

Proposal Report: Forecasting Bitcoin Transaction Fees

Partner: Trilemma Foundation

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

2025-05-09

Table of contents

Introduction	1
Data Overview	1
Data Science Techniques	1
Forecasting Methodology	1
Workflow & Timeline	3
References	3

Introduction

Data Overview

Data Science Techniques

Forecasting Methodology

The overall feature structure reflects both temporal dependencies and external influences, such as time-of-day effects, network congestion, and market conditions. Since our data contains both time-series patterns and contextual signals from network and market activity, we deliberately combine autoregressive, regression-based, and deep learning models to address different aspects of this complexity.

To tackle the challenges of predicting Bitcoin fee rates—like changing patterns over time, external factors, and non-linear behavior—we start with a few simple but solid baseline models. To understand how things like mempool congestion or the day of the week affect fees, we use **Linear Regression**. It's not meant for forecasting, but it gives us a clearer idea of which

external factors are actually influencing the fee rates. Then, to capture how fee rates depend on their own past values over time, we use **ARIMA**, which is well-suited for modeling time-based patterns and serves as a good starting point for temporal forecasting. Since fee rates often show short-term trends and recurring daily patterns, we use **Holt-Winters Exponential Smoothing (HWES)** to capture these seasonal behaviors without needing additional input features. Finally, to deal with more complex nonlinear interactions, we bring in **XGBoost**, a powerful tree-based model that works well with structured data and can handle both historical and external features.

As the project moves forward, we need models that can better deal with the complexity and unpredictability of the data. For example, we want to understand broader patterns like seasonal cycles with holiday effects in fee rates. That's where **Prophet** comes in—it breaks the time series into interpretable parts like trend and seasonality, giving us a clearer picture of the overall structure. While it's more suited for daily data and doesn't respond well to sudden changes, it's still helpful for spotting big-picture patterns. We also need a way to model multiple related time series—like our detailed fee histogram bins—while accounting for uncertainty. **DeepAR** is a good fit for this, since it learns from sequences and produces probabilistic forecasts, which is especially useful in such a volatile environment. Lastly, to make sense of all the moving parts in our high-dimensional dataset, we use the **Temporal Fusion Transformer (TFT)**. It can focus on the most important features at each point in time using attention and variable selection, making it powerful for capturing complex relationships. That said, it's a heavy model that takes more computation and needs careful tuning to avoid overfitting.

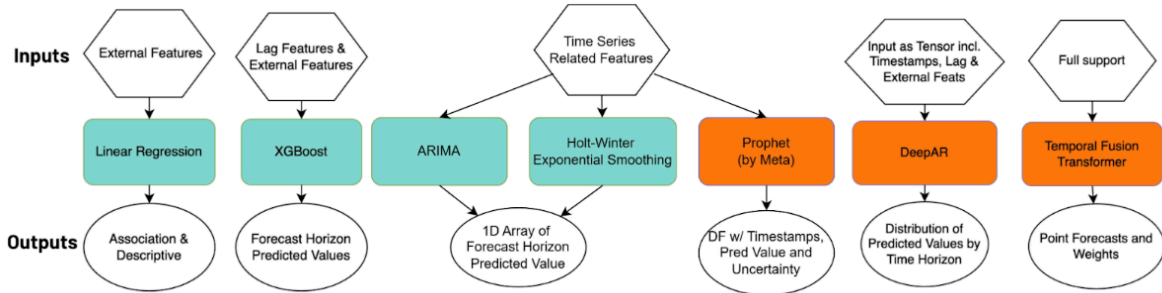


Figure 1: Diagram of model inputs and outputs for baseline and advanced forecasting models.

To assess model performance, we use **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)** as our primary evaluation metrics. MAE provides an interpretable average of prediction errors in the same unit as the target variable, while RMSE penalizes larger deviations more strongly, making it useful for identifying models that are sensitive to extreme fluctuations. For probabilistic models like DeepAR and TFT, we also evaluate the **calibration of predicted confidence intervals** to ensure that they reliably reflect observed

variability in fee rates. This is critical for making informed, risk-sensitive decisions in high-variance network environments.

Workflow & Timeline

References