Final Report: Forecasting Bitcoin Transaction Fees

Partner: Trilemma Foundation

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

2025-06-25

Table of contents

1 Executive Summary **2**

2 Introduction **2**

3 Data Science Techniques **4**
3.1 Dataset Overview . . . 4
3.2 Feature Preprocessing . . . 5
3.3 Models . . . 7
3.4 Evaluation Metrics . . . 12
3.5 Stakeholder Impact & Ethical Considerations . . . 13

4 Data Product and Results **14**
4.1 Results: Model Performance and Forecast Visualization . . . 14
4.2 Data Product Overview . . . 16
4.3 Value Proposition and Strengths . . . 17
4.4 Limitations and Design Trade-Offs . . . 17

5 Conclusion and Recommendations **17**

6 Appendix **18**
Appendix A: Terminology . . . 18

References **19**

1 Executive Summary

Bitcoin transaction fees are highly volatile and event-driven, with single-day spikes exceeding $78 million and inefficiencies costing users hundreds of millions. 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 21%–27%. 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[1] fluctuate sharply due to irregular, off-chain events—such as network congestion, protocol updates, or Non-Fungible Token (NFT) inscription surges—rather than stable, recurring patterns. This fluctuation has led to daily transaction fees reaching over $78 million (Harper 2024), inefficient fee spending accumulating to over $272 million, and average 7.4 times more miner income variance (Basu et al. 2019). As a result, short-term fee selection has become a high-stakes challenge for both 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 A 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.

This challenge was initially raised by our capstone partner Trilemma Capital, whose mission focuses on serving industry talent and advancing Bitcoin infrastructure through data science. To bridge this gap, we started by examining the temporal structure of transaction fees, which revealed key challenges: strong daily cycles, right-skewed distributions, and sudden congestion spikes. These patterns exposed the limits of naïve point forecasts and motivated a shift toward volatility-aware, time-sensitive modeling.

We therefore evaluated a range of models capturing diverse 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 TFT (Lim et al. 2019).

A central contribution of our work is a custom loss function designed to capture spike sensitivity and relative error fairness, addressing limitations of standard metrics like MAE and RMSE. This, along with lag-based features, careful preprocessing, and robust cross-validation, enabled more meaningful model comparison and a strong foundation for real-world deployment in volatile blockchain environments. Among the models tested, TFT consistently outperformed others, improving key evaluation metrics by 21%–27% and demonstrating strong capacity to model irregular fee dynamics 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 methods used in our analysis, including data overview, preprocessing, model development, hyperparameter tuning, and evaluation strategies for fee forecasting.

3.1 Dataset Overview

We worked with a time series dataset constructed from 5-minute snapshots of the Bitcoin mempool (mempool.space), collected between March 5 and May 12 2025 (Figure 1). The mempool is a real-time queue of unconfirmed transactions that wait to be included in a block, which proves that the network has validated and recorded them on the blockchain. Each snapshot captures the network state at a specific moment, forming a time series that reflects both blockchain activity and shifting market demand—the key drivers of transaction fee dynamics.

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

Our primary target variable, the recommended *fastestFee* (in sats/vByte), represents the rate required for near-immediate confirmation and serves as the core signal for modeling. The dataset includes 61 features covering mempool activity, transaction volume, block statistics, mining difficulty, BTC price, and related indicators. For clarity, we grouped these into four categories (see Table 1), each representing a distinct aspect of network or market behavior that influences 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

We conducted exploratory correlation analysis to understand the data and validate feature relevance. While not used for formal feature selection, the analysis confirmed that several predictors carry meaningful signals for the target variable, *fastestFee*.

Moreover, we excluded features with extremely high correlation to the target to prevent data leakage. For instance, *halfHourFee* showed a correlation of 0.99 with *fastestFee*. These features were removed from all models except DeepAR and TFT, where architectural constraints made exclusion more difficult.

We also applied several feature engineering steps. The distribution of *fastestFee* was highly right skewed, with most values concentrated at the low end and a few extreme spikes (Figure 2). This skew can reduce model stability and violate assumptions in some forecasting methods.

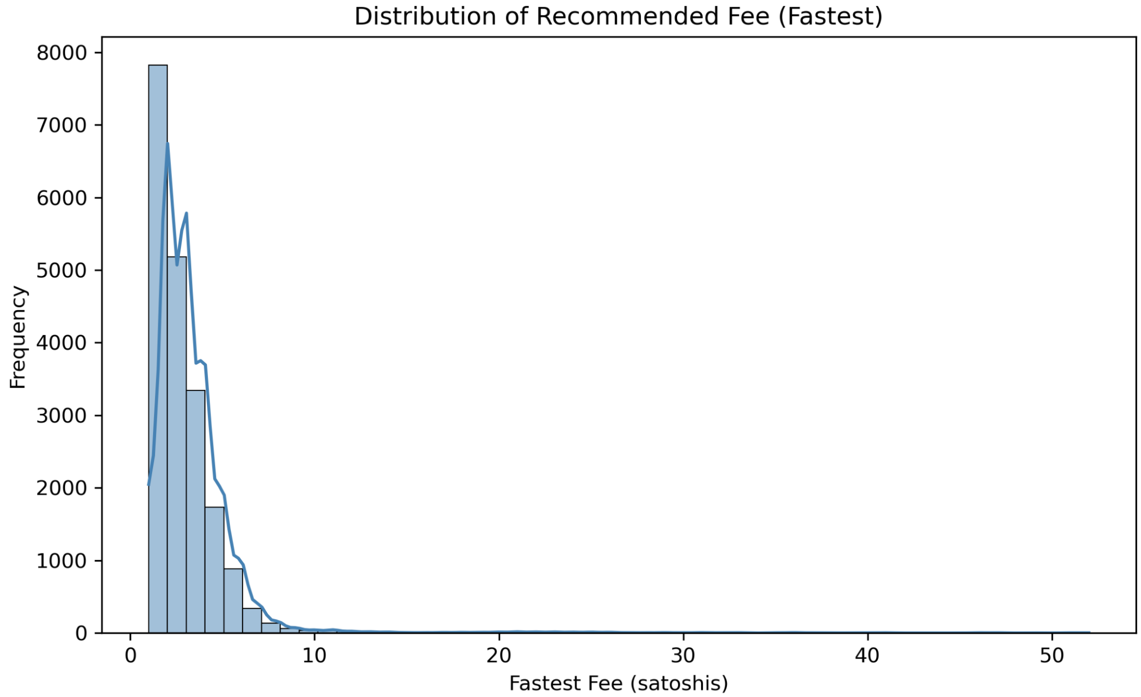

Figure 2: The distribution of fastestFee shows strong right skew.

To address this, we selectively applied a logarithmic transformation to *fastestFee* to compress large values, stabilize variance, and make the series more suitable for modeling.

Furthermore, we resampled the original 5-minute data into 15-minute intervals to reduce noise and improve short-term signal clarity. Excessively short intervals introduce high-frequency noise, while overly long intervals risk losing important dynamics. Resampling also reduces the data volume, easing the computational load during tuning and training. To find the right balance, we tested multiple frequencies and assessed their predictive strength using the decay ratio.

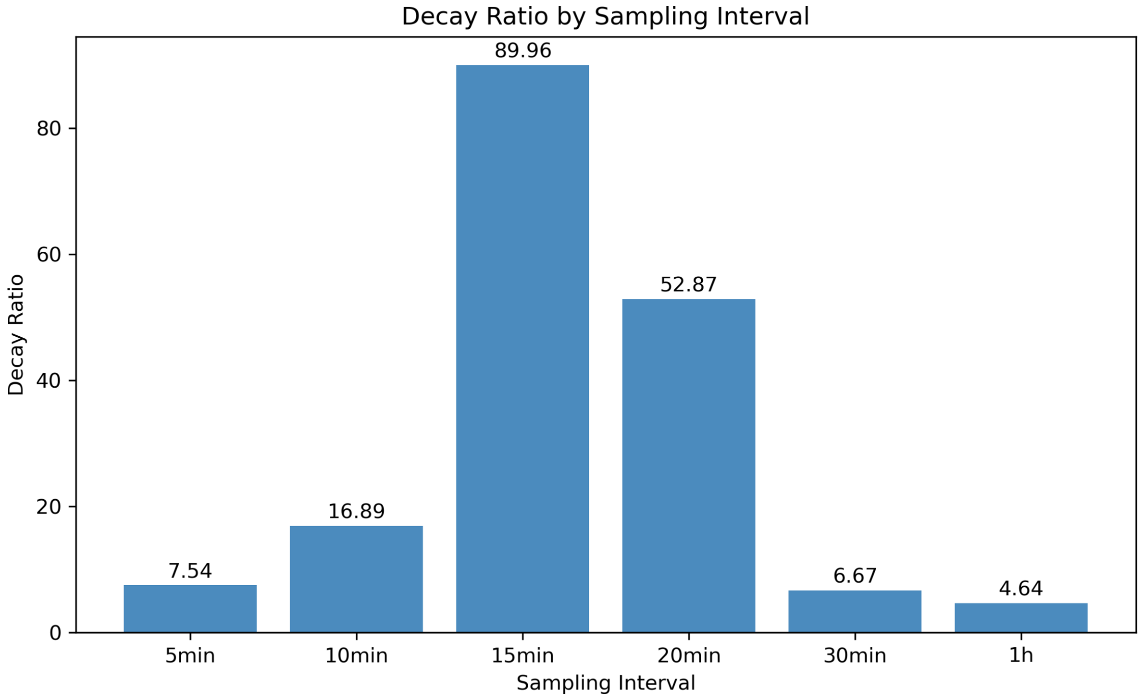

Figure 3: Decay ratio by sampling interval. Higher ratios indicate stronger AR(1)-like structure and better short-term predictability.

As shown in Figure 3, the 15-minute interval yielded the highest decay ratio, indicating the strongest short-term autocorrelation. This supported its selection as the default sampling interval.

Finally, we created lagged features and rolling aggregates to help models capture temporal dependencies. The final feature set retained most original variables, excluding those flagged for leakage, to balance completeness and modeling integrity.

3.3 Models

As part of our modeling strategy, we implemented a series of progressively refined models. Each was selected to address limitations of its predecessor, forming an iterative pipeline summarized in Figure 4. Below, we describe each model, outline its contributions and remaining challenges.

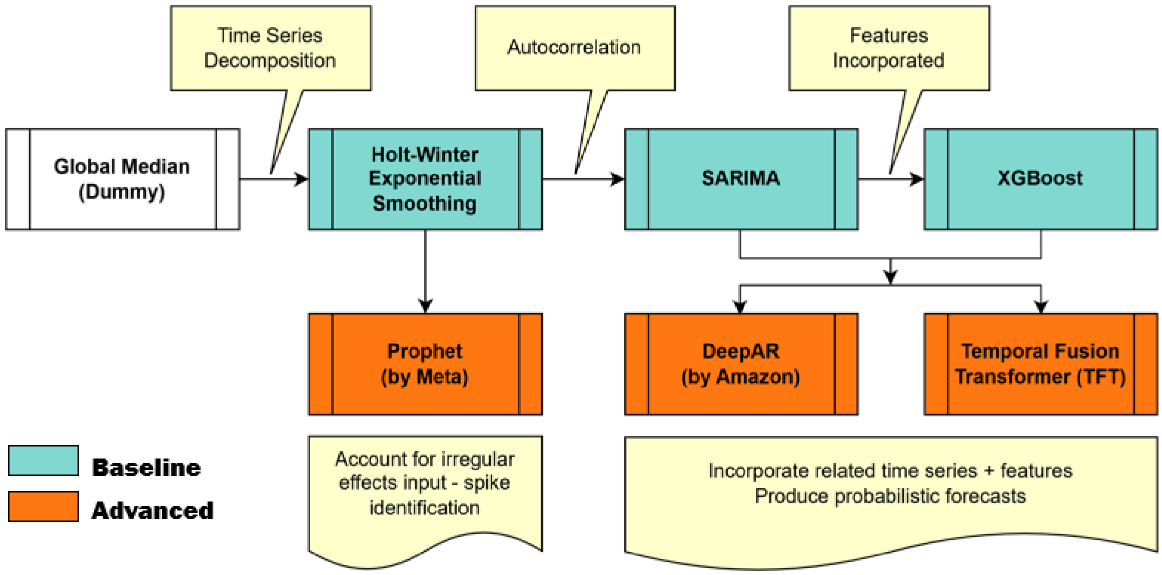

Figure 4: Model progression from simple baselines to deep learning, each addressing prior gaps.

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), approximately 88.9% of all transactions fell into the 1–2 sats/vB bin, and including 2–3 sats/vB raised the cumulative share above 95.5%. This made the median a surprisingly strong benchmark. Although the model ignored all temporal and contextual information, it served as a useful reference for evaluating task difficulty. No training or tuning was required.

3.3.2 Holt-Winters Exponential Smoothing (HWES)

We then implemented HWES to model the clear trend and seasonality revealed by our decomposition analysis (Figure 5). HWES supports both additive and multiplicative structures, along with optional damped trends. We used grid search to optimize these components.

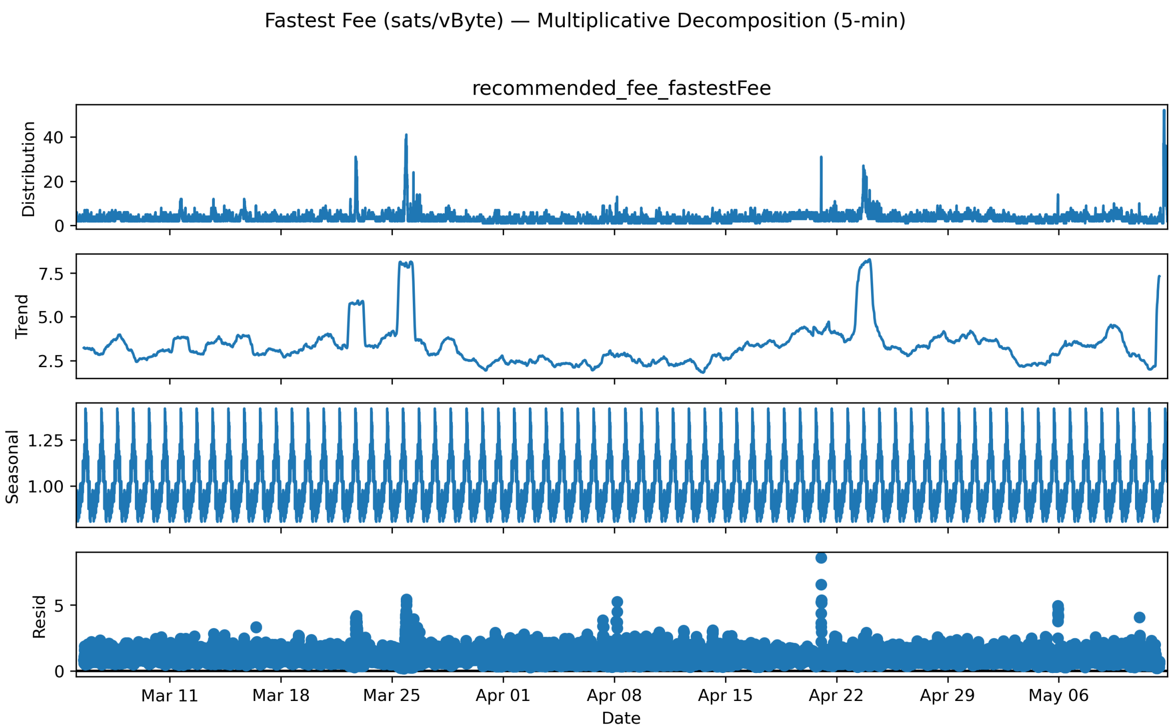

Figure 5: FastestFee exhibits multiplicative trend and seasonality structure.

However, autocorrelation patterns observed in ACF and PACF plots (Figure 6) indicated that HWES failed to capture deeper temporal dependencies, limiting its forecasting accuracy.

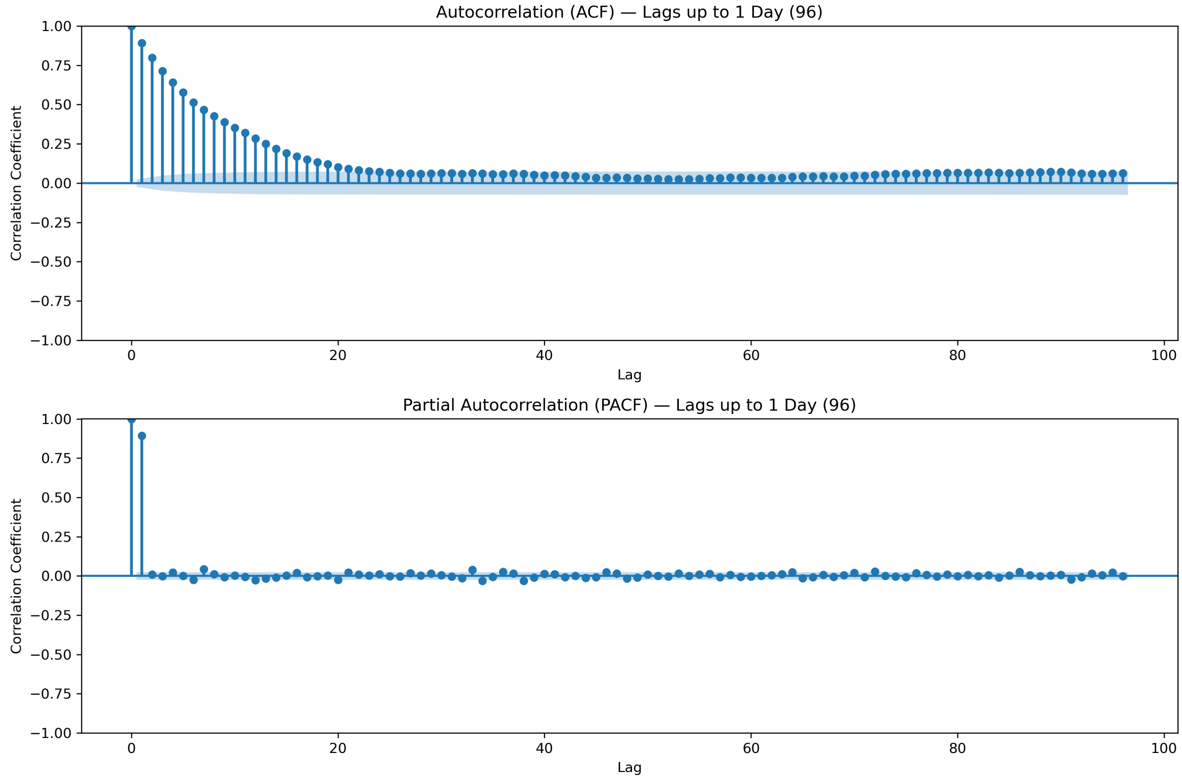

Figure 6: Residual autocorrelation highlights unmodeled temporal dependencies.

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. Based on exploratory analysis—strong short-lag autocorrelation, partial autocorrelation structure, and daily seasonality—we manually selected parameters to reflect short-term dynamics and recurring cycles. However, SARIMA lacks support for exogenous variables, limiting our ability to incorporate other signals such as transaction volume or mempool congestion.

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* (0.60), *mempool count* (0.64), and *projected block fees* (0.87) were strongly correlated with the *fastestFee*. XGBoost, a gradient-boosted tree model, captured non-linear interactions among features and extended our modeling beyond univariate structures. It also enabled detailed feature importance analysis. We tuned it using randomized search over tree depth, learning rate, and regularization parameters.

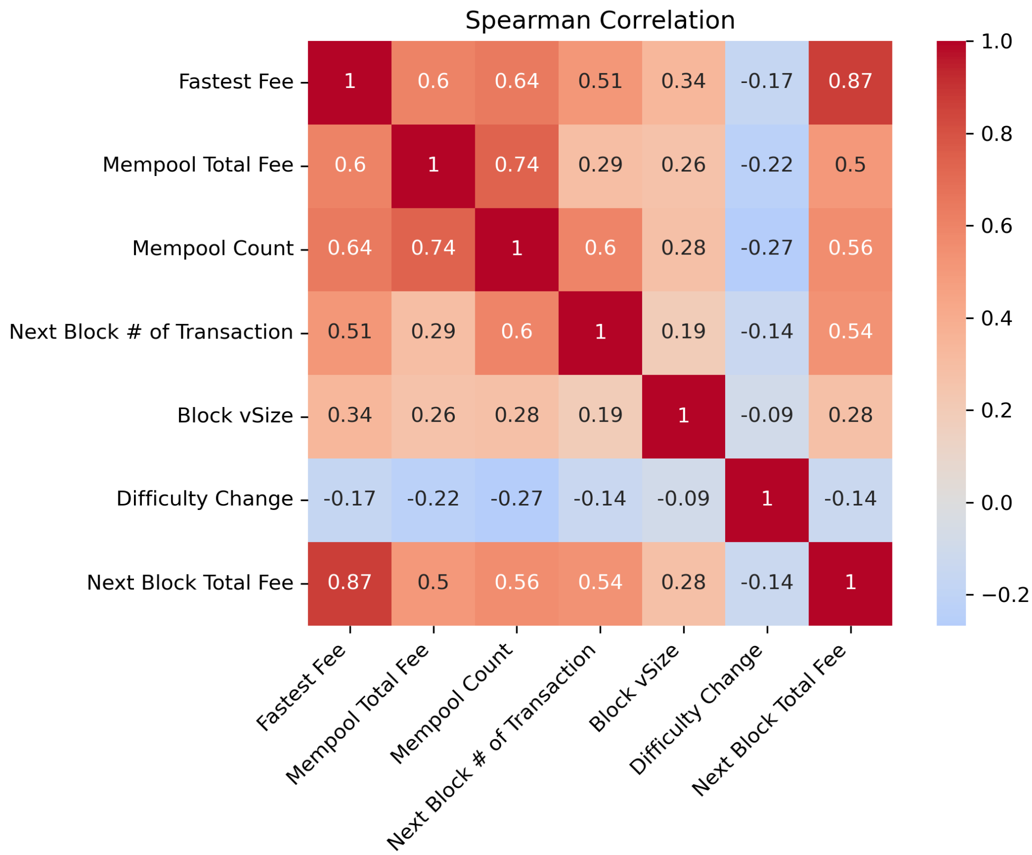

Figure 7: Strong correlations observed between target fee rate and key explanatory variables.

3.3.5 Prophet

Then we moved to advanced models. First we extended the HWES baseline using Prophet to better handle abrupt changes in fee trends. Like HWES, Prophet models trend and seasonality, but adds flexibility through automatic changepoint detection and customizable seasonal effects. We configured hourly, daily, and weekly seasonalities to reflect recurring fee cycles and tuned changepoint sensitivity to capture sudden shifts, such as batch fee spikes or market congestion.

3.3.6 DeepAR

Building on SARIMA's autoregressive structure and XGBoost's feature integration, we explored DeepAR—an LSTM-based model designed for probabilistic sequence forecasting. DeepAR captures non-linear temporal dependencies and learns global patterns across sequences, while supporting both time-varying and static covariates. We used PyTorch Lightning's *Trainer* for training management and *Tuner* for hyperparameter optimization, including learning rate and hidden units. Early stopping was applied to prevent overfitting and ensure stable convergence. This setup enabled more expressive and robust modeling of sequential fee dynamics.

3.3.7 Temporal Fusion Transformer (TFT)

Finally, we introduced TFT, the most advanced model in our pipeline. TFT extends DeepAR by incorporating explicit variable selection, multi-horizon attention, and interpretable outputs, while preserving support for time-varying covariates and static metadata. These enhancements allow it to capture richer dependencies and reveal feature importance more effectively. We carefully optimized the architecture by tuning variable selection layers, gated residual networks, and learning rate schedules. TFT enabled us to model complex interactions and produce multi-horizon probabilistic forecasts with greater interpretability.

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 pipeline: first forecasting key indicators, then feeding them into the fee model. However, due to the absence of reliable leading signals, the partner's emphasis on fully leveraging existing features, and the risk of compounding errors in multi-stage setups, we prioritized 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.

We used multiple evaluation metrics to balance interpretability and relevance to fee volatility. MAPE provided intuitive, percentage-based errors for assessing relative accuracy across fee levels. RMSE complemented this by penalizing large errors more heavily, making it better suited for detecting sharp spikes and for users seeking 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 functions: Le Guen & Thome propose a distortion-based loss aligned with temporal patterns (Guen and Thome 2019), Wang et al. show improved spike detection using custom losses in extreme-value settings (Z. Wang et al. 2024), and Lopez stresses aligning evaluation with volatility-driven business objectives (Lopez 2001).

Inspired by these studies, we crafted a custom loss combining 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 for capturing event-driven Bitcoin congestion. A breakdown of the 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 key stakeholder needs. End users aim to avoid high-fee periods, wallet providers require timely and interpretable forecasts, and miners may optimize revenue with improved 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 issues through open access, transparent design, and clear communication of model limitations. Broader concerns like fairness, miner incentives, and malicious mempool behavior lie beyond our current scope and merit future attention.

4 Data Product and Results

This section presents model performance results and forecast visualizations, followed by a discussion of the final data product—its deliverables, target users, use cases, and extensibility.

4.1 Results: Model Performance and Forecast Visualization

We evaluated 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.

We evaluated six forecasting models—HWES, SARIMA, Prophet, XGBoost, DeepAR, and TFT—for their ability to capture next day (24-hour) Bitcoin fee dynamics, with a focus on high-frequency volatility. We also included a global median model for baseline comparison, as summarized in Table 3.

Table 3: Model performance on test data.

	TFT	Prophet	DeepAR	XGBoost	HWES	SARIMA	Median
custom_loss	1.777	2.428	2.548	2.585	2.594	2.612	2.665
rmse	0.942	1.343	1.153	1.226	1.423	1.2	1.109
mape	0.347	0.581	0.51	0.55	0.468	0.547	0.508

We used HWES and SARIMA as simple, fast-to-train baselines for coarse fee trend forecasting. Both captured broad day-level seasonality but missed sharp intraday dips and spikes (see Figure 8). HWES recorded a custom loss of 2.59, RMSE of 1.42, and MAPE of 0.47. SARIMA performed similarly (2.61 custom loss, 1.20 RMSE, 0.55 MAPE). Both models failed to improve on the global median baseline, which posted 2.67 custom loss, 1.11 RMSE, and 0.51 MAPE.

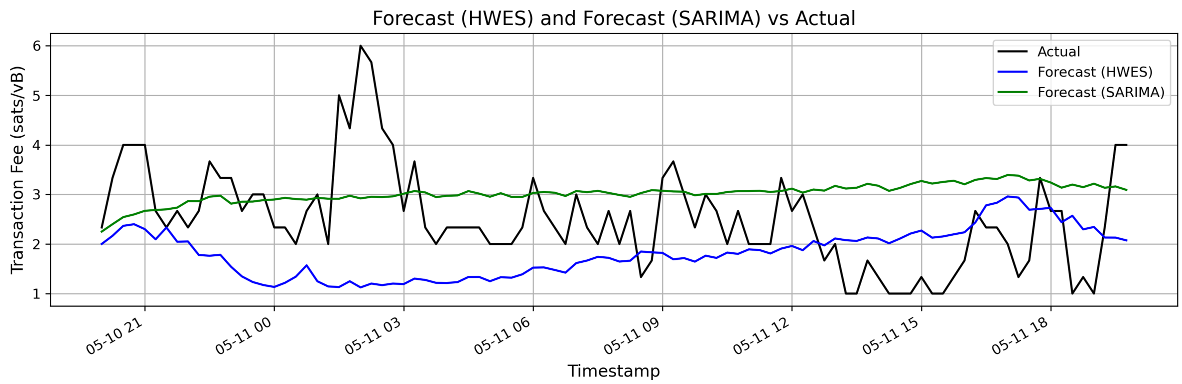

Figure 8: HWES and SARIMA forecasts vs actual fee on test data

We used Prophet for its simplicity, interpretability, and built-in support for trend shifts, seasonality, and holiday effects—features well-suited to capturing macro-level fee patterns with minimal tuning. However, it oversmoothed sharp fee fluctuations and produced the highest MAPE (0.58) among all models. Its custom loss (2.43) and RMSE (1.34) placed it only marginally ahead of the global median (2.67, 1.11), highlighting its limited advantage. As shown in Figure 9, its forecasts lagged behind actual spikes and decayed slowly.

We selected XGBoost to model nonlinear interactions between features like mempool congestion and lagged fees. It slightly outperformed Prophet, with a custom loss of 2.59, RMSE of 1.23, and MAPE of 0.55, but remained on par with classical baselines. It tended to underestimate sharp fee jumps and often produced flat or conservative predictions in volatile regions (Figure 9).

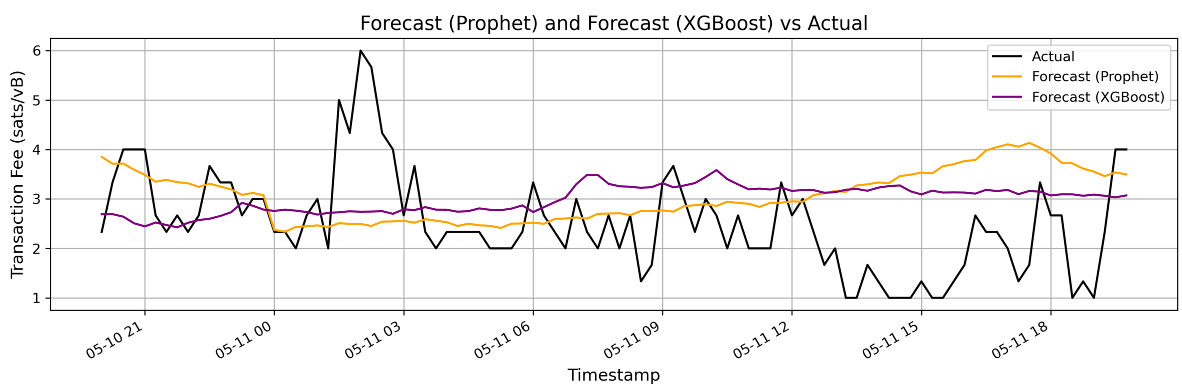

Figure 9: Prophet and XGBoost forecasts vs actual fee on test data

The neural models marked a meaningful shift in modeling capacity, enabling the system to learn from richer temporal patterns and nonlinear interactions. We used DeepAR to capture sequential dependencies through autoregressive recurrence. It provided a path toward probabilistic forecasting and outperformed all classical models with a custom loss of 2.55 and RMSE of 1.15.

We chose TFT for its architecture, which combines attention mechanisms, gating, and variable selection. This design enabled it to track both the magnitude and timing of sudden fee surges. TFT delivered the strongest performance on high-frequency volatility, making it well-suited for fine-grained, urgency-tiered fee forecasts with real-time planning value (see Figure 10).

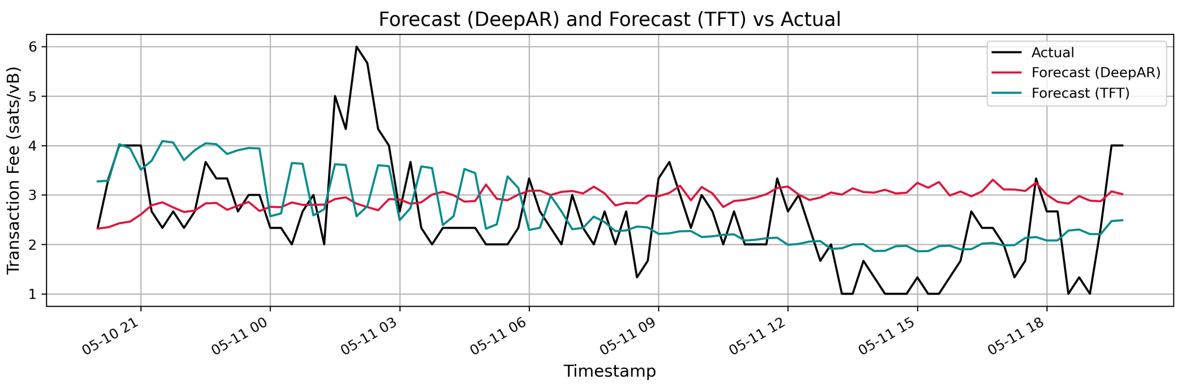

Figure 10: DeepAR and TFT forecasts vs actual fee on test data

Quantitatively, TFT reduced the bespoke volatility-aware loss from approximately 2.43—the best among non-neural models—to 1.78, representing a 27% improvement. It also lowered RMSE from the 1.20–1.42 range to 0.94, yielding a 21% relative gain. MAPE followed the same trend, reaching 0.35 and marking a meaningful 26% gain in relative accuracy. Together, the results show that TFT achieved the strongest balance of accuracy and responsiveness to congestion-driven fee spikes.

4.2 Data Product Overview

The data product consists of three core components: Jupyter notebooks that guide users through EDA, modeling, and findings; Python scripts that define a structured pipeline for reproducible training; and an open-source GitHub repository with clear documentation to support collaboration and long-term extensibility.

Its design is intentionally modular to serve diverse use cases. General users and institutions can rely on the 24-hour forecasts to plan transactions and reduce fee costs. Learners and educators engage with a transparent, step-by-step forecasting walkthrough. Technical developers and our partner benefit from a scalable, reproducible pipeline and workflow.

4.3 Value Proposition and Strengths

Beyond addressing the analytical limitation of forecast horizon by extending it from 10 to 60 minutes to a full 24 hours, we introduced forecasts across multiple urgency tiers (fastest, economy, minimum), giving users actionable insight aligned with cost sensitivity and transaction timing. We also delivered significant value on the development side by emphasizing transparency, modularity, and extensibility. Our scripts, notebooks, and open-source repository enhance accessibility through clear narratives, offer developers greater customization flexibility, and enable community contribution—advancing open infrastructure-grade Bitcoin research in line with our partner's mission.

4.4 Limitations and Design Trade-Offs

Several constraints affect both product's performance and accessibility. TFT powers the pipeline but raises entry barriers, often requiring GPUs or cloud resources. Regular retraining will be necessary as network conditions evolve, but automation is best deferred until model accuracy stabilizes. Confidence intervals and real-time APIs are valuable but not yet justified due to forecast noise and current development priorities. Reactive inputs like mempool congestion and block composition limit foresight and slow model maturity. To remain agile, the script-first design supports rapid iteration, modular updates, and future extensions such as adaptive loss tuning, hybrid pipelines, and scalable deployment.

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 (TFT) is best-suited to capturing irregular spike patterns and outperformed other models in RMSE, MAPE and a the volatility-sensitive loss by 21%–27%. 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 two-month dataset limits exposure to rare but impactful events. Key features that help explain fee behaviors are lagging and reduce predictive lead time. While deep models like TFT are effective, they require regular retraining and high compute, which may constrain deployment. The custom loss function, though aligned with project goals, remains static and could be further tuned for 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

Appendix A: 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.