Final Report: Forecasting Bitcoin Transaction Fees

Partner: Trilemma Foundation

Jenny (Yuci) Zhang, Tengwei Wang, Ximin Xu, Yajing Liu

2025-06-25

Table of contents

1 Executive Summary

1.1 Subsection (Use Two Hashes)

2 Introduction

2.1 Subsection (Use Two Hashes)

2.2 Subsection (Use Two Hashes)

3 Data Science Techniques

3.1 Subsection (Use Two Hashes)

3.2 Subsection (Use Two Hashes)

3.3 Subsection (Use Two Hashes)

3.4 Subsection (Use Two Hashes)

4 Data Product and Results

4.1 Overview of the Data Product

This data product directly supports Trilemma Capital's mission of serving industry talent and advancing Bitcoin infrastructure through data science, both educational and technical. Its design intentionally tailors to three core audiences. General users and institutions can rely on the 24-hour forecasts to plan transactions and reduce fee costs. Learners and educators receive a transparent, step-by-step walkthrough of Bitcoin-fee forecasting and time-series methodology. Industry experts and partners see infrastructure-grade modeling practice embodied in a modular pipeline and well-documented repository. The product is purposefully modular. Jupyter notebooks guide users through EDA, modeling decisions, and final TFT results. Python scripts implement a structured pipeline for reproducible experiments and easy re-training on new data. Finally, the open-source GitHub repository—with clear documentation—enables collaboration, scalability, and long-term extensibility.

4.2 Results

Model comparison tables and the forecast plots illustrate how predictive fidelity improves as we progress from classical statistics to deep learning. Baselines such as HWES and SARIMA track the day-level seasonal drift but miss the sharp dips and spikes that dominate the fee landscape, as shown in Figure 1.

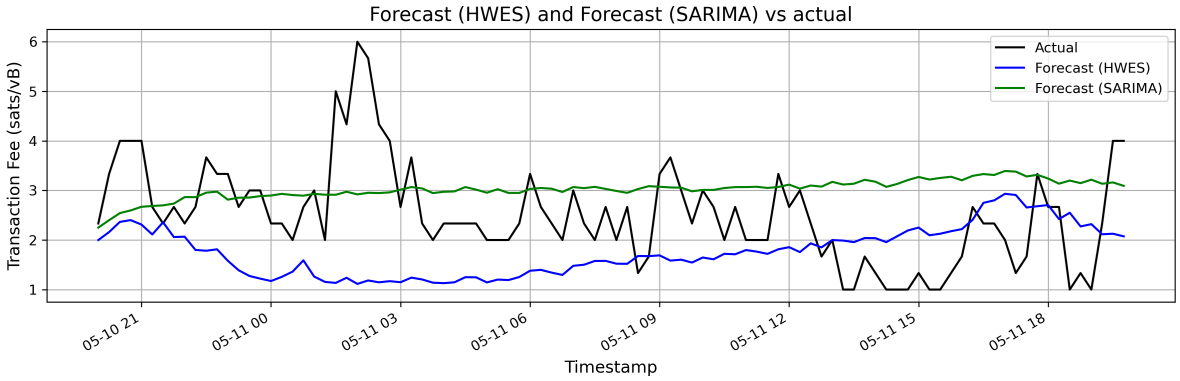

Figure 1: HWES and SARIMA forecasts vs actual fee (test set)

Prophet, with its flexible trend and built-in seasonality terms, improves the global fit but still smooths over most intraday spikes, yielding a custom loss of about 2.43 and an RMSE just about 1.34. XGBoost, which ingests engineered lags and mempool signals, pushes average error lower than any of the purely statistical models. This model yields an RMSE of 1.04, but continues to understate high-frequency volatility, as shown in Figure 2. The custom loss of XGBoost is 2.20, indicating that while the model captures general trends, it still struggles to fully account for the sharp fee spikes and intraday volatility present in the data.

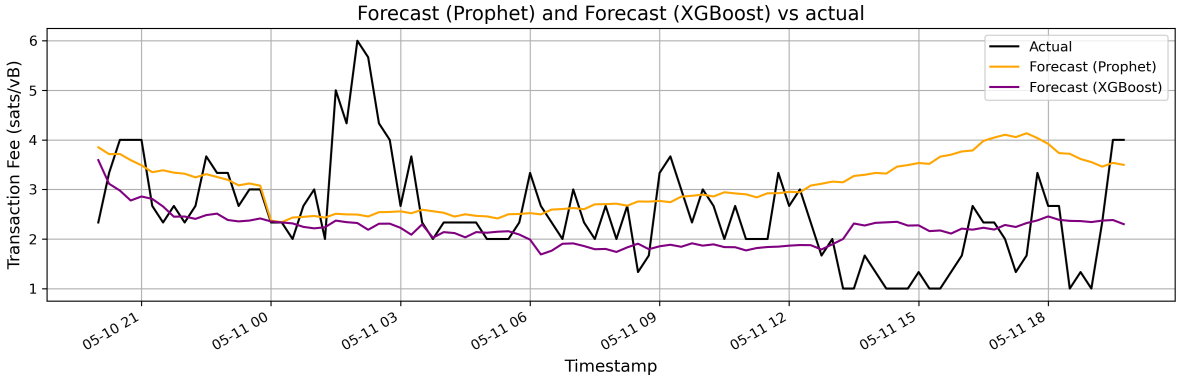

Figure 2: Prophet and XGBoost forecasts vs actual fee (test set)

The neural models represent a meaningful shift in modeling capacity. While DeepAR introduces sequential awareness through recurrence, it falls short of Prophet and XGBoost in short-term fee tracking, with a custom loss of 2.55 and RMSE of 1.15. It is the TFT that best synchronises with both the amplitude and timing of sudden fee surges (darkcyan versus black traces in Figure 3).

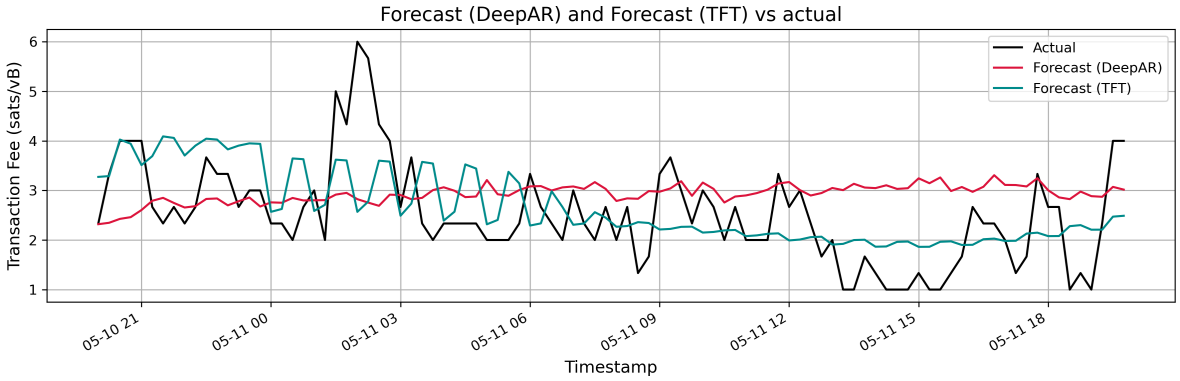

Figure 3: DeepAR and TFT forecasts vs actual fee (test set)

Quantitatively, TFT reduces the bespoke volatility-aware loss from about 2.20, the lowest among the baselines, to 1.78, representing a 19% improvement. It also brings RMSE down from the 1.04–1.44 range to 0.94, a relative gain of 9%. MAE, MAPE, and distribution-shape penalties follow the same pattern, confirming that TFT strikes the strongest balance between overall accuracy and sensitivity to congestion-driven shocks.

4.3 Strength and Competitive Edge

The product's chief strength is that it predicts rather than reacts. Where mainstream tools like estimatesmartfee or Mempool.space publish heuristics for the next few blocks, this project delivers reproducible code that extends the horizon to an entire day and outputs tiered curves—fastest, half-hour, hourly, economy, and minimum—so users can weigh urgency against cost. The product is fully transparent: every notebook, script, and utility function is openly shared on GitHub and documented with detailed inline comments. This openness allows peers to audit modeling assumptions, reproduce plots, and validate performance metrics with ease. Its modular design further reinforces this transparency. Developers can experiment freely—by changing the loss function, introducing new input features, or retraining a specific model—without disrupting the broader pipeline. The educational structure of the notebooks sets this product apart. It not only walks newcomers through key time-series modeling concepts and trade-offs but also provides experienced users with a clear path to automated, command-line execution. Together, these qualities establish Trilemma not as a provider of short-term

heuristics, but as a builder of forward-looking, evidence-based infrastructure for the Bitcoin ecosystem.

4.4 Limitations

Despite its strengths, the product faces several limitations. First, the models do not yet incorporate real-time external signals, such as exchange outflows or policy announcements. This is making them less responsive to abrupt market events that drive sudden fee spikes. Second, The loss function uses fixed weights, which limits its ability to adapt dynamically to changing market regimes. Third, predictive intervals are not yet implemented, which may reduce confidence for risk-sensitive users. Fourth, the deep learning models, particularly transformers, require GPU acceleration or cloud infrastructure, which can raise the barrier for experimentation and increase the cost of continuous deployment. Forecast performance will also degrade over time unless models are periodically retrained to reflect evolving network conditions. Lastly, while the notebooks are well-documented and educational, the absence of a live dashboard or public API limits accessibility for non-technical users, who would need to develop a custom interface to integrate the forecasts into practical workflows.

4.5 Future Directions

These limitations above, however, reflect intentional trade-offs made to balance flexibility, transparency, and cost. Keeping each model in a separate script makes training costs transparent and allows teams to execute only the components they need. This modular design supports flexible development but comes at the expense of centralized orchestration, such as a unified Makefile. Looking ahead, enhancements like adaptive loss weighting, uncertainty quantification, and real-time signal ingestion could improve resilience to edge cases. However, each would introduce added computational cost and complexity. While a real-time API remains an appealing long-term goal, the high cost of hosting transformer models continuously makes it impractical today. In the meantime, scheduled batch forecasts offer a pragmatic balance between accuracy, interpretability, and operational efficiency.

5 Conclusion and Recommendations

5.1 Subsection (Use Two Hashes)

5.2 Subsection (Use Two Hashes)

6 Appendix

6.1 Terminology

Term	Definition
Bitcoin	Unit of currency is called "bitcoin" with a small b, and system is called "Bitcoin," with a capital B. "bitcoin" is a virtual currency (cryptocurrency) designed to act as money and a form of payment outside the control of any one person, group, or entity (i.e. decentralized).
Bitcoin Address	"1DSrfJdB2AnWaFNgSbv3MZC2m74996JafV" An encoded base58-check version of a public key 160-bit hash consists of a string of letters and numbers. Think of it analogous to an email address when sending someone an email.
Blockchain	A decentralized digital ledger that records transactions across a network of computers, making it transparent, immutable, and resistant to tampering. Technology used by Bitcoin.
Fees	The sender of a transaction often includes a fee to the network for processing the requested transaction. Most transactions require a minimum fee of 0.5 mBTC (millibitcoin) = 0.0005 BTC. Typical unit measurement in satoshi/bytes.
Hash	A function that converts an input of letters and numbers into an encrypted output of a fixed length. The hash is irreversible, meaning it cannot be decrypted back to the original input. Hashes are used in Bitcoin to create blocks and verify transactions.
Mempool	The bitcoin Mempool (memory pool) is a collection of all transaction data in a block that have been verified by Bitcoin nodes, but are not yet confirmed.
Mining / Miner	A process/network node that finds valid proof of work for new blocks, by repeated hashing.
Node	Refers to blockchain stakeholders and their devices that keep a copy of the distributed ledger and serve as communication points within the network. Major purpose is to verify the validity of the transactions within a particular blockchain.
Proof-of-Work	A piece of data that requires significant computation to find; In bitcoin, miners must find a numeric solution to the SHA256 algorithm that meets a network-wide target, the difficulty target.

Satoshi	The smallest denomination of bitcoin that can be recorded on the blockchain. 1 Bitcoin is equivalent to 100 million satoshis, named after the creator of Bitcoin, Satoshi Nakamoto.

Table 1: Key Terms and Definitions in Bitcoin and Blockchain (Alphabetically Ordered)

7 References