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[1] are highly volatile and event-driven, with annual spending exceeding billions and single-day spikes reaching over $78 million (Harper 2024). A study by Basu et al. (Basu et al. 2019) found that inefficiencies in fee estimation led to over $272 million in avoidable user spendings, and miners could have reduced fee income variance by an average factor of 7.4 times. Despite this, 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 24 hours in advance, with the primary goal of identifying high-volatility fee periods and a secondary goal of estimating fee levels within those windows.

We explored multiple modeling approaches, including classical time series methods, tree-based models, and deep learning architectures. Standard models captured trend and seasonality but smoothed over the sharp fee spikes that matter most in our context, which are statistical outliers but carry real world significance. To address this, we designed a custom composite loss function that accounts for spike shape, timing, volatility, and accuracy. Guided by this loss, the tailored Temporal Fusion Transformer (TFT) demonstrated the best performance, capturing complex dependencies and improving key evaluation metrics by 25–35%. Despite constraints such as limited historical coverage, lagging input features, and the computational demands 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 Non-Fungible Token (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 has made important progress in identifying key determinants of Bitcoin fees, such as network congestion and transaction complexity (Fan and Liu 2020). Building on this foundation, more recent research has explored theoretical models of fee formation (Glaser, Weber, and Pentz 2023) and applied machine learning techniques to short-term fee estimation, such as using mempool-based features to track congestion in real time (Nair, Bautista, and Goenka 2022). Tools like Bitcoin Core's estimatesmartfee (Bitcoin Core Docs n.d.) and Mempool.space (Bitcoin Explorer n.d.) have also been widely adopted for near-term guidance, offering fast, lightweight heuristics suited for integration into wallets and clients.

[1] Refer to Appendix (Section 6) for formal definitions of Bitcoin-related terms.

However, these approaches remain limited in scope: most focus only on the next 1–6 blocks (~10 minutes per block) and rely exclusively on recent block data. Their methodologies are often opaque, unresponsive to external volatility triggers, and not designed to anticipate sudden shifts. As a result, most public tools and research underemphasize the temporal structure and volatility dynamics that matter most to users planning beyond the immediate horizon (Li et al. 2020), leaving a critical planning gap unaddressed.

Our exploratory analysis revealed key challenges in fee forecasting: strong daily cycles, right-skewed distributions, and sudden congestion spikes. These patterns exposed the limitations of naive point forecasts and motivated a shift toward volatility-aware, time-sensitive modeling. To address this, we evaluated a diverse set of forecasting models reflecting different temporal dynamics: decomposition methods such as Holt-Winters Exponential Smoothing (HWES) (Holt 1957; Winters 1960) and Prophet (Taylor and Letham 2018); autoregressive model Seasonal Autoregressive Integrated Moving Average (SARIMA) (Box et al. 2015); tree-based learner XGBoost (Chen and Guestrin 2016); and deep learning architectures including DeepAR (Salinas, Flunkert, and Gasthaus 2017) and the Temporal Fusion Transformer (TFT) (Lim et al. 2019).

A key contribution of our work is the custom loss function tailored to capture spike sensitivity and relative error fairness. This enabled more meaningful model comparisons beyond traditional metrics like MAE and RMSE. Combined with lag features, careful preprocessing, and robust cross-validation, our modeling pipeline offers a strong foundation for real-world deployment in volatile blockchain conditions.

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 3 outlines the data science techniques used, including preprocessing, model selection, and evaluation strategy. Section 4 presents the data product and results, detailing model performance, intended use, and the system's extensibility. Section 5 concludes with key takeaways, limitations, and recommendations for future development.

3 Data Science Techniques

This section outlines the core data and techniques used in our analysis, including data overview, preprocessing, model development, hyperparameter optimization, 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)[1], captured between March 5 and May 12 2025 (Figure 1). 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 they 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.

Our time series dataset consists of 61 features capturing mempool activity, transaction volume, block statistics, mining difficulty, BTC price, and other relevant indicators. To make these inputs more interpretable, we organized them into four high-level categories, as shown in @#tbl-features. Each category reflects a distinct dimension of network behavior or market condition relevant to fee dynamics

Table 1: Input features from the dataset.

Category	# of Features	Description
Mempool Congestion	40	Mempool congestion and fee distribution indicators
Block Metrics	5	Projected mempool block statistics
Difficulty Metrics	10	Mining difficulty adjustment progress and projection
BTC Price	1	Market BTC price in various currencies

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 that had extremely high correlation with the target (e.g., halfHourFee, exhibiting a 0.98 correlation with fastestFee). 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 fastest-Fee and found it to be highly right-skewed, with most values clustered at the low end and a few extreme spikes shown in Figure 2. This heavy skew can hinder model stability and violate common assumptions.

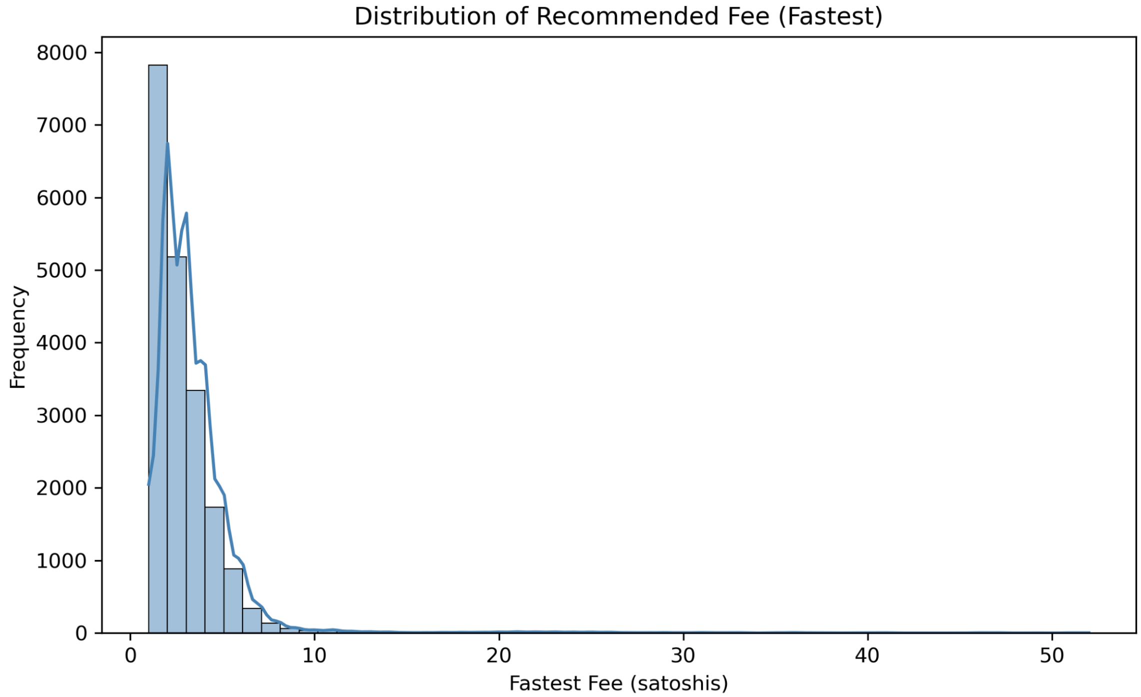


Figure 2: The distribution of fastestFee shows strong right skew.

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.

To reduce noise and enhance short-term signal clarity, we resampled the original 5-minute data into 15-minute intervals. Extremely short intervals introduce high-frequency noise, making patterns harder for models to learn, while overly long intervals risk losing short-term dynamics. As resampling effectively reduces the number of data points, it eases the computational burden during hyperparameter tuning and model training, enabling more efficient use of resources. To strike a balance, we tested multiple resampling frequencies and evaluated their predictive strength using the decay ratio.

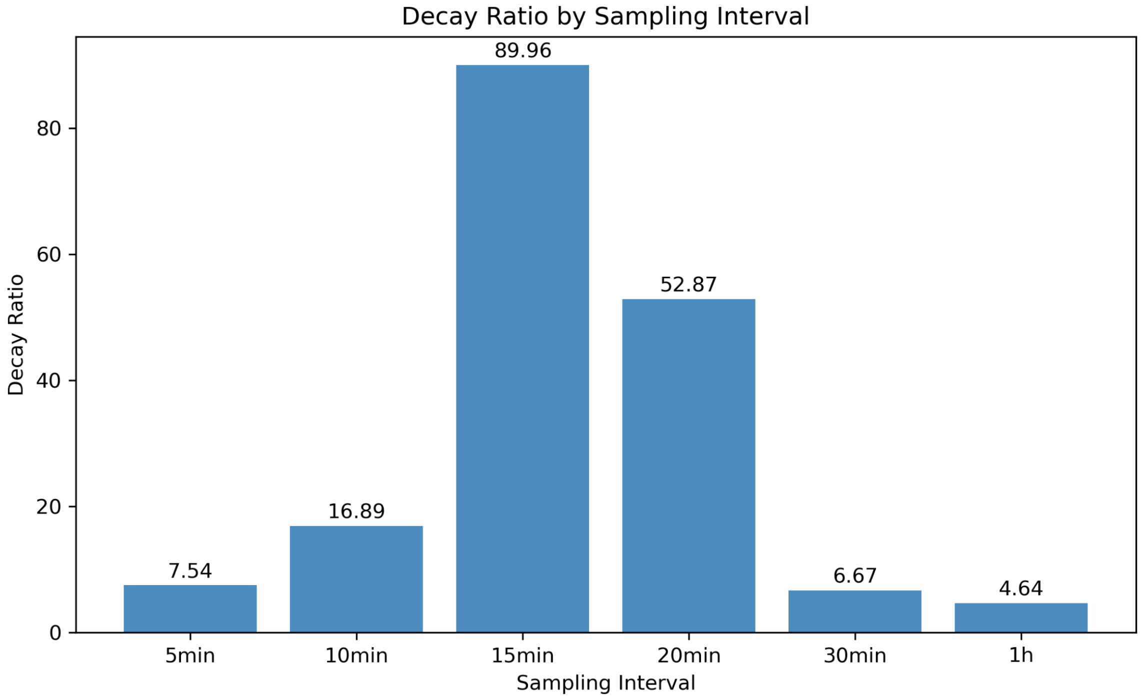


Figure 3: Decay ratio by sampling interval. Higher ratios indicate stronger AR(1)-like structure and better short-term predictability.

As shown in Figure 3, the 15-minute interval achieved the highest decay ratio, indicating the strongest short-term autocorrelation structure. This supports its use as the default sampling interval in our modeling pipeline.

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 progressive series of models. Each model was chosen to overcome specific limitations identified in its predecessor, forming an iterative pipeline that is summarized in Figure 4. Below, we describe each model, its contributions, and where it ultimately fell short.

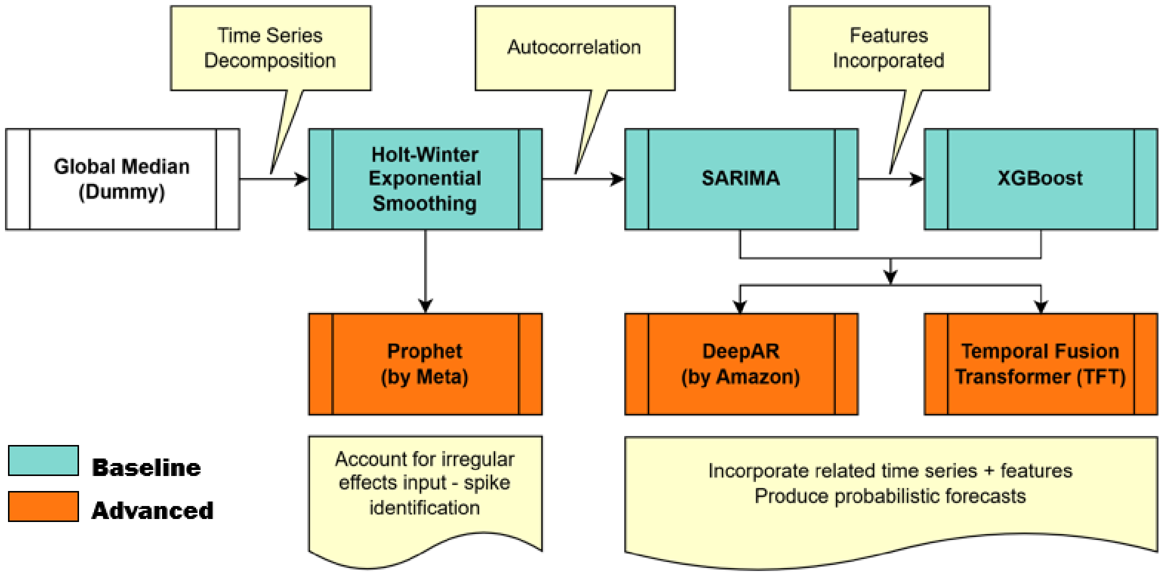


Figure 4: Model progression from simple baselines to deep learning, each addressing prior gaps.

3.3.1 Dummy Model (Global Median)

We began with a dummy model that predicted a constant value: the global median of the fee rate series. As shown in the distribution of fastest fee rates (Figure 2), over 92.5% of observed values clustered between 2–3 satoshis/byte, making the median a surprisingly strong benchmark. While this model ignored all temporal and contextual variation, it provided a reference point to quantify the inherent difficulty of the forecasting task. No training or hyperparameter tuning was required for this static approach.

3.3.2 Holt-Winters Exponential Smoothing (HWES)

We then implemented Holt-Winters Exponential Smoothing given clear seasonality and trend observed in our decomposition analysis (Figure 5). This method supports both additive and multiplicative structures, as well as optional damped trends. We performed a grid search to identify the optimal configuration of these components. However, residual autocorrelation observed in ACF and PACF plots (Figure 6) indicated that HWES failed to capture deeper temporal dependencies, limiting its forecasting accuracy.

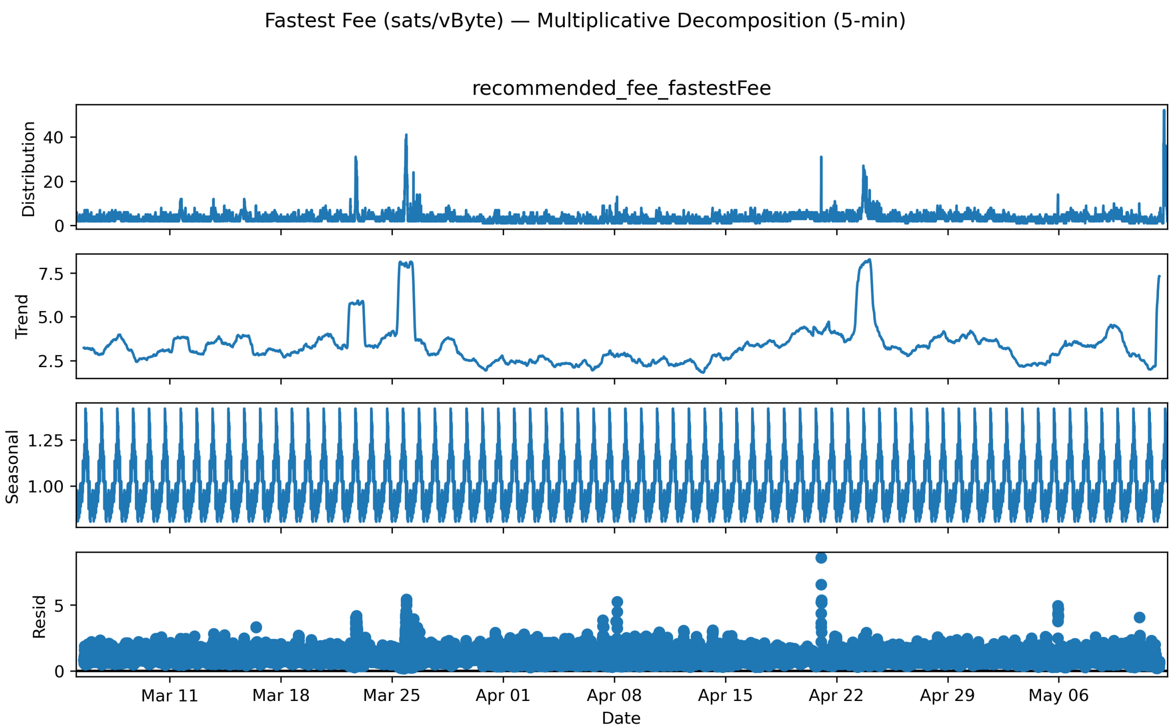


Figure 5: FastestFee exhibits multiplicative trend and seasonality structure.

3.3.3 SARIMA

To address the temporal dependencies missed by HWES, we introduced SARIMA that combines autoregressive, differencing, and moving average components to model time-dependent and seasonal patterns. Drawing on exploratory analysis, including strong autocorrelation at short lags, partial autocorrelation structure, and daily seasonality, we manually selected parameters to reflect short-term dynamics and recurring cycles in the fee series. However, SARIMA does not support exogenous variables, which limited our ability to incorporate external factors such as transaction volume or mempool congestion.

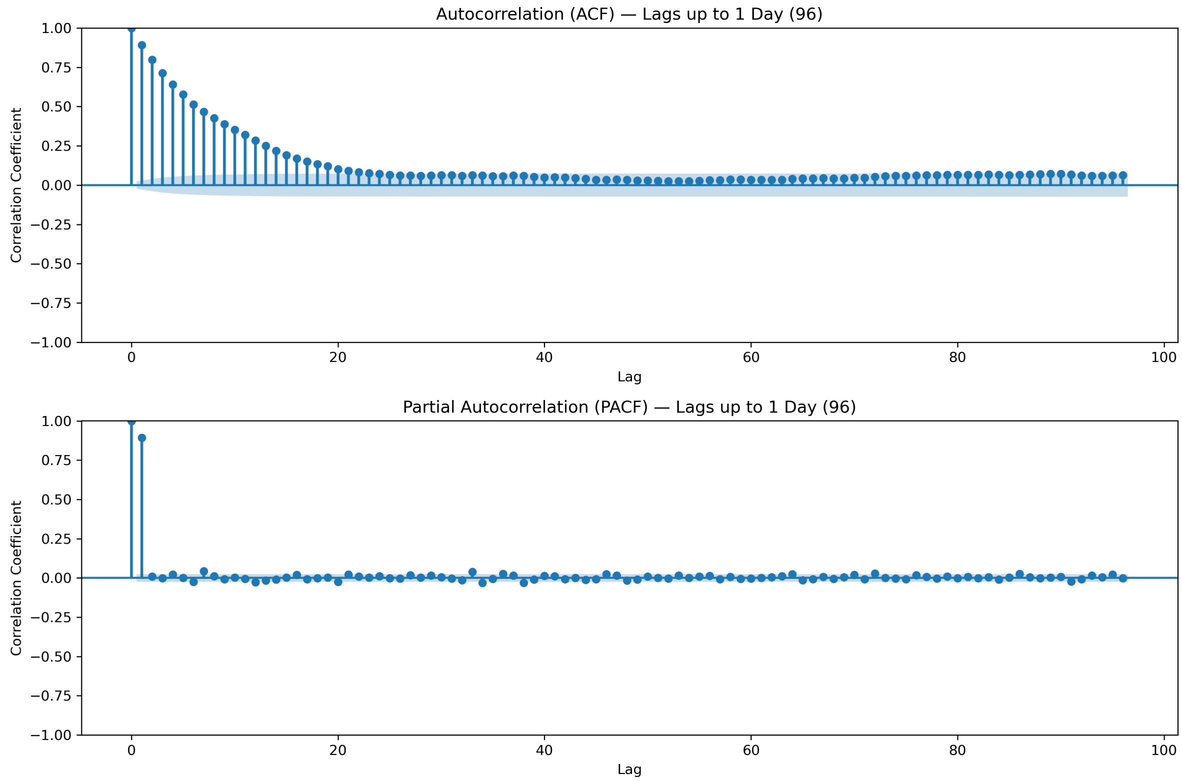
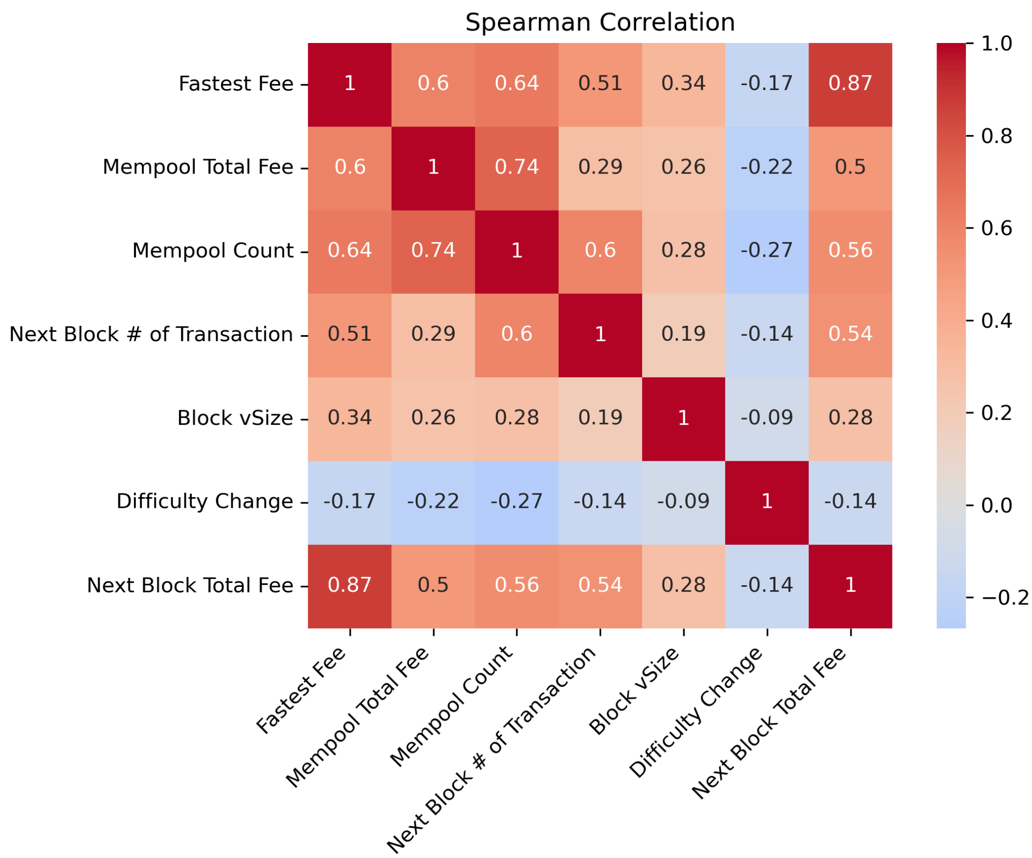


Figure 6: Residual autocorrelation highlights unmodeled temporal dependencies.

3.3.4 XGBoost

We next adopted XGBoost to leverage a broader set of concurrent features. As shown in the correlation heatmap (Figure 7), variables such as mempool total fee, mempool count, and projected block fees were strongly correlated with the fastest fee rate. XGBoost, a gradient-boosted tree model, enabled us to capture non-linear interactions among these features. It expanded our modeling capacity beyond univariate structures and allowed fine-grained feature importance analysis. We tuned the model using randomized search over tree depth, learning rate, and regularization terms.



	Fastest Fee	Mempool Total Fee	Mempool Count	Next Block # of Transaction	Block vSize	Difficulty Change	Next Block Total Fee
Fastest Fee	1	0.6	0.64	0.51	0.34	-0.17	0.87
Mempool Total Fee	0.6	1	0.74	0.29	0.26	-0.22	0.5
Mempool Count	0.64	0.74	1	0.6	0.28	-0.27	0.56
Next Block # of Transaction	0.51	0.29	0.6	1	0.19	-0.14	0.54
Block vSize	0.34	0.26	0.28	0.19	1	-0.09	0.28
Difficulty Change	-0.17	-0.22	-0.27	-0.14	-0.09	1	-0.14
Next Block Total Fee	0.87	0.5	0.56	0.54	0.28	-0.14	1

Figure 7: Strong correlations observed between target fee rate and key explanatory variables.

3.3.5 Prophet

Then we moved to advanced models. First we extended the HWES baseline with Prophet to better handle abrupt changes in fee trends. While both methods model trend and seasonality, Prophet improves flexibility by incorporating automatic changepoint detection and user-defined seasonal effects. We configured Prophet with hourly, daily, and weekly seasonalities to reflect recurring fee cycles, and tuned changepoint sensitivity to capture sudden demand shifts such as batch fee spikes or market congestion.

3.3.6 DeepAR

Building on the autoregressive structure of SARIMA and the feature integration strength of XGBoost, we explored DeepAR—an LSTM-based model developed by Amazon for probabilistic sequence forecasting. Unlike traditional models, DeepAR learns global patterns across sequences and captures non-linear temporal dependencies, while also allowing the inclusion of time-varying and static covariates. We used PyTorch Lightning's Trainer to manage training configuration and Tuner to automatically search for optimal hyperparameters, including learning rate and hidden units. Early stopping was applied to prevent overfitting and stabilize convergence. This setup allowed us to model sequential fee dynamics more expressively and robustly.

3.3.7 Temporal Fusion Transformer (TFT)

Finally, we introduced the Temporal Fusion Transformer (TFT), the most advanced model in our pipeline. TFT builds on DeepAR's ability to incorporate time-varying covariates and static metadata, while advancing the architecture with explicit variable selection, multi-horizon attention, and interpretable output mechanisms. This allows TFT to capture richer dependencies and highlight feature importance in ways DeepAR cannot. We carefully optimized the TFT architecture, including variable selection layers, gated residual networks, and learning rate schedules. This model allowed us to capture complex interactions and feature importances while generating multi-horizon probabilistic forecasts.

3.3.8 Considered Alternatives and Limitations

Our efforts extended beyond conventional forecasting once it became clear that traditional models failed to capture spike behavior. We reframed the problem as a classification task to predict spike occurrences within a time window, but the rarity and unpredictability of these events made the approach unreliable. We also explored methods like Distributed Lag Non-Linear Models (DLNM) and Fourier transforms, but their assumptions did not fit the data, as fee spikes lacked consistent leading signals and showed no periodic structure.

A major limitation in our modeling was the reliance on lagged exogenous features, which constrained the model's ability to anticipate sudden fee spikes. To address this, we considered a two-stage forecasting setup in which key indicators would be predicted first and then fed into the fee model to enable more forward-looking forecasts. However, given the lack of reliable leading signals, the partner's emphasis on fully utilizing existing features, and the risk of compounding errors in multi-stage pipelines, we chose to prioritize model depth over input expansion within the project's limited timeframe.

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 2.

Table 2: Breakdown of custom loss function components.

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

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

This project evaluates six forecasting models—HWES, SARIMA, Prophet, XGBoost, DeepAR, and Temporal Fusion Transformer (TFT)—for their effectiveness in capturing short-term Bitcoin transaction fee dynamics, particularly high-frequency volatility. For baseline comparison, a global median model is also included, as summarized in Table 3.

Table 3: Model performance on test data.

	TFT	Prophet	DeepAR	HWES	XGBoost	SARIMA	Median
custom_loss	1.78	2.43	2.55	2.59	2.59	2.61	2.67
rmse	0.94	1.34	1.15	1.42	1.23	1.2	1.11
mae	0.75	1.01	0.92	1.13	1	0.97	0.91
mape	0.35	0.58	0.51	0.47	0.55	0.55	0.51

HWES and SARIMA were included as simple, fast-to-train baselines that provide coarse-grained fee trend forecasts. They are able to approximate day-level seasonal drifts but fail to track sharp intraday dips and spikes (see Figure 8). While their predictions are stable, they perform poorly on short-term volatility metrics: HWES records a custom loss of 2.59 and RMSE of 1.42, while SARIMA fares in a similar level with a custom loss of 2.61 and RMSE of 1.20. These results underscore the limitations of traditional models in capturing high-frequency dynamics in transaction fee patterns. In fact, both HWES and SARIMA perform no better than the global median baseline, which records a custom loss of 2.67 and an RMSE of 1.11.

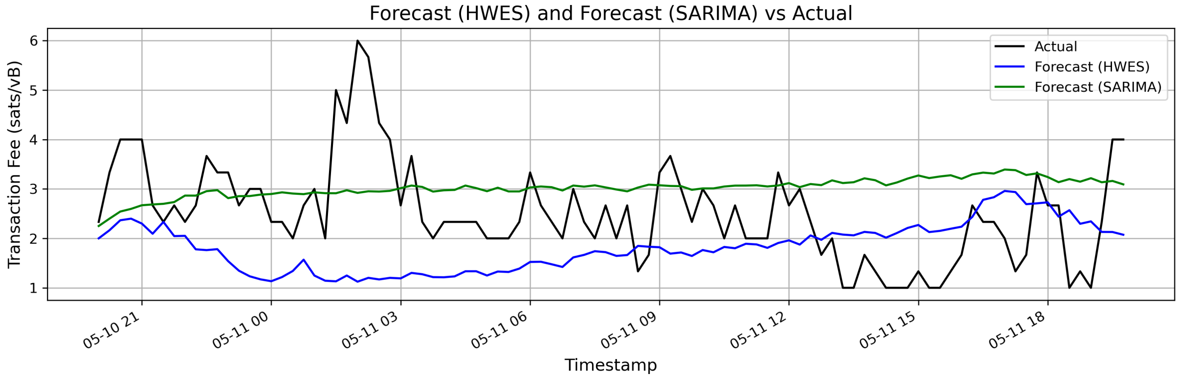


Figure 8: HWES and SARIMA forecasts vs actual fee on test data

Prophet was chosen for its ease of use, interpretability, and built-in support for trend shifts, seasonality, and holiday effects—features well-suited for capturing macro-level fee patterns with minimal tuning. It benefits from flexible trend components and seasonal priors, yielding improved global fits. However, it still oversmooths transient spikes. Its custom loss is 2.43, and RMSE is 1.34.

XGBoost was selected for its ability to model nonlinear interactions between engineered features such as lagged fees and mempool congestion signals, while remaining efficient to train and tune. It outperforms all classical models, achieving a custom loss of 2.59 and RMSE of 1.23. However, it tends to underestimate sharp fee jumps and often produces flat or conservative predictions in highly volatile regions (see Figure 9).

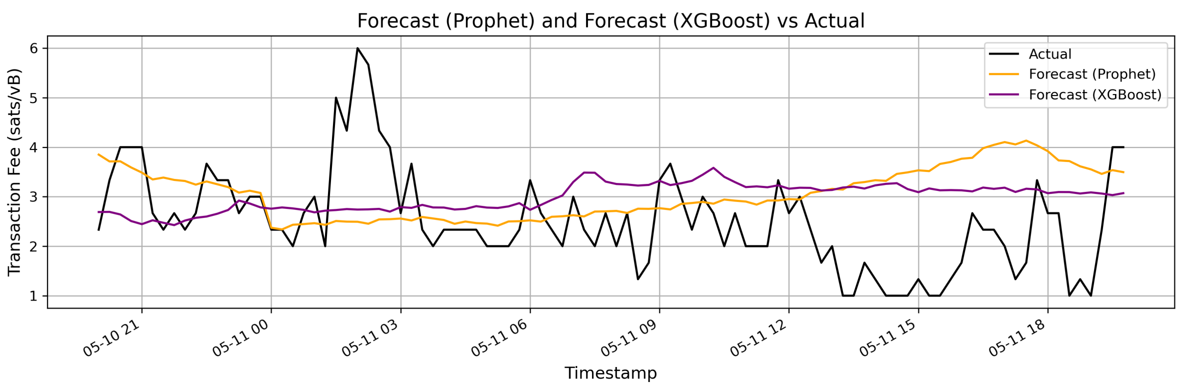


Figure 9: Prophet and XGBoost forecasts vs actual fee on test data

The neural models represent a meaningful shift in modeling capacity, enabling the system to learn from richer temporal patterns and non-linear interactions. DeepAR was selected for its ability to capture sequential dependencies through autoregressive recurrence, offering a pathway toward future probabilistic forecasting, with a custom loss of 2.55 and an RMSE of 1.15.

TFT was chosen for its architecture that combines attention mechanisms, gating, and variable selection—allowing it to track both the magnitude and timing of sudden fee surges. It demonstrates the strongest performance on high-frequency volatility, making it ideal for fine-grained, urgency-tiered fee forecasts with real-time planning value (see Figure 10).

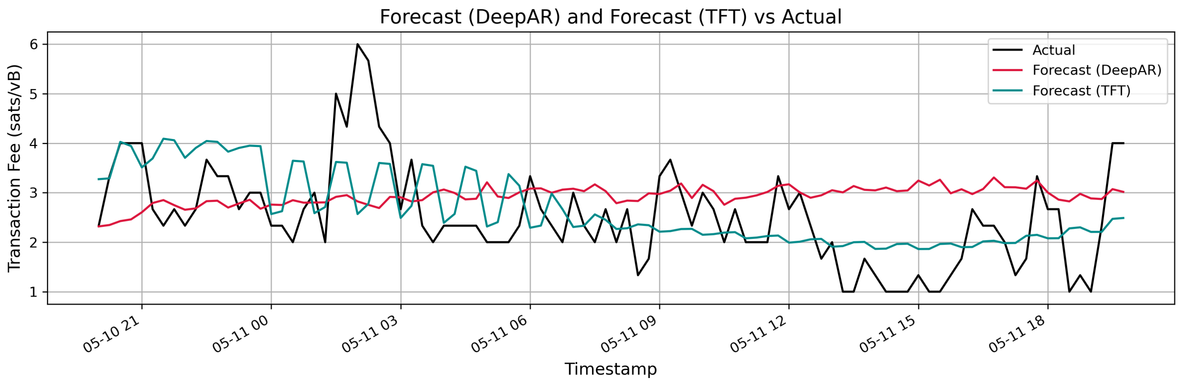


Figure 10: DeepAR and TFT forecasts vs actual fee on test data

Quantitatively, TFT reduces the bespoke volatility-aware loss from about 2.43, the lowest among the non-neural models, to 1.78, representing a 27% improvement. It also brings RMSE down from the 1.20–1.42 range to 0.94, yielding a 21% relative gain. MAE and MAPE follow the same pattern, confirming that TFT strikes the strongest balance between overall accuracy and sensitivity to congestion-driven shocks.

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 Strengths

Our product addresses key limitations of current Bitcoin fee tools like estimatesmartfee and Mempool.space, which offer only 10 to 60 minute forecasts with a single suggested rate and minimal transparency into volatility drivers or user-specific needs. In contrast, our system delivers 24-hour forecasts across multiple urgency tiers (fastest, economy, minimum), giving users actionable insight aligned with cost sensitivity and transaction timing. On the development side, we emphasized transparency, modularity, and extensibility. Jupyter notebooks explain modeling choices in an accessible narrative format, making our approach easy to audit and adapt. The pipeline's modular structure allows developers to swap models, adjust hyperparameters, or introduce new features without reworking the entire system. Finally, by releasing the code as an open-source GitHub repository, we enable community contribution and peer validation—supporting our partner's infrastructure mission and fostering long-term adaptability.

4.4 Limitations and Design Trade-Offs

While the product emphasizes flexibility, insight, and transparency, several constraints affect both its performance and accessibility. Most notably, deep learning models like TFT power the pipeline but also raise the barrier to entry, often requiring GPU access or cloud resources beyond the reach of smaller teams. On top of that, regular retraining will be necessary as network conditions shift, but automation is better introduced once models reach stable accuracy. One key challenge is that most current inputs, such as mempool congestion or block composition, are reactive. This limits the system's ability to anticipate sudden fee shifts and slows overall model maturity. Similarly, confidence intervals and real-time APIs are valuable for deployment, but not yet warranted given forecast noise and development priorities. To stay agile, the project remains script-first and avoids full orchestration. This lightweight backbone supports rapid experimentation, modular upgrades without entire pipeline overhaul, and future extensions like adaptive loss tuning, hybrid pipelines, and scalable serving layers.

5 Conclusion and Recommendations

This project sets out to forecast Bitcoin transaction fees 24 hours ahead. To ensure models respond appropriately to sudden fee surges, we developed a custom loss function that emphasizes spike behavior and structural accuracy. Through extensive experimentation across statistical models, tree-based learners, and deep sequence models, we found that Temporal Fusion Transformer is best-suited to capturing irregular spike patterns and outperformed other models in RMSE, MAPE and a the volatility-sensitive loss by 25–35%. The resulting system offers a flexible and transparent foundation that moves beyond average-point to support users, wallets, and infrastructure providers in making 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 due to limited time for deeper research and could be better tuned to balance volatility sensitivity and accuracy. Future work could expand historical coverage, incorporate off-chain or leading indicators, and refine the custom loss function for better spike detection. In addition, hybrid pipelines that first predict intermediate signals and feed them into the final forecasting model, along with uncertainty-aware forecasts, may enhance robustness and usability over time.

6 Appendix

6.1 Terminology

Table 4: Key Terms and Definitions in Bitcoin and Blockchain (Alphabetically Ordered)

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.
Bitcoin Core	The reference implementation of the Bitcoin protocol. It is the official software used to run a full Bitcoin node, validate transactions, and maintain a copy of the entire blockchain.
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.
Non-Fungible Token (NFT)	A unique digital asset stored on a blockchain that represents ownership of something like art, music, or collectibles. Unlike cryptocurrencies such as Bitcoin, NFTs are one-of-a-kind and cannot be exchanged on a one-to-one basis.
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.

References

Basu, Debojyoti, David Easley, Maureen O'Hara, and Seth Weinberg. 2019. "Towards a Functional Fee Market for Cryptocurrencies." In *Proceedings of the 2019 ACM Conference on Economics and Computation*, 433–51. ACM. https://doi.org/10.1145/3328526.3329585.

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.

Glaser, Michael, Sophia Weber, and Carl Pentz. 2023. "Determination of Equilibrium Transaction Fees in the Bitcoin Network." *Journal of Financial Intermediation* 53: 100248. https://doi.org/10.1016/j.jfi.2023.100248.

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.

Nair, Nithin, Camille A. Bautista, and Harsh Goenka. 2022. "On Optimizing Transaction Fees in Bitcoin Using AI." In *Proceedings of the ACM Conference on Advances in Financial Technologies (AFT)*, 85–97. https://doi.org/10.1145/3528669.

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.