 2026. <https://x.com/bubblemaps/status/2027718004193300791>
- [6] Chainalysis: *The Data Accuracy Flywheel: How Chainalysis Detects and Prevents Errors*. Chainalysis Blog, 16 January 2024. <https://www.chainalysis.com/blog/chainalysis-data-accuracy/>.
- [7] Jacob Cohen: *Statistical Power Analysis for the Behavioral Sciences*. 2nd edition, Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.
- [8] CoinMarketCap Academy: Polymarket Activity Surges While Kalshi Leads in Volume (2025), <https://coinmarketcap.com/academy/article/polymarket-activity-surges-while-kalshi-leads-in-volume>, Accessed: 2025-11-13
- [9] Eugene F. Fama and Kenneth R. French: Luck versus Skill in the Cross-Section of Mutual Fund Returns. *The Journal of Finance*, 65(5):1915–1947, 2010.

- [10] Lawrence R. Glosten and Paul R. Milgrom: Bid, Ask and Transaction Prices in a Specialist Market with Heterogeneously Informed Traders. *Journal of Financial Economics*, 14(1):71–100, 1985.
- [11] Roberto Gómez-Cram, Yunhan Guo, Theis Ingerslev Jensen, and Howard Kung: Prediction Market Accuracy: Crowd Wisdom or Informed Minority? SSRN Working Paper 6617059, 2026. <https://ssrn.com/abstract=6617059>
- [12] Sanford J. Grossman and Joseph E. Stiglitz: On the Impossibility of Informationally Efficient Markets. *The American Economic Review*, 70(3):393–408, 1980.
- [13] The Guardian: On Polymarket, ‘privileged’ users made millions betting on war strikes and diplomatic strategy. What did they know beforehand? *The Guardian*, 30 January 2026. <https://www.theguardian.com/society/ng-interactive/2026/jan/30/polymarket-prediction-markets-betting>
- [14] Ben Horney: Warren Buffett Calls Sports Betting a ‘Tax on Stupidity’. *Front Office Sports*, April 2026. <https://frontofficesports.com/warren-buffett-calls-sports-betting-a-tax-on-stupidity/>
- [15] Gian Jaeger, Wang Ngai Yeung, and Renaud Lambiotte: Needles in a Haystack: Using Forensic Network Science to Uncover Insider Trading. *arXiv preprint arXiv:2512.18918*, 2025.
- [16] Richard James, Henry Leung, and Artem Prokhorov: A machine learning attack on illegal trading. In *Journal of Banking & Finance*, vol. 148, art. 106735, Elsevier, 2023. Available at <https://doi.org/10.1016/j.jbankfin.2022.106735>
- [17] Marcin Kacperczyk and Emiliano S. Pagnotta: Chasing Private Information. *The Review of Financial Studies*, 32(11):4997–5047, 2019.
- [18] Jaynti Kanani, Sandeep Nailwal, and Anurag Arjun: Matic Network Whitepaper. 2019. <https://github.com/maticnetwork/whitepaper>
- [19] Leslie Kish: *Survey Sampling*. John Wiley & Sons, New York, 1965.
- [20] Albert S. Kyle: Continuous Auctions and Insider Trading. *Econometrica*, 53(6):1315–1335, 1985.
- [21] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio: Basic Notions for the Analysis of Large Two-Mode Networks. *Social Networks*, 30(1):31–48, 2008.
- [22] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou: Isolation Forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08)*, pp. 413–422, IEEE, 2008.
- [23] Lookonchain: On-chain analysis of the *Maduro out by January 31* market. Post on X, 2026. <https://x.com/lookonchain/status/2007639475497881625>

- [24] Lookonchain (citing ZachXBT): On-chain analysis of trading in the *Axiom insider trading* market. Post on X, 2026. <https://x.com/lookonchain/status/2027044945572573534>
- [25] Piero Mazzarisi, Adele Ravagnani, Paola Deriu, Fabrizio Lillo, Francesca Medda, and Antonio Russo: A machine learning approach to support decision in insider trading detection. In *EPJ Data Science*, vol. 13, no. 66, Springer, 2024. Available at <https://doi.org/10.1140/epjds/s13688-024-00500-2>
- [26] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage: A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In *Proceedings of the 2013 Internet Measurement Conference (IMC '13)*, pp. 127–140, ACM, 2013.
- [27] Lisa K. Meulbroek: An Empirical Analysis of Illegal Insider Trading. *The Journal of Finance*, 47(5):1661–1699, 1992.
- [28] Joshua Mitts and Barak Ofir: From Iran to Taylor Swift: Informed Trading in Prediction Markets. SSRN Working Paper 6426778, 2026.
- [29] M. E. J. Newman: Scientific Collaboration Networks. II. Shortest Paths, Weighted Networks, and Centrality. *Physical Review E*, 64(1):016132, 2001.
- [30] Hunter Ng, Lin Peng, Yubo Tao, and Dexin Zhou: Price Discovery and Trading in Modern Prediction Markets. *SSRN Working Paper 5331995*, 2026. Available at <https://ssrn.com/abstract=5331995>
- [31] Judea Pearl: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [32] Polymarket: *Conditional Token Framework*. Polymarket Documentation, accessed 8 June 2026. <https://docs.polymarket.com/trading/ctf/overview>.
- [33] Polymarket: Market Integrity, Polymarket Policy, accessed June 2026. <https://integrity.polymarket.com/>
- [34] Reuters: *Italy’s Consob tests AI for market supervision, insider trading detection*. Reuters, 25 June 2024. <https://www.reuters.com/technology/artificial-intelligence/italys-consob-tests-ai-market-supervision-insider-trading-detection-2024-06-25/>.
- [35] Dorit Ron and Adi Shamir: Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security (FC 2013)*, Lecture Notes in Computer Science 7859, pp. 6–24, Springer, 2013.
- [36] Allen Sirolly, Hongyao Ma, Yash Kanoria, and Rajiv Sethi: Network-Based Detection of Wash Trading. *Columbia Business School Research Paper No. 5714122*, 2025. Available at <https://ssrn.com/abstract=5714122>

- [37] Boaz Sobrado: Alleged Insider Nets \$1 Million on Polymarket in 24 Hours. *Forbes*, 4 December 2025. <https://www.forbes.com/sites/boazsobrado/2025/12/04/alleged-insider-nets-1-million-on-polymarket-in-24-hours/>
- [38] U.S. Commodity Futures Trading Commission: CFTC Charges U.S. Service Member with Insider Trading in Nicolás Maduro-Related Event Contracts, 2026 <https://www.cftc.gov/PressRoom/PressReleases/9217-26>
- [39] UC San Diego Today: *UC San Diego Computer Scientists Earn IMC Test-of-time Award for Pivotal Bitcoin Crime Fighting Research*. UC San Diego Today, 12 December 2024. <https://today.ucsd.edu/story/computer-scientists-earn-test-of-time-award-for-pivotal-bitcoin-crime-fighting-research>.
- [40] UMA: *Welcome to UMA*. UMA Documentation, accessed 8 June 2026. <https://docs.uma.xyz/>.
- [41] United States Court of Appeals for the Second Circuit: *United States v. Ulbricht*, No. 15-1815, 858 F.3d 71, 31 May 2017. <https://law.justia.com/cases/federal/appellate-courts/ca2/15-1815/15-1815-2017-05-31.html>.
- [42] Friedhelm Victor and Andrea Marie Weintraud: Detecting and Quantifying Wash Trading on Decentralized Cryptocurrency Exchanges. In *Proceedings of the Web Conference 2021 (WWW '21)*, pp. 23–32, ACM, 2021.
- [43] The Wall Street Journal: A Mystery Trader Scored With Prescient Bets on the Nobel Peace Prize *The Wall Street Journal*, 2025. <https://www.wsj.com/finance/stocks/polymarket-nobel-peace-prize-bets-c34ee0c8>
- [44] warproxxx: `poly_data` — Polymarket historical trade-data collection tooling. GitHub repository. v1. [https://github.com/warproxxx/poly\\_data/tree/d4bfeb240b286b0775f5b3b3d4cc95eb83599a47](https://github.com/warproxxx/poly_data/tree/d4bfeb240b286b0775f5b3b3d4cc95eb83599a47)
- [45] B. P. Welford: Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, 4(3):419–420, 1962.
- [46] Justin Wolfers and Eric Zitzewitz: Prediction Markets. *Journal of Economic Perspectives*, 18(2):107–126, 2004.
- [47] Gavin Wood: Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Project Yellow Paper*, 2014. <https://ethereum.github.io/yellowpaper/paper.pdf>
- [48] Yahoo Finance: ZachXBT’s Axiom Exposé Had a Bigger Problem: Prediction Market Insiders *Yahoo Finance*, 2026. <https://finance.yahoo.com/news/zachxbt-axiom-expos-had-bigger-073842441.html>

- [49] Tao Yan, Chuanshan Huang, Claudio J. Tessone: Tracing Cross-chain Transactions between EVM-based Blockchains: An Analysis of Ethereum–Polygon Bridges. *arXiv preprint arXiv:2504.15449*, 2025.

# Appendix A

## Technical details

### A.1 Glossary of Terms

**Smart contract.** A program deployed on a blockchain that executes automatically when its conditions are met, without a trusted intermediary. Its code and state are public and its execution is enforced by the network rather than by any single party.

**UMA Optimistic Oracle.** The smart-contract system Polymarket uses to report a market’s real-world outcome onto the blockchain. Resolution is *optimistic*: a proposer asserts the outcome and posts a bond, and the assertion is accepted automatically once a challenge period elapses with no dispute. If a party disputes it (also posting a bond), the question escalates to UMA’s Data Verification Mechanism, where holders of the UMA token vote on the correct outcome and the losing side of the dispute forfeits its bond. The design assumes most outcomes are uncontroversial, so only contested questions incur the cost of a vote.

**Conditional Token Framework (CTF).** The Gnosis smart-contract standard under which each market outcome is issued as an ERC-1155 token. It enforces the invariant that one YES and one NO token can always be merged to redeem exactly \$1.00, which is what lets a token price be read as a probability.

### A.2 Trade-Level Detector Details

This appendix records the preconditions, scaling constants, and the shared filter omitted from Section 3.5. The confidence functions themselves are stated in the main text (Eqs. 3.4–3.9); search ranges are in Appendix A.3.

**Shared non-directional-trader filter.** A wallet is suppressed as likely non-directional in market  $m$  if its directional ratio  $r_{\text{dir}}(w, m) < r_{\text{nd}}^*$  and its total market volume  $V_{w,m} \geq V_{\text{nd}}^*$

(defaults 0.5 and \$10,000). The volume guard is essential: without it a wallet with a single buy and sell of any size would be suppressed, whereas the filter is only meaningful once enough trading history exists to establish the pattern. The filter is consulted by the Volume Anomaly and Extreme Position detectors only.

**Volume Anomaly.** A minimum trade count in the rolling window is required before the baseline mean and standard deviation are treated as stable; below it the detector returns no signal.

**Probability Impact.** Each sub-score is scaled linearly against twice its own threshold and clipped to  $[0, 1]$ . The detector requires a known previous price for the traded outcome, and the log-odds sub-score additionally requires both prices to lie away from the  $\{0, 1\}$  boundaries (where the logit diverges); when that condition fails the raw-probability sub-score can still fire alone.

**Accumulation.** The conjunctive filter is deliberate: large one-sided buys also characterise retail momentum and liquidity-seeking institutional traders, and requiring outcome concentration, outcome-level directional persistence, and position magnitude simultaneously (Eq. 3.6) isolates the pattern of interest. Defaults are  $c_{\text{out}}^* = 0.9$ ,  $r_{\text{dir}}^* = 0.8$ , and  $A^* = \$5,000$ .

**Extreme Position.** The trade must fall in a defined tail and meet a minimum notional. The size term saturates at \$5,000 of notional, so capital commitment beyond that point does not further raise the size component of Eq. 3.7.

**Contra-Outcome Silence.** The median normal gap  $g_{\text{med}}$  is taken over a rolling window of the most recent contra-outcome inter-trade gaps recorded in `DetectionContext`. A minimum number of recorded gaps is required for a stable median, and the elapsed gap must not exceed a configured maximum age, distinguishing a quiet contra side from a market that has simply gone inactive; a minimum notional also applies.

## A.3 Parameter Search Grids

This appendix records the search space explored by the multi-start coordinate descent of Section 4.5. The grids below define the candidate values for each block; the optimiser enumerates them as described in Section 4.5 and the selected per-window configurations are the result of that search, not values set by hand. All ranges are the active search grid; the deployed defaults in the live system are a point within these ranges.

### A.3.1 Trade-Level Detector Grids

Each detector block is searched as a full Cartesian product of its parameter values; the rightmost column gives the resulting number of candidate configurations per block. The alert threshold is swept jointly with the blocks as a one-dimensional grid of 15 values spanning  $[0.45, 0.90]$ .

Table A.1: Trade-level detector search grids. Each row lists the discrete values swept for one parameter; the block configuration count is the product of the per-parameter value counts.

| Block / parameter                          | Search values                               | Configs |
|--------------------------------------------|---------------------------------------------|---------|
| <b>Volume Anomaly</b>                      |                                             | 810     |
| lookback window (h)                        | 1, 2, 4, 8, 24, 48                          |         |
| min. trades for baseline                   | 1, 3, 8                                     |         |
| z-score threshold                          | 2.5, 3.0, 3.5, 4.0, 5.0                     |         |
| min. absolute notional                     | \$50, \$200, \$500                          |         |
| max. confidence $c_{\max}$                 | 0.02, 0.05, 0.10                            |         |
| <b>Probability Impact</b>                  |                                             | 400     |
| min. $\Delta p$                            | 0.005, 0.010, 0.020, 0.035, 0.055           |         |
| min. $\Delta$ log-odds                     | 0.20, 0.35, 0.50, 0.70, 0.90                |         |
| min. notional                              | \$250, \$600, \$1500, \$3000                |         |
| max. confidence $c_{\max}$                 | 0.10, 0.25, 0.40, 0.55                      |         |
| <b>Accumulation</b>                        |                                             | 600     |
| min. accumulation                          | \$500, \$1500, \$3500, \$8000, \$20000      |         |
| min. directional ratio                     | 0.83, 0.88, 0.92, 0.95, 0.97, 0.99          |         |
| max. confidence $c_{\max}$                 | 0.02, 0.05, 0.10, 0.18                      |         |
| min. outcome concentration                 | 0.78, 0.85, 0.91, 0.96, 0.99                |         |
| <b>Extreme Position</b>                    |                                             | 210     |
| tail threshold $\tau$                      | 0.08, 0.12, 0.15, 0.18, 0.22, 0.27, 0.33    |         |
| min. notional                              | \$150, \$400, \$800, \$1500, \$3000, \$6000 |         |
| max. confidence $c_{\max}$                 | 0.20, 0.40, 0.60, 0.80, 0.95                |         |
| <b>Contra-Outcome Silence</b>              |                                             | 720     |
| min. gap samples                           | 3, 8, 20                                    |         |
| silence threshold $\rho^*$                 | 2.0, 3.5, 5.5, 8.0                          |         |
| min. notional                              | \$500, \$1500, \$4000                       |         |
| max. contra age (min)                      | 10, 30, 90, 240, 480                        |         |
| max. confidence $c_{\max}$                 | 0.05, 0.12, 0.25, 0.40                      |         |
| <b>Alert threshold <math>\alpha</math></b> | 15 values in $[0.45, 0.90]$                 | 15      |

### A.3.2 Clustering and Jump-Anticipation Grids

The clustering and jump-anticipation blocks are searched differently from the detector blocks. Rather than a full Cartesian product, parameters are grouped (graph construction, pruning, and boost) and swept one parameter at a time within each group, holding the rest of the block at its current best. This keeps the second-layer search tractable given that each candidate requires a causal boost replay (Section 3.11.2) rather than an  $O(1)$

cache lookup. The hard outcome- and direction-constraints of Section 3.8 are encoded in the bucket key and are therefore not tunable parameters.

Table A.2: Clustering and jump-anticipation search grids. Parameters within each block are swept independently (one parameter varied at a time), not as a Cartesian product.

| Block / parameter                             | Search values                    |
|-----------------------------------------------|----------------------------------|
| <b>Clustering - graph construction</b>        |                                  |
| bucket size (s)                               | 180, 240, 300, 360, 480          |
| size normaliser $S$                           | \$5000, \$8000, \$10000, \$15000 |
| max. size multiplier $M$                      | 3.0, 5.0, 7.0                    |
| <b>Clustering - pruning</b>                   |                                  |
| $k$ -core                                     | 2, 3                             |
| min. edge weight $w_{\min}^*$                 | 0.3, 0.6, 1.0, 1.5, 2.5, 4.0     |
| <b>Clustering - boost</b>                     |                                  |
| max. boost factor $B_{\max}$                  | 1.5, 2.0, 2.5                    |
| size weight $\lambda_s$                       | 0.3, 0.4, 0.5                    |
| density weight $\lambda_d$                    | 0.15, 0.20, 0.25                 |
| ownership bonus $\lambda_o$                   | 0.1, 0.2, 0.3, 0.4, 0.6          |
| size normaliser $N_s$                         | 30, 50, 70                       |
| <b>Jump anticipation - detection</b>          |                                  |
| jump threshold $\Delta p_{\text{jump}}^*$     | 0.03, 0.05, 0.07, 0.10           |
| jump window (min) $T_{\text{jump}}$           | 15, 30, 60                       |
| <b>Jump anticipation - lookback</b>           |                                  |
| pre-jump lookback (min) $T_{\text{lookback}}$ | 30, 60, 90                       |
| min. pre-jump trades $N_{\text{pre}}^*$       | 1, 2, 3                          |
| <b>Jump anticipation - boost</b>              |                                  |
| max. boost factor $B_{\max}^{\text{JA}}$      | 1.5, 2.0, 2.5, 3.0               |
| min. trade notional                           | \$0, \$500, \$1000               |

## A.4 Trade-Oriented Objective: Derivation

This derives the lower-confidence-bound weighted-return objective of Section 4.4.2. Let the flagged BUY set be  $\{(r_i, n_i)\}$ , where  $r_i$  is the resolution return of trade  $i$  (Equation 4.5) and  $n_i$  its notional size. The notional-weighted mean return is

$$\bar{r}_w = \frac{\sum_i n_i r_i}{\sum_i n_i}, \quad (\text{A.1})$$

and the notional-weighted variance is

$$s_w^2 = \frac{\sum_i n_i (r_i - \bar{r}_w)^2}{\sum_i n_i}. \quad (\text{A.2})$$

Because the trades are weighted rather than equally informative, the relevant sample size is not the raw count but the effective sample size (Kish's effective  $n$  [19]),

$$n_{\text{eff}} = \frac{(\sum_i n_i)^2}{\sum_i n_i^2}, \quad (\text{A.3})$$

which equals the trade count when all notionals are equal and falls toward one as the weighted mean becomes dominated by a single very large trade. The weighted standard error is

$$\text{SE}_w = \sqrt{\frac{s_w^2}{n_{\text{eff}}}}, \quad (\text{A.4})$$

and the objective is the one-sided lower bound

$$\mathcal{O}_{\text{trade}} = \bar{r}_w - z_{0.95} \text{SE}_w, \quad z_{0.95} = 1.64. \quad (\text{A.5})$$

A configuration whose apparent profitability rests on too few trades, or is concentrated in too few large ones, has a small  $n_{\text{eff}}$ , a large standard error, and a correspondingly punished score, driving the optimiser toward configurations whose flagged-trade edge is both positive and reliably estimated.