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

Bitcoin transaction fees are highly volatile and event-driven, with annual spending exceeding billions and single-day spikes reaching over $78 million (Harper 2024). Most existing tools provide only short-horizon, heuristic-based estimates for the next few blocks (<1 hour), offering limited foresight for users aiming to optimize transaction cost or timing. This project addresses that gap by forecasting fee volatility up to 24 hours ahead, with the primary goal of identifying high-volatility periods in the fastestFee tier and a secondary goal of estimating fee magnitude within those windows.

Multiple modeling approaches were explored, including classical time series methods, tree-based models, and deep learning architectures. Traditional models captured seasonality but failed to anticipate sharp spikes. The Temporal Fusion Transformer (TFT) demonstrated the best performance, capturing complex dependencies and improving both RMSE and a custom volatility-sensitive loss by 25–35%. The final product is a modular forecasting system that combines narrative-driven notebooks with executable model pipelines and open-source infrastructure for ongoing development. While limitations remain—such as limited historical data, reliance on reactive features, and the computational cost of deep learning— the system provides a practical foundation for long-horizon fee forecasting and contributes to advancing infrastructure in the Bitcoin ecosystem.

2 Introduction

In the Bitcoin network, transaction fees fluctuate sharply due to congestion, shifting incentives, and user behavior. These spikes are driven by irregular, event-based shocks — such as NFT inscription surges, exchange batching, or market volatility — rather than recurring patterns (Harper 2024). Fee data is marked by abrupt jumps and heavily influenced by off-chain events, making short-term fee selection a major challenge for users and infrastructure providers (M. Wang et al. 2025).

Prior work highlights network congestion and transaction complexity as critical drivers of transaction fees (Fan and Liu 2020). However, existing tools like Bitcoin Core's estimatesmartfee (Bitcoin Core Docs n.d.) and Mempool.space offer short-term, reactive guidance based on recent block data (Bitcoin Explorer n.d.), typically covering only the next 1–6 blocks. These methods are opaque, insensitive to external drivers of volatility, and provide no insight beyond the immediate horizon. Most public tools and research treat fee prediction as a static regression problem, overlooking the timing and structure of volatility — and leaving a critical planning gap unaddressed (Li et al. 2020).

Our exploratory analysis reinforced the limitations of these approaches, revealing patterns such as daily cycles, short-term dependencies, and sudden spikes. To address the forecasting gap, we reframe fee prediction as a volatility-first, time-sensitive problem. For longer horizons, the timing and shape of fee spikes are often more actionable than precise point estimates. To capture these dynamics, we evaluate a diverse set of models: Holt-Winters Exponential Smoothing (HWES) (Holt 1957; Winters 1960), SARIMA (Box et al. 2015), XGBoost (Chen and Guestrin 2016), Prophet (Taylor and Letham 2018), DeepAR (Salinas, Flunkert, and Gasthaus 2017), and the Temporal Fusion Transformer (TFT) (Lim et al. 2019). These represent a progression from classical time series models to tree-based regressors and deep learning architectures.

We assess these models using both standard metrics (e.g., RMSE, MAPE) and a custom composite loss designed to penalize deviation from spike shape, timing, and volatility. We find that TFT significantly outperforms other models, improving key evaluation metrics by 25–35% over baseline approaches. This demonstrates its ability to model non-periodic fee behavior and support forward-looking decision-making. The remainder of this report is organized as follows: Section III outlines the data science techniques used, including preprocessing, model selection, and evaluation strategy. Section IV presents the data product and results, detailing model performance, intended use, and the system's extensibility. Section V concludes with key takeaways, limitations, and recommendations for future development.

3 Data Science Techniques

This section outlines the core techniques used in our analysis, including data preprocessing, model development, and evaluation strategies that guided our approach to fee forecasting.

3.1 Dataset Overview

We worked with a time series dataset constructed from 5-minute snapshots of the Bitcoin mempool (mempool.space, Figure 1), captured between March 5 and May 2025. The mempool is a real-time queue of unconfirmed transactions waiting to be included in a block, which serves as proof that the transactions have been validated and recorded on the blockchain. Each snapshot captures the network state at a given point in time and includes 61 features such as mempool congestion, transaction volume, block production, mining difficulty, and BTC price. These snapshots collectively form a time series that reflects both blockchain-level activity and evolving market demand conditions—capturing the key drivers of transaction fee dynamics.

Figure 1: Visual illustration of Bitcoin mempool.space activity.

3.2 Feature Preprocessing

To prepare the data, we conducted exploratory correlation analysis between features and the target variable fastestFee, which confirmed that several predictors contained useful signals. This was not used for formal feature selection, but helped validate feature relevance.

To prevent target leakage, we removed a few features (e.g., hourFee) that had extremely high correlation with the target (over 0.97). These were excluded from all models except DeepAR and TFT, where model constraints made removal more complex.

We also applied several feature engineering steps. We first examined the distribution of fastestFee and found it to be highly right-skewed (see Figure 2), with most values clustered at the low end and a few extreme spikes. This heavy skew can hinder model stability and violate common assumptions.

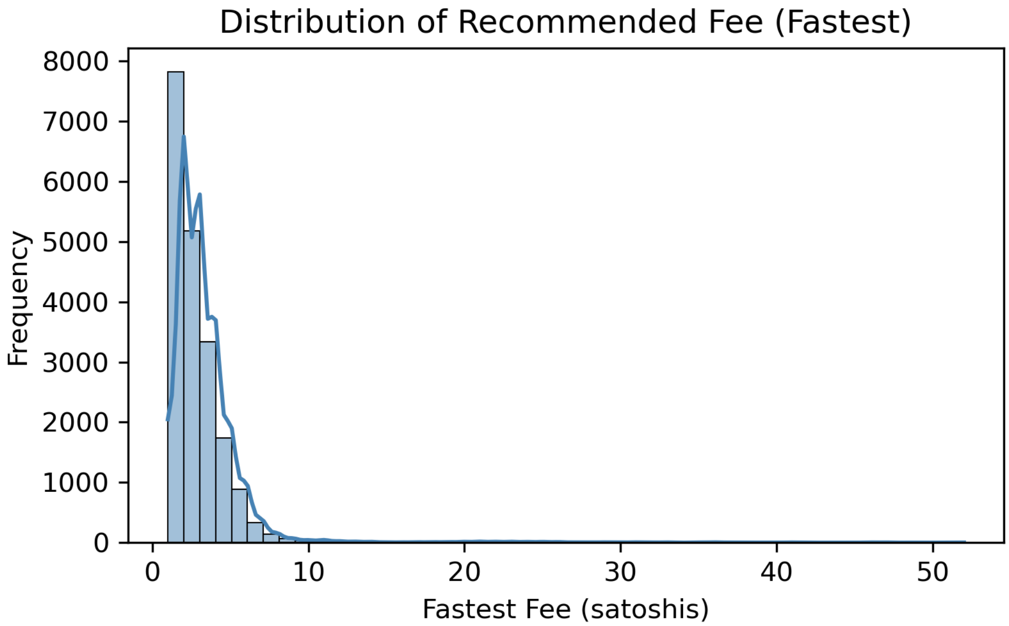

Figure 2: Original distribution of fastestFee shows strong right skew.

To mitigate this, we applied a logarithmic transformation to fastestFee, which compresses large values and helps stabilize variance across the series—making it more suitable for forecasting. Second, we resampled the original 5-minute data into 15-minute intervals to reduce noise and enhance short-term signal clarity.

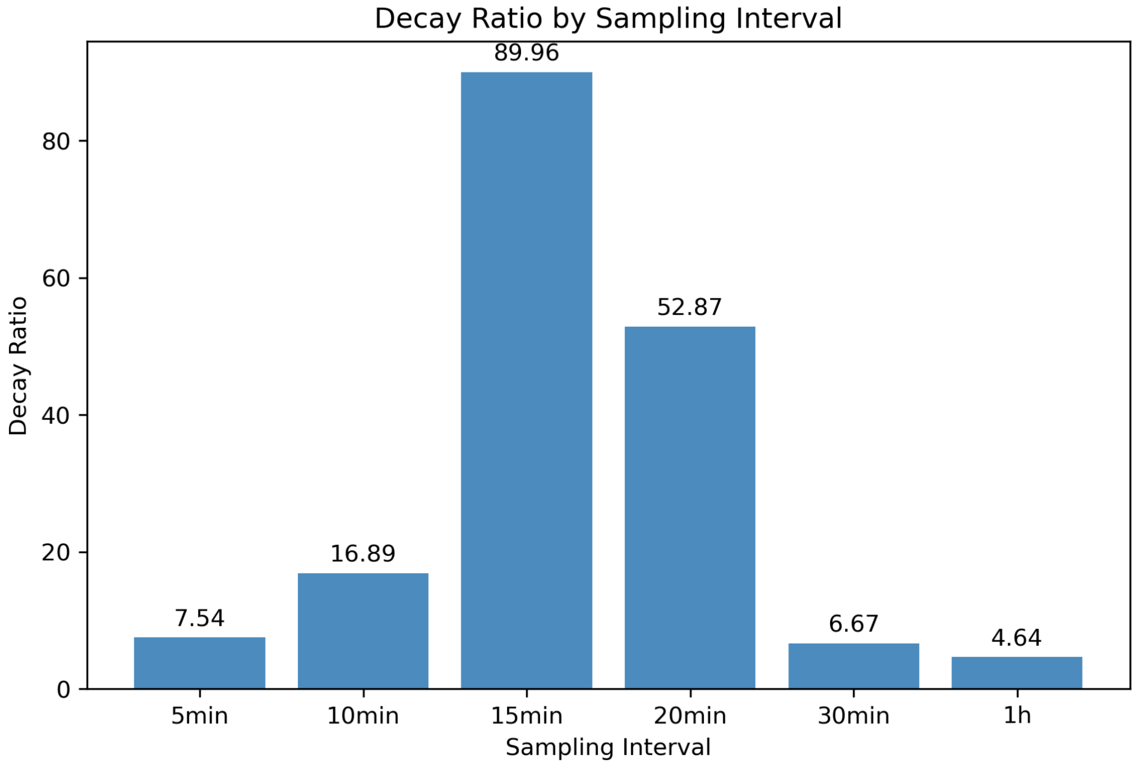

Figure 3: Decay ratio by sampling interval. Higher ratios indicate stronger AR(1)-like structure and better short-term predictability.

This choice was motivated by decay ratio analysis, which evaluates the signal strength of short-term temporal dependencies. As shown in Figure 3, the 15-minute interval yielded the strongest AR(1)-like pattern among tested frequencies, supporting its selection for resampling. Finally, we created lagged variables and rolling aggregations to help models capture short-term temporal dependencies. Finally, we created lagged variables and rolling aggregates to help models capture temporal dependencies. The final feature set retained most original variables except those flagged for leakage, balancing completeness with modeling integrity.

3.3 Models

To understand and anticipate Bitcoin transaction fee rate dynamics, we implemented a sequence of models. Each of the models were selected for its ability to address specific limitations observed in the previous ones. Below, we walk through these choices, results, and where each model fell short.

3.3.1 Dummy Model (Global Median)

We began with a simple dummy model. From Figure 2, it could be illustrated that over 92.5% of observed fee rates fell within a narrow band of 2–3 satoshis/byte, so the median itself was already a strong first guess. Thus, our dummy model always predicted the global median transaction fee rate. Although it had no predictive power, it served as a baseline to measure improvements from more sophisticated approaches. This model completely ignored any temporal or external structure in the data, but offered a useful starting point to quantify how difficult the prediction task really was.

3.3.2 Holt-Winters Exponential Smoothing (HWES)

Given clear seasonality observed in our decomposition analysis Figure 4, Holt-Winters Exponential Smoothing was a natural next step. It captured seasonality fairly well and improved upon the dummy model. However, analysis of the residuals in Figure 5 revealed persistent autocorrelation, suggesting that the model failed to account for important lag effects or hidden patterns beyond periodic behavior.

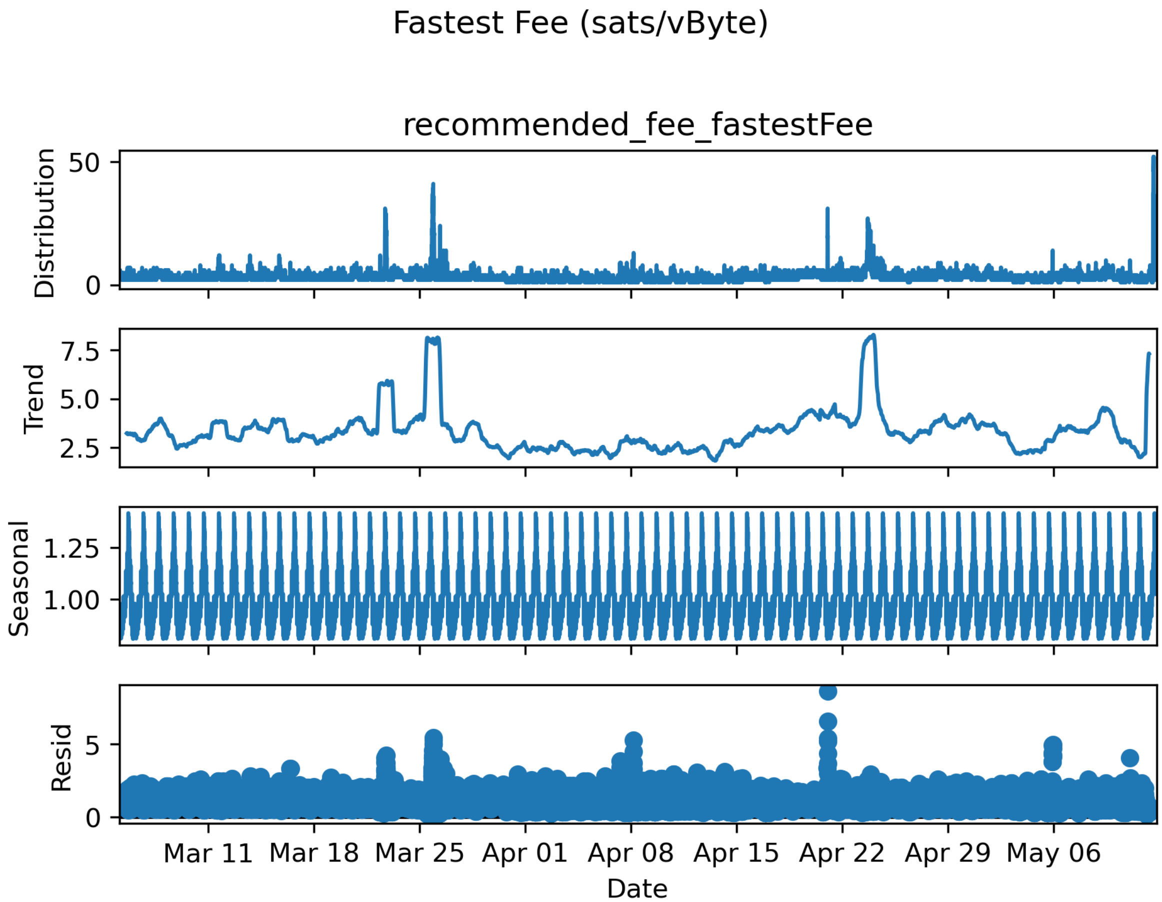

Figure 4: Multiplicative decomposition of recommended fastest fee rate

3.3.3 SARIMA

To address the temporal dependencies missed by HWES, we adopted SARIMA, which supported autoregressive and seasonal differencing components. Because of its capacity to learn from prior values, SARIMA produced a better short-term fit. However, its univariate nature prevented us from including important exogenous features like transaction count, mempool congestion, or size distributions. That limited its practical usefulness in a multi-variable environment.

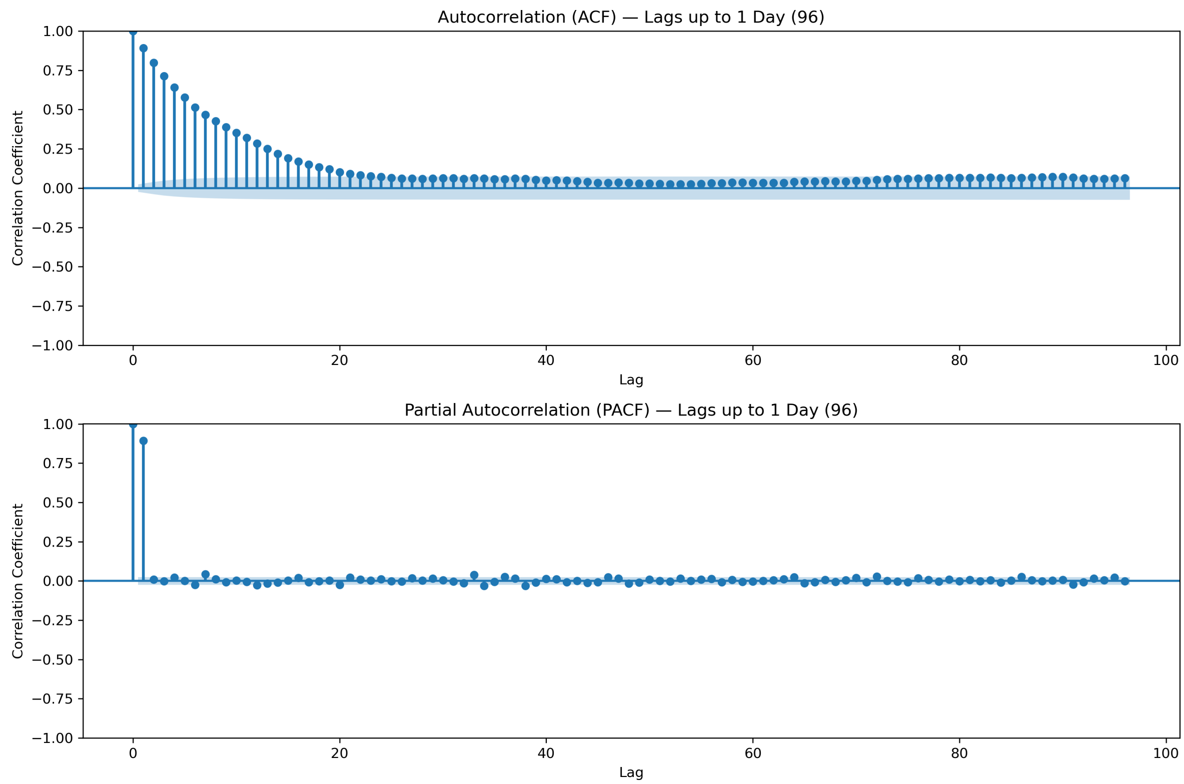

Figure 5: ACF and PACF

3.3.4 XGBoost

To move beyond univariate models, we adopted XGBoost, which allowed us to leverage the full multivariate feature set. Correlation analysis in Figure 6 showed that recommended fastest fee rate was significantly influenced by several concurrent indicators. XGBoost is a powerful tree-based model capable of incorporating a broad array of features. It significantly expanded our input space and enabled non-linear interactions among variables. While its numerical performance improved, especially on average error metrics like MAPE, the model still struggled with volatility. It produced smooth and flat outputs that failed to capture sudden fee spikes. However, those spikes were precisely the events most critical for users.

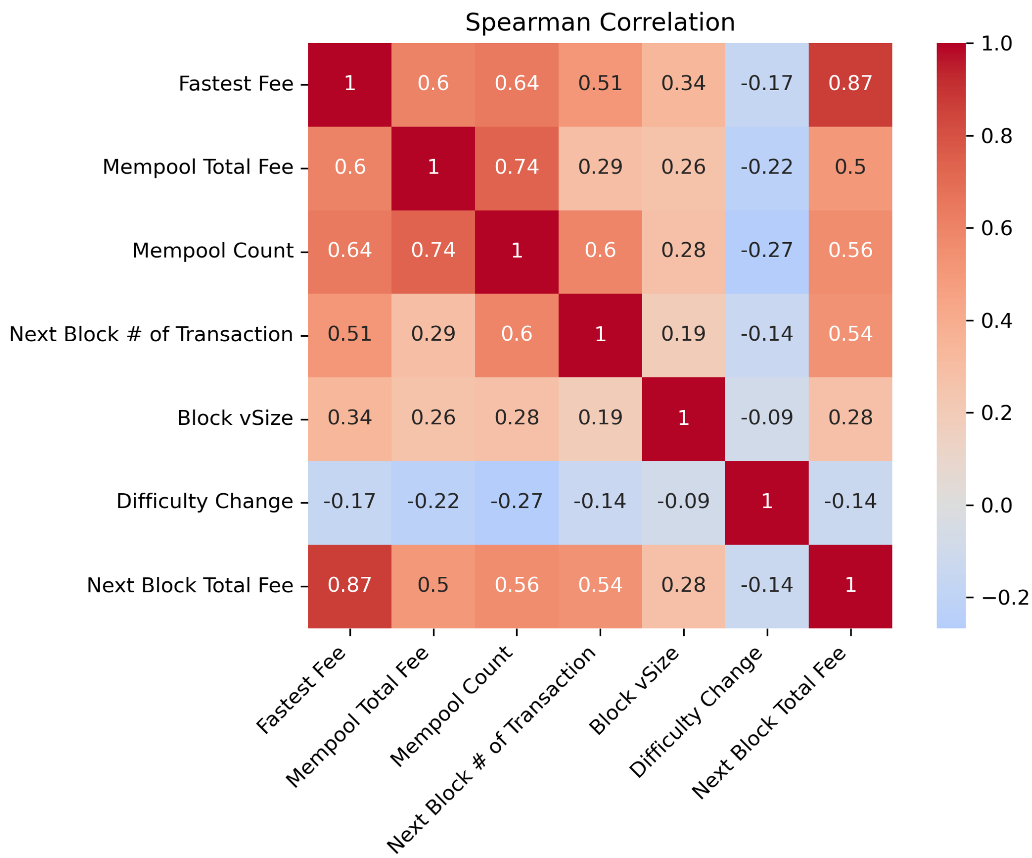

Figure 6: Spearman correlation heatmap

3.3.5 Prophet

We explored Facebook's Prophet model to take advantage of its flexibility in modeling seasonality, changepoints, and custom events. Prophet brought in useful priors for time series with irregular behavior and offered a simple interface for integrating domain knowledge. Unfortunately, it still smoothed over the spikes and underperformed in capturing real-time fee volatility. Its strength in trend estimation did not translate well to our highly reactive use case.

3.3.6 DeepAR

To try more dynamic modeling, we implemented DeepAR, which was an LSTM-based autoregressive forecasting model. In theory, DeepAR should have leveraged temporal context more effectively and handled sequential data better. However, the outputs were unstable and noisy. The results often failed to align with real-world fee movements. The model demonstrated limited generalizability, and its probabilistic forecasts often appeared more random than informative.

3.3.7 Temporal Fusion Transformer (TFT)

Our final and most advanced model was the Temporal Fusion Transformer(TFT). TFT was designed to integrate static covariates, time-varying features, attention mechanisms, and variable selection into a unified deep learning architecture. Among all models, TFT came closest to capturing both overall volatility and individual spike events. It successfully learned temporal dependencies, responded to feature relevance dynamically, and produced the most realistic forecasts. While computationally expensive and complex to tune, its performance and interpretability made it the strongest candidate for this forecasting task.

3.3.8 Considered Alternatives and Limitations

A major limitation of our approach is the reliance on lagged exogenous features (e.g., mempool stats), which limits the model's ability to anticipate fee spikes. One potential improvement is a two-stage setup: first, forecast future values of key features; then, feed those predictions into the fee model for better forward-looking performance.

We did not pursue this path due to both practical and strategic reasons. The partner emphasized focusing on existing features before adding external signals and noted prior attempts at using sentiment data (e.g., scraped news, tweets) yielded poor results. Multi-stage forecasting also risks compounding errors, and with limited time, we prioritized evaluating diverse model architectures over expanding feature pipelines.

3.4 Evaluation Metrics

We used multiple metrics to evaluate model performance, selected to balance interpretability and relevance to fee volatility. MAPE was chosen for its intuitive, percentage-based output, helping stakeholders assess relative accuracy across fee levels. RMSE complemented this by penalizing large errors more heavily, making it better suited for detecting and differentiating sharp fee spikes—crucial for users aiming to avoid overpayment during congestion.

To address the limitations of standard loss functions in modeling Bitcoin fee spikes, we developed a custom composite loss tailored to the problem's volatility-first nature. Traditional losses such as MAE tend to reward average accuracy while under-penalizing models that miss high-volatility patterns or smooth out sudden transitions. Recent literature underscores the need for shape- and time-aware loss formulations in forecasting tasks. Le Guen & Thome introduce a distortion-based loss that aligns predictions with observed temporal patterns (Guen and Thome 2019), Wang et al. demonstrate that custom loss functions enhance spike detection in extreme-value settings (Z. Wang et al. 2024), and Lopez highlights the importance of aligning evaluation with volatility-specific business objectives (Lopez 2001).

Inspired by these, we crafted a loss that combines three components: base error (MAE), volatility mismatch (standard deviation loss), and spike timing deviation (difference in normalized series structure). This formulation explicitly encourages models to preserve both the timing and magnitude of fee surges—crucial in the context of event-driven Bitcoin congestion. A breakdown of the loss components is shown in Table 1.

Component	**Base Loss**	**Std Loss**	**Deviation Error**
Calculation	`y_pred - y_true`	`std_pred - std_true`	`(y_pred - ȳ_pred) - (y_true - ȳ_true)`
Captures	Raw error	Overall volatility mismatch	Dynamic (pointwise) pattern mismatch
Relevance to Spikes	Underweights spikes	Penalizes smoothing	Captures spike timing

Table 1: Breakdown of custom loss function components.

3.5 Stakeholder Impact & Ethical Considerations

By prioritizing volatility and spike timing, our evaluation metrics better reflect the needs of key stakeholders. End users want to avoid high-fee periods, wallet providers need timely and interpretable forecasts, and miners may optimize revenue through better visibility. However, ethical risks exist: users may over-rely on predictions, forecasts may be exploited, and unequal access could widen fee disparities. We mitigate these risks through open access, transparent design, and clear communication of model limitations. Broader concerns like fairness, miner incentives, and malicious mempool behavior remain outside our current scope and merit future attention.

4 Data Product and Results

This section presents results from our analysis, including model performance comparisons and forecast visualizations, followed by a discussion of the final data product—its intended users, applications, and extensibility.

4.1 Results: Model Performance and Forecast Visualization

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 7.

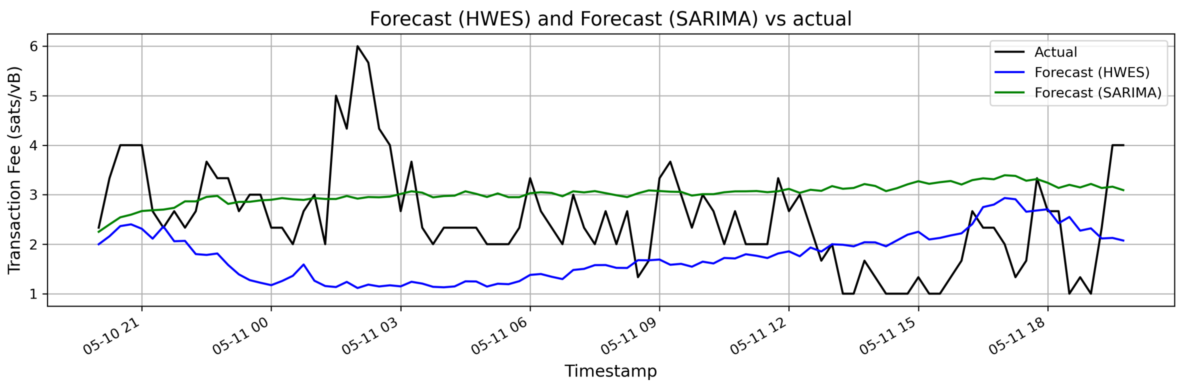

Figure 7: HWES and SARIMA forecasts vs actual fee (test set)

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 8. 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.

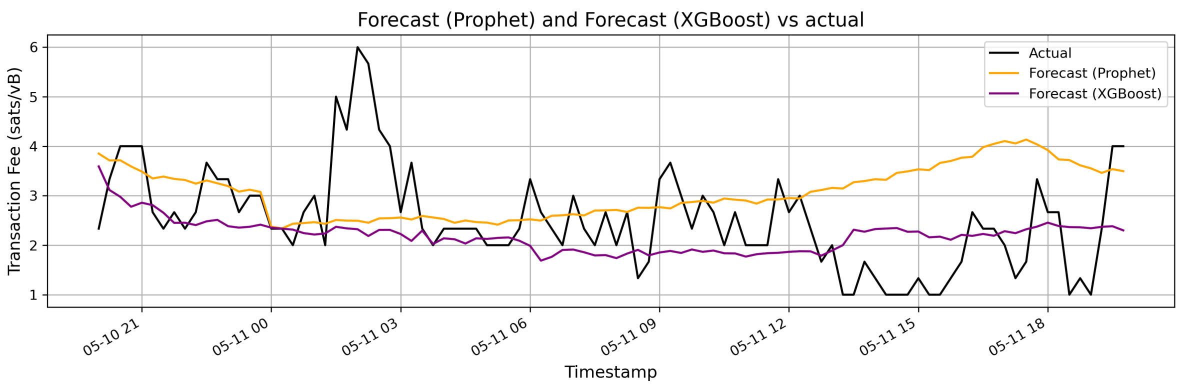

Figure 8: Prophet and XGBoost forecasts vs actual fee (test set)

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 9).

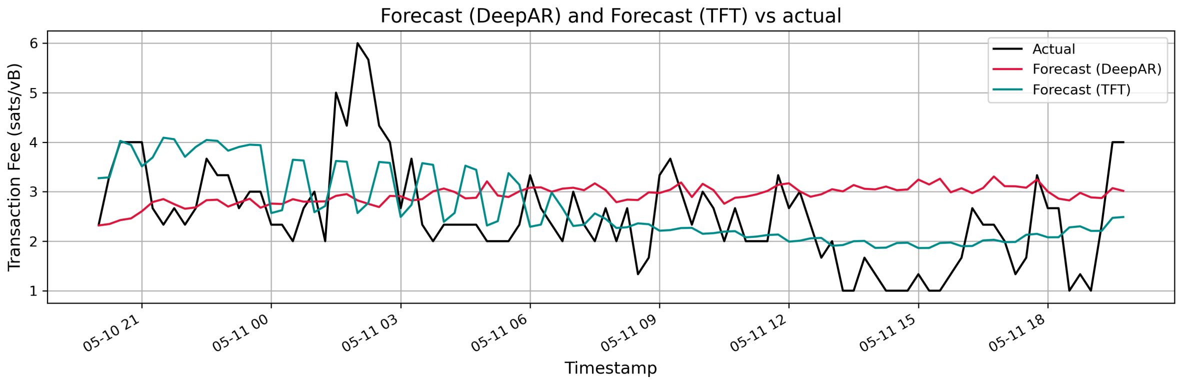

Figure 9: DeepAR and TFT forecasts vs actual fee (test set)

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, as shown in the comprehensive comparison in Table 2.

	TFT	XGBoost	Prophet	DeepAR	HWES	SARIMA
custom_loss	1.78	2.2	2.43	2.55	2.61	2.61
rmse	0.94	1.04	1.34	1.15	1.44	1.2
mae	0.75	0.78	1.01	0.92	1.16	0.97
mape	0.35	0.35	0.58	0.51	0.47	0.55

Table 2: Model performance on test data.

4.2 Data Product Overview

The 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.

4.3 Value Proposition and Justification

Our product addresses the key limitation of current Bitcoin fee tools, which are reactive, opaque, and confined to short-term horizons. Tools like estimatesmartfee and Mempool.space forecast fees only 10 - 60 minutes ahead, offering single-rate suggestions without transparency into volatility drivers or user-specific needs. In contrast, our system produces 24-hour forecasts across multiple urgency tiers, helping users align timing with cost tolerance. We chose Jupyter notebooks to explain modeling decisions and trade-offs in an accessible format. Modular pipeline scripts offer reproducibility and extensibility for advanced users, while an open-source GitHub repository invites transparency and community contribution—aligned with our partner's infrastructure mission.

4.4 Product Strengths and Limitations

The product balances analytical depth with real-world usability. It offers tiered, volatility-aware forecasts that are actionable for users ranging from individuals to technical infrastructure teams. Tiered fee outputs (fastest, economy, minimum) accommodate different urgency and cost preferences, offering flexibility not found in existing one-size-fits-all tools. On the development side, its modular structure promotes extensibility—developers can swap models, tune hyperparameters, or introduce new features without disrupting the pipeline. Full transparency also enables peers to audit modeling assumptions, reproduce visualizations, and verify performance results. The inclusion of a custom loss function aligns model behavior with real-world fee dynamics, supporting better planning around cost and urgency trade-offs.

Nonetheless, the system has limitations that affect both performance and accessibility. From a usability standpoint, deep learning models like TFT demand significant compute resources and are not easily accessible to users without GPUs. More critically, model performance will degrade over time unless retrained regularly to adapt to shifting network dynamics. On the input side, models currently rely on lagging indicators and do not incorporate real-time external signals—such as exchange outflows or policy announcements—making them less responsive to abrupt market events that trigger sudden fee spikes. Additionally, the absence of uncertainty estimates may limit users' confidence in edge cases, and the lack of a real-time dashboard or API could restrict adoption by users expecting more interactive or integrated experiences.

4.5 Design Trade-Offs

We made design choices that prioritize flexibility over complexity. For example, we avoided creating a Makefile to run all models at once due to compute constraints. Probabilistic outputs were also omitted, as current magnitude forecasts are too noisy to support reliable uncertainty estimates. Likewise, real-time API deployment was ruled out due to cost and performance constraints of transformer-based models. Instead, the product supports batch forecasts and modular experimentation as a foundation for future enhancements, such as adaptive loss tuning, hybrid model pipelines, or more scalable delivery formats.

5 Conclusion and Recommendations

This project set out to address the challenge of forecasting Bitcoin transaction fees—a notoriously volatile and difficult-to-predict signal shaped by network congestion, user incentives, and unpredictable events. Recognizing the limitations of existing tools that offer only near-term, reactive guidance, we reframed the problem as one of volatility forecasting over a 24-hour horizon. Through extensive experimentation across statistical models, tree-based learners, and deep sequence models, we found that modern deep learning architectures—particularly the Temporal Fusion Transformer—are well-suited to capturing irregular spike patterns. The resulting system moves beyond average-point estimation to provide actionable foresight, helping users, wallets, and infrastructure providers make more informed decisions around transaction timing.

Despite these gains, several constraints remain. The dataset used spans just two months, limiting exposure to rare but impactful phenomena. Many features that help explain fee behavior—such as mempool congestion or block composition—are available only with a lag, reducing our ability to predict true leading signals. Furthermore, while deep models like TFT are powerful, they require regular retraining and high computational resources, which may constrain deployment or accessibility. Our custom loss function, though aligned with business needs, remains static and could be better tuned to balance volatility sensitivity and accuracy.

Looking forward, future work could explore alternate data sources and modeling strategies that address these constraints. First, longer historical coverage or the inclusion of off-chain indicators may improve predictive range. Second, deeper exploration and adaptive tuning of the custom loss function may further align model behavior with user pain points around volatility and timing. Third, future approaches might benefit from hybrid pipelines—where intermediate signals (e.g., mempool congestion or block inclusion rates) are predicted first and then fed into the final fee forecasting model. Finally, probabilistic forecasting and uncertainty quantification could add value once magnitude prediction improves.

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

References

Bitcoin Core Docs. n.d. "Estimatesmartfee." https://developer.bitcoin.org/reference/rpc/estimatesmartfee.html.

Bitcoin Explorer. n.d. "Mempool.space." https://mempool.space/.

Box, George E. P., Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2015. *Time Series Analysis: Forecasting and Control*. 5th ed. Wiley.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. https://doi.org/10.1145/2939672.2939785.

Fan, Ying, and Xiaotong Liu. 2020. "The Determinants of Bitcoin Transaction Fees: Evidence from the Mempool." *Finance Research Letters* 35: 101289. https://doi.org/10.1016/j.frl.2019.101289.

Guen, Vincent Le, and Nicolas Thome. 2019. "Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models." In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 4191–4203. https://doi.org/10.5555/3454287.3454664.

Harper, Colin. 2024. "Bitcoin Transaction Fees Hit Record Levels After Halving — Here's Why." https://www.forbes.com/sites/colinharper/2024/04/22/bitcoin-transaction-fees-hit-record-levels-after-halving---heres-why/.

Holt, Charles C. 1957. "Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages." *International Journal of Forecasting* 20 (1): 5–22. https://doi.org/10.1016/0169-2070(94)00023-X.

Li, Xiaoqi, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 2020. "A Survey on the Security of Blockchain Systems." *Future Generation Computer Systems* 107: 841–53. https://doi.org/10.1016/j.future.2017.08.020.

Lim, Bryan, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. 2019. "Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting." *arXiv Preprint arXiv:1912.09363*. https://arxiv.org/abs/1912.09363.

Lopez, Jose A. 2001. "Evaluating the Predictive Accuracy of Volatility Models." *Journal of Forecasting* 20 (2): 87–109. https://doi.org/10.1002/1099-131X(200103)20:2%3C87::AID-FOR782%3E3.0.CO;2-7.

Salinas, David, Valentin Flunkert, and Jan Gasthaus. 2017. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks." *arXiv Preprint arXiv:1704.04110*. https://arxiv.org/abs/1704.04110.

Taylor, Sean J., and Benjamin Letham. 2018. “Forecasting at Scale.” *The American Statistician* 72 (1): 37–45. https://doi.org/10.1080/00031305.2017.1380080.

Wang, Minxing, Pavel Braslavski, Vyacheslav Manevich, and Dmitry Ignatov. 2025. “Bitcoin Ordinals: Bitcoin Price and Transaction Fee Rate Predictions.” *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3541302.

Wang, Ziyang, Masahiro Mae, Takeshi Yamane, Masato Ajisaka, Tatsuya Nakata, and Ryuji Matsuhashi. 2024. “Novel Custom Loss Functions and Metrics for Reinforced Forecasting of High and Low Day-ahead Electricity Prices Using CNN-LSTM.” *Energies* 17 (19): 4885. https://doi.org/10.3390/en17194885.

Winters, Peter R. 1960. “Forecasting Sales by Exponentially Weighted Moving Averages.” *Management Science* 6 (3): 324–42. https://doi.org/10.1287/mnsc.6.3.324.