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 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 models—including classical baselines (HWES, SARIMA), machine learning models (XGBoost, Prophet), and deep learning architectures (DeepAR, TFT)—each suited to different temporal dynamics (Holt 1957; Winters 1960; Box et al. 2015; Chen and Guestrin 2016; Taylor and Letham 2018; Salinas, Flunkert, and Gasthaus 2017; 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 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), captured between March 5 and May 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 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.

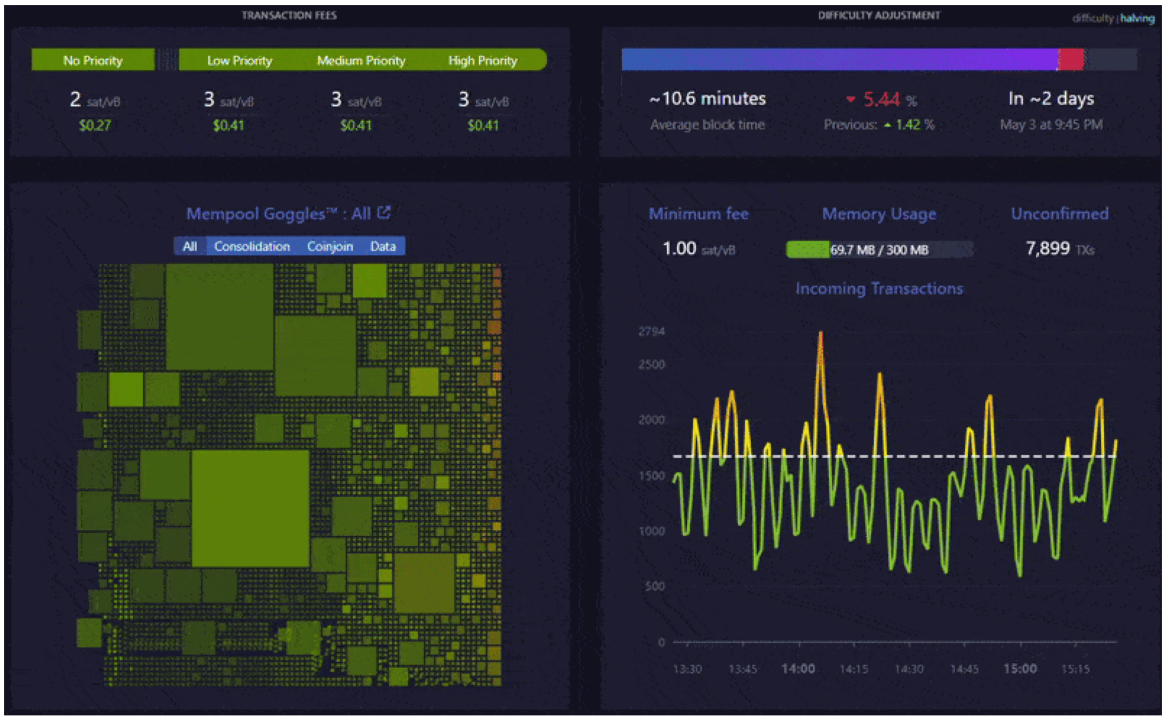

Figure 1: Visual illustration of Bitcoin mempool.space activity.

To better understand the input features used for modeling, we grouped the features in our dataset into four major categories, as shown in @#tbl-features. These categories capture different aspects of network activity and market conditions that influence fee dynamics.

Table 1: Input features from the dataset.

Category	**Description**
Mempool Congestion	Mempool congestion and fee distribution indicators
Block Metrics	Projected mempool block statistics
Difficulty Metrics	Mining difficulty adjustment progress and projection
BTC Price	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 fastestFee 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.

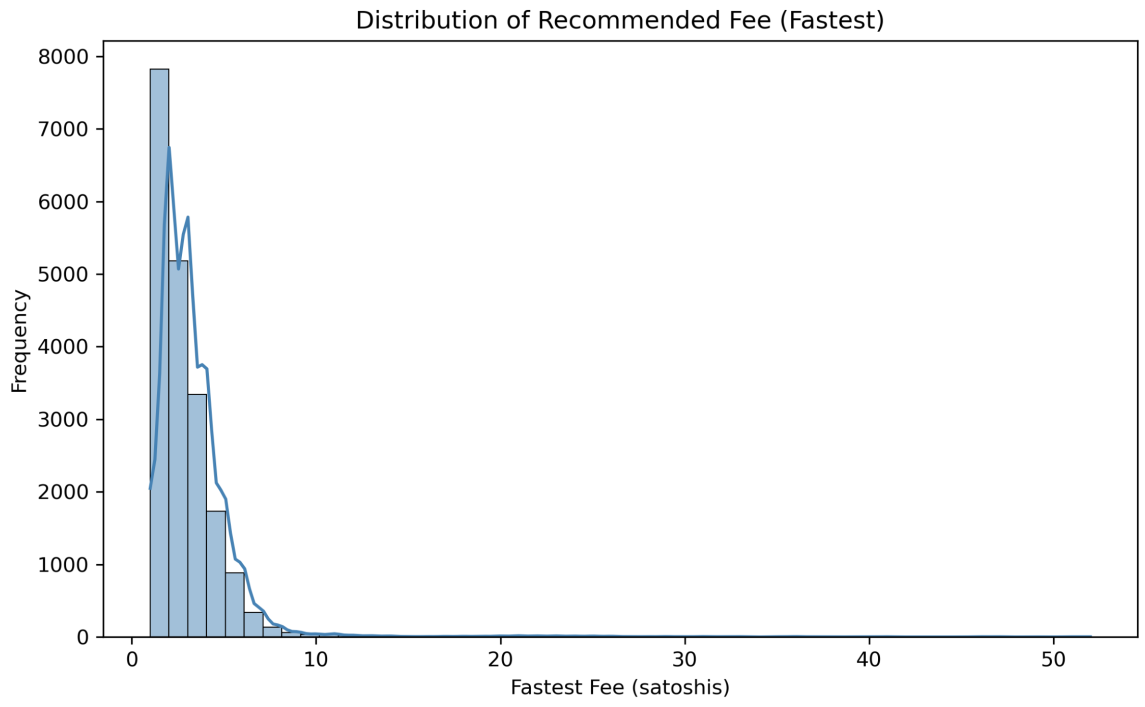

Figure 2: The distribution of fastestFee shows strong right skew.

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. Using the raw granularity is not always optimal: if the interval is too short, the signal between time steps becomes unstable and harder for models to learn; if too long, crucial short-term dynamics may be lost. To strike a balance, we tested multiple resampling frequencies and evaluated their predictive strength using the decay ratio.

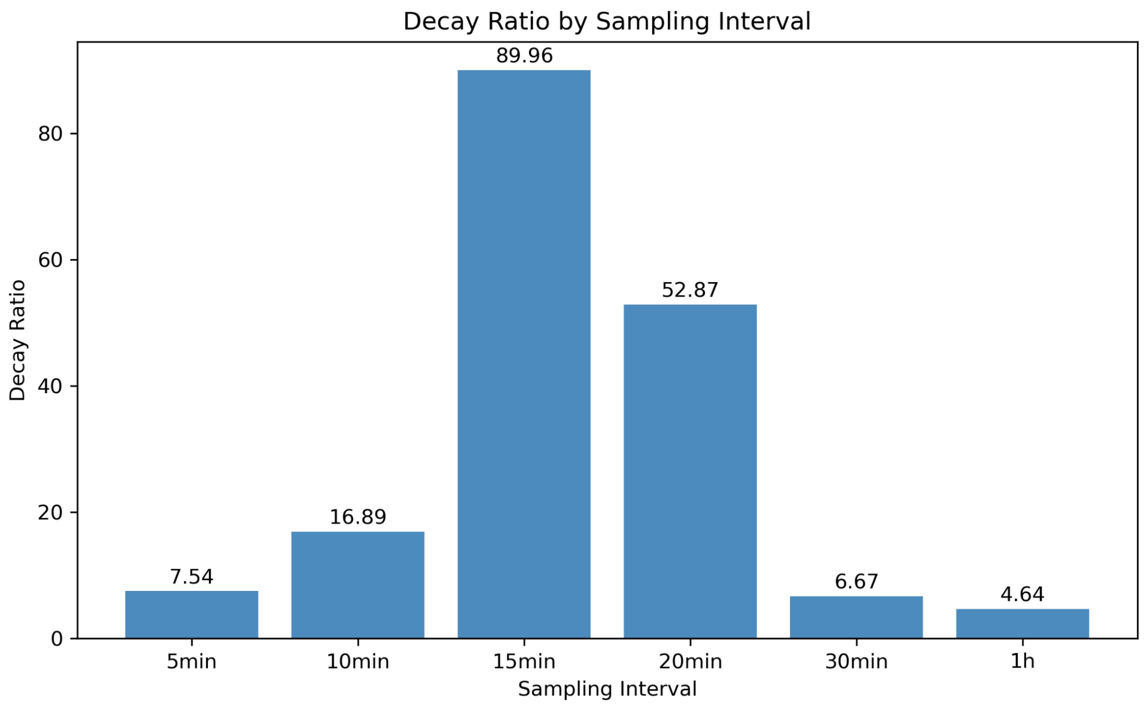

Figure 3: Decay ratio by sampling interval. Higher ratios indicate stronger AR(1)-like structure and better short-term predictability.

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 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 that always predicted the global median fee rate. It could be observed in the distribution of fastest fee rates (Figure 2) that over 92.5% of observed fee rates fell within a narrow band of 2–3 satoshis/byte. Thus, the median itself was already a strong baseline. Although it ignored all temporal or external patterns, we could still quantify how difficult the prediction task really was. No hyperparameter tuning was needed for this static model.

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. This time series method accounted for both trend and periodic fluctuations. We used grid search to determine trend components, seasonal components and trend options. However, autocorrelation analysis of residuals (Figure 5) showed persistant autocorrelation. The result suggested that HWES did not fully capture temporal dependencies.

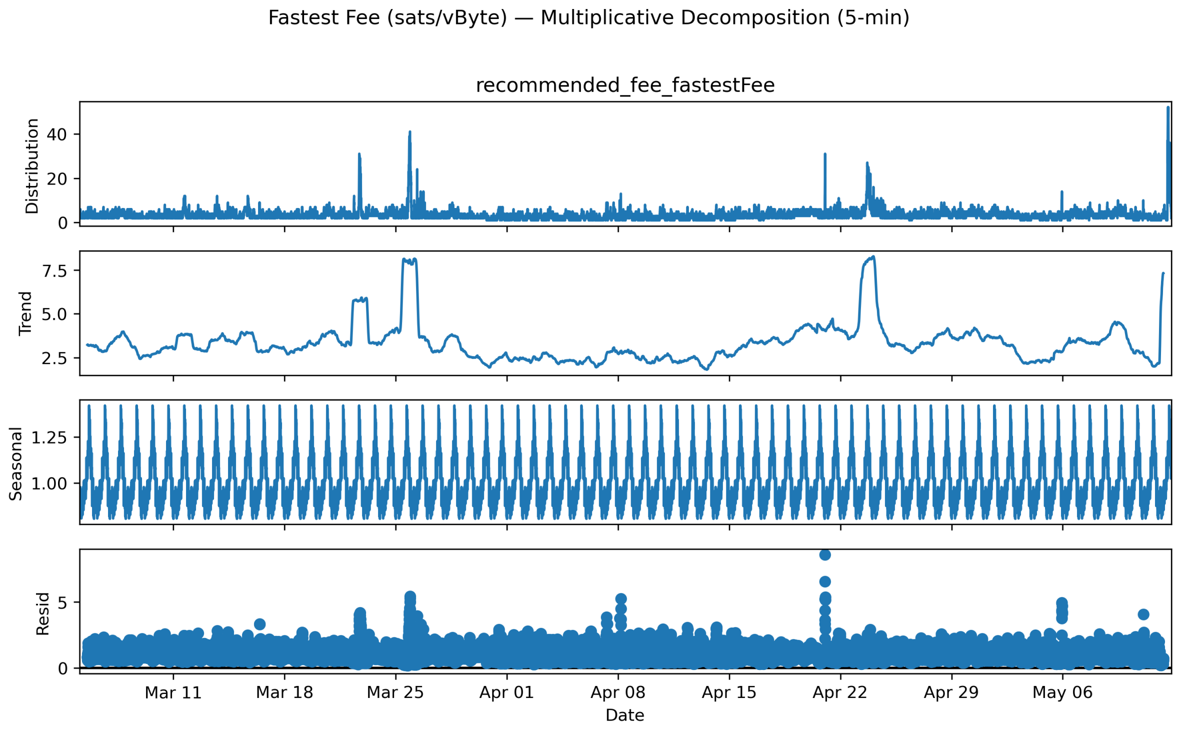

Figure 4: Multiplicative decomposition of recommended fastest fee rate

3.3.3 SARIMA

To address the temporal dependencies missed by HWES, we introduced SARIMA to better handle autocorrelation and lagged temporal structures. SARIMA used autoregressive, differencing and moving average components to model time-dependent patterns. Based on the analysis of EDA including strong autocorrelation at short lags, partial autocorrelation and daily seasonality, we select appropriate hyperparameters manually to capture both short-term dynamics and daily seasonal cycles in the fee series. However, this model prevented us from using exogenous features such as transaction counts or mempool congestion.

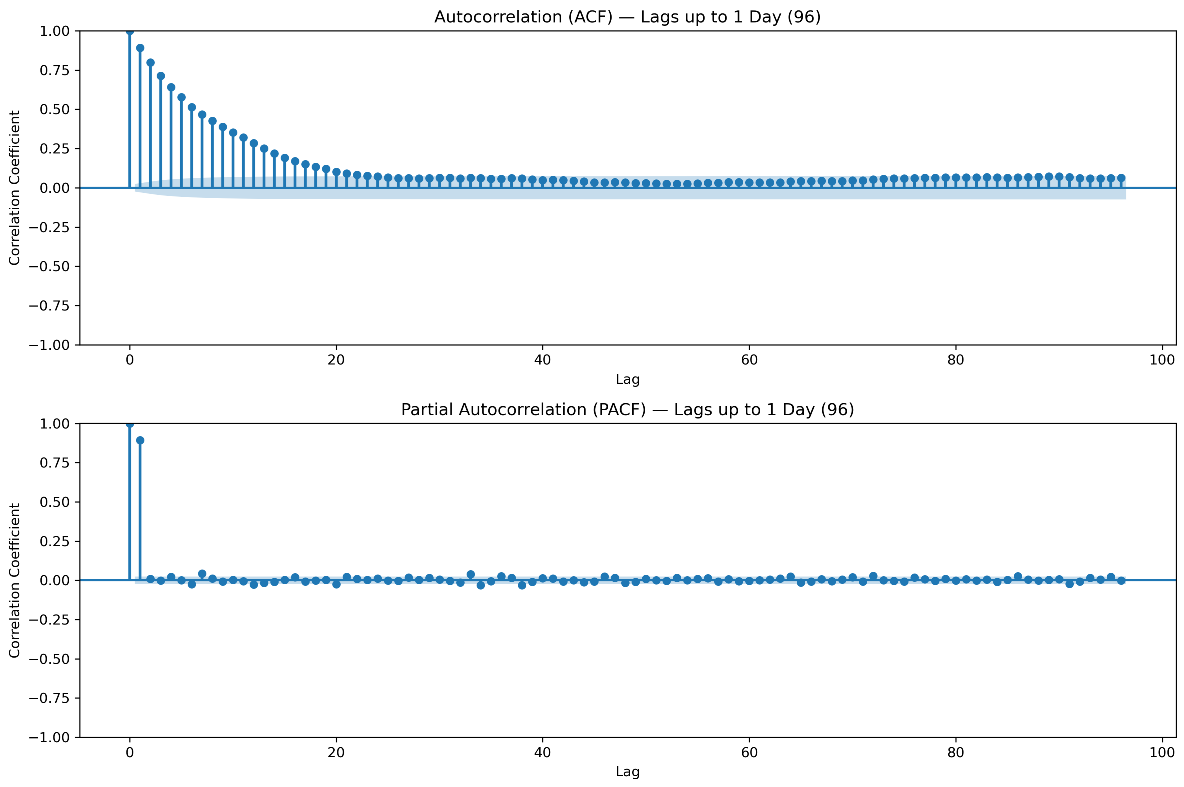

Figure 5: ACF and PACF

3.3.4 XGBoost

To incorporate a richer feature set, we adopted XGBoost. The correlation heatmap (Figure 6) showed strong correlations between recommended fastest fee rate and several concurrent indicators like mempool total fee, mempool count or next block total fee. 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. We performed hyper-parameter tuning using randomized search over tree depth, learning rate and regularization terms.

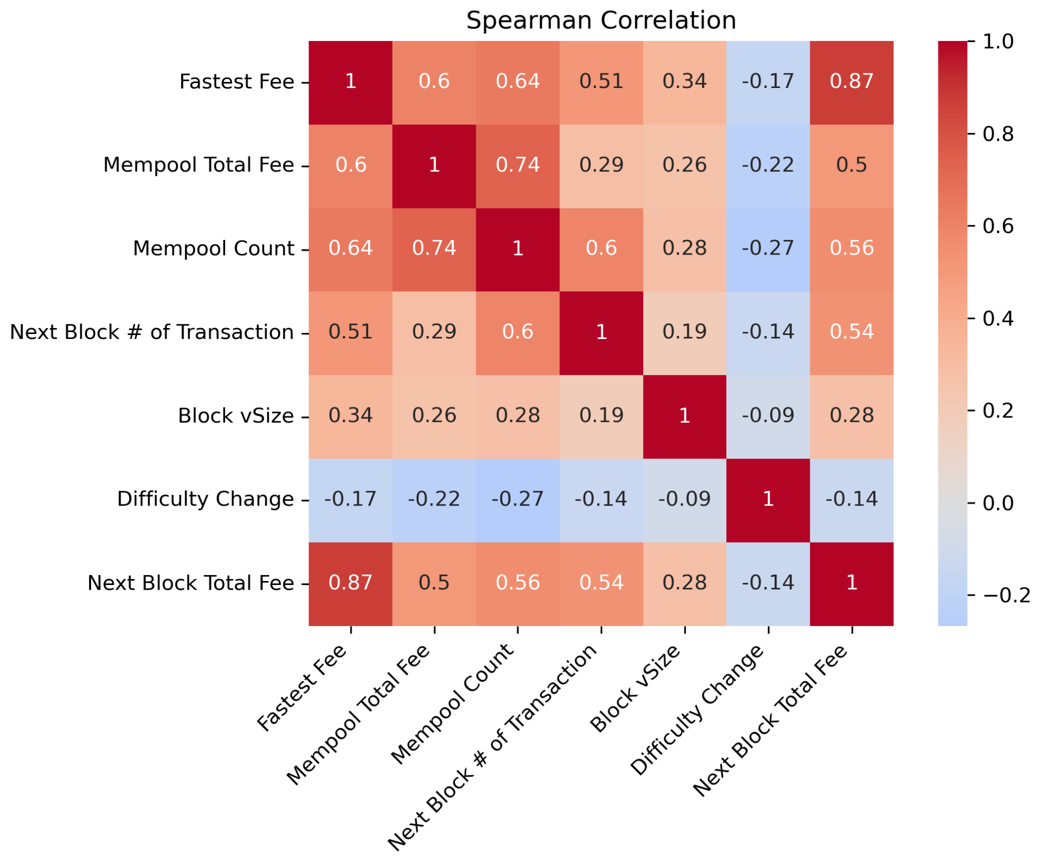

Figure 6: Spearman correlation heatmap

3.3.5 Prophet

Then we moved to advanced models. The first one we tried was Meta's Prophet model. It had the ability to model trend shifts, 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. We configured custom seasonal components including hourly, daily and weekly ones and tuned changepoint and seasonality parameters to adapt to the reactive nature of fee dynamics.

3.3.6 DeepAR

To try more dynamic modeling, we implemented DeepAR. It was an LSTM-based autoregressive forecasting model. In theory, DeepAR should have leveraged temporal context more effectively and handled sequential data better. To optimize the model, we used PyTorch Lightning's "Trainer" to manage training configuration, ran "Tuner" to automatically find optimal hyperparameters and applied early stop to reduce overfitting.

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. We carefully configured the final TFT model architecture, training callbacks, optimization settings and learning rate scheduler to ensure stable training, efficient convergence and robust generalization.

3.3.8 Considered Alternatives and Limitations

A major limitation of our approach is the reliance on lagged exogenous features. They 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 yielded poor results. Multi-stage forecasting also risks compounding errors. Thus, 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 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 7). 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.

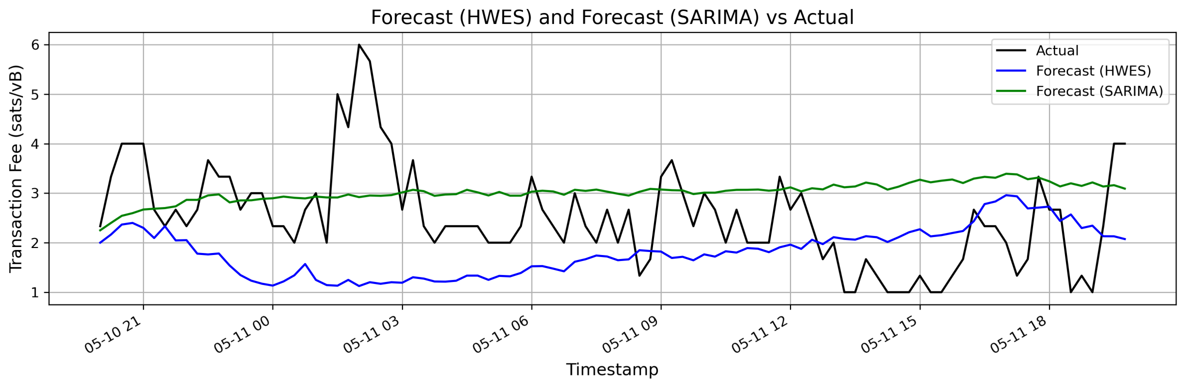

Figure 7: HWES and SARIMA forecasts vs actual fee on test data

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 8).

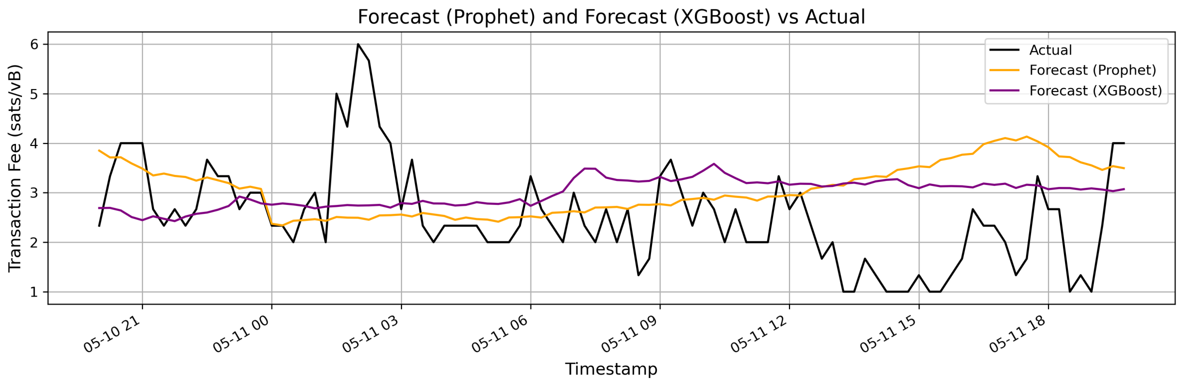

Figure 8: Prophet and XGBoost forecasts vs actual fee on test data

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 9).

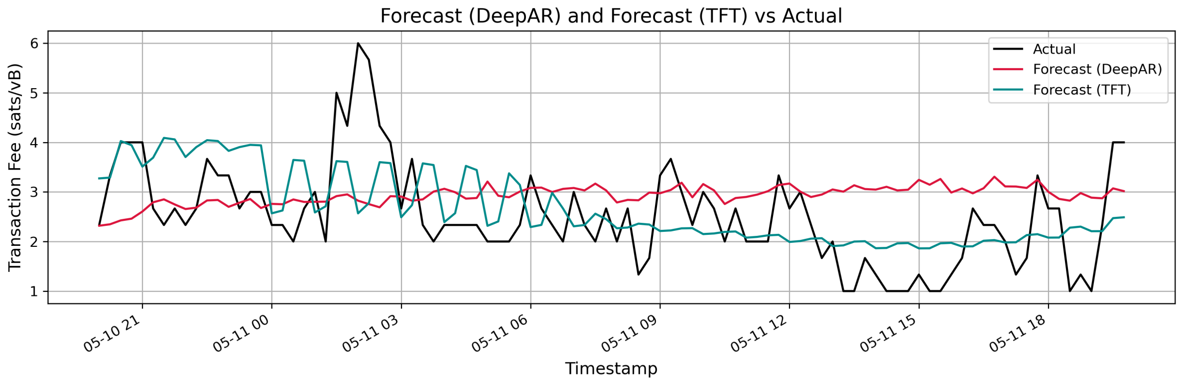

Figure 9: DeepAR and TFT forecasts vs actual fee on test data

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 the key limitations of existing Bitcoin fee estimation tools, which are typically reactive, opaque, and constrained to very short-term horizons. Services like estimatesmartfee(Bitcoin Core Docs n.d.) and Mempool.space(Bitcoin Explorer n.d.) provide only 10–60 minute forecasts and offer single-rate suggestions, with no visibility into volatility drivers or user-specific needs. In contrast, our system produces 24-hour forecasts across multiple urgency tiers (fastest, economy, minimum), helping users better align transaction timing with cost sensitivity. These tiered, volatility-aware outputs provide actionable guidance for both individuals and technical teams, offering flexibility not found in conventional one-size-fits-all tools. We prioritized both usability and transparency. Jupyter notebooks communicate modeling decisions and trade-offs in an accessible, narrative format. At the same time, our modular pipeline structure enables reproducibility and extensibility—developers can swap models, retune parameters, or add new features without reworking the entire system. A custom loss function also aligns model behavior with real-world fee dynamics, improving planning around congestion and urgency. Finally, by releasing the code as an open-source GitHub repository, we invite community validation and collaboration. This transparency supports our partner's infrastructure mission and ensures long-term trust and 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 such as TFT power the pipeline but also raise the cost of entry. Running them almost always requires GPU hardware or paid cloud instances, which puts the system out of reach for many smaller teams and individual developers. On top of that, the models need frequent retraining as Bitcoin network conditions change, so an automated pipeline will eventually be necessary to avoid performance drift. Besides, the feature set is still mostly reactive. It relies on mempool state and recent fee history, but does not yet use forward-looking signals such as exchange flows, protocol-upgrade announcements, or market news. Without those leading indicators, the system can miss abrupt fee spikes. This limitation is felt even more because the forecasts provide only point estimates; users have no confidence bands to guide decisions when volatility is high. Since time and computing resources are tight, the project stays script-first and keeps orchestration to a minimum. It skips tools like Makefiles, end-to-end automation, and real-time APIs so that contributors can iterate quickly. This lightweight backbone makes it easier to swap models, widen hyper-parameter searches, or add new features without overhauling the entire pipeline. When resources allow, uncertainty quantification, adaptive loss functions like fine tuning a best weight, or scalable serving layers can be added on top.

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

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.
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.

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.