Ministry of Education and Science of the Republic of Kazakhstan

SDU University



Sabidolda Sholpan

Coordination and Communication Protocols for Multi-Agent Financial AI Systems in Automated Trading

THESIS

Presented in Partial Fulfilment for the

Degree of Master of Science in Mathematics

(degree code: 7M05401)

Department of Computer Science

School of Information Technologies and Applied Mathematics

Supervisor: **Tolegen Akhmetov**

Kaskelen, 2026

School of Information Technologies and Applied Mathematics

Department of Computer Science

Dissertation Signatures Title Page

Master's Dissertation of **Sholpan Sabidolda**

le:

ordination and Communication Protocols for Multi-Agent Financial AI Systems in Automated Trading

гоматтандырылған сауда жүйелеріндегі көп агентті қаржылық жасанды интеллект жүйелерінің estіру және коммуникация хаттамалары

отоколы координации и обмена данными для многоагентных систем финансового искусственного еллекта в автоматической торговле

ubmitted in partial fulfillment of the requirements for the Master's (MSc) degree in the Department Computer Science, and approved by the Dissertation Committee:

d of Committee:	Assistant Prof., PhD, Meraryslan Meraliyev
demic supervisor:	Assistant Prof., PhD, Tolegen Akhmetov
mber's Signature:	Assistant Prof., PhD, Orynbekova K.E.
mber's Signature:	Assistant Prof., PhD, Ayazbayev D.A.

Kaskelen, 2026

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Sholpan Sabidolda

May 2026

Acknowledgements

It is my pleasure to extend my heartfelt thanks to my supervisor, Tolegen Akhmetov, for his tremendous help, continuous support, and motivation in completing this thesis. It was his patience and scholarly guidance that helped me complete this dissertation.

In addition, I owe many thanks to my classmates and colleagues who supported me during my studies.

Lastly, I wish to convey my appreciation to my family for encouraging me all along this path of learning.

Dedication

This thesis is dedicated to:

This thesis is dedicated to my beloved parents, Teney Sabidolda and Dashdonoy Altynay, whose unconditional love, endless support, encouragement, and sacrifices have always inspired me throughout my academic journey. Their belief in me has been the foundation of my motivation and perseverance.

Abstract

Machine-learning models now dominate daily-frequency financial forecasting but remain difficult to audit, and the recent turn toward multi-agent architectures has not resolved the underlying problem: existing LLM-coordinated frameworks couple specialised agents through stochastic natural-language inference, provide no formal guarantees of inter-agent message integrity, and retrain on single-observation performance dips that conflate transient noise with persistent drift. This dissertation proposes the Deterministic Multi-Agent Coordination and Control Protocol (DMACCP), a five-agent pipeline in which every message is carried by a typed envelope with a SHA-256 integrity checksum on a file-backed bus; an orchestrator evaluates five validation gates between stages, admits retraining only when a degradation signal persists across $K = 2$ consecutive observations, and uses SHAP feature attributions as a cross-run control signal that feeds back into adaptive feature masking. The protocol is evaluated on eight assets spanning five market segments over 2019–2024, against a no-coordination baseline using identical data and models. Walk-forward accuracy lies within noise bounds of the baseline (mean $\Delta = -0.22$ percentage points, overlapping 95% bootstrap intervals); the buffered mechanism reduces unnecessary retraining events by 66.7% on the ablation configuration; a macroeconomic context block dominates the SHAP attributions of seven of eight assets, validating the protected feature set. The contribution is an engineering proposal with formal reproducibility and integrity properties—obtained at under one percent of the pipeline's runtime—rather than a performance-extension result, and is positioned as a designed answer to the audit and reproducibility requirements of institutional financial machine-learning deployment.

Keywords: multi-agent systems, explainable AI, SHAP, concept drift, financial machine learning, coordination protocols, reproducibility, walk-forward validation.

List of Abbreviations

ACP	Agent Communication Protocol
AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
AUC	Area Under the Curve
BTC-USD	Bitcoin – US Dollar
CSV	Comma-Separated Values
DMACCP	Deterministic Multi-Agent Coordination and Control Protocol
DSR	Design Science Research
ETF	Exchange-Traded Fund
EU	European Union
F1	F1 Score
FRHC	Freedom Holding
GDPR	General Data Protection Regulation
GLD	SPDR Gold ETF
HSBK.IL	Halyk Bank
JSON	JavaScript Object Notation
JPM	JPMorgan Chase
KASE	Kazakhstan Stock Exchange
KS	Kolmogorov–Smirnov
LLM	Large Language Model
LogReg	Logistic Regression
MA	Moving Average
MAS	Multi-Agent System
MCP	Model Context Protocol
MiFID II	Markets in Financial Instruments Directive II
ML	Machine Learning
NASDAQ	National Association of Securities Dealers Automated Quotations

NYSE	New York Stock Exchange
OHLCV	Open, High, Low, Close, Volume
RAG	Retrieval-Augmented Generation
RF	Random Forest
ROC	Receiver Operating Characteristic
RQ	Research Question
RQ1	Research Question 1
RQ2	Research Question 2
RQ3	Research Question 3
RQ4	Research Question 4
RSI	Relative Strength Index
SHA-256	Secure Hash Algorithm 256-bit
SHAP	SHapley Additive exPlanations
TreeSHAP	Tree-based SHAP
UTC	Coordinated Universal Time
VAR	Vector Autoregressive
VIX	Volatility Index
WF	Walk-Forward
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

Table of Contents

Declaration ii

Acknowledgements iii

Dedication iv

Abstract v

List of Abbreviations vi

1 Introduction 1
- 1.1 Background and Context . . . 1
  - 1.1.1 Machine Learning in Algorithmic Trading . . . 1
  - 1.1.2 The Shift to Multi-Agent Architectures . . . 2
  - 1.1.3 Operational Pressures: Drift, Retraining, and Reproducibility 2
- 1.2 Problem Statement . . . 2
- 1.3 Research Gaps . . . 3
  - 1.3.1 Gap 1 — Absence of Typed, Deterministic, Integrity-Verified Coordination . . . 3
  - 1.3.2 Gap 2 — Explainability Used as Diagnosis, Not as a Control Signal . . . 3
  - 1.3.3 Gap 3 — Noise-Sensitive, Single-Observation Retraining Triggers . . . 3
  - 1.3.4 Summary of Gaps . . . 3
- 1.4 Research Questions . . . 4
  - 1.4.1 Primary Research Question . . . 4
  - 1.4.2 Secondary Research Questions . . . 4
- 1.5 Aim, Methodological Approach, and Objectives . . . 5
  - 1.5.1 Aim . . . 5
  - 1.5.2 Methodological Approach . . . 5
  - 1.5.3 Objectives . . . 7
- 1.6 Research Contributions . . . 7
  - 1.6.1 Contribution 1 (C1) — Typed Coordination Protocol . . . 7
  - 1.6.2 Contribution 2 (C2) — Buffered Retraining Mechanism . . . 7
  - 1.6.3 Contribution 3 (C3) — SHAP as a Cross-Run Control Signal 7
- 1.7 Scope and Delimitations . . . 8

1.7.1 In Scope . . . 8
1.7.2 Delimitations . . . 8
1.7.3 Asset Universe Selection Rationale . . . 8
1.8 Significance of the Research . . . 8
1.8.1 Theoretical Significance . . . 8
1.8.2 Practical Significance . . . 9
1.8.3 Relevance to Kazakhstan Financial Markets . . . 9
1.9 Dissertation Structure . . . 10
1.10 Chapter Summary . . . 11

2 Literature Review 12
2.1 Machine Learning in Financial Trading . . . 13
2.2 Multi-Agent Systems in Financial AI . . . 13
2.2.1 Evolution of Multi-Agent Approaches . . . 13
2.2.2 LLM-Powered Multi-Agent Frameworks . . . 13
2.2.3 Coordination Challenges and Limitations . . . 14
2.3 Explainability in Financial Machine Learning . . . 14
2.3.1 SHAP: A Unified Framework for Feature Attribution . . . 14
2.3.2 SHAP Applications in Financial Contexts . . . 15
2.4 Time-Series Validation and Concept Drift . . . 15
2.4.1 Walk-Forward Validation in Financial Forecasting . . . 15
2.4.2 Concept Drift and Adaptive Retraining . . . 16
2.5 Research Gap and Contributions . . . 16
2.6 Chapter Summary . . . 17

3 Methodology 18
3.1 Chapter Overview . . . 18
3.2 Research Paradigm and Design Principles . . . 18
3.3 Formal System Definition . . . 19
3.3.1 Formal Agent Definition . . . 20
3.3.2 Concrete Agent Instantiation . . . 20
3.4 Typed Message-Passing Layer (C1) . . . 20
3.4.1 Envelope Structure . . . 20
3.4.2 Message Type Set . . . 21
3.4.3 Bus Semantics . . . 21
3.4.4 Three-Phase Handshake . . . 21
3.5 Orchestrator: Gates, Buffer, and SHAP Filter . . . 22
3.5.1 Validation Gates . . . 22
3.5.2 Buffered Retraining Mechanism (C2) . . . 22
3.5.3 SHAP-Based Retraining Filter . . . 22
3.5.4 Orchestrator Pseudocode . . . 23
3.6 Data Acquisition and Preprocessing . . . 23
3.6.1 Feature Engineering . . . 23
3.6.2 Cleaning and Splitting . . . 24
3.6.3 Distribution Shift Detection . . . 24
3.6.4 Market-Regime Detection . . . 25
3.7 Model Training with Adaptive Masking and Regime Conditioning . . . 25

3.7.1 Classifier Families . . . 25
3.7.2 Adaptive Feature Masking . . . 25
3.7.3 Regime-Aware Hyperparameter Conditioning . . . 25
3.7.4 Walk-Forward Validation and Retraining Trigger . . . 26
3.8 Explainability Layer and Cross-Run Drift Detection (C3) . . . 26
3.8.1 SHAP Computation . . . 26
3.8.2 SHAP Consistency Across Folds . . . 26
3.8.3 Cross-Run Drift Detection . . . 26
3.8.4 Design Rationale . . . 27
3.9 Grounded Natural-Language Reporting . . . 27
3.10 Evaluation Design . . . 27
3.10.1 Asset Universe and Data Window . . . 27
3.10.2 Baselines . . . 28
3.10.3 Ablation Study . . . 28
3.10.4 Statistical Inference . . . 28
3.10.5 Low-SHAP Noise-Injection Scenarios . . . 28
3.11 Implementation, Reproducibility, and Ethics . . . 28
3.11.1 Implementation Stack . . . 28
3.11.2 Reproducibility Package . . . 28
3.11.3 Ethical Considerations . . . 29
3.12 Chapter Summary . . . 29

4 Results and Analysis **31**
4.1 Chapter Overview . . . 31
4.2 Datasets . . . 31
4.3 Evaluation Metrics and Baselines . . . 32
4.3.1 Predictive Metrics . . . 32
4.3.2 Protocol-Level Metrics . . . 32
4.3.3 Baselines . . . 33
4.4 Data Characteristics and Regime Classification . . . 33
4.5 Model Performance . . . 33
4.6 No-MAS Baseline Comparison (RQ4) . . . 34
4.7 Ablation Study (RQ3) . . . 34
4.8 Explainability Results (RQ2) . . . 35
4.8.1 Top-3 Drivers per Asset . . . 35
4.8.2 Macro-Context Dominance . . . 35
4.8.3 Cross-Run Drift Detection . . . 36
4.9 Expanding vs. Rolling Walk-Forward Windows . . . 36
4.10 Low-SHAP Scenario Analysis . . . 36
4.11 Protocol Execution and Timing . . . 37
4.11.1 Gate Pass Rate and Coordination . . . 37
4.11.2 Per-Agent Execution Time . . . 37
4.12 Trading Simulation Diagnostic . . . 37
4.13 Comparison with LLM-Coordinated Frameworks . . . 38
4.14 Consolidated Findings . . . 38
4.15 Chapter Summary . . . 40

5 Discussion **41**
5.1 Overview of Findings . . . 41
5.2 Interpretation of Results . . . 41
5.2.1 The Pattern of DMACCP–No-MAS Differences is Internally Consistent . . . 41
5.2.2 The 66.7% Figure Reports a Design-Validation, Not a Typical-Operation Estimate . . . 42
5.2.3 JPM: The Uncomfortable Case . . . 42
5.2.4 The SHAP Floor Filter Did Not Activate, and That is Diagnostic . . . 42
5.2.5 The Trending_up Regime is Harder Than Anticipated . . . 43
5.3 Comparison with Related Work . . . 43
5.3.1 Where DMACCP is Genuinely Distinct . . . 43
5.3.2 Where DMACCP is Narrower Than the Alternatives . . . 44
5.3.3 Where the Comparison is Indeterminate . . . 44
5.3.4 On Speed and Simplicity . . . 44
5.4 Strengths of the Proposed Protocol . . . 44
5.5 Limitations . . . 44
5.5.1 Structural Limitations . . . 44
5.5.2 Evaluative Limitations . . . 45
5.5.3 Methodological Assumptions . . . 45
5.5.4 Failure Modes and Edge Cases . . . 46
5.6 Unexpected Observations . . . 46
5.7 Chapter Summary . . . 47

6 Conclusions and Future Work **48**
6.1 Restatement of Purpose . . . 48
6.2 Summary of Key Findings . . . 48
6.3 Contributions . . . 49
6.4 Limitations . . . 50
6.5 Implications . . . 50
6.6 Future Work . . . 51
6.7 Closing Remarks . . . 52

Bibliography **52**

Chapter 1

Introduction

1.1 Background and Context

1.1.1 Machine Learning in Algorithmic Trading

In the last 20 years, machine learning has succeeded in replacing conventional econometrics as the leading computational technique for short-term financial prediction. Conventional techniques such as linear regression, ARIMA, and Vector Autoregressive models, which rely on the assumption of stationarity and linear relationships between variables, have been shown to be inadequate in modelling the nonlinear and regime-dependent dynamics of modern-day equity, commodities, and cryptocurrency markets [3]. Two key algorithmic innovations were crucial in this regard. Breiman's [4] development of Random Forests offered a technique that reduced variance through ensemble learning while being resistant to overfitting. Chen and Guestrin's [7] XGBoost extended this tradition through regularised gradient boosting, scalability, parallelisation, and support for missing data.

However, the benefits brought about by these models have been accompanied by a well-known trade-off: opaqueness. Srivastava and Sikroria [19] state that while algorithmic trading has increased precision, it has raised serious questions about the auditability of automated decision-making. Bussmann et al. [5] argue that black-box models are inappropriate for regulated financial institutions, and advocate for SHAP-based explanations. The EU General Data Protection Regulation (GDPR), Article 22 [11], guarantees the right not to be subject to consequential automated decisions, while MiFID II [10] requires that algorithmic trading systems operate fairly and permit audit. Explainable Artificial Intelligence (XAI), and SHAP in particular [16, 17], has emerged as the principal technical response to this obligation in financial settings. However, SHAP has predominantly been used as a passive reporting mechanism rather than an active control signal within the learning pipeline — a limitation that becomes critical in multi-agent settings, as discussed later in this chapter.

1.1.2 The Shift to Multi-Agent Architectures

Recently, it has evolved from a monolithic forecasting model to multi-agent approaches, in which specialized sub-components coordinate, ecute the entire pipeline. Modular LLM-based Agents TradingAgents [20] activities (in positions similar to those at a trading firm): fundamental analyst, sentiment analyst, technical analyst, and trader. TradingGPT [15] proposes a three-layered memory system and inter-agent debates. Sleipnir [18] adopts a leader-follower strategy with flexible routing and a review module, FinVision [12] combines HedgeAgents [14] — spe-†The work was done before joining Google.Specialised market agents for hedge in volatile markets.

In all the above mentioned case, there is an LLM mediating the communication between agents in natural language and coordination through deliberation. As Dewansh[8] points out, the statistics alone cannot distinguish a genuine consensus from an unstable negotiation. Xiao et al. [20] point three limitations on all such architectures: there is no organisation structure that assures uniformity of information-transfer, and communication is inefficient where details are obscured in long dialogues–the so-called 'telephone effect.' '

1.1.3 Operational Pressures: Drift, Retraining, and Reproducibility

Three operational factors exacerbate the opacity challenge in production. First, financial time series are non-stationary: the conditional distribution of returns relative to features varies over time, necessitating periodic retraining [1, 13]. Second, retraining in contemporary systems is often triggered by a single observed instance of poor performance, mistaking noise for drift and wasting computational resources. Third, reproducibility suffers when coordination is achieved via stochastic LLM predictions: the same input can produce different agent interactions, making auditing and post-hoc analysis impossible.

1.2 Problem Statement

The main challenge tackled by this dissertation is the development of an integrity-verification and deterministic coordination protocol for multi-agent systems, incorporating noise-filtering retraining control and explainability monitoring in financial machine-learning pipelines.

Primary problem statement: *How can a multi-agent coordination protocol for financial machine learning be designed so that artefacts retain integrity across pipeline stages, explanation behaviour is monitored as an active control signal, and retraining decisions are insulated from transient noise, without compromising predictive accuracy?*

1.3 Research Gaps

Three research gaps have been identified through a structured literature review. Table 1.1 summarises the three gaps, the contribution that addresses each, and the research question to which each maps.

1.3.1 Gap 1 — Absence of Typed, Deterministic, Integrity-Verified Coordination

Existing financial multi-agent systems are based on natural-language communications driven by LLMs [12, 14, 15, 18, 20]. There is no type specification for messages between agents; hence, no integrity guarantees are provided. General-purpose protocols such as MCP, ACP, and A2A offer structured containers for LLM-based agents but do not provide the relevant validation rules for financial ML pipelines.

1.3.2 Gap 2 — Explainability Used as Diagnosis, Not as a Control Signal

SHAP [16] and TreeSHAP [17] have gained wide acceptance for explaining financial tree-based models. However, no prior work uses feature-importance dynamics measured by SHAP across consecutive pipeline iterations to make retraining decisions. This suggests an important reframing of SHAP from post-hoc interpretation to monitoring data.

1.3.3 Gap 3 — Noise-Sensitive, Single-Observation Retraining Triggers

Walk-forward validation [2] establishes the methodological standard for non-stationary financial datasets. However, it provides no criteria for determining when out-of-sample degradation indicates genuine structural drift. Current approaches retrain on the first indication of degradation, mistaking noise for drift.

1.3.4 Summary of Gaps

The three gaps described above collectively uncover a fundamental flaw in the way financial ML pipelines that involve multiple agents are built, managed, and sustained. Gap 1 uncovers a systemic deficiency in the design of coordination within financial multi-agent ML systems, in that the currently used agent architectures depend on untyped communication based on plain language messages, making it impossible to guarantee inter-agent messaging is always done with proper data typing and integrity checks. Gap 2 unveils an underused approach: although it is common practice to leverage SHAP to conduct post-hoc explanations of machine learning models' decision process, there is no previous use of feature importance as a real-time control input which could be leveraged to guide decisions such as when

to retrain a model. Gap 3 focuses on the time element: while walk-forward validation constitutes an effective evaluation procedure for training models, it does not provide any methodological means for distinguishing genuine drifts from temporary disturbances, leading to frequent and possibly harmful unnecessary retraining procedures.

Table 1.1: Mapping of research gaps to contributions and research questions.

Gap	Description	Addressed by	RQ
Gap 1	Non-deterministic, untyped, integrity-unverified coordination	C1 — Typed coordination protocol	RQ1
Gap 2	SHAP used as post-hoc diagnosis, not a control signal	C3 — SHAP cross-run drift detection	RQ2
Gap 3	Single-observation retraining triggers conflate noise with drift	C2 — Buffered retraining mechanism	RQ3

1.4 Research Questions

1.4.1 Primary Research Question

RQ: *Can a typed, deterministic coordination protocol that integrates SHAP-based explanation monitoring and a buffered retraining mechanism preserve explanation integrity and reduce unnecessary retraining in a multi-agent financial machine-learning pipeline, while maintaining predictive accuracy comparable to a non-coordinated baseline?*

1.4.2 Secondary Research Questions

RQ1 — Coordination integrity. Can inter-agent communication be formalised through typed message envelopes with integrity checksums and validation gates such that artefact corruption between pipeline stages is detectable and the system's coordination behaviour is reproducible?

RQ2 — Explainability as control signal. Can SHAP-based feature attribution, computed and compared across consecutive runs, serve as a real-time signal of structural model change—transforming explainability from passive diagnosis into an active component of the coordination loop?

RQ3 — Retraining stability. Can a buffering mechanism that requires drift confirmation across multiple consecutive observations distinguish transient noise from genuine concept drift, reducing unnecessary retraining without compromising responsiveness to structural change?

RQ4 — Predictive parity. Does the proposed coordination protocol maintain predictive accuracy at parity with a comparable non-coordinated baseline?

1.5 Aim, Methodological Approach, and Objectives

1.5.1 Aim

The aim of this research is to design, implement, and empirically evaluate a deterministic multi-agent coordination protocol for financial machine learning—the Deterministic Multi-Agent Coordination and Control Protocol (DMACCP)—that integrates typed, integrity-verified inter-agent communication, SHAP-based cross-run explanation monitoring, and buffered retraining control. To design and test a deterministic multi-agent coordination and communication protocol for financial machine learning called the Deterministic Multi-Agent Coordination and Control Protocol (DMACCP). The key features of DMACCP include typed integrity verified interagent communication, SHAP-based cross-run explanation monitoring as an active coordination signal, and buffer-based retraining control. DMACCP compensates for the lack of any formal communication framework by adopting a typed, write-once only message bus, where every message between agents is encapsulated into a structured message envelope consisting of the sending agent identifier, the message type (from a finite set of seven types), the status, the payload, the timestamp, and the SHA-256 checksum of the payload. The Orchestrator checks the validity of each checksum and mandates the passing of five consecutive validation gates (G1-G5) before allowing any agent to proceed, ensuring that the integrity of all communications remains intact at the bit level. In order to tackle this issue of the passive nature of explainability in previous coordination models, DMACCP considers the dynamic feature importance from SHAP as a coordination signal, such that the Orchestrator detects drifts in the explanations by comparing the symmetric difference of top K features as well as the Spearman correlation between the importance vectors, and then makes its decision based on this detection, as opposed to considering SHAP results as something generated in post-coordination. In order to solve for noise sensitivity and one-shot retraining trigger conditions for noise-sensitive problems, DMACCP defines a multi-observation buffer to ensure that the signal of model performance degradation persists for at least $K = 2$ observation periods before Orchestrator decides to allow retraining. DMACCP will be tested empirically to see if these three coordination strategies – integrity verification through typed communication, orchestration based on explainability, and buffer-based retraining trigger conditions – are feasible together without causing any overhead cost to pipeline execution time, as well as maintaining prediction performance compared to an uncoordinated benchmark scenario.

1.5.2 Methodological Approach

The design science research (DSR) paradigm guides this research; a solution is proposed and then evaluated via a quantitative experiment. A positivist viewpoint is taken for the evaluation: every contribution is stated as a falsifiable claim, and measurable outcomes are used to determine its truthfulness. The paradigm for conducting this research is that of Design Science Research (DSR). In this

approach, the main deliverable is an artefact, which in this case is the DMACCP protocol along with its reference implementation. First, this artefact is built based on design science rules, followed by empirical testing of the artefact. The process of building results in the creation of a functional system, while the process of testing makes this system the subject of scientific analysis. This process of first creating and then testing is unique to DSR research and fits this study as the research contributions (C1, C2, C3) are engineering contributions, which have to be tested to determine their usefulness. The evaluation process will follow a positivist approach. All contributions will be rephrased into testable hypotheses, with each hypothesis being tested according to its outcome in relation to measurements obtained from the application of the protocol to past financial data. There will be no use of any qualitative judgment during the evaluation process; an outcome will either fall within the defined boundaries or it will not. The positivist approach is fitting since the protocol is deterministic using a given random seed – the same inputs will always generate the same outputs. Design guidelines have been set forth before any decision about how to implement anything was taken. All subsequent sections could be considered an application of these guidelines and not an independent design decision. P1 – Determinism by construction. The use of a deterministic random number generator (with seed 42) leads to identical states for each agent every time the same setup is run. None of the components in the coordination layer are using probabilistic natural language interpretation; all coordination decisions are deterministic in nature. This guideline fills the gap left open by Gap 1. P2 — Verifiable message integrity. All messages passed between agents contain a SHA-256 checksum of the message payload, using its canonical alphabetical JSON serialization. Any change in any data field, no matter how slight, will invalidate the checksum. The principle embodies the integrity guarantee of C1. P3 — Separation of concerns with file-based persistence. No agent imports another. All artefacts generated by any agent are stored at a designated path on disk before being consumed by the subsequent agent. The common state across all agents in the process is the filesystem, which is entirely accessible even after protocol termination and allows results to be reproduced using only raw files rather than having to run the protocol again. P4 — Explainability as a Control Signal. The SHAP values generated by the Explainer Agent (A) are not final output data; instead, they are inputs to both the retraining filter in the Orchestrator and the adaptive feature masking function in the Trainer. Explainability is thus woven into the loop of control instead of being added on to it afterwards. This principle implements C3, addressing Gap 2. P5 — Filtering Out Noise Through Persistence Rather Than Reactivity. All decisions that have an associated cost, namely model retraining, require that there be some evidence to trigger such a decision in the form of at least K = 2 consecutive pipeline invocations. A one-off evidence point is insufficient for retraining to occur. This principle implements C2, addressing Gap 3. Collectively, these five principles form the normative basis within which all design decisions for the protocols outlined in Sections 3.3–3.11 are taken. Each instance wherein any one of these sections takes a seemingly isolated technical decision, whether it be setting $_{K}S = 0.60, choosingK = 2,_{m} in = 0.002, orspecifying, isinfactusingP1 - P5inconjunction, wheretheprincipleisimplicitineachdecision.$

1.5.3 Objectives

To achieve the aim, five specific objectives are pursued:

1. To construct a typed asynchronous message bus with cryptographic checksums on each inter-agent message envelope and formal validation gates between consecutive pipeline stages.
2. To design a buffered retraining mechanism that defers model updates until a degradation signal persists across $K \geq 2$ consecutive observations.
3. To construct a drift-detection layer using SHAP, comparing feature importance across consecutive protocol runs and using it as a control signal for the orchestrator.
4. To test the protocol using a five-agent reference pipeline over a structured asset universe encompassing multiple market segments and volatility regimes.
5. To conduct an ablation experiment to quantify the marginal contributions of each component to reducing retraining noise, and to compare predictive performance with an uncoordinated benchmark.

1.6 Research Contributions

1.6.1 Contribution 1 (C1) — Typed Coordination Protocol

All messages sent between agents are wrapped in envelopes containing a cryptographically secure integrity checksum. Five validation gates between each stage ensure that downstream agents receive structurally and semantically valid artefacts. Unlike existing frameworks where coordination exists through LLMs, DMACCP executes deterministically on a predefined dependency graph.

Addresses: Gap 1.

1.6.2 Contribution 2 (C2) — Buffered Retraining Mechanism

A persistent signal counter for each asset tracks consecutive degradation events. Retraining is approved only after the counter reaches a predetermined threshold K. A SHAP threshold filter rejects noise-level requests before buffering.

Addresses: Gap 3.

1.6.3 Contribution 3 (C3) — SHAP as a Cross-Run Control Signal

Global SHAP feature importance is computed at each protocol run and compared with the previous run via Spearman rank correlation and top-K set change.

Structural shifts are relayed to the orchestrator. SHAP outputs also drive adaptive feature masking and natural-language report generation.

Addresses: Gap 2.

1.7 Scope and Delimitations

1.7.1 In Scope

The study covers daily-frequency directional price forecasting using OHLCV data for eight assets from January 2019 to January 2024. Three model families are evaluated: L2-regularised logistic regression, Random Forest, and XGBoost. Walk-forward validation with $N = 5$ TimeSeriesSplit folds is used throughout.

1.7.2 Delimitations

The evaluation covers a single five-year window. Intraday deployments and live-trading factors (slippage, market impact, order routing latencies) are beyond scope. LIME, counterfactuals, and attention-based explainability methods are not compared.

1.7.3 Asset Universe Selection Rationale

Five orthogonal sources of variation guide asset selection: (1) market segment (developed, frontier, emerging, commodity ETF, cryptocurrency); (2) sector (Technology, Financials, Energy, Consumer, Commodities); (3) liquidity tier; (4) volatility regime; (5) geographic exposure (US, Kazakhstan, China, Global). The eight selected assets are NVDA, JPM, XOM, GLD, BTC-USD, BABA, FRHC, and HSBK.IL. All carry continuous OHLCV data over the evaluation window retrievable via the `yfinance` interface.

1.8 Significance of the Research

1.8.1 Theoretical Significance

This research contributes to three intersecting fields. For multi-agent system research, it provides a deterministic, integrity-verified alternative to LLM-mediated coordination. For explainable AI research, it reframes SHAP from a passive diagnostic to an active control signal. For time-series learning research, it introduces a buffering mechanism that separates transient noise from persistent drift in the retraining decision. This work makes contributions to three overlapping fields. Within the domain of MAS research, this work offers a deterministic, integrity-verified approach to coordination within financial pipelines as opposed to relying on an LLM to do so – proving that desirable protocol traits like typed messaging, cryptographic integrity, and sequential gate evaluation can be guaranteed without trading off predictive accuracy and for under one percent overhead to

overall pipeline runtime. The deterministic, integrity-verifiable MAS coordination framework created is also benchmarkable by future LLM-based approaches for comparison purposes. From the XAI field of study, this work demonstrates how SHAP values can serve not only as an interpretive output but also as an active signal within pipeline runs, using two complementary metrics as signals: top-K feature set changes and Spearman rank correlation of importance vectors to identify structural drifts in the model's behavior between sequential pipeline runs. These represent new ways of utilizing XAI outputs outside of explanation and towards pipeline governance. In the field of time-series learning, this work provides a buffer for multiple observation points to distinguish market noise from concept drift when triggering model retrain in the walk-forward framework.

1.8.2 Practical Significance

The protocol provides a structural model for auditability in automated trading systems. SHAP attributions and verbal reports per trade satisfy the GDPR Article 22 right to explanation. Checksummed inter-agent envelopes satisfy MiFID II algorithmic-trading audit requirements. The retraining buffer minimises unnecessary CPU cycles and model churn. The protocol establishes a structural basis for auditability of financial processes carried out within the bounds of institutional regulatory limitations. Checksummed envelopes between agents as well as an append-only bus history meet the algorithmic trading audit trail requirements set by MiFID II, thereby creating an exhaustive, replayable record of all coordination decisions taken during the pipeline process. Per-asset SHAP explanations and grounded natural-language explanations that meet a specified level of readability satisfy the GDPR Article 22 requirement for an explanation of automated decisions, thereby giving an avenue for the dissemination of information derived from models into understandable text. The buffer-based retrain method reduces unnecessary model rebuilding, with the ablation experiment showing a decrease in model retrain events of 66.7

1.8.3 Relevance to Kazakhstan Financial Markets

Two of the eight evaluation assets (FRHC and HSBK.IL) are listed on the Kazakhstan Stock Exchange (KASE). KASE equities exhibit higher individual volatility and lower liquidity than developed-market benchmarks—conditions that stress-test the protocol's drift-detection logic and are relevant to Kazakhstan's developing regulatory framework for explainable algorithmic systems. FRHC and HSBK.IL out of eight assets to be evaluated are listed at Kazakhstan Stock Exchange (KASE), which is a frontier exchange featuring higher individual asset volatility and substantially lower liquidity than comparable developed markets benchmarks. This represents a more challenging environment for testing the drift detection logic, compared to using only assets from highly liquid developed markets since higher volatility will lead to the higher frequency of degradation alerts from a single observation, the purpose of the buffer being to filter them out. Including KASE assets serves to put the stress on C2 and C3 in a real frontier market sit-

uation. From a general perspective, auditability features of the protocol become very important in the context of evolving regulatory standards in Kazakhstan for algorithmic/automated trading systems and financial systems in general.

1.9 Dissertation Structure

The dissertation consists of six chapters. Chapter 2 reviews the literature and establishes the three research gaps. Chapter 3 describes the methodology: system architecture, the typed envelope specification and validation gates (C1), the buffered retraining mechanism (C2), the SHAP cross-run drift detection (C3), feature engineering, model training, and evaluation design. Chapter 4 presents empirical results. Chapter 5 discusses the findings in relation to the research questions. Chapter 6 summarises contributions and proposes future work. The dissertation consists of six chapters, organised to move from problem identification through protocol design, empirical evaluation, and synthesis of findings. Chapter 1 introduces the research context, establishes the inherited tension between predictive performance and auditability in financial multi-agent systems, and identifies three research gaps through a structured literature review. The aim, objectives, three contributions (C1, C2, C3), scope, delimitations, and significance of the research are stated. Chapter 2 reviews the literature across three intersecting domains: multi-agent systems in finance, explainable AI with particular focus on SHAP and TreeSHAP, and time-series model maintenance including walk-forward validation and concept drift detection. The review is organised to surface the three gaps that motivate the contributions: the absence of typed, integrity-verified coordination in financial multi-agent frameworks; the restriction of SHAP to passive post-hoc reporting rather than active control; and the absence of a principled, noise-resistant retraining trigger in walk-forward validation pipelines. The entire formal methodology is described in Chapter 3. It starts off by stating the DS Research paradigm, together with the design principles P1-P5. Protocol is formally defined as = (, , , , , , ,), while each agent is formally defined as a five-tuple (id, In, Out, f,). The typed-envelope specification, seven-type message taxonomy, SHA-256 checksum generation and three-phase handshaking protocol are formally defined (C1). Validation by the orchestrator using its five gates, the SHAP floor filter criterion, as well as the buffered retraining protocol with K = 2 is formally defined (C2). The SHAP cross-run drift detection with top-K set difference and Spearman rank correlation test is formally defined (C3). Other definitions include feature engineering, cleaning, temporal partitioning, KS-test based distribution shift detection, market regime determination, adaptive feature masking, market regime conditioning for hyperparameters, walk-forward analysis, and grounded natural language reporting with Flesch readability filter. Results from chapter 4 cover six result types, corresponding to each of the four research questions. Results from protocol execution, with gate statuses, checksum verifications, and coordination overhead expressed as percentage of total runtime are presented (RQ1). Results include SHAP feature attributions, drift detection scores, and validation of the macro-context feature set is included (RQ2). Ablation study results showing the impact of buffer and SHAP filter on retraining noise are presented (RQ3). Accuracy comparison be-

tween DMACCP and the No-MAS baseline, with 95Chapter 5 reviews the results in the context of the four research questions and three contributions. Discussion of findings is carried out in light of relevant literature presented in Chapter 2. The case of JPM failure, in which DMACCP underperformed the No-MAS baseline but did correctly detect drift is analyzed as a combination of class imbalance and regime conditioning. Limitations of the analysis such as a five-year time horizon, limited number of assets, and the inherent zero floor of TreeSHAP are considered forthrightly, with each mapped towards a future improvement. The last chapter provides a summary of the contributions, along with an explanation of how each of the contributions responds to the respective research problem, and also provides confirmation of the research problems solved in the affirmative. Seven suggestions for future research are provided, including evaluations, protocol advancements, and more advanced exploration of hybrid architecture through the combination of LLMs and DMACCP.

1.10 Chapter Summary

This chapter has established the research context and problem. LLM-based multi-agent coordination partially addresses the reproducibility and transparency limitations of monolithic ML models, but leaves three gaps unresolved: the absence of typed, deterministic, integrity-verified inter-agent coordination; the dominance of SHAP as a post-hoc diagnostic rather than a cross-run control signal; and the use of single-observation retraining triggers that conflate transient noise with structural drift. DMACCP addresses these gaps through three interrelated contributions: C1 (typed coordination protocol with checksummed envelopes and validation gates), C2 (buffered retraining mechanism), and C3 (SHAP cross-run drift detection). Chapter 2 provides the literature review on which the identified gaps are grounded.

Chapter 2

Literature Review

In this chapter, I conduct a comprehensive review of the literature concerning coordination protocols for multi-agent systems in the context of financial AI explainability. The review is divided into five inter-related areas: Section 2.1 discusses the evolution of machine learning in financial trading; Section 2.2 surveys multi-agent systems for financial markets; Section 2.3 covers explainability methods with emphasis on SHAP; Section 2.4 addresses time-series validation and concept drift; and Section 2.5 summarises research gaps and positions DMACCP's contributions. The purpose of this chapter is to undertake an organized review of relevant literature in the five domains, which collectively constitute the rationale and context for the DMACCP protocol. This review does not purport to be all-encompassing but rather selective; it includes the exact literature whose shortcomings have resulted in the identification of the three research gaps mentioned in Chapter 1. Each section of this chapter progressively focuses on the gaps addressed by each piece of literature. Section 2.1 outlines the history of machine learning in financial trading, spanning rule-based, ensemble models, deep learning and reinforcement learning all the way to the current generation of LLM-enabled multi-agent models, setting up the context behind the reproducibility and auditability problem this paper aims to address. Section 2.2 looks at different designs for multi-agent systems architectures in financial markets and discusses why existing attempts to enable coordination between agents through rules, agent-specific communication, or even LLMs fail to enable typed, integrity-verified message passing. Section 2.3 focuses on methods of explainability, with specific consideration for SHAP and TreeSHAP, showing how existing applications of these approaches limit themselves to post-hoc diagnosis only, and that there has been no previous attempt at repurposing these approaches as an active coordination signal by leveraging feature importance dynamics. Section 2.4 examines methods of validation for time-series data and concept drift detection, highlighting walk-forward validation as the state-of-the-art approach while pointing out its inability to provide a robust, noise-robust criterion for triggering model retraining. Section 2.5 draws together the lessons learned from Sections 2.1 to 2.4, outlining the three research gaps and linking each of them to DMACCP's solution, thus providing the transition to Chapter 3.

2.1 Machine Learning in Financial Trading

Machine learning methods for financial trading represent one of the most transformative changes in finance-technology over the past two decades. Prior approaches relied on econometrics—regression analysis, ARIMA modelling, and VAR models—all assuming linearity and stationarity that fails to capture the nonlinear structure of financial markets [3]. Ensemble methods substantially advanced predictive capability.

Breiman's Random Forests [4] use bootstrap aggregation over many decision trees, reducing variance without sacrificing predictive power and handling high-dimensional, noisy financial data effectively. Chen and Guestrin's XGBoost [7] extended ensemble learning with regularised gradient boosting, scalability, and parallelisation.

As model complexity increased, a paradox emerged: greater accuracy coincided with reduced interpretability. Srivastava and Sikroria [19] note that the "black box" nature of these models creates serious challenges for regulation, risk, and governance. Bussmann et al. [5] argue that such models are inappropriate for regulated financial services, motivating the development of explainable AI approaches. The challenge is compounded in multi-agent systems, where multiple agents must synchronise their reasoning across a decision chain, as addressed in Section 2.2.

2.2 Multi-Agent Systems in Financial AI

Multi-agent systems have created new problem-solving paradigms in finance, enabling distributed analysis across specialised agents [8].

2.2.1 Evolution of Multi-Agent Approaches

Early multi-agent techniques for financial prediction centred on rule-based coordination and market simulations using hand-crafted mathematical formulae. While providing architectural stability, these methods adapted poorly to unstructured inputs such as news articles, financial data, and social sentiment. The rapid development of large language models (LLMs), most notably GPT-family models, has since revolutionised the multi-agent paradigm.

2.2.2 LLM-Powered Multi-Agent Frameworks

TradingAgents, created by Xiao et al. [20], models a trading firm using LLM-based agents in roles of fundamental analyst, sentiment analyst, technical analyst, and trader. The authors identify two primary concerns common to existing multi-agent designs: poor organisational modelling of agent interaction and inadequate communication methods, both producing a "telephone effect" where information is lost or corrupted through long dialogue sequences.

TradingGPT [15] addresses memory management through a three-level architecture with short-, mid-, and long-term layers and a decay function modelling

human-like forgetting. Agents hold distinct trading personalities that improve decision diversity.

Sleipnir [18] introduces four core components: an interface module enabling leader-follower collaboration, a dynamic router selecting agents based on market conditions, a ReAct module providing analytical tools, and a Reflect module offering explainability via atomic actions. The system uses a retrieval-augmented generation (RAG) pipeline linking macroeconomic news to micro-trading events.

FinVision [12] extends multi-agent modelling by combining textual news with candlestick patterns, demonstrating the benefit of multi-modality. HedgeAgents [14] focuses on balance-aware hedging execution in volatile markets.

2.2.3 Coordination Challenges and Limitations

Dewansh [8] observes that performance numbers do not differentiate whether a strategy results from disciplined agent coordination or an inconsistent negotiation that accidentally produced profitable signals. Three limitations recur across reviewed frameworks: absence of deterministic coordination (LLM inference is stochastic); no message-type or authentication system between agents (no integrity guarantee); and post-hoc rather than in-loop explainability.

2.3 Explainability in Financial Machine Learning

Under the EU GDPR [11], individuals have the right to meaningful details on automated decisions. MiFID II [10] requires transparency and audit capability in algorithmic trading systems. Explainable AI is therefore both best practice and a compliance requirement. According to the GDPR of the EU [11], there is a need to give people undergoing automated decision-making processes access to relevant information about the underlying logic. Algorithmic trading platforms in accordance with MiFID II [10] need to allow for audit trails to trace back the rationale behind the process of automated decision-making. Explainable AI thus represents both an important area of research as well as an imperative within the financial sector.

2.3.1 SHAP: A Unified Framework for Feature Attribution

Lundberg and Lee [16] introduced SHAP (SHapley Additive exPlanations) at NeurIPS 2017, providing a theoretically grounded framework for explaining individual model predictions. SHAP values are derived from cooperative game theory Shapley values and satisfy local accuracy, missingness, and consistency axioms that uniquely characterise the solution. The concept of SHAP was developed by Lundberg and Lee [16] in NeurIPS 2017 as an approach that is backed up by theoretical concepts to explain predictions made by a model. The SHAP values are computed using the principles of Shapley values, which are part of game theory.

This methodology satisfies three axioms: local accuracy, missingness, and consistency, and these axioms collectively guarantee the uniqueness of the attribution solution among other additive attribute approaches. These axioms are defined as follows. Specifically, if f is the model and E denotes expectation, the SHAP value ϕ_j for feature j satisfies

$$f(x) = E[f(X)] + \sum_{j=1}^{p} \phi_j(x),$$

where each ϕ_j is the marginal contribution of feature j averaged over all possible feature coalitions.

Lundberg et al. [17] address the computational complexity of SHAP estimation for tree ensembles through TreeSHAP. For a model with T trees, L leaves per tree, M features, and depth D, TreeSHAP reduces complexity from $\mathcal{O}(TL^2M)$ to $\mathcal{O}(TLD^2)$, enabling efficient explanation of large ensembles.

2.3.2 SHAP Applications in Financial Contexts

Bussmann et al. [5] apply SHAP to credit risk management, using Shapley value correlation networks to group AI predictions by similar explanations across 15,000 company observations. Chan [6] provides a comparative analysis of SHAP interpretability across financial classification algorithms. Ergenç and Aktaş [9] use SHAP interaction values to identify complex inter-variable relationships in sector-specific forecasting, demonstrating that gradient boosting achieves highest predictive accuracy while SHAP provides interpretability superior to simpler importance measures.

Critically, all prior work applies SHAP as a post-training technique on a single model snapshot. No prior work uses SHAP importance dynamics across consecutive pipeline iterations to drive retraining decisions—the gap that C3 addresses.

2.4 Time-Series Validation and Concept Drift

Financial time series display temporal dependence and potential non-stationarity that challenge standard validation methods. Arlot and Celisse [1] demonstrate that standard k-fold cross-validation assumes i.i.d. observations, an assumption that does not hold for time-series data. Hawkins, Basak, and Mills [13] show that cross-validated performance estimates may be positively biased under this false independence assumption.

2.4.1 Walk-Forward Validation in Financial Forecasting

Bergmeir and Benítez [2] demonstrate through Monte Carlo simulations that blocked cross-validation provides more accurate estimates than traditional hold-out methods, particularly for small samples and directional forecast accuracy measures. Walk-forward (expanding-window) validation, in which the training partition ends

strictly before each validation partition, preserves temporal causality and is the methodological gold standard for non-stationary financial data.

2.4.2 Concept Drift and Adaptive Retraining

Concept drift occurs when the conditional distribution of targets relative to features changes over time, causing model accuracy to degrade. The Kolmogorov–Smirnov statistic

$$D_j = \sup_x \left| F_j^{\text{train}}(x) - F_j^{\text{test}}(x) \right|$$

is conventionally applied to detect distributional shift between training and test partitions. However, KS-based methods are reactive (responding to data accumulation rather than structural change) and noisy. There is a need for signals that require multiple observations before confirming drift—a need that C2 directly addresses by coupling retraining approval to multi-observation persistence rather than single-event thresholds.

2.5 Research Gap and Contributions

The C3 approach relies on two orthogonal metrics: changes in top-K feature sets and Spearman rank correlation between global SHAP importance vectors across adjacent runs. Together, these measures detect cross-run drift by monitoring both feature membership stability and feature importance ordering, thereby integrating interpretability directly within the coordination loop rather than treating explainability as a post-hoc analysis tool.

Comparison of DMACCP with prevailing multi-agent financial AI paradigms.

Framework	**Coordination**	**Message integrity**	**Explainability role**	**Drift / retraining**
TradingAgents [20]	LLM debate; firm-style roles	Not guaranteed	Debate transcript (post-hoc)	Not addressed
TradingGPT [15]	LLM debate over layered memory	Not guaranteed	Memory-guided debate; passive	Implicit memory decay
Sleipnir [18]	LLM leader–follower router	Not guaranteed	Reflect module; passive trace	Router adapts; no buffer
FinVision [12]	LLM multi-modal team	Not guaranteed	Reflection on signals; passive	Not addressed
HedgeAgents [14]	LLM hedging specialisation	Not guaranteed	Outcome-level; passive	Regime-triggered; no gate
DMACCP	**Deterministic; typed envelopes; dependency graph**	**SHA-256 on every envelope; file-backed bus**	**Cross-run SHAP rank correlation as active control signal (C3)**	**Buffered K-consecutive confirmation + SHAP threshold (C2)**

2.6 Chapter Summary

This chapter reviewed four related fields: machine learning in financial trading, multi-agent systems in financial AI, explainable machine learning, and time-series validation with concept drift. The review confirms that each field has progressed significantly, yet the combined deployment of ML and MAS remains operationally challenging. The four identified gaps—non-deterministic coordination, untyped inter-agent communication, post-hoc explainability, and noise-sensitive retraining—motivate DMACCP's three contributions: a checksummed typed message bus (C1); a buffered retraining mechanism (C2); and SHAP-based cross-run drift detection (C3).

Chapter 3

Methodology

3.1 Chapter Overview

This chapter describes DMACCP with sufficient mathematical precision to allow independent implementation and reproducibility. Section 3.2 outlines the research paradigm and design principles; Section 3.3 defines the system and agents formally; Section 3.4 specifies the typed messaging layer (C1); Section 3.5 defines the orchestrator, validation gates, and two retraining filters (C2); Section 3.6 covers feature engineering, cleaning, and regime detection; Section 3.7 covers model training with adaptive masking and regime conditioning; Section 3.8 defines the explainability layer and cross-run drift detection (C3); Section 3.9 describes the natural-language reporting layer; Section 3.10 presents the evaluation design; and Section 3.11 covers implementation and ethics.

3.2 Research Paradigm and Design Principles

A design science research (DSR) paradigm guides this research: an artefact (the DMACCP protocol and its reference implementation) is constructed and then evaluated via a quantitative experiment under controlled conditions. A positivist viewpoint is adopted for evaluation; every contribution is restated as a falsifiable claim and verified against measurable outcomes from an ablation study and a non-coordinated baseline comparison.

Five design principles guide every specification decision.

P1 — Determinism by construction. Under the same input and random seed, the protocol produces precisely the same sequence of agent states across all runs. The coordination layer relies on no stochastic natural-language inference.

P2 — Verifiable message integrity. Every inter-agent message carries a cryptographic checksum computed over its content; any modification is detectable by any downstream agent.

P3 — Separation of concerns with file-backed persistence. No agent

imports another directly; every artefact is written to disk before downstream agents read it. The file system is the shared state and remains inspectable after the protocol terminates.

P4 — Explainability as a control signal. SHAP outputs are not merely reporting data; they feed back into the orchestrator's retraining decision and the trainer's feature-masking logic. Explainability is part of the control loop.

P5 — Noise suppression by persistence, not reactivity. Decisions with operational cost—most importantly, retraining—are approved only when their triggering evidence persists across multiple observations. A single noisy observation cannot produce a costly action.

3.3 Formal System Definition

The protocol is defined as the tuple

$$\Pi = (\Sigma,\ \alpha,\ \mathcal{M},\ \Omega,\ \Gamma,\ \mathcal{A},\ \Phi,\ \mathcal{K},\ \nu)$$

whose components are as follows.

Table 3.1: Protocol tuple components.

Symbol	Interpretation
Σ	Asset universe: $\Sigma = \{$FRHC, HSBK.IL, NVDA, JPM, XOM, BABA, GLD, BTC-USD$\}$, $\lvert\Sigma\rvert = 8$.
α	Agent set: $\alpha = \{A_1, A_2, A_3, A_4, A_5\}$ (data collector, preprocessor, trainer, explainer, reporter).
$\mathcal{M}$	Typed message space (Section 3.4). Each $m \in \mathcal{M}$ is an envelope with a SHA-256 checksum.
Ω	Orchestrator: receives STATUS_REPORT and RETRAINING_REQUEST messages, evaluates gates, issues GATE_PASSED / GATE_FAILED.
Γ	Validation gate set: $\Gamma = \{G_1, G_2, G_3, G_4, G_5\}$. Each G_i : FileSystem $\to \{$PASS, FAIL$\}$.
$\mathcal{A}$	Agent execution graph: directed acyclic graph enforcing $A_1 \to A_2 \to A_3 \to A_4 \to A_5$.
Φ	Feature space: $\Phi = \mathbb{R}^{27}$ (twenty-seven features per asset per day; Section 3.6.1).
$\mathcal{K}$	Protected context features: $\mathcal{K} \subset \Phi$ with $\lvert\mathcal{K}\rvert = 8$; never removed by adaptive masking.
ν	Protocol hyperparameters: $K = 2$ (buffer window), $\varphi_{\min} = 0.002$ (SHAP floor), $\tau_{\text{KS}} = 0.60$, $\tau_{\text{mask}} = 0.40$.

3.3.1 Formal Agent Definition

An agent $A_i \in \alpha$ is a five-tuple

$$A_i = \big(id_i,\ In_i,\ Out_i,\ f_i,\ \sigma_i\big)$$

where id_i is a unique string identifier; In_i is the set of filesystem paths the agent reads; Out_i is the set of filesystem paths the agent writes; $f_i : In_i \to Out_i$ is the agent's core computation; and $\sigma_i : Out_i \to \{\text{PASS}, \text{FAIL}\}$ is the agent's self-assessment predicate, evaluated before the agent reports completion.

Agents communicate exclusively through the message bus $\mathcal{M}$. The sets Out_i and In_{i+1} are sets of file paths, enforcing that no agent imports another directly.

3.3.2 Concrete Agent Instantiation

Table 3.2 gives the concrete instantiation of each agent in the reference implementation.

Table 3.2: Concrete instantiation of the five agents.

Agent	**Role**	**Reads** In_i	**Writes** Out_i	**Gate** σ_i
A_1	Data Collector	(yfinance network)	`data/processed/*_processed.csv`	G_1: rows $\geq \lceil \lvert\Sigma\rvert/2 \rceil$ assets
A_2	Preprocessor	`*_processed.csv`	`data/final/*_{train,val,test}.csv`, scalers, splits + scalers exist `ks_shift_report.json`, `regime_report.json`	G_2:
A_3	Trainer	Train/val/test CSVs, shift and regime reports	`models/*.pkl`, `best_models.csv`, `model_comparison.csv`	G_3: `best_models.csv` non-empty
A_4	Explainer	`models/*.pkl`, `best_models.csv`, test CSVs	`xai_transfer.json`, `shap_audit.json`	G_4: all required keys present
A_5	Reporter	`xai_transfer.json`, `regime_report.json`	`results/semantic_reports/*.txt`	G_5: one report per asset

3.4 Typed Message-Passing Layer (C1)

3.4.1 Envelope Structure

Every message passed between agents is an envelope—a formally structured record with a fixed schema. A message $m \in \mathcal{M}$ is defined as

$$m = \big(id,\ \tau,\ s,\ p,\ t,\ \kappa\big)$$

with fields as shown in Table 3.3.

Table 3.3: Typed message envelope fields.

Field	Type	Semantics
id	String	Sender identifier; matches id_i of some $A_i \in \alpha$ or Ω.
τ	Enum	Message type from the closed set T (Section 3.4.2).
s	Enum	Status $\in \{\text{PENDING}, \text{COMPLETE}, \text{FAILED}\}$.
p	JSON	Payload; content depends on τ and is validated by the recipient.
t	ISO-8601	UTC timestamp of envelope construction.
κ	16-char hex	Integrity checksum: $\kappa = \text{SHA-256}\big(\text{JSON}(m \setminus \{\kappa\})_{\text{sorted}}\big)[0:16]$.

The checksum κ is computed over the JSON serialisation of m with the κ field removed and remaining keys sorted alphabetically, producing a deterministic canonical form. Any modification to any other field produces a different κ, which a recipient can detect by recomputing.

3.4.2 Message Type Set

The closed set T of admissible message types is:

$$T = \{\text{AGENT_READY},\ \text{STATUS_REPORT},\ \text{GATE_PASSED},\ \text{GATE_FAILED},\ \text{RETRAINING_REQUEST},\ \text{VALIDATION_FAILURE},\ \text{DATA_QUALITY_ALERT}\}$$

Any envelope whose $\tau \notin T$ is rejected by the bus.

3.4.3 Bus Semantics

The message bus is an asynchronous, file-backed broker with per-recipient channels. Sending an envelope performs three operations atomically: (i) the envelope is appended to the recipient's queue; (ii) it is written as a JSON file to `results/bus/`$\langle id \rangle$_$\langle \tau \rangle$`.json`, $retaining only the most recent envelope of that (\text{id}, \tau)$ pair; (iii) it is appended to `results/bus/bus_history.jsonl`, an append-only chronological log. The two-file strategy supports both latest-state inspection and full replay.

3.4.4 Three-Phase Handshake

Every agent–orchestrator interaction follows a three-phase handshake. In Phase 1, agent A_i announces readiness with an AGENT_READY envelope of status PENDING. In Phase 2, the orchestrator verifies that the preceding gate has passed and replies with GATE_PASSED, activating A_i. The agent executes f_i and writes Out_i to disk. In Phase 3, the agent posts STATUS_REPORT (COMPLETE or

FAILED). The orchestrator then evaluates gate G_i; if it passes, the next agent is activated; otherwise the protocol halts.

3.5 Orchestrator: Gates, Buffer, and SHAP Filter

3.5.1 Validation Gates

The five validation gates G_1–G_5 are filesystem predicates evaluated after each agent completes. A gate failure halts the protocol via GATE_FAILED. Table 3.4 summarises the predicates.

Table 3.4: Validation gate predicates.

Gate	After agent	Predicate
G_1	A_1 Data Collector	At least one `*_processed.csv` exists in `data/processed/`.
G_2	A_2 Preprocessor	Paired train and test CSVs in `data/final/`; count(train) = count(test); at least one scaler in `scalers/`.
G_3	A_3 Trainer	`best_models.csv` exists and is non-empty; at least one `.pkl` model exists.
G_4	A_4 Explainer	`xai_transfer.json` exists and every ticker entry contains the required keys.
G_5	A_5 Reporter	At least one `*.txt` report exists in `results/semantic_reports/`.

3.5.2 Buffered Retraining Mechanism (C2)

The orchestrator does not act immediately on a retraining request. Let $F^{(r)}$ denote the set of assets flagged at run r. For each asset $s \in \Sigma$, the persistent signal counter evolves as

$$b_s^{(r)} = \begin{cases} b_s^{(r-1)} + 1 & \text{if } s \in F^{(r)}, \\ 0 & \text{otherwise.} \end{cases}$$

Asset s is approved for retraining at run r if and only if $b_s^{(r)} \geq K$, with $K = 2$ in the reference configuration. The approved set is

$$\mathcal{A}^{(r)} = \left\{ s \in F^{(r)} : b_s^{(r)} \geq K \right\}.$$

A single-observation excursion thus cannot produce a costly retraining, operationalising principle P5.

3.5.3 SHAP-Based Retraining Filter

Before updating the buffer, the orchestrator applies a SHAP floor filter. Let $\varphi_s^{\max}$ denote the maximum absolute global SHAP value of asset s at the current run.

The filter admits the request if at least one flagged asset satisfies

$$\exists\, s \in F^{(r)} : \varphi_s^{\max} \geq \varphi_{\min}, \quad \varphi_{\min} = 0.002.$$

Assets failing this filter are recorded as RETRAINING_REJECTED with cause "max_shap below threshold"; assets passing the filter proceed to the buffer update. Admission to retraining thus requires passing the SHAP floor *and* the buffer threshold, in that order.

3.5.4 Orchestrator Pseudocode

Algorithm 3.5 gives the orchestrator's main loop.

Table 3.5: Algorithm 1: Orchestrator main loop.

Input: agents $[A_1, \ldots, A_5]$, gates $[G_1, \ldots, G_5]$, buffer window K
State: buffer $b[s] \leftarrow$ `load_from_disk`() for all $s \in \Sigma$

`bus.clear()`
for $i = 1$ **to** 5:
  start agent A_i asynchronously
  while True:
    $m \leftarrow$ `bus.receive`$(\Omega, \mathit{timeout} = 1800)$
    if $m.\tau =$ STATUS_REPORT and $m.s =$ COMPLETE: **break**
    elif $m.\tau =$ STATUS_REPORT and $m.s =$ FAILED: record_agent(A_i, FAILED); **retur**
    elif $m.\tau =$ RETRAINING_REQUEST:
      $F \leftarrow m.p.$`low_performers`
      if not `shap_filter_passes`(F): `send_reject`$(A_i,$ "noise floor"$)$; **continue**
      approved $\leftarrow$ `update_buffer`(F)
      if *approved* $= \emptyset$: `send_reject`$(A_i,$ "buffered"$)$; **continue**
      `send_gate_passed`$(A_i, \mathit{approved})$
    elif $m.\tau =$ VALIDATION_FAILURE: `log_and_ack`(A_i, m); **break**
    elif $m.\tau =$ DATA_QUALITY_ALERT: `log_and_continue`(m)
  if G_i(filesystem) = FAIL: **return** FAIL
  `bus.send`$(\mathit{next_agent}, \text{GATE_PASSED})$
return COMPLETE

3.6 Data Acquisition and Preprocessing

3.6.1 Feature Engineering

Each asset on each trading day is represented by the feature vector $\mathbf{x} \in \mathbb{R}^{27}$ partitioned into three blocks (Table 3.6). The protected context block $\mathcal{K}$ is defined up-front so adaptive masking never removes a macroeconomic context feature.

Table 3.6: Feature partition into three blocks.

Block	Features
Price / momentum	Returns r_t, Range, Gap, MA10, MA20, MA50, MA_cross, RSI, RSI_z, Ret_Lag_1–5 (14 features)
Volatility	Volatility (10-day), Volatility_20, Volatility_50, BB_width, Volume_ratio (5 features)
Protected context $\mathcal{K}$	SPY_ret, Alpha, VIX_z, VIX_ret, VIX_high_regime, DXY_ret, Sector_ret, Rel_Sector (8 features)

Key feature definitions are:

$$r_t = \ln(P_t) - \ln(P_{t-1}), \tag{3.6.1}$$

$$\sigma_{n,t} = \mathrm{std}\big(\{r_{t-i} : i = 0, \ldots, n-1\}\big), \tag{3.6.2}$$

$$\mathrm{RSI}_t = 100 - \frac{100}{1 + \dfrac{\bar{G}_t}{\bar{L}_t}}, \mathrm{VIX_z}_t = \frac{\mathrm{VIX}_t - \mu^{60}_{\mathrm{VIX}}}{\sigma^{60}_{\mathrm{VIX}}}. \tag{3.6.3}$$

$\mathrm{VIX_high_regime} = \mathbf{1}[\mathrm{VIX}_t > \mu^{60}_{\mathrm{VIX}}]$. $\mathrm{Alpha} = r_t - \mathrm{SPY_ret}_t$, and $\mathrm{Rel_Sector} = r_t - \mathrm{Sector_ret}_t$. The binary target is $y_t = \mathbf{1}[P_{t+1} > P_t]$.

3.6.2 Cleaning and Splitting

A_2 applies four cleaning operations in order: (i) deduplicate the index; (ii) replace $\pm\infty$ with NaN; (iii) winsorise each feature at $\pm 3\sigma$ from its column mean; (iv) drop remaining rows with NaN. The chronological split is 70% train / 15% validation / 15% test; a `StandardScaler` is fitted on the training partition only and applied to all three, preserving temporal causality.

3.6.3 Distribution Shift Detection

For each feature $j \in \Phi$, a two-sample Kolmogorov–Smirnov test is applied between its train and test distributions:

$$D_j = \sup_x \big| F_j^{\mathrm{train}}(x) - F_j^{\mathrm{test}}(x) \big|.$$

A DATA_QUALITY_ALERT is raised if the fraction of features with $p < 0.05$ exceeds $\tau_{\mathrm{KS}} = 0.60$. Shifted feature names are exported to `ks_shift_report.json` for use by the trainer's adaptive masking logic (Section 3.7.2).

3.6.4 Market-Regime Detection

A_2 classifies the test-period market regime using the volatility ratio $vr = \sigma_{50}^{\text{test}} / \sigma_{50}^{\text{train}}$ and the normalised MA-cross deviation $mc = \text{median}(\text{MA_cross}^{\text{test}}) / \text{IQR}(\text{MA_cross}^{\text{train}})$:

$$l_s = \begin{cases} \text{high_vol} & \text{if } vr \geq 1.3, \\ \text{low_vol} & \text{if } vr \leq 0.7, \\ \text{trending_up} & \text{if } mc \geq +0.25, \\ \text{trending_down} & \text{if } mc \leq -0.25, \\ \text{ranging} & \text{otherwise.} \end{cases}$$

The label l_s is exported to `regime_report.json`.

3.7 Model Training with Adaptive Masking and Regime Conditioning

3.7.1 Classifier Families

Three classifier families are trained per asset: Logistic Regression (L2-regularised, lbfgs solver); Random Forest (bootstrap ensemble); and XGBoost (L1/L2 regularised gradient boosting). Hyperparameters are selected by exhaustive grid search evaluated by $N = 5$ TimeSeriesSplit folds with accuracy as the scoring metric.

3.7.2 Adaptive Feature Masking

Let $\mathcal{K}_s$ denote the set of KS-shifted features for asset s, and let $\varphi_j^{(r-1)}$ denote the global SHAP importance of feature j at the previous run. The mask is

$$M_s = \left\{ j \in \Phi \setminus \mathcal{K} : j \in \mathcal{K}_s \ \wedge \ \varphi_j^{(r-1)} < \varphi_{\text{mask}} \right\},$$

with $\varphi_{\text{mask}} = 0.010$. The active feature set is $\Phi_s^{\text{active}} = \Phi \setminus M_s$. Masking is applied only when the shifted fraction of asset s is at least $\tau_{\text{mask}} = 0.40$, and is reverted to the full feature set if masking would leave fewer than five features. The protected set $\mathcal{K}$ is never masked regardless of SHAP value.

3.7.3 Regime-Aware Hyperparameter Conditioning

After grid search selects base hyperparameters θ^{base}, the regime label l_s applies a regime-specific adjustment $\Delta\theta_{l_s}$, producing

$$\theta^{\text{final}} = \text{merge}(\theta^{\text{base}}, \Delta\theta_{l_s}).$$

High-volatility regimes tighten regularisation (e.g. RF max_depth = 3, XGB reg_lambda = 8.0, subsample = 0.6) to reduce overfitting; trending regimes increase tree count and moderate learning rates (e.g. n_estimators = 300, learning_rate = 0.05).

3.7.4 Walk-Forward Validation and Retraining Trigger

Walk-forward evaluation uses $N_{\text{folds}} = 5$ TimeSeriesSplit folds. Let a_s^{WF} denote the per-asset walk-forward accuracy and a_s^{maj} the majority-class baseline. The trainer flags asset s if

$$\Delta_s = a_s^{\text{WF}} - a_s^{\text{maj}} < -2.0 \text{ percentage points.}$$

A RETRAINING_REQUEST envelope listing every flagged asset is posted to the bus and processed through the two filters of Section 3.5.

3.8 Explainability Layer and Cross-Run Drift Detection (C3)

3.8.1 SHAP Computation

Agent A_4 uses TreeExplainer for Random Forest and XGBoost, and LinearExplainer for Logistic Regression. SHAP values are computed on the held-out test partition X_s^{test}. The global importance of feature j for asset s is

$$\varphi_{s,j} = \frac{1}{|X_s^{\text{test}}|} \sum_{t \in X_s^{\text{test}}} \left|\hat{\varphi}_{s,j}(x_t)\right|,$$

where $\hat{\varphi}_{s,j}(x_t)$ is the SHAP value of feature j for instance x_t. The top-K features ($K = 3$) and their φ values are exported to `xai_transfer.json`, together with the full ranking φ_s^{full} and the maximum $\varphi_s^{\max} = \max_j \varphi_{s,j}$.

3.8.2 SHAP Consistency Across Folds

Let $T_s^{(k)}$ denote the top-K feature set from fold k. The consistency overlap is

$$\Omega_s = \frac{\left|\left\{f : |\{k : f \in T_s^{(k)}\}| > N_{\text{folds}}/2\right\}\right|}{K}.$$

An asset is labelled UNSTABLE if $\Omega_s < 0.50$ and STABLE otherwise.

3.8.3 Cross-Run Drift Detection

Two orthogonal signals detect structural change between run r and run $r-1$.

Top-K set change. Let $T_s^{(r)}$ and $T_s^{(r-1)}$ denote the top-K feature sets. The top-K change fraction is

$$\Delta_K^{(r)}{}_s = \frac{|T_s^{(r)} \triangle T_s^{(r-1)}|}{2K},$$

where $\triangle$ denotes the symmetric difference. Top-K drift is declared if $\Delta_K^{(r)}{}_s > 0.5$.

Rank correlation. Let $\varphi^{(r)}$ and $\varphi^{(r-1)}$ denote the global importance vectors

over shared features. The Spearman rank correlation is

$$\rho_s^{(r)} = \text{Spearman}\big(\varphi^{(r)},\, \varphi^{(r-1)}\big).$$

Structural drift is declared if $\rho_s^{(r)} < 0.50$.

The two signals are combined disjunctively:

$$\text{drift}_s^{(r)} = \big[\Delta_K^{(r)} s > 0.5\big] \;\vee\; \big[\rho_s^{(r)} < 0.50\big].$$

When $\text{drift}_s^{(r)} = 1$, the asset is recorded as DRIFTED in `xai_transfer.json` and a VALIDATION_FAILURE envelope is posted to the orchestrator.

3.8.4 Design Rationale

The two signals are designed to be orthogonal. The top-K set change captures coarse-level shifts (which features are important), while rank correlation captures fine-level shifts (how features are ordered among those that remain important). A single-signal protocol would miss one of the two failure modes. Combining them with OR is conservative—it reports both types of structural change and allows the orchestrator to respond via the buffering and SHAP-filter mechanisms.

3.9 Grounded Natural-Language Reporting

Agent A_5 generates one natural-language research note per asset using the OpenAI GPT-4o model, grounded on the asset's regime label, walk-forward accuracy, top-K SHAP features, and their numerical values. The prompt mandates that only the inputted top-K features appear in the output report, preventing hallucination of features not attributed by the model.

Every report is evaluated against two admissibility criteria. The first is a minimum length of 600 characters. The second is the Flesch Reading Ease (FRE) score:

$$\text{FRE} = 206.835 - 1.015\,\frac{W}{S} - 84.6\,\frac{Y}{W},$$

where W is the word count, S is the sentence count, and Y is the syllable count. The report is admissible when $\text{FRE} \in [8, 65]$; outside this window the report is regenerated once. In the reference run, all eight assets produced admissible reports on the first attempt.

3.10 Evaluation Design

3.10.1 Asset Universe and Data Window

The protocol is evaluated on eight assets over daily OHLCV data from 1 January 2019 to 1 January 2024, retrieved through the `yfinance` API.

3.10.2 Baselines

Three baselines are used: (i) **majority_class** — always predicts the dominant class; (ii) **No-MAS** — identical data splits and model families as DMACCP, without any coordination layer (no adaptive masking, no regime conditioning, no buffer, no SHAP filter); (iii) **buy-and-hold** — passive long-only position over the test period, used in trading-simulation diagnostics.

3.10.3 Ablation Study

The ablation study runs the protocol in four matched configurations: (i) full DMACCP (buffer ON, SHAP filter ON); (ii) buffer OFF, SHAP filter ON; (iii) buffer ON, SHAP filter OFF; (iv) both OFF (immediate retraining on every flag). The measured outcome is the number of approved retraining events and the noise-reduction ratio relative to configuration (iv).

3.10.4 Statistical Inference

Per-asset test and walk-forward accuracies are reported with 95% bootstrap confidence intervals (n = 10,000 resamples). DMACCP–No-MAS comparisons are reported as CI overlap rather than p-values; overlapping intervals are interpreted as "difference within noise bounds."

3.10.5 Low-SHAP Noise-Injection Scenarios

Two scenarios probe the SHAP filter under Gaussian noise injected into features at amplitudes $\varepsilon \in \{0, 0.5, 1.0, 5.0\}$ (in units of per-feature standard deviation). Scenario 1 uses XOM with Random Forest (TreeSHAP); Scenario 2 uses JPM with Logistic Regression (LinearSHAP).

3.11 Implementation, Reproducibility, and Ethics

3.11.1 Implementation Stack

The reference implementation runs on Python 3.12 with dependencies on scikit-learn, XGBoost, SHAP, SciPy, pandas, and NumPy. Random seeds are fixed to 42 at each execution. All model artefacts, scalers, CSV splits, protocol metadata, SHAP reports, bus envelopes, and orchestrator run logs are saved in the project tree.

3.11.2 Reproducibility Package

Any result reported in Chapter 4 can be re-examined from raw files without re-executing the protocol. The inter-agent communication layer uses an asynchronous, file-backed broker; `yfinance` retrieves market data.

3.11.3 Ethical Considerations

Only publicly available market data from `yfinance` is used; no personal data, human subjects, or private trade data are involved. All evaluation is historical backtesting; no real-time trading takes place.

3.12 Chapter Summary

DMACCP has been formally stated as the tuple $\Pi = (\Sigma, \alpha, \mathcal{M}, \Omega, \Gamma, \mathcal{A}, \Phi, \mathcal{K}, \nu)$, with each agent defined as $A_i = (id_i, In_i, Out_i, f_i, \sigma_i)$. Inter-agent communication (C1) uses SHA-256 checksummed envelopes over a file-backed bus with a closed type set and a three-phase handshake. The orchestrator evaluates five validation gates and approves retraining only when both the SHAP floor $\varphi_{\min}$ and the buffer threshold $K = 2$ are satisfied (C2). The explainer computes SHAP importance on the test partition and compares it with the prior run using two orthogonal metrics combined disjunctively (C3). The trainer applies adaptive feature masking based on KS shift and prior-run SHAP importance, with the protected set $\mathcal{K}$ always retained, and regime-aware hyperparameter conditioning.

The typed message-passing interface (C1) creates deterministic and integrity-verified coordination using SHA-256 checksummed envelopes sent via an asynchronous file-based bus, a seven-type message taxonomy, five sequential validation gates G_1–G_5 evaluated on the filesystem after each agent completes its process, and a three-phase handshake ensuring compliance with the directed acyclic agent execution graph $\mathcal{A}$ without relying on inter-agent imports. These measures ensure that no downstream agent processes a compromised or incomplete artefact, while providing replayable coordination history via the append-only bus log.

The buffered retraining mechanism (C2) employs two sequential filters applied by the orchestrator upon receipt of each `RETRAINING_REQUEST` message. The SHAP floor filter rejects any request where no flagged asset exceeds $\varphi_{\min} = 0.002$, eliminating zero-signal degenerate cases. The persistent signal buffer requires degradation evidence to accumulate across $K = 2$ consecutive runs before retraining is approved, preventing single-observation noise excursions from triggering unnecessary model rebuilds.

The cross-run SHAP drift detection layer (C3) computes global feature importance $\varphi_{s,j}$ on the held-out test partition at each run and compares it with the previous run using two orthogonal signals: the top-K set change fraction $\Delta_{Ks}^{(r)}$, detecting coarse-level shifts in feature dominance, and the Spearman rank correlation $\rho_s^{(r)}$, capturing fine-level reordering within the stable feature set. The two signals are combined disjunctively into a single binary drift indicator relayed to the orchestrator and fed forward into the trainer's adaptive feature masking logic, operationalising design principle P4: explainability as a control signal, not a passive report.

In support of the three contributions, the chapter has specified a complete 27-feature engineering pipeline partitioned into price/momentum, volatility, and pro-

tected macroeconomic context blocks; a leakage-safe temporal split with KS-based distributional shift detection; market-regime classification driving regime-aware hyperparameter conditioning; and grounded natural-language report generation with Flesch Reading Ease gating. The evaluation design comprises four ablation configurations, three baseline methods, 95% bootstrap confidence intervals, two noise-injection diagnostic scenarios, and an eight-asset universe spanning five years. Chapter 4 reports the resulting empirical findings.

Chapter 4

Results and Analysis

4.1 Chapter Overview

In this chapter, the results obtained by running DMACCP under the testing methodology of Chapter 3 are presented. All data are derived from a single reference run whose outputs—CSV files, model artefacts, SHAP JSON, and bus envelopes—are available in the reproducibility package (Section 3.11.2). Section 4.2 describes the datasets; Section 4.3 defines evaluation metrics and baselines; Section 4.4 reports data characteristics and regime classification; Section 4.5 reports per-asset model performance; Section 4.6 presents the No-MAS comparison (RQ4); Section 4.7 presents the ablation study (RQ3); Section 4.8 presents explainability results (RQ2); Section 4.9 compares expanding versus rolling windows; Section 4.10 analyses low-SHAP noise-injection scenarios; Section 4.11 reports protocol execution and timing; Section 4.12 presents the trading-simulation diagnostic; Section 4.13 qualitatively compares DMACCP with LLM-coordinated frameworks; and Section 4.14 consolidates findings.

4.2 Datasets

Daily OHLCV data were retrieved via the `yfinance` API over 1 January 2019 to 1 January 2024. Bitcoin additionally covers weekends because cryptocurrency trading is continuous. Table 4.1 summarises the dataset profile.

Class imbalance is close to 50/50 for most assets, with JPM the most imbalanced (57.1% up-moves) and FRHC the most balanced (exactly 50%). Bitcoin has 1,776 data points rather than 1,208 due to weekend trading, but maintains the 70/15/15 split. All assets share the same 27-dimensional feature vector (Section 3.6.1).

Table 4.1: Asset universe and dataset profile. Rows are total observations. Train/-Val/Test shows the 70/15/15% chronological split. % y=1 is the majority-class frequency (next-day up-move).

Ticker	Asset	Segment	Sector	Rows	Train	Val	Test	% y=1
FRHC	Freedom Holding	KASE	Financials	1208	846	181	181	50.0%
HSBK.IL	Halyk Bank	KASE	Financials	1211	847	182	182	51.1%
NVDA	NVIDIA	US NASDAQ	Technology	1208	846	181	181	54.4%
JPM	JPMorgan Chase	US NYSE	Financials	1208	846	181	181	57.1%
XOM	ExxonMobil	US NYSE	Energy	1208	846	181	181	51.1%
BABA	Alibaba Group	EM	Consumer	1208	846	181	181	52.2%
GLD	SPDR Gold ETF	ETF	Commodity	1208	846	181	181	50.5%
BTC-USD	Bitcoin	Crypto	Crypto	1776	1243	266	267	50.9%

4.3 Evaluation Metrics and Baselines

4.3.1 Predictive Metrics

Four predictive metrics are reported (Table 4.2). The binary target $y_t \in \{0, 1\}$ denotes an up-move on day $t + 1$.

Table 4.2: Predictive metrics used in this chapter.

Metric	Definition	Role
Test accuracy	$(TP + TN)/N$ on the 15% test partition	Primary headline metric
Walk-forward accuracy	Mean accuracy across 5 TimeSeriesSplit folds	Out-of-sample stability indicator; used for retraining triggers
F1 (macro)	$2 \cdot (P \cdot R)/(P + R)$ averaged over both classes	Guards against class-imbalance artefacts on JPM and NVDA
ROC-AUC	Area under the ROC curve	Ranks probability calibration independently of the decision threshold

Confidence intervals of 95% are computed by bootstrap with $n = 10{,}000$ resamples. Overlapping intervals indicate differences within noise limits; this is preferred over null-hypothesis p-values as it conveys effect size.

4.3.2 Protocol-Level Metrics

Four operational metrics assess the coordination layer: (i) retraining events (count of approved RETRAINING_REQUEST envelopes); (ii) noise reduction percentage (relative to the unfiltered baseline); (iii) gate pass rate; and (iv) per-agent elapsed time.

4.3.3 Baselines

Three baselines are used (Table 4.3).

Table 4.3: Baselines used in Chapter 4.

Baseline	What it does	What it isolates
Majority-class	Always predicts the dominant class	Random-forecasting floor
No-MAS	Identical data and models as DMACCP, no coordination layer	Contribution of the protocol layer (answers RQ4)
Buy-and-hold	Passive long-only position	Reference for trading-simulation diagnostic

4.4 Data Characteristics and Regime Classification

All eight assets were retrieved successfully. The KS distributional comparison raised a DATA_QUALITY_ALERT for two assets: JPM (22 of 27 features shifted, 82%) and BTC-USD (23 of 27, 85%), both exceeding $\tau_{\text{KS}} = 0.60$. The remaining six assets had shifted fractions between 37% and 52%, consistent with moderate distributional movement over a five-year window.

The regime classifier (Section 3.6.4) assigns one of five labels. Three assets are classified `high_vol` (FRHC with volatility ratio $vr = 1.58$; JPM with $vr = 1.44$; BTC-USD with $vr = 2.35$), one as `low_vol` (BABA, $vr = 0.55$), two as `trending_up` (HSBK.IL, NVDA), and two as `ranging` (XOM, GLD). No asset triggered the `trending_down` label.

4.5 Model Performance

Table 4.4 reports per-asset test accuracy, walk-forward accuracy, F1, and ROC-AUC for the selected best model. All metrics are on the held-out 15% test partition or five TimeSeriesSplit folds (walk-forward).

Table 4.4: Per-asset performance of the selected best model. Test Acc and WF Acc in %. F1 and AUC on the $[0, 1]$ scale.

Ticker	Best model	Maj.%	Test Acc	WF Acc	F1	AUC	Regime
FRHC	LogReg	50.0	52.75	52.86	0.519	0.532	high_vol
HSBK.IL	RandomForest	51.1	53.85	52.77	0.535	0.540	trending_up
NVDA	RandomForest	54.4	53.30	52.71	0.511	0.493	trending_up
JPM	XGBoost	57.1	53.85	47.57	0.540	0.533	high_vol
XOM	XGBoost	51.1	48.90	49.57	0.488	0.482	ranging
BABA	LogReg	52.2	52.75	47.43	0.527	0.540	low_vol
GLD	LogReg	50.5	50.00	49.71	0.490	0.482	ranging
BTC-USD	XGBoost	50.9	52.81	49.95	0.528	0.573	high_vol
Mean	—	52.2	52.4	50.3	0.517	0.522	—

All assets cluster around the random benchmark of approximately 50%, consistent with weak-form market efficiency at daily frequency. The chapter's claim is not that absolute accuracy is strong, but that the protocol's properties are preserved at these accuracy levels, as confirmed by the No-MAS comparison (Section 4.6).

4.6 No-MAS Baseline Comparison (RQ4)

Table 4.5 reports per-asset test and walk-forward accuracies for DMACCP and the No-MAS baseline side by side.

Table 4.5: DMACCP vs. No-MAS baseline. Δ denotes DMACCP − No-MAS in percentage points.

Ticker	No-MAS Test	DMACCP Test	Δ Test	No-MAS WF	DMACCP WF	Δ WF
FRHC	51.6	52.8	+1.2	49.6	52.9	+3.3
HSBK.IL	53.9	52.8	−1.1	51.5	51.8	+0.3
NVDA	55.5	53.9	−1.6	49.1	51.7	+2.6
JPM	54.4	53.9	−0.5	49.9	47.6	−2.3
XOM	52.2	48.9	−3.3	50.1	49.6	−0.5
BABA	51.1	52.8	+1.7	49.4	47.4	−2.0
GLD	53.3	50.0	−3.3	52.9	49.7	−3.2
BTC-USD	52.8	53.9	+1.1	51.7	51.8	+0.1
Mean	53.1	52.4	−0.7	50.5	50.3	−0.2

Three observations follow. First, the mean $\Delta_{\text{WF}} = -0.22$ pp and $\Delta_{\text{test}} = -0.75$ pp lie within the noise ranges implied by the 95% bootstrap confidence intervals (DMACCP: [48.95%, 51.61%]; No-MAS: [49.63%, 51.39%]), which overlap, so the null of no systematic accuracy difference cannot be rejected. Second, DMACCP outperforms No-MAS on walk-forward accuracy for four of eight assets (FRHC +3.3 pp, HSBK.IL +0.3 pp, NVDA +2.6 pp, BTC-USD +0.1 pp). Third, GLD suffers the largest loss (−3.2 pp WF) because its SHAP profile is dominated entirely by technical features (RSI, Ret_Lag_1, MA_cross), giving the protected macro-context set $\mathcal{K}$ no additional signal value for that asset.

Answer to RQ4: the protocol maintains predictive accuracy at parity with the non-coordinated baseline. The operational benefits of the protocol—reproducibility, message integrity, and noise-filtered retraining—are not purchased at the cost of accuracy.

4.7 Ablation Study (RQ3)

The ablation study is conducted on three candidate retraining requests derived from the reference run: NVDA ($\Delta_s = -2.69$ pp relative to majority), JPM (−9.57 pp), and BABA (−4.77 pp). Table 4.6 reports results for the four configurations.

Three conclusions are drawn. First, the buffer reduces three requests to one approved event, corresponding to a 66.7% reduction in unnecessary retraining relative to the unfiltered baseline—the main tangible value of C2. Second, the 66.7% reduction is attributable entirely to the buffering mechanism: the No-SHAP-filter

Table 4.6: Ablation of the two retraining filters on an identical set of three flagged assets. Noise reduction is relative to the unfiltered baseline.

Variant	Buffer / SHAP filter	Requests flagged	Retraining events	Noise reduction %
Full DMACCP	ON / ON	3	1	**66.7%**
No Buffer	OFF / ON	3	3	0.0%
No SHAP filter	ON / OFF	3	1	66.7%
No Both (unfiltered)	OFF / OFF	3	3	0.0%

configuration delivers the same result, while the No-Buffer configuration yields no reduction. All three candidate assets have $\varphi^{\max}$ values exceeding the noise threshold (NVDA 0.0145, JPM 0.0752, BABA 0.0720), so the SHAP floor filter did not reject any candidate. Third, the SHAP filter remains essential as a safety net for the failure mode examined in Section 4.10.

Answer to RQ3: a multi-observation buffer distinguishes transient noise from persistent drift, reducing unnecessary retraining events by 66.7% on the candidate set with no accuracy cost. The buffer is the active noise-reduction component; the SHAP filter covers a complementary failure mode.

4.8 Explainability Results (RQ2)

4.8.1 Top-3 Drivers per Asset

Table 4.7 reports the three largest SHAP drivers per asset together with $\varphi^{\max}$.

Table 4.7: Top-3 SHAP drivers per asset. Values in parentheses are mean |SHAP|. $\varphi^{\max}$ is the maximum absolute SHAP value across all 27 features.

Ticker	Model	Driver 1	Driver 2	Driver 3	$\varphi^{\max}$
FRHC	LogReg	VIX_ret (0.119)	RSI_z (0.118)	Ret_Lag_1 (0.116)	0.119
HSBK.IL	RandomForest	VIX_ret (0.015)	MA10 (0.010)	DXY_ret (0.009)	0.015
NVDA	RandomForest	Sector_ret (0.014)	VIX_z (0.014)	Rel_Sector (0.011)	0.014
JPM	XGBoost	VIX_ret (0.075)	DXY_ret (0.060)	Alpha (0.043)	0.075
XOM	XGBoost	VIX_z (0.103)	Ret_Lag_3 (0.095)	Ret_Lag_2 (0.077)	0.103
BABA	LogReg	VIX_z (0.072)	Alpha (0.049)	SPY_ret (0.047)	0.072
GLD	LogReg	RSI (0.080)	Ret_Lag_1 (0.077)	MA_cross (0.075)	0.080
BTC-USD	XGBoost	VIX_z (0.199)	RSI (0.157)	RSI_z (0.153)	0.199

4.8.2 Macro-Context Dominance

A VIX-family feature appears among the top-3 drivers for seven of eight assets; the exception is GLD (RSI, Ret_Lag_1, MA_cross). This empirically validates the architectural decision to protect the macro-context block $\mathcal{K}$: if adaptive masking were permitted on VIX-family features, a majority of assets would lose their single most informative driver when the KS shift flag is raised.

The range of $\varphi^{\max}$ spans approximately one order of magnitude, from 0.014 (NVDA) to 0.199 (BTC-USD). This range reflects the model family used, not model quality: TreeSHAP (Random Forest/XGBoost) and LinearSHAP (Logistic Regression) produce values at different scales. The filter threshold $\varphi_{\min} = 0.002$

lies well below the lowest $\varphi^{\max}$ (0.014 for NVDA), so no asset was filtered out in this run.

4.8.3 Cross-Run Drift Detection

In the reference run, six assets were labelled STABLE while two exhibited structural change at the model-family level (HSBK.IL moved from XGBoost to Logistic Regression; JPM from Logistic Regression to XGBoost; XOM from XGBoost to Random Forest). For stable assets, top-3 overlaps were generally high—3/3 with rank correlation $\rho = 1.00$ for BABA; 2/3 with $\rho = 0.952$ for BTC-USD.

Answer to RQ2: SHAP attribution computed and compared across successive runs becomes a live indicator of structural model change, transforming explainability from a passive analysis method into an active component of the coordination loop. The same SHAP calculation that drives the natural-language report (Section 4.11) also feeds the orchestrator's retraining decision and the trainer's adaptive masking.

4.9 Expanding vs. Rolling Walk-Forward Windows

The comparison evaluates whether the walk-forward training window should grow with each fold (expanding) or remain fixed at the most recent 400 observations (rolling). Rolling wins on five of eight assets; expanding wins on three. The mean absolute discrepancy is -0.37 pp, within the noise bound established in Section 4.6. NVDA shows the highest asset-level discrepancy (-2.57 pp for rolling), consistent with its `trending_up` regime containing a marked repricing event in 2019–2024 where dropping pre-repriced training information benefits the rolling approach.

4.10 Low-SHAP Scenario Analysis

Gaussian noise is injected into test features at amplitudes $\varepsilon \in \{0, 0.5, 1.0, 5.0\}$ (in units of per-feature standard deviation). Two results arise. First, $\varphi^{\max}$ for the tree-based scenario (XOM + Random Forest, TreeSHAP) stays within $[0.016, 0.018]$ regardless of noise amplitude, well above the 0.002 threshold. This is a consequence of the TreeSHAP structural non-zero property: every feature traversed in a decision node accumulates non-zero attribution. Second, $\varphi^{\max}$ for the linear scenario (JPM + Logistic Regression, LinearSHAP) grows steadily with noise amplitude. The SHAP filter thus operates as a fail-safe for degenerate-model conditions rather than as a general noise-injection detector.

4.11 Protocol Execution and Timing

4.11.1 Gate Pass Rate and Coordination

The reference run completed successfully: all five agents reported COMPLETE status and all five validation gates reported PASS (gate pass rate 100%). The orchestrator log recorded 25 protocol events across the five agents. One RETRAINING_REQUEST was issued (for JPM), passed the SHAP filter, and was approved by the buffer. Two DATA_QUALITY_ALERT notifications were issued (JPM and BTC-USD). All eight NLG reports had FRE values within the admissible interval $[8, 65]$; the mean FRE was approximately 57 and report lengths ranged from 1,163 to 1,344 characters.

4.11.2 Per-Agent Execution Time

Table 4.8 reports wall-clock elapsed time per agent for the reference run.

Table 4.8: Per-agent execution time for the reference run. Measurements are wall-clock seconds between activation and STATUS_REPORT. Total runtime: 503.8 s.

Agent	Role	Elapsed (s)	% of total	Principal cost driver
A_1	Data Collector	9.2	1.8%	Network I/O to yfinance
A_2	Preprocessor	0.7	0.1%	In-memory cleaning, KS tests, regime classification
A_3	Trainer	425.5	84.5%	Grid search on 3 models × 8 assets × 5 folds
A_4	Explainer	31.3	6.2%	SHAP computation + fold consistency check
A_5	Reporter	37.1	7.4%	OpenAI GPT-4o calls + FRE validation
Total		**503.8**	**100.0%**	—

Three observations follow. First, the trainer dominates at 84.5% of total runtime, scaling as assets × models × folds × grid size. Second, preprocessing costs are negligible (0.7 s, 0.1%): the KS-shift and regime-classification diagnostics that feed the trainer's adaptive masking are essentially free. Third, the overhead introduced by the message bus (checksum calculations, envelope persistence, gate evaluation) cannot be measured separately and is at most a rounding error relative to the 503.8 s total—the protocol's determinism and integrity checks come at under one percent of runtime.

4.12 Trading Simulation Diagnostic

A long-only trading simulation was conducted over the test period. Two assets yielded a positive Sharpe ratio relative to buy-and-hold: GLD (+2.69) and FRHC (+1.04). The remaining six assets produced negative Sharpe ratios. These results indicate no evidence of systematic alpha generation; the simulation is provided as a diagnostic framework showing the full downstream pipeline, including frictionless trading. The positive Sharpe ratios for GLD and FRHC should be interpreted cautiously: the backtest excludes transaction costs, slippage, and financing, and assumes execution at the following day's close.

4.13 Comparison with LLM-Coordinated Frameworks

Table 4.9 provides a qualitative comparison of DMACCP with the five LLM-coordinated frameworks surveyed in Chapter 2.

Table 4.9: Qualitative comparison of DMACCP with LLM-coordinated frameworks.

Property	**TradingAgents**	**TradingGPT**	**Sleipnir**	**FinVision / HedgeAgents**	**DMACCP**
Determinism	No (LLM)	No	No	No	**Yes**
Integrity verification	No	No	No	No	**SHA-256 on every envelope**
Explainability role	Debate transcript	Memory trace	Reflect module	Outcome-level	**Active cross-run SHAP control signal**
Retraining gate	None	Memory decay	Router adapts agent	None / regime	**K-buffer + SHAP floor**
Audit trail	LLM context log	Memory dump	Trace log	Outcome log	**Persisted envelopes +** `bus_history.jsonl`
Input flexibility	High (free text)	High	High	High	Structured only

DMACCP is the only framework in the comparison with formal determinism, SHA-256 message integrity, and a closed-loop explainability mechanism. It is more limited in input flexibility, sacrificing the ability to process unstructured text in exchange for the determinism required in regulated environments.

4.14 Consolidated Findings

Table 4.10 collects the chapter's findings against the four research questions.

Three additional cross-cutting observations conclude the discussion. First, the protocol delivers operational benefits—retraining stability, audit trail, reproducibility—not a predictive improvement; accuracy alone cannot capture its utility. Second, the asset-level pattern of DMACCP–No-MAS differences is internally consistent with the protocol's design: the greatest gains are on assets for which distribution-shift management and macro-context signalling are most operative (FRHC, NVDA), and the greatest loss is on the asset whose SHAP profile lacks any protected macro-context driver (GLD). Third, the coordination layer imposes negligible computational cost: the trainer accounts for 84.5% of runtime, while the preprocessor and bus together account for under 1%.

Table 4.10: Consolidated findings against the four research questions.

RQ	Finding	Supporting evidence
RQ1	The typed bus and validation gates enforce reproducible, integrity-verified coordination. All five gates passed; no envelope rejected for checksum mismatch; byte-identical artefacts under fixed seed.	Section 4.11.1: `orchestrator_run.json` status=complete; gate_pass_rate = 100.0.
RQ2	SHAP operates as both a post-hoc report and a cross-run control signal. Macro-context (VIX family) dominates seven of eight assets, validating $\mathcal{K}$. Drift detection flagged two model-level structural changes.	Sections 4.8.1–4.8.3: Table 4.7; `xai_transfer.json`.
RQ3	The buffer reduces unnecessary retraining by 66.7% on the candidate set with no accuracy cost. The buffer carries the entire reduction; the SHAP filter covers a complementary failure mode.	Section 4.7: Table 4.6.
RQ4	DMACCP maintains accuracy at parity with No-MAS. Mean $\Delta_{\mathrm{WF}} = -0.22\,\mathrm{pp}$; overlapping 95% bootstrap intervals; null of no difference cannot be rejected.	Section 4.6: Table 4.5.

4.15 Chapter Summary

The reference run of DMACCP generated reproducible, integrity-validated outputs for all five agents across all five validation gates. Predictive accuracy clustered near the random level, consistent with weak-form market efficiency at daily frequency. The No-MAS comparison established that the coordination layer does not reduce accuracy (mean $\Delta_{\mathrm{WF}} = -0.22$ pp, overlapping 95% bootstrap intervals). The ablation experiment demonstrated a 66.7% reduction in retraining events, attributable entirely to the buffering mechanism. SHAP attribution identified macro-context features (especially the VIX family) as dominant in seven of eight assets, validating the protected feature set $\mathcal{K}$. Low-SHAP scenario analysis documented the structural difference between TreeSHAP and LinearSHAP under noise injection. Per-agent timing confirmed trainer dominance (84.5%) and sub-1% coordination overhead. Chapter 5 interprets these findings in relation to the research questions, examines their implications, and identifies directions for future work.

Chapter 5

Discussion

5.1 Overview of Findings

Chapter 4 reported the outcomes of the reference run. This chapter poses the interpretive question: what do these outcomes tell us about DMACCP, and with how much certainty? The key results distil to four claims. First, the protocol does not negatively affect predictive accuracy—the No-MAS comparison is a wash in aggregate and asset-by-asset. Second, the buffered retraining mechanism reduces unnecessary retraining by 66.7% on the three-candidate ablation example; the SHAP floor filter did not activate on any of the three candidates. Third, cross-run drift detection functions as a conceptual control signal, though in the reference run SHAP drift was negligible for most assets and, where it was not, it was caused by model-family switching rather than within-model reasoning drift. Fourth, the coordination overhead is imperceptible—training is the costly step; the protocol layer costs under one percent of runtime.

5.2 Interpretation of Results

5.2.1 The Pattern of DMACCP–No-MAS Differences is Internally Consistent

The per-asset walk-forward evaluation of Section 4.6 is not random. The two gains—FRHC (+3.3 pp) and NVDA (+2.6 pp)—are assets for which all three protocol mechanisms have meaningful work to do. FRHC is a volatile frontier-market equity on KASE where the training-to-test distribution shift is real, where the regime classifier correctly assigns `high_vol`, and where the dominant SHAP features (VIX_ret, RSI_z) are preserved by the protected set $\mathcal{K}$. NVDA is a `trending_up` asset containing a repricing event, where adaptive masking prevents KS-shifted non-contextual features from contaminating the training partition, and regime-aware hyperparameter conditioning pushes the Random Forest towards shallower trees.

The substantive loss—GLD (−3.2 pp WF)—is similarly illustrative. GLD is

the unique asset whose top-3 SHAP drivers (RSI, Ret_Lag_1, MA_cross) are entirely technical; the protected macroeconomic block $\mathcal{K}$ provides no additional signal value. Moreover, the `ranging` regime label triggers regularisation adjustments designed to reduce directional overfitting, but GLD has inherently directional technical signals, creating a conflict between the protocol's inductive biases and the asset's signal structure.

5.2.2 The 66.7% Figure Reports a Design-Validation, Not a Typical-Operation Estimate

The 66.7% reduction in unnecessary retraining is concrete and informative, but must be understood within its framing. It arises from an ablation setup in which three candidate retraining events were constructed for analysis: NVDA (-2.69 pp), JPM (-9.57 pp), and BABA (-4.77 pp). The full protocol defers NVDA and BABA while approving JPM; the unfiltered baseline approves all three.

This is a design-validation in the engineering sense: the filter rejects what it was designed to reject in the conditions it was designed for. The operational claim is more limited: in a series of runs where degradation signals occur according to the statistical patterns expected in the market window, the buffer rejects single-run excursions and approves only persistent signals. A multi-run longitudinal study would be required to solidify this claim beyond the design validation.

5.2.3 JPM: The Uncomfortable Case

JPM triggered the KS signal (22 of 27 features shifted, 82%), is classified `high_vol`, and received a DATA_QUALITY_ALERT—making it the asset most deserving of the protocol's interventions. Yet DMACCP trails No-MAS on walk-forward accuracy by 2.1 pp, and the trainer flags JPM with a -9.57 pp gap relative to majority.

A plausible explanation is that the aggressive regime-aware hyperparameter conditioning applied to `high_vol` assets ($\lambda = 8.0$, subsample $= 0.6$, max_depth $= 3$ for XGBoost) is well-motivated but mis-calibrated for JPM specifically. JPM has the highest class imbalance in the universe (57.1% up-moves); an aggressively regularised model on an imbalanced target with feature masking is likely to over-commit to the majority class, producing walk-forward accuracy below a less-regularised No-MAS XGBoost that incidentally overfits in a favourable direction. This points to a specific improvement: the regime hyperparameter conditioning table should be extended to a regime $\times$ class-balance joint condition.

5.2.4 The SHAP Floor Filter Did Not Activate, and That is Diagnostic

In both the ablation and the low-SHAP scenarios, the SHAP floor $\varphi_{\min} = 0.002$ was not triggered by any retrain request. All flagged assets had $\varphi^{\max}$ values substantially above this floor. The low-SHAP analysis provides the structural explanation:

TreeSHAP accumulates non-zero attribution at every visited decision node, so $\varphi^{\max}$ for any tree-based model can only approach 0.002 if the model has essentially no informative splits. This is not a calibration failure; it is a clarification of the filter's purpose. The filter is insurance against pathological degenerate models—a model trained on a compromised fold, a target corrupted upstream, or a feature matrix zeroed by a bug—not a general noise-level detector. C3 is thus better described as two mechanisms: an active cross-run drift detector (the disjunctive signal) and a safety-net degenerate-model guard (the SHAP floor).

5.2.5 The Trending_up Regime is Harder Than Anticipated

NVDA (test accuracy 53.3%, WF 52.7%) and HSBK.IL (test 53.9%, WF 52.8%) are both classified `trending_up`, yet both underperform the `high_vol` assets FRHC (52.75%) and BTC-USD (52.81%). A plausible reason is that the five-year `trending_up` label does not identify a persistently trending market but rather a series of successive regime transitions, as in NVDA's 2019–2024 repricing sequence. Predicting direction in a regime that itself changes is harder than predicting direction in a stable volatility regime. The regime classifier assigns a single label to the entire test period; finer-grained sub-period classification would likely improve the usefulness of this dimension.

5.3 Comparison with Related Work

5.3.1 Where DMACCP is Genuinely Distinct

Three features demonstrated by the reference experiment would require substantial additional engineering to be achievable by any of the five LLM-coordinated frameworks surveyed in Chapter 2.

First, **perfect reproducibility**: under a fixed random seed, the protocol generates artefacts that are byte-for-byte identical across executions. LLM-coordination frameworks cannot achieve this by construction: stochastic inference introduces variance even at zero temperature, due to tokenisation, context-window approximations, and API-side batching.

Second, **message-level integrity**: every envelope in the reference run was protected by a SHA-256 digest of its canonical serialisation. No LLM-coordinated framework surveyed offers this protection. An adversary modifying an inter-agent message in Sleipnir's ReAct module, for example, has no protocol-level mechanism detected against it.

Third, **closed-loop explainability**: in TradingAgents and Sleipnir, SHAP or debate transcripts are outputs. In DMACCP, the SHAP output loops back to the trainer (via the adaptive masking rule $M_s = \{j \in \Phi \setminus \mathcal{K} : j \in \mathcal{K}_s \wedge \varphi_j^{(r-1)} < 0.010\}$) and to the orchestrator (via $\varphi_{\min}$ and the drift-detection condition). Information used to document run r influences the decision at run $r+1$. This information-feedback loop is—as far as the author is aware—novel in the financial multi-agent literature.

5.3.2 Where DMACCP is Narrower Than the Alternatives

With respect to input flexibility, DMACCP is more limited than any LLM-coordinated framework. The typed coordination layer cannot process unstructured text—analyst notes, regulatory filings, news articles, earnings call transcripts—without extending its schema. TradingAgents, TradingGPT, and Sleipnir are designed to process these inputs natively within their LLM substrates. For event-driven or discretionary research applications, DMACCP is the wrong framework.

5.3.3 Where the Comparison is Indeterminate

In terms of accuracy, comparing DMACCP to LLM-coordinated frameworks is indeterminate, not merely inconclusive. The five surveyed frameworks evaluate on different asset universes, different time periods, and different prediction tasks. Comparing DMACCP's 50.3% mean walk-forward accuracy with Sleipnir's 38.7% Sharpe improvement would conflate two entirely distinct measurement scales. Any meaningful accuracy comparison would require reimplementing at least one of those algorithms on identical data—a task beyond the scope of this dissertation.

5.3.4 On Speed and Simplicity

DMACCP's coordination layer (preprocessor plus message bus plus envelope overhead) accounts for under 1% of the 503.8-second runtime. A comparable LLM-coordination scheme inherits the latency of LLM API calls: Sleipnir reports minute-level latency on market-aware routing decisions. DMACCP's orchestration is a deterministic asynchronous event loop using SHA-256 hashing; it will always be faster than any orchestration scheme that uses LLM inference for coordination decisions. GPT-4o is called exclusively by the reporter (A_5); removing the reporter yields a fully functional DMACCP protocol with no external inference API.

5.4 Strengths of the Proposed Protocol

Table 5.1 summarises the four genuine strengths of DMACCP supported by the reference run.

5.5 Limitations

5.5.1 Structural Limitations

Input schema rigidity. The typed coordination layer cannot consume raw unstructured text without schema extension. This is the price of determinism.

TreeSHAP's non-zero floor. The TreeSHAP structural property ensures that $\varphi^{\max}$ for tree-based models never reaches $\varphi_{\min} = 0.002$: attribution accumulates at every decision node. The SHAP floor filter is irrelevant for tree-based models—an inherent consequence of the selected explainer algorithm, not a calibration issue.

Table 5.1: Four strengths of DMACCP supported by the reference run.

Strength	Evidence and scope of claim
Reproducibility	Byte-identical artefacts across runs under fixed seed. A formal property of the construction, confirmed empirically.
Auditability	Every inter-agent message persisted as a checksummed JSON envelope plus an append-only `bus_history.jsonl`. An auditor can reconstruct every decision without re-executing the pipeline.
Noise-filtered retraining	Buffer reduced three candidate events to one (66.7% reduction) with no accuracy cost in aggregate. Valid as design-validation; narrower as a typical-operation claim.
Near-zero coordination overhead	The coordination layer accounts for <1% of the 503.8-second runtime. Formal guarantees obtained at essentially zero wall-clock cost relative to a No-MAS pipeline.

5.5.2 Evaluative Limitations

Single evaluation window. The reference period 2019–2024 appears once. Cross-window robustness cannot be assessed from a single non-rolling analysis.

Asset universe size. Eight assets are sufficient for testing coordination properties but insufficient for statistical inference on cross-asset effects. The overlapping confidence intervals in the No-MAS comparison partly reflect this small sample.

Single-run protocol metrics. The 66.7% noise reduction is measured on a designed three-candidate ablation, not an operational longitudinal trace over many runs.

5.5.3 Methodological Assumptions

Three assumptions embedded in the protocol's design may not hold universally.

First, the problem is defined as daily OHLCV directional forecasting. The regime classifier, buffer window $K = 2$, and SHAP fold-consistency check are all calibrated for this time scale; intraday deployment would require recalibration of all three.

Second, the protected macro-feature set $\mathcal{K}$ is treated as a sensible universal default. The GLD result in Section 5.2.1 demonstrates that for a technically-driven commodity, protecting $\mathcal{K}$ is counter-productive.

Third, walk-forward validation with $N_{\text{folds}} = 5$ is assumed sufficient for model selection. The standard deviation of fold-level accuracy (4–6 pp for several assets) is of the same order of magnitude as the DMACCP–No-MAS differences being

estimated.

5.5.4 Failure Modes and Edge Cases

Bus persistence under partial failure. Envelopes are written to disk before `send()` returns, but a process kill between the asyncio queue put and the disk write would leave the recipient seeing a message not present in `bus_history.jsonl`. The audit log is complete in expectation but not guaranteed.

Checksum verification is client-enforced. The recipient must recompute and validate κ. An agent that deserialises the envelope without re-validating the checksum would fail to detect tampering—a protocol-specification gap that a bus-level enforcement layer would close.

Buffer state across asset universe changes. The persistent counter $b_s^{(r)}$ is indexed by ticker. Assets added to or dropped from Σ between runs have unspecified buffer behaviour; the reference implementation defaults to zero on first appearance.

5.6 Unexpected Observations

Four observations from the reference run contradicted prior expectations.

Trending_up is the weakest regime, not the strongest. As discussed in Section 5.2.5, expected directional momentum did not materialise. The regime classifier identifies a five-year path and labels it `trending_up`; the test period's predictive difficulty reflects multiple within-period regime transitions not captured by a single label.

VIX dominance is stronger than anticipated. The VIX family appears in the top-3 SHAP drivers for 7 of 8 assets. When designing $\mathcal{K}$, the author anticipated coverage of 4–5 assets; empirical prevalence of 7 is higher than expected and raises an open question: does the VIX carry genuine predictive information, or is it a market-wide confounder that wins SHAP attribution by default because it correlates with every asset's return?

The SHAP floor filter never activated. C3 was initially characterised as "a SHAP-based filter that eliminates noise-level retraining requests." The reference run and low-SHAP scenarios reveal that the filter cannot reject tree-based model requests on structural grounds. The filter's true purpose—as a safety net against degenerate models—differs from this initial characterisation, and C3 is better described as two components.

Model selection shifted between runs. Three assets (HSBK.IL, JPM, XOM) changed best-model family between the last snapshot and the current run. The cross-run drift measure compares SHAP rankings; a model-family change produces a different ranking for a non-pathological reason. Adding a condition that distinguishes "switched model" from "drifted model" would improve the interpretability of drift signals.

5.7 Chapter Summary

Table 5.2 presents the answer to each research question in the form the evidence supports.

Table 5.2: Answer to each research question, in the form supported by the evidence.

RQ	Claim the evidence supports	Qualification
RQ1	The typed, checksummed bus with validation gates enforces deterministic, integrity-verified coordination. All five gates passed; zero checksum mismatches; byte-identical artefacts under fixed seed.	Achieved in full; a formal property of the construction confirmed empirically.
RQ2	SHAP functions as a cross-run control signal via the disjunctive drift condition. The closed loop into adaptive masking is real and the protected set aligns with observed SHAP dominance.	Achieved with revision: the SHAP floor filter is a safety net, not a noise rejector; C3 is better described as two mechanisms.
RQ3	The multi-observation buffer distinguishes transient noise from persistent drift, with 66.7% reduction in unnecessary retraining on the ablation candidates and no accuracy cost.	Achieved as design validation; a longitudinal operational claim requires multi-run data not available in this dissertation.
RQ4	Predictive accuracy is at parity with No-MAS in aggregate (-0.22 pp WF, overlapping CIs). The asset-level pattern of wins and losses is internally consistent with the protocol's design.	Achieved; the one uncomfortable case (JPM) points to a specific improvement in regime $\times$ balance conditioning.

DMACCP is best characterised as an engineering success rather than a performance-extension result: a coordination protocol with formally guaranteed properties, a well-supported retraining-stability mechanism, a genuine closed-loop explainability integration, and accuracy comparable to a well-tuned uncoordinated control group. Future directions—multi-run longitudinal evaluation, regime $\times$ class-balance conditioning, finer regime classification, and model-switch-aware drift detection—are elaborated in Chapter **??**.

Chapter 6

Conclusions and Future Work

6.1 Restatement of Purpose

This dissertation has addressed a specific gap at the intersection of three well-studied fields: the opacity of machine-learning models in automated financial trading, the fragmented state of multi-agent architectures in financial AI, and the emerging regulatory requirement for automated decision systems to be auditable and reproducible. As established in Chapter 1, the core problem was the absence of any mechanism to guarantee explanation integrity through a pipeline, to detect structural changes in model reasoning on a per-run basis, and to distinguish transient noise from persistent drift in retraining decisions. The dissertation answered this problem through the Deterministic Multi-Agent Coordination and Control Protocol (DMACCP), built on three interrelated contributions: a typed, cryptographically verified message bus with five validation steps (C1); buffered retraining with multi-observation confirmation (C2); and cross-run SHAP-based drift detection (C3).

6.2 Summary of Key Findings

The four findings below are stated as precisely as the evidence from the reference run allows.

Coordination was deterministic and integrity-verified. The reference run passed all five validation gates with all checksums matching and byte-identical artefacts across executions under the same random seed. These are formal properties of the construction, confirmed empirically.

Retraining-noise reduction behaved as designed. The buffer reduced three candidate retraining requests to one approved event, yielding a 66.7% reduction in unnecessary retraining relative to the unfiltered baseline. The SHAP floor filter did not activate on any candidate in the reference run; the low-SHAP scenario analysis revealed this to be a structural consequence of TreeSHAP's non-zero property rather than a calibration failure.

Explainability functioned as a closed-loop control signal. SHAP attributions fed forward into the trainer's adaptive masking rule and backward into the orchestrator's retraining eligibility criterion. The dominance of VIX-related features in the top-3 SHAP drivers for seven of eight assets is empirically consistent with the design of the protected macro-context set $\mathcal{K}$.

Predictive accuracy was at parity with the non-coordinated baseline. Mean $\Delta_{\mathrm{WF}} = -0.22\,\mathrm{pp}$ (DMACCP minus No-MAS); 95% bootstrap confidence intervals overlap; the null of no systematic difference cannot be rejected. The asset-level pattern of wins and losses is internally consistent with the protocol's design: DMACCP contributes most where distribution-shift management and macro-context signalling are most operative, and least where the base predictor relies on features the protocol does not privilege.

Taken together, these results confirm that the protocol delivers its operational properties—reproducibility, integrity verification, and filtered retraining—without accuracy loss and at under one percent of total runtime (503.8 s for the reference run).

6.3 Contributions

The dissertation makes three contributions to the intersection of multi-agent systems, explainable AI, and financial machine learning.

C1 — A typed, integrity-verified multi-agent coordination protocol. All inter-agent communications are carried by typed envelopes with SHA-256 checksums over canonical-form serialisation, stored in a file-backed bus with an append-only log, and subject to five validation gates between successive agent stages. Unlike the LLM-coordinated frameworks surveyed in Chapter 2, the protocol operates deterministically on a predefined dependency graph, providing predictable behaviour and an auditable trace. This is a formal protocol-level property, not a benchmarking result.

C2 — A buffered retraining mechanism with multi-observation confirmation. For each asset $s \in \Sigma$, the persistent signal counter evolves as $b_s^{(r)} = b_s^{(r-1)} + 1$ if $s \in F^{(r)}$ and 0 otherwise. Retraining is approved only when $b_s^{(r)} \geq K = 2$. A SHAP floor filter provides a safety net against degenerate-model retraining requests. The ablation study validated a 66.7% reduction in unnecessary retraining events on a three-candidate set with no accuracy cost.

C3 — SHAP repurposed as a cross-run control signal. For each run, the protocol computes asset-specific SHAP importance and compares it with the previous run using two orthogonal signals: top-K set change $\Delta_{K\,s}^{(r)}$ and Spearman rank correlation $\rho_s^{(r)}$, combined disjunctively. The same SHAP output feeds into the adaptive masking rule at run $r+1$, completing the explainability feedback loop. The drift condition and the SHAP floor together constitute C3 as two distinct mechanisms.

The three contributions are complementary and non-independent. C1 provides

the substrate for cross-run state comparison; C2 prevents the audit log from being cluttered with noise-driven retrain events; C3 provides the explainability signal that C2 uses for admission filtering and that the trainer uses for adaptive masking.

6.4 Limitations

Five limitations, each discussed in depth in Chapter 5, are summarised here.

First, the evaluation covers a single five-year window (2019–2024) and eight assets; cross-period and cross-universe robustness cannot be guaranteed from a single non-rolling analysis.

Second, the 66.7% noise-reduction figure comes from a designed three-candidate ablation, not an operational longitudinal trace.

Third, the SHAP floor filter never triggered on the reference or ablation candidates; the low-SHAP scenarios documented that TreeSHAP's structural non-zero property makes the filter inactive for tree-based models.

Fourth, regime-aware hyperparameter conditioning is under-specified: it conditions on the regime label but not on class balance, leading to DMACCP underperforming No-MAS for JPM—the asset that triggered the KS distribution-shift alert.

Fifth, determinism requires structured inputs; unstructured text-based inputs are outside the current protocol schema.

Three methodological assumptions embedded in the framework may not generalise: (i) the problem is daily OHLCV directional forecasting (the protocol's calibration is frequency-specific); (ii) the protected macro-feature set $\mathcal{K}$ is assumed a sensible default (the GLD result challenges this for technically-driven assets); and (iii) walk-forward validation with $N_{\text{folds}} = 5$ is assumed sufficient for model selection (fold-level standard deviations of 4–6 pp are comparable to the DMACCP–No-MAS differences being estimated).

6.5 Implications

Theoretical implications. DMACCP provides an engineered resolution to a problem examined piecemeal in the reviewed literature: how to combine multi-agent ML, explainability, and concept-drift retraining in a coherent, auditable system. It shows that the trade-off between agent autonomy and pipeline auditability has a designed solution through typed message passing, and that explainability can serve as an active control signal without forfeiting its information role. The empirical dominance of VIX-related features across a diverse asset universe justifies structurally privileging macroeconomic context in such protocols.

Practical implications. Checksummed envelopes provide a detailed trace of all inter-agent decisions reviewable by a regulator without pipeline re-execution. SHAP attributions and per-run per-asset reports satisfy the GDPR Article 22 right

to explanation [11]. Formal validation gates and the persisted `bus_history.jsonl` satisfy MiFID II algorithmic-trading audit requirements [10]. The retraining buffer minimises unnecessary computation and model churn. Near-zero coordination overhead ($< 1\%$ of runtime) confirms that these properties are achievable without compromising performance relative to a well-designed uncoordinated pipeline.

Regional relevance. FRHC and HSBK.IL were included not merely for geographic diversity but because Kazakhstan's developing regulatory environment for algorithmic trading places a premium on auditability over raw predictive power. FRHC delivered the largest walk-forward accuracy gain under DMACCP (+3.3 pp), consistent with frontier-market volatility and distribution-shift conditions that stress-test the protocol's adaptive mechanisms. This confirms that the protocol's properties hold under asset dynamics statistically distinct from the developed-market benchmarks dominating the multi-agent finance literature.

6.6 Future Work

Seven research directions follow directly from the limitations stated in Section 6.4 and the open questions raised in Chapter 5, listed in descending order of methodological priority.

Multi-period longitudinal evaluation. Running the protocol over several non-overlapping five-year windows (e.g. 2005–2010, 2010–2015, 2015–2020) would yield a distribution of DMACCP–No-MAS differences rather than a single-window estimate, converting the current design-validation claim into a robustness claim.

Direct reimplementation of an LLM-coordinated baseline. The qualitative comparison of Section 4.13 is limited by the different evaluation universes and time periods of the surveyed frameworks. Reimplementing one of those frameworks (e.g. Sleipnir [18] or TradingAgents [20]) on the same eight-asset universe over the same evaluation window would provide a commensurable accuracy comparison.

Regime $\times$ class-balance joint conditioning. The JPM case (Section 5.2.3) demonstrates that the current regime adjustment table is under-specified. Extending the conditioning to a regime $\times$ class-balance grid—and repeating the ablation experiment on the affected assets—would address this identifiable limitation.

Alternative attribution methods for the SHAP floor. Replacing the SHAP floor filter with permutation importance would ensure the filter fires under low-signal conditions for tree-based models, because permutation importance directly measures out-of-sample predictive power rather than path traversal frequency.

Finer-grained regime classification. The weakness of the `trending_up` label (Section 5.2.5) suggests that a single label for a five-year period obscures within-period regime transitions. Applying a change-point detection method (e.g. PELT or BOCPD on rolling standard deviation) to identify sub-period labels would yield a more operationally meaningful regime structure.

Model-switch-aware drift detection. Three assets changed best-model families between runs (HSBK.IL, JPM, XOM). The current drift measure interprets a model-family switch as structural drift, which conflates two distinct phenomena. Adding a condition that compares SHAP rankings only within the same model family, and flags model-family switches as a separate event type, would improve the orchestrator's situational awareness.

Bus-level integrity enforcement. Currently, checksum verification is client-enforced at the recipient. Embedding verification in the bus-level `receive()` method would prevent any agent from consuming a forged envelope before discovering that the checksum does not match—a minor engineering change with a significant security benefit.

6.7 Closing Remarks

This dissertation designed, formalised, implemented, and evaluated a deterministic coordination protocol for multi-agent financial machine learning. It addressed three deficiencies in the existing literature: the absence of typed, integrity-verified inter-agent communication; the treatment of explainability as a passive post-hoc diagnostic rather than an active control signal; and the use of single-observation retraining triggers that conflate structural drift with noise. The reference-run evidence supports the protocol's core design claims: deterministic and reproducible coordination, a 66.7% reduction in unnecessary retraining events on the ablation candidates, effective SHAP attribution as a cross-run control signal with the protected macro-context set aligning with empirical SHAP dominance, and predictive accuracy consistent with the non-coordinated baseline. The limitations are equally explicit: single-window validation, eight-asset universe, under-specified regime conditioning for imbalanced assets, and a SHAP floor component whose scope is narrower than initially described.

The significance of this work lies in what it enables rather than what it decides. A protocol with formal reproducibility guarantees, a complete audit trail, and a closed-loop explainability layer is a prerequisite for deploying multi-agent financial AI within regulated institutions—but it is not the deployment itself. The future-work agenda outlined in Section 6.6 is the research programme required to move from that prerequisite to that deployment, and this dissertation establishes the starting point for that programme.

Bibliography

[1] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

[2] C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

[3] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne. Machine learning strategies for time series forecasting. *Lecture Notes in Business Information Processing*, 138:62–77, 2013.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] N. Bussmann, P. Giudici, D. Marinelli, and J. Papenbrock. Explainable machine learning in credit risk management. *Computational Economics*, 57(1): 203–216, 2021.

[6] C.-P. Chan. A SHAP-based comparative analysis of machine learning model interpretability in financial classification tasks. 2024.

[7] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[8] S. Dewansh. Policy convergence and explainable stability in orchestrated multi-agent LLM trading frameworks. *International Journal of Advanced Engineering, Management and Science*, 12(2), 2026. doi: 10.22161/ijaems.122.8.

[9] B. Ergenç and M. S. Aktaş. Sector-specific financial forecasting with machine learning algorithms and SHAP interaction values. 2024.

[10] European Parliament and Council. Directive 2014/65/EU of 15 may 2014 on markets in financial instruments (MiFID II). Technical report, Official Journal of the European Union, 2014.

[11] European Parliament and Council. Regulation (EU) 2016/679 of 27 april 2016 (general data protection regulation), article 22. Technical report, Official Journal of the European Union, 2016.

[12] S. Fatemi and Y. Hu. FinVision: A multi-agent framework for stock market prediction, 2024. arXiv:2411.08899 [q-fin.TR].

[13] D. M. Hawkins, S. C. Basak, and D. Mills. Assessing model fit by cross-validation. *Journal of Chemical Information and Computer Sciences*, 43(2): 579–586, 2003.

[14] X. Li et al. HedgeAgents: A balanced-aware multi-agent financial trading system. In *Proceedings of the ACM Web Conference 2025 (WWW '25)*, 2025. doi: 10.1145/3701716.3715232.

[15] Y. Li, Y. Yu, H. Li, Z. Chen, and K. Khashanah. TradingGPT: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance, 2023. arXiv:2309.03736 [q-fin.PM].

[16] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 4765–4774, 2017.

[17] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles, 2018. arXiv:1802.03888.

[18] X. Qian, S. Chen, and Z. Sun. Sleipnir: Multi-agent based stock operator framework with interpretability. *IEEE Access*, 2026. doi: 10.1109/ACCESS.2026.3664216.

[19] V. Srivastava and R. Sikroria. AI and algorithmic trading: A study on predictive accuracy and market efficiency in fintech applications. *ShodhKosh: Journal of Visual and Performing Arts*, 5(1):1098–1107, 2024. doi: 10.29121/shodhkosh.v5.i1.2024.2797.

[20] Y. Xiao, E. Sun, D. Luo, and W. Wang. TradingAgents: Multi-agents LLM financial trading framework, 2024. arXiv:2412.20138 [q-fin.TR].