

PLUTUS OPEN SOURCE

Breaking Barriers in Algorithmic Trading

An-Dan Nguyen
ALGOTRADE
Ho Chi Minh City, Vietnam
andan@algotrade.vn

Quang-Khoi Ta
ALGOTRADE
Ho Chi Minh City, Vietnam
khoi.ta@algotrade.vn

Duy-Anh Vo
ALGOTRADE
Ho Chi Minh City, Vietnam
anh.vo@algotrade.vn

Abstract—Algorithmic trading has long been an opaque, fragmented domain, guarded by secrecy and built around proprietary systems. In contrast to the open, collaborative evolution in fields like machine learning or software engineering, the algorithmic trading ecosystem has been slow to adopt reproducibility, standardization, and shared infrastructure.

This paper introduces PLUTUS Open Source, an initiative sponsored by ALGOTRADE to reshape this landscape through openness, structure, and collaboration. PLUTUS combines a reproducibility standard, a modular development framework, and a growing suite of community-built reference strategies. The project provides a systematic approach to designing, testing, and documenting trading algorithms, regardless of the user’s technical or financial background.

We outline the motivation behind the initiative, present its foundational structure, and showcase working examples that adhere to the PLUTUS standard. We also invite the broader research and trading communities to contribute, iterate, and help build a transparent and inclusive future for algorithmic trading.

Index Terms—algorithmic trading, reproducibility, open-source, framework, standard

I. INTRODUCTION

Closed doors, secret signals, and proprietary systems have long defined algorithmic trading. While other fields, such as artificial intelligence and data science, have benefited from open-source collaboration, reproducible research, and community-driven innovation, algorithmic trading remains an outlier. Fragmented, opaque, and often hostile to newcomers, the field has mainly progressed in isolation, reinforcing a cycle of exclusivity and secrecy.

PLUTUS Open Source is an effort to change that reality.

Sponsored by ALGOTRADE, PLUTUS is not just another open-source repository. It is a structured, community-first initiative to bring standardization, reproducibility, and accessibility to a field that has traditionally resisted all three. The goal is to reimagine what algorithmic trading could look like if the barriers to entry were lowered, best practices were codified, and high-quality research became both replicable and extensible.

PLUTUS introduces a reproducibility standard, a modular development framework, and a living repository of sample strategies, each fully documented, version-controlled, and aligned with a 9-step development process. But more importantly, PLUTUS is a platform for learning and contribution:

a bridge for programmers without finance backgrounds and finance professionals without engineering skills.

This paper introduces the motivations and structure behind PLUTUS Open Source. We explore its reproducibility framework, present working examples, and outline how contributors can engage with and expand the initiative. If you’ve ever wondered how to build credibility in a field built on secrecy, or how to start contributing meaningfully to its future, this paper is your blueprint.

II. MOTIVATION

Despite algorithmic trading’s technological sophistication, the field remains largely inaccessible to those outside tightly controlled professional circles. The lack of openness is not just cultural; it’s structural. There is no shared framework for development, no common language for collaboration, and no unified pathway for newcomers to learn and contribute meaningfully.

At the same time, the demand for such a structure has never been greater. The rise of quantitative finance, the availability of data, and the global reach of digital markets have drawn a new generation of builders to the space: developers, data scientists, mathematicians, engineers, and finance professionals. Each brings a unique perspective, but also faces unique barriers.

We observe two dominant audience types entering the field:

- **Tech learners:** Individuals with strong programming and analytical backgrounds, often proficient in machine learning, statistics, or systems design. These users can automate workflows and scale infrastructure, but often lack financial domain knowledge, making it challenging to form sound trading hypotheses or interpret market behavior.
- **Finance learners:** Investors, traders, students, and professionals trained in market dynamics, behavioral finance, or investment strategy. These users can form promising ideas, but often lack the engineering skills to test, scale, or validate their approaches algorithmically.

PLUTUS Open Source is designed to bridge this divide.

- For technical users, PLUTUS provides a clear structure, reproducibility standards, and tooling to build and iterate on strategies quickly, accelerating their learning curve and surfacing real signals from noise.
- For finance-native users, it offers ready-made automation infrastructure and reproducible templates, reducing the

friction involved in transforming investment logic into working code.

However, the motivation for PLUTUS goes beyond onboarding. It is about credibility. Reproducibility becomes the cornerstone of trust in a field where performance often hides behind cherry-picked backtests and unverifiable claims. A standardized, community-maintained framework ensures that strategies can be shared, discussed, stress-tested, and improved like code in any mature open-source discipline.

PLUTUS is not trying to replace proprietary edge. It is trying to raise the floor.

The more projects that follow shared processes, disclose testing details, and invite peer scrutiny, the faster the field matures, and the more inclusive, credible, and innovative it becomes.

III. BACKGROUND AND RELATED WORKS

The landscape of algorithmic trading research has historically operated in silos, split between proprietary institutional models and fragmented academic output. Industry systems tend to be closed, inaccessible, and tied to confidential infrastructure. While often innovative, academic studies typically suffer from reproducibility limitations due to unavailable datasets, undocumented assumptions, or missing source code [1] [2].

This is not unique to finance. Across the broader scientific community, a growing body of literature has documented what is now widely referred to as the reproducibility crisis, the realization that many published findings cannot be independently verified, even by experts within the same field [3] [4]. Factors contributing to this problem include poor documentation, selection bias, data dredging, and the lack of incentives for publishing negative or null results.

By contrast, machine learning and data science fields have rapidly evolved through open-source ecosystems. Platforms like TensorFlow [5], PyTorch [6], and scikit-learn [7] have demonstrated how community-driven frameworks, standardized pipelines, and shared benchmarks accelerate progress and lower entry barriers. These projects have also formalized reproducibility norms, enabling collaborative validation and iteration across institutions.

Several efforts have attempted to bring similar openness to quantitative trading:

- QuantConnect [8] and Zipline [9] provide robust infrastructure for backtesting and execution, but their focus is global and generic. They do not enforce reproducibility standards or cater to emerging markets with unique structural constraints.
- Backtrader [10], bt, and other Python libraries offer basic simulation and portfolio modeling functionality, but lack standardization in documentation, results reporting, or research transparency.
- Many academic papers introduce promising strategy formulations without accessible datasets or executable code, limiting their practical value and undermining replicability [11].

More importantly, none of these tools explicitly address the unique needs of localized markets, such as Vietnam's stock exchange, where data availability, liquidity structure, and institutional behaviors differ significantly from U.S. or European contexts. As a result, researchers and retail participants in these markets are left to reverse-engineer infrastructure or rely on closed platforms that offer little transparency or extensibility.

What's missing is a framework that acts more like a protocol than a toolkit, one that enforces structure while remaining flexible, supports localized market conditions, and bridges the gap between raw experimentation and verifiable results.

PLUTUS Open Source is designed to fill that void.

Built from the ground up with reproducibility, modularity, and accessibility as core design principles, PLUTUS provides not only infrastructure but a shared language for strategy research. It draws inspiration from DevOps practices and open science methodology, applying them to algorithmic trading with a focus on regional inclusion and cross-disciplinary participation.

IV. THE PLUTUS INITIATIVE

The PLUTUS Open Source¹ project was launched with a clear, practical vision: to bring openness, reproducibility, and standardization to a field traditionally dominated by secrecy and fragmentation. Sponsored by ALGOTRADE, PLUTUS is not just an infrastructure tool but a cultural initiative aimed at transforming how algorithmic trading research is created, shared, and evaluated.

At its heart, PLUTUS is designed to answer one core question:

What would algorithmic trading look like if it embraced the open-source principles that transformed fields like machine learning and software development?

To answer that, PLUTUS is built around four core objectives:

A. The Four Core Objectives of PLUTUS

1) *Standardize the Practice*: PLUTUS provides a unified framework of tools, design patterns, and naming conventions so developers, researchers, and traders can speak the same language. Making high-quality practices repeatable reduces the learning curve for newcomers and increases interoperability across teams.

2) *Lower the Entry Barrier*: Through structured templates, documentation, and reference implementations, PLUTUS offers a guided on-ramp for people entering the field, whether from finance or tech. The framework enables individuals to go from concept to prototype without reinventing infrastructure.

3) *Enable Reproducibility and Transparency*: PLUTUS introduces clear expectations for project documentation and verification. Each project can be tested and audited using a consistent methodology, reducing ambiguity and reinforcing the credibility of results.

¹<https://github.com/algotrade-plutus>

4) *Promote Innovation and Fairness:* By open-sourcing high-quality strategies, workflows, and educational materials, PLUTUS helps level the playing field. It invites participation from students, independent quants, and researchers around the world, accelerating experimentation and idea exchange across borders and backgrounds.

B. The PLUTUS Standard

The PLUTUS Standard is a set of conventions, documentation requirements, and evaluation metrics designed to ensure that trading research is understandable, testable, and reproducible. It draws from practices standard in open-source software engineering, such as templated README structures, version control discipline, and result reproducibility. It adapts them to the workflow of algorithmic strategy development.

Central to this standard is the PLUTUS Compliance Score, a rubric that grades a project's adherence to reproducibility best practices. Projects are encouraged to present results using a consistent structure, including documented assumptions, configuration files, code execution steps, and performance reporting.

C. The Framework (and Future Platform)

Beyond guidelines, PLUTUS offers an extensible framework to accelerate the research and development lifecycle. This includes:

- Modular pipelines for each phase of the 9-step development process
- Strategy templates that are plug-and-play for backtesting, optimization, and validation
- Integration support for paper trading, performance analytics, and deployment tools

The framework acts as the glue between ideas and implementation, giving both individuals and teams a structured, extensible environment to build within. Over time, this foundation will evolve into a full-featured platform that supports continuous strategy iteration, collaboration, and deployment.

D. The Process and Knowledge Base

Finally, PLUTUS is a process. It promotes not just tools, but a shared way of thinking about research. The initiative provides:

- Educational materials for practitioners and students
- Contributor guides and project templates for new teams
- A growing repository of PROTO sample projects that follow the standard

This knowledge base is continuously expanded by the community, ensuring that PLUTUS remains accessible, current, and useful at every level of experience, from first-time traders to institutional quants.

E. Who PLUTUS Is For

PLUTUS is designed to meet users where they are. It's not exclusive to elite quants or professional developers. It is built for:

- Technologists (programmers, data scientists, statisticians) who want to explore trading but lack domain experience
- Finance professionals (traders, investors, fund managers, students) who want to codify and test strategies at scale
- Educators and researchers looking for a transparent way to publish, validate, and extend their findings

Whether building your first bot or formalizing a multi-strategy portfolio, PLUTUS offers a structure that scales with your ambition.

V. THE REPRODUCIBILITY STANDARD OF PLUTUS

Reproducibility is foundational to credible scientific progress, but it is often lacking in algorithmic trading. Strategy write-ups rarely include code, backtests are frequently non-verifiable, and even well-meaning academic research usually lacks the documentation or environment controls needed to replicate results.

PLUTUS addresses this head-on by introducing a Reproducibility Standard tailored to the realities of trading system development. It sets clear expectations for documentation, execution paths, and result validation. The goal is to make independent verification not only possible but routine.

A. Scope of the Standard

The PLUTUS standard² focuses on reproducibility through Step 7: Paper Trading of the 9-Step Development Process for Trading Strategies³. A project is considered reproducible under PLUTUS if:

- All results can be replicated using the provided source code and instructions
- No direct communication with the author is needed to execute or validate the system
- Reported results in the documentation match the outputs when the code is run independently
- Data sources, processing steps, and parameter settings are fully disclosed

This standard is enforced through a structured repository format, a reproducibility checklist, and an optional scoring system to evaluate compliance.

B. Required Project Structure

Every PLUTUS-compliant repository must contain a *README.md* file that follows a standard structure and provides a full blueprint for reproducing the project's findings. The required sections include:

- Abstract: A summary of the hypothesis, methods, and key results
- Introduction: Motivation, goals, and overview
- Related Work (optional): Prior strategies or concepts informing the project
- Trading Hypotheses (Step 1): Description of the edge being pursued

²<https://github.com/algotrader-plutus/plutus-guideline>

³<https://hub.algotrade.vn/knowledge-hub/steps-to-develop-a-trading-algorithm/>

- Data (Steps 2–3): Collection, processing, and structure
- Implementation (Steps 4–6+): How to run, configure, and test the code
- Backtesting & Optimization: In-sample and out-of-sample result reporting
- Paper Trading (optional Step 7): Results from simulated real-time deployment
- Conclusion (optional): Final observations or future ideas
- References: Datasets, APIs, libraries, or academic sources
- Final Report or Paper (optional): Deeper methodology and results discussion

Each section is not just a placeholder; it is meant to provide concrete, step-by-step instructions for full project replication.

C. Why This Matters

Every repo is its own little universe without a standard, with different structures, inconsistent terminology, and unclear replication paths. PLUTUS fixes that. The goal is not to constrain creativity but to standardize the fundamentals, just like version control, unit testing, and style guides do in software.

Reproducibility builds credibility. It's the foundation for discussion, collaboration, and innovation. The more projects that follow the PLUTUS Standard, the richer and more useful the entire ecosystem becomes.

We do not just want working strategies; we want strategies you can verify, tweak, extend, and build on. That's how fields move forward.

And in trading, moving forward is everything.

D. Bridging into Practice

The next two sections present real-world projects developed under the PLUTUS standard to illustrate reproducibility in action. These are not toy examples. They are fully replicable strategies tested across multiple phases, documented in detail, and shared openly for reuse and adaptation.

Together, they form the foundation of the PROTO series: a growing collection of reference implementations designed to serve as templates for real-world, production-ready trading research.

VI. PROTO: SMART BETA

This section presents a working reference implementation: PROTO: Smart Beta⁴ to bring the PLUTUS standard to life. This strategy exemplifies how reproducibility, structure, and transparency can be applied to a long-term, factor-based investing strategy, making it functional, shareable, testable, and extensible.

PROTO: Smart Beta was developed using the full PLUTUS development process and published with complete documentation and code under the standard repository structure. Its goal is not to outperform every benchmark, but to serve as a clean, minimal template for building and validating systematic investing strategies.

⁴<https://github.com/algotrade-plutus/ProtoSmartBeta>

A. Trading Hypothesis

This strategy is built on classic value investing principles: firms with lower valuation ratios and consistent dividend payouts are more likely to outperform over the long term. Two signals are used to screen the equity universe:

- Low Price-to-Earnings (P/E) Ratio
- Positive Dividend Yield (DY)

The core hypothesis is that by filtering for undervalued and income-generating stocks, one can construct a stable, outperforming portfolio in inefficient or emerging markets.

B. Strategy Rules

At the end of each month:

- All current positions are exited.
- Stocks are screened based on the following:
 - P/E ratio between 0 and 15
 - DY greater than 0.01
- The portfolio is rebalanced equally among all qualified stocks.
- A transaction fee of 0.035% is applied on each buy and sell.
- Risk-free rate is set at 6% annually
- The chosen benchmark is VNINDEX

The rules are intentionally minimal to emphasize transparency and ease of replication. All logic is implemented via parameterized functions and configuration files.

C. Data

- Source: Algotrade internal database
- Period: Jan 1, 2022 – Jan 1, 2025
- Asset Universe: Vietnamese listed equities
- Tools: SQL-based data collection, Python preprocessing

Raw data is filtered for missing values, aligned monthly, and exported as .csv files for backtesting pipelines. Feature columns include financial statement metrics and market prices.

D. In-Sample Backtesting

- Period: Jan 1, 2019 – Jan 1, 2022
- Benchmark: VNINDEX
- Frequency: Monthly rebalancing

Metric	Value
Sharpe Ratio	1.2971
Information Ratio	0.0298
Sortino Ratio	1.7297
Maximum Drawdown (MDD)	-28.28%

TABLE I
IN-SAMPLE BACKTESTING PERFORMANCE

1) *Performance Metrics:* These metrics show strong performance during the training period. The positive Sharpe and Sortino ratios suggest favorable risk-adjusted returns, with adequate downside protection. Although the information ratio remains modest, it indicates that the strategy has some predictive power over the benchmark. The drawdown, while notable, is within acceptable bounds for a monthly-rebalanced equity strategy in an emerging market.

2) *Charts*: The following charts provide a visual overview of the strategy’s in-sample performance. The NAV curve shows the equity trajectory of the strategy, while the drawdown curve illustrates peak-to-trough losses during the period. Output was

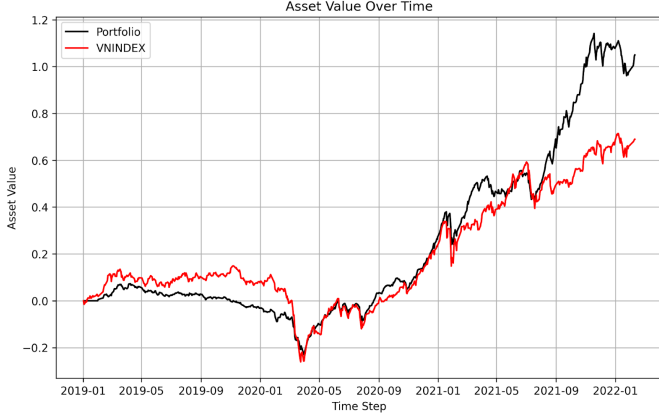


Fig. 1. Smart Beta NAV Chart In-sample

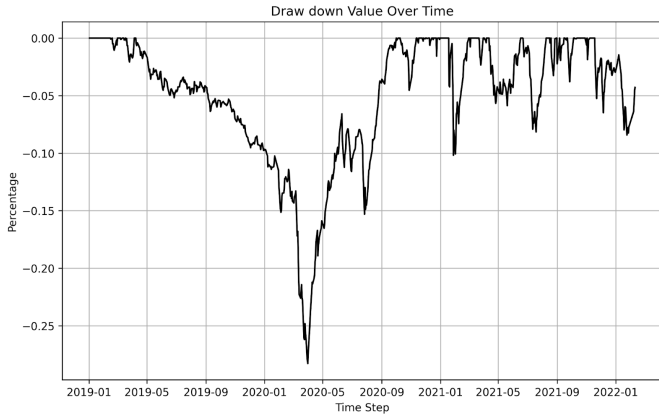


Fig. 2. Smart Beta Drawdown Chart In-sample

tested on multiple runs to ensure determinism.

E. Optimization

The optimization process focused on identifying the best range bounds for the P/E ratio and Dividend Yield to fine-tune the strategy’s screening criteria. These ranges define the eligibility window for stocks at each rebalance.

The optimization was performed using Optuna, a modern hyperparameter optimization framework. The objective function was to maximize the Sharpe Ratio of the resulting portfolio over the in-sample training period, subject to diversification and data sufficiency constraints.

Parameter	Lower Bound	Upper Bound
P/E Ratio	0	15
Dividend Yield	0.01	$+\infty$

TABLE II

SMART BETA: OPTIMIZED PARAMETERS

A stock qualifies if:

- Its P/E ratio is between 0 and 15
- Its dividend yield is greater than 1%

These bounds reflect a traditional value investing logic: excluding speculative or overpriced stocks while filtering out firms without meaningful income payouts. The resulting configuration was used in the out-of-sample backtest.

F. Out-of-Sample Backtesting

- Period: Jan 1, 2022 – Jan 1, 2024
- Configuration: Optimized thresholds from prior step

Metric	Value
Sharpe Ratio	-0.6420
Information Ratio	0.0206
Sortino Ratio	-0.7986
Maximum Drawdown (MDD)	-47.15%

TABLE III

OUT-OF-SAMPLE BACKTESTING PERFORMANCE

1) *Performance Metrics*: The out-of-sample results show a marked decline in performance, with negative Sharpe and Sortino ratios indicating poor risk-adjusted returns and asymmetric downside. However, the strategy remained reproducible and structurally sound across reruns. The Information Ratio, though weak, suggests limited relative signal persistence, meriting future refinements or market condition adjustments.

Although performance declines compared to the in-sample performance, the results show the strength of the chosen Smart Beta strategy when outperforming the benchmark in the out-of-sample periods. That reflects the strategy’s nature. The sharp decrease in Sharpe and Sortino ratios can be rationalized through the benchmark’s downtrend in the same period.

2) *Charts*: Below are the equity curve and drawdown trajectory for the out-of-sample testing window. These charts allow readers to observe changes in volatility, drawdown behavior, and recovery patterns post-optimization.

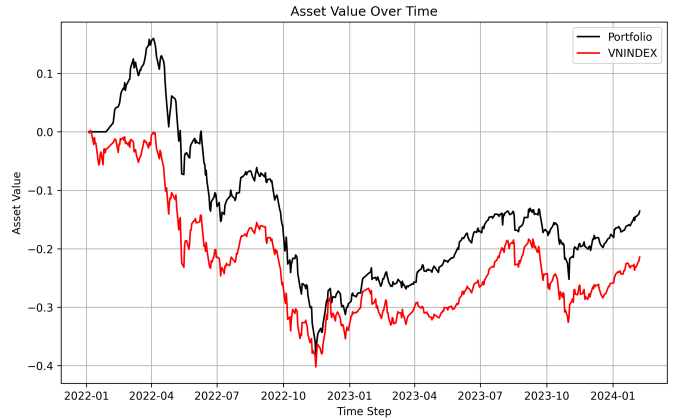


Fig. 3. Smart Beta NAV Chart Out-of-Sample

G. Paper Trading

No live or delayed data paper trading was conducted for this implementation. All experiments were contained within historical simulations.

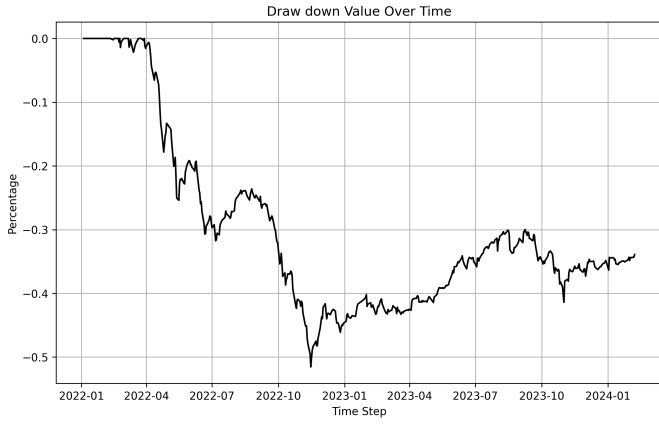


Fig. 4. Smart Beta Drawdown Chart Out-of-Sample

H. Conclusion

PROTO: Smart Beta shows how even traditional strategies can be elevated through structure and transparency. By focusing on simple signals, modular code, and thorough documentation, the strategy becomes a tool for learning, not just a performance product.

As part of the growing PROTO series, it provides a foundational template for building and refining Smart Beta portfolios using the PLUTUS standard.

VII. PROTO: MARKET MAKER

While PROTO: Smart Beta demonstrates PLUTUS in the context of long-term portfolio strategies, PROTO: Market Maker⁵ shifts focus to the microstructure domain, where trading decisions occur in seconds, not months.

Market making is one of the most essential forms of algorithmic execution. It involves continuously posting buy and sell limit orders, capturing spread while managing inventory risk. PROTO: Market Maker exemplifies how even low-latency strategies can be built and shared using the PLUTUS standard.

A. Hypothesis

The strategy operates under the assumption that inventory-aware quoting, adjusting bid and ask levels dynamically based on exposure, enables greater stability and risk control over time. When inventory becomes imbalanced (long or short), the system alters quote distances to encourage mean reversion toward neutrality.

This mechanism discourages directional drift, improves turnover distribution, and promotes long-term resilience even in volatile conditions.

B. Strategy Rules

The bid and ask prices are determined using the following formulas:

- $bid = mid_{price} - step \times (\max(inventory, 0) \times 0.02 + 1)$
- $ask = mid_{price} - step \times (\min(inventory, 0) \times 0.02 - 1)$

⁵<https://github.com/algotrade-plutus/ProtoMarketMaker>

Where:

- Inventory is the net position
- Step is a configurable distance from mid-price
- Quotes are refreshed every 15 seconds or immediately upon a fill
- Each trade incurs a fee of 0.2 points, deducted from the execution price

This design allows the strategy to rebalance exposure passively without relying on price prediction. Execution is rule-based, deterministic, and traceable.

There are some notes about the current implementation:

- The strategy is currently demonstrated in the VN30F1M instrument. Hence, the fee is based on points (not percentage)
- The initial capital to test the strategy is 500M Vietnamese Dong.

C. Data

- Source: Algotrade internal database
- Period: Jan 1, 2022 – Apr 29, 2025
- Structure: Tick-level close prices, bid-ask spreads, execution logs
- Cleaning: Gaps forward-filled; timestamp normalization applied

This dataset enables near real-time simulation with sufficient market depth granularity to evaluate inventory dynamics and quote placement efficiency.

D. In-Sample Backtesting

- Period: Jan 1, 2022 – Jan 1, 2023
- Execution: Every 15 seconds
- Inventory: No daily reset
- Fee Model: 0.2 points per trade
- Step Size: 1.8

1) *Performance Metrics*: The in-sample results indicate strong risk-adjusted performance under stable market conditions. The high Sharpe and Sortino ratios reflect consistent, high-quality returns with minimal downside volatility. Given the high-frequency nature and inventory exposure, the drawdown of $\sim 19\%$ is acceptable.

Metric	Value
Sharpe Ratio	1.5619
Sortino Ratio	2.3335
Maximum Drawdown (MDD)	-18.91%

TABLE IV

IN-SAMPLE BACKTESTING PERFORMANCE

2) *Charts*: The NAV curve illustrates steady growth over the test period, while the drawdown curve shows moderate but recoverable losses. The inventory plot confirms that the quoting logic effectively managed exposure, frequently returning the strategy to near-neutral inventory.

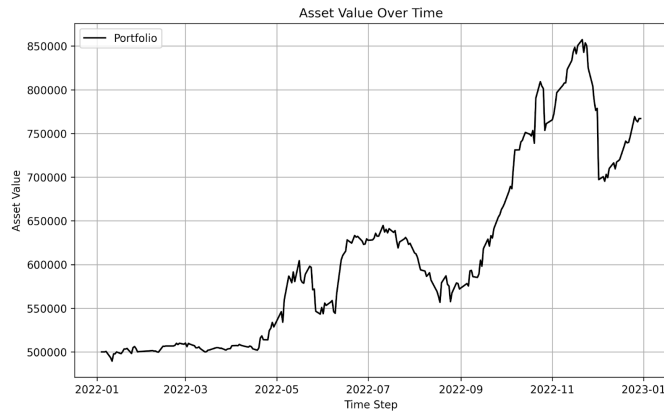


Fig. 5. Market Maker NAV Chart In-sample

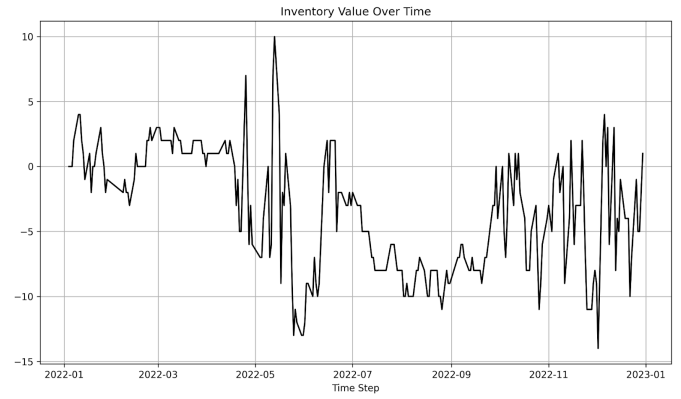


Fig. 7. Market Maker Daily Inventory Chart In-sample

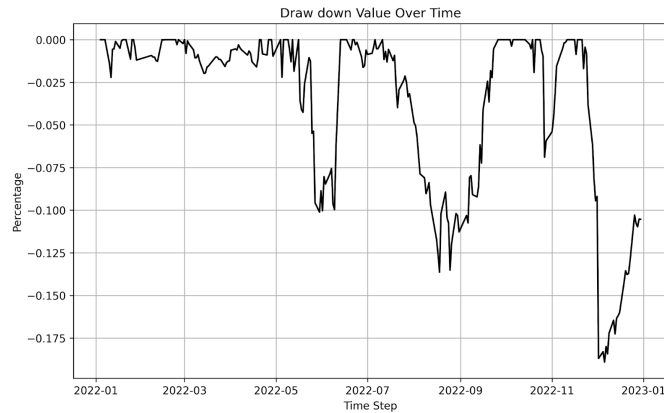


Fig. 6. Market Maker Drawdown Chart In-sample

E. Optimization

The step size parameter controls the trade-off between fill probability and profit per trade.

A randomized optimization routine was run using seed 2025, with the objective function set to maximize the Sharpe Ratio during the in-sample period. This ensured the resulting configuration balanced profitability and risk across varying inventory exposures.

- Step Size: 2.940955612440289

This value balanced stability, fill frequency, and fee tolerance during the in-sample phase.

F. Out-of-Sample Backtesting

- Period: Jan 2, 2024 – Apr 29, 2025
- Config: Step size from prior optimization retained
- Market Regime: Slightly higher volatility, mild directional bias

1) *Performance Metrics*: In the out-of-sample phase, the strategy struggled to maintain profitability. Both the Sharpe and Sortino ratios turned negative, reflecting weaker risk-adjusted returns and more persistent drawdowns. Nevertheless, the system's internal mechanics, such as inventory controls

and execution cadence, remained stable, and the maximum drawdown of $\sim 21\%$ shows that risk was still bounded.

Metric	Value
Sharpe Ratio	-0.0536
Sortino Ratio	-0.0673
Maximum Drawdown (MDD)	-21.37%

TABLE V

IN-SAMPLE BACKTESTING PERFORMANCE

2) *Charts*: The NAV curve shows a mostly sideways and slightly deteriorating trajectory. The drawdown curve reflects choppy equity performance, while the inventory plot remains stable, suggesting that the quoting logic still functioned as designed even under less favorable market conditions.

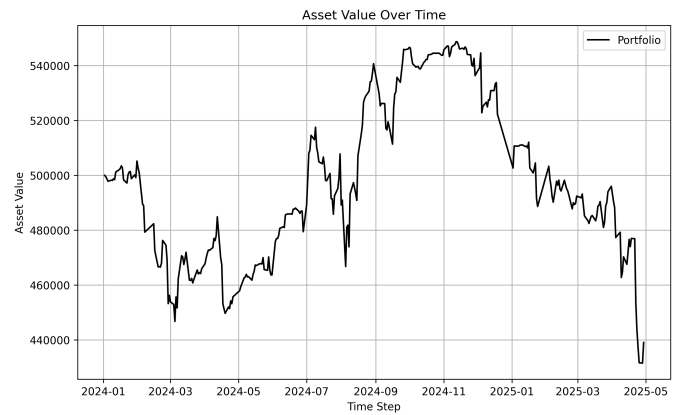


Fig. 8. Market Maker NAV Chart Out-of-Sample

G. Paper Trading

No live market deployment has been conducted yet. This strategy is currently validated only through historical simulation.

H. Conclusion

PROTO: Market Maker illustrates that reproducibility can extend into the domain of fast-moving execution strategies. A PLUTUS-compliant implementation can make trading logic

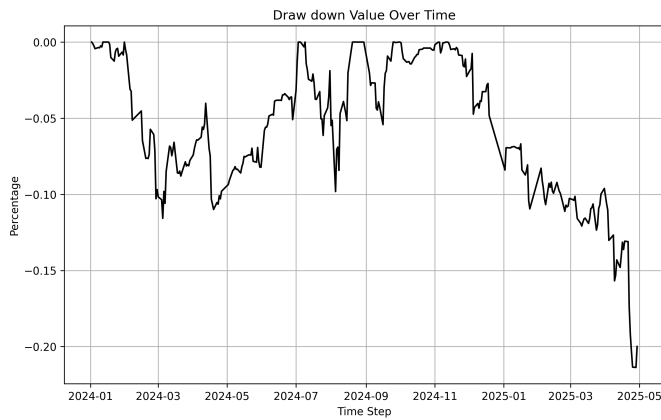


Fig. 9. Market Maker Drawdown Chart Out-of-Sample

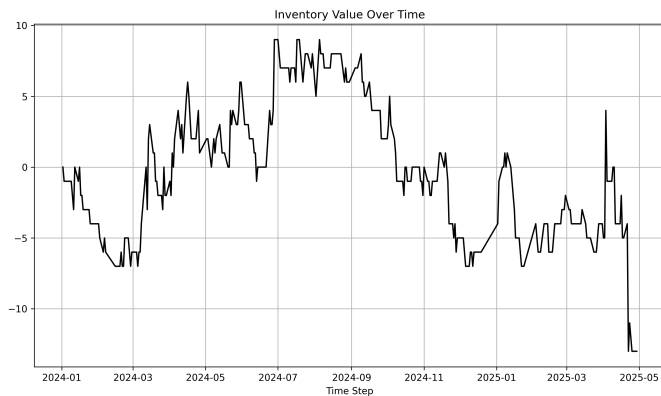


Fig. 10. Market Maker Daily Inventory Chart Out-of-Sample

transparent, measurable, and extensible even when faced with tick-level complexity and sensitive parameters.

Together with PROTO: Smart Beta, this strategy rounds out the initial PROTO series, a growing suite of reference implementations designed to serve as launchpads for deeper exploration and collaboration.

VIII. CONTRIBUTING TO PLUTUS: BUILDING TOGETHER

PLUTUS is not just a framework. It is a movement. Like any open-source ecosystem, its long-term success hinges not on a single company but on the collective effort of contributors worldwide.

This paper presents PLUTUS as a structured, reproducible, and transparent approach to algorithmic trading research. But its full potential will only be realized when builders of all backgrounds engage with it by using, extending, testing, and evolving the ecosystem.

Whether you are a quant developer, a student writing your first strategy, or a finance professional looking to scale your ideas, PLUTUS has a role for you.

A. Why Contributions Matter

The problems PLUTUS addresses, fragmentation, irreproducibility, and opacity, are not just technical. They are sys-

temic. Fixing them requires more than software. It requires a culture change.

When you contribute to PLUTUS, you help:

- Raise the standard for trading research quality
- Make learning and collaboration easier for others Create momentum around open best practices
- Expand the diversity of ideas and strategies available in the field

In short, you don't just add code. You help reshape the ecosystem.

B. Ways to Contribute

1) *Develop PLUTUS-Compliant Projects*: Apply the reproducibility standard to your algorithmic strategies. Use the README.md structure, follow the 9-step methodology, and report your results. This is the easiest and most impactful way to participate, by showing what good looks like.

2) *Contribute to the PLUTUS Framework*: Help improve the shared infrastructure. You can:

- Fix bugs or optimize components
- Add new strategy templates, data loaders, or evaluators
- Improve documentation and developer onboarding
- Integrate external tools, APIs, or execution engines

Think of this as DevOps for quantitative finance.

3) *Refine the PLUTUS Standard*: The reproducibility rubric isn't static. You can:

- Propose improvements to the documentation structure
- Suggest better evaluation metrics or compliance scoring
- Stress-test edge cases through unconventional strategies

The standard should evolve alongside the community.

4) *Improve Sample Projects*: The PROTO series is where many newcomers start. You can:

- Add test cases
- Expand across markets or asset classes
- Fix broken steps or outdated dependencies
- Translate documentation for broader access

These are low-barrier, high-impact contributions.

5) *Join the Conversation*: Collaboration doesn't just happen in code. Join forums, review pull requests, host study groups, write explainers, or mentor others entering the field.

PLUTUS is a tool, but it's also a conversation. And conversations need participants.

C. From Users to Stewards

Every open project begins with a small group of maintainers. But the best ones outgrow them. Our goal with PLUTUS is to create a system that scales beyond us.

By contributing today, you're helping build:

- A globally accessible body of trading knowledge
- A shared protocol for evaluating and exchanging ideas
- A foundation for more credible, inclusive, and accelerated innovation

Whether you publish your first backtest or are architecting multi-strategy portfolios, your voice matters. The future of algorithmic trading doesn't belong behind paywalls or in black boxes. It belongs to the people who built it together.

IX. LOOKING AHEAD

PLUTUS Open Source began as a response to the silence, the locked doors, the unverifiable results, the isolated notebooks. But it has quickly become something much more: a blueprint for how algorithmic trading can evolve when openness becomes a first principle.

This paper presents PLUTUS as a reproducible, community-driven framework for algorithmic trading research. It introduces a practical standard, a modular development process, and a growing body of reference projects, all designed to encourage transparency, rigor, and collaboration in a domain that has long resisted all three.

But PLUTUS is not finished. It is barely beginning.

The roadmap ahead includes:

- Expansion of the PROTO series to cover more strategy classes: arbitrage, grid trading, volatility overlays, multi-asset models, and more
- Development of a hosted platform for strategy testing, compliance scoring, and paper trading integration
- Once strategies have been developed and rigorously validated using the PLUTUS platform and standard, they can be seamlessly integrated with any brokerage that supports PLUTUS Open Source, allowing practitioners to deploy and realize their discovered alphas with minimal friction
- Creation of learning tracks for both technical and financial audiences, helping more people break into the field with confidence
- Continued refinement of the standard as community feedback grows

Most importantly, PLUTUS will grow through participation.

This project is for you if you have ever struggled to verify a strategy, explain your logic, or find a credible path into the field. If you've ever wanted to teach, share, or build something you're proud of, this community is for you.

Let us build the future of algorithmic trading, not in secret, but in the open.

REFERENCES

- [1] D. H. Bailey, J. M. Borwein, M. López de Prado, and Q. J. Zhu, "The probability of backtest overfitting," *Journal of Computational Finance*, vol. 20, no. 4, pp. 39–69, 2014.
- [2] J. P. Ioannidis, "Why most published research findings are false," *PLoS Medicine*, vol. 2, no. 8, p. e124, 2005.
- [3] O. S. Collaboration, "Estimating the reproducibility of psychological science," *Science*, vol. 349, no. 6251, p. aac4716, 2015.
- [4] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, vol. 533, no. 7604, pp. 452–454, 2016.
- [5] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems." <https://www.tensorflow.org>, 2016.
- [6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [8] QuantConnect, "Quantconnect platform." <https://www.quantconnect.com>, 2024.
- [9] Quantopian, "Zipline backtesting library (archived)." <https://www.zipline.io>. Accessed: 2024-05-20.

- [10] Backtrader, "Backtrader documentation." <https://www.backtrader.com>, 2024.
- [11] M. López de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.