HASHRATE INDEX PREMIUM

CONFIDENTIAL: FOR HASHRATE INDEX PREMIUM SUBSCRIBERS ONLY

Forecasting Transaction Fees

For Bitcoin Miners, Hosters, Lenders and Hashrate Traders

By: Hamza Ali and Ben Harper

About Hashrate Index

Hashrate Index is a Bitcoin mining data, analytics and research platform. Our platform offers novel data sets that enable miners, traders, content creators, and investors to gain key insights into the mining industry to generate alpha. Hashrate Index is a product of Luxor Technologies, a Bitcoin mining software and services company.

Bitcoin is at once highly predictable and highly unpredictable. With relative certainty, we can measure the days between difficulty epochs and the months between halvings, and Bitcoin's emissions schedule is set in stone. Conversely, we have no idea what Bitcoin's price will be tomorrow or in a year, though we can of course throw out a guess or two.

Similarly, we can't measure demand for blockspace until that demand materializes. But we can devise a method for projecting transaction fees into the future to give miners a better shot at anticipating transaction fee volatility.

The proliferation of Ordinal Theory and the advent of inscriptions on Bitcoin highlight this need. At the beginning of May 2023, a new OP_CODE format sprung to life on Bitcoin, called BRC-20 (in playful reference to Ethereum's ERC-20 standard). The BRC-20 token standard gave inscribers a new way to mint non-fungible tokens on Bitcoin. The activity of "minting" BRC20 tokens caused transaction fees to spike, which in turn led to an ephemeral and unexpected surge in hashprice. Transaction fee bidding wars were so intense that some block rewards were over 12.5 BTC – greater than last halving epoch's block subsidy. Hashprice topped out at $129/PH/day on May 8 from the fee action, a 72% increase from the week prior.

This profitability boost was short lived, though. As transaction fee volume receded and Bitcoin's price slipped from $29,000 to $27,000, hashprice's meteoric rise cratered into a swift decline over the course of the week. Even so, transaction fee volumes are still much higher than they were last year, or even in February and March when the inscriptions mania kicked off.

This volatility has Bitcoin miners, hosting providers, lenders, and hashrate forward traders all wondering what comes next.

In this report we cover:

- How transaction fees function in the Bitcoin network
- The supply and demand factors that affect fees on the Bitcoin network
- How transaction fees have behaved historically
- Models we can use to forecast Bitcoin transaction fees

For Hashrate Index Premium Gold subscribers, we also provide:

- New forecasting methods for Bitcoin network transaction fees
- Updated hashrate supply and demand model projections and sensitivity tables
- Premium Hashrate Index Quarterly reports
- Hashrate Index API

This is the first iteration of our reports for Forecasting Transaction Fees, and we have made this report public to share our research to the mining community for transparency and feedback. Future versions of this report will be available only to Hashrate Index Premium Gold and Platinum subscribers. For more information on our premium research and data, please visit this page.

Executive Summary

To forecast Bitcoin transaction fees, we employ qualitative and quantitative techniques. In particular, we develop:

- Univariate time series forecasting techniques for periods of low activity, fees and volatility,
- Univariate volatility forecasting techniques and multivariate causal methods to signal upcoming periods of high activity, fees and volatility, and
- A case study to look at the development of transaction fees and MEV on the Ethereum network.

These techniques provide readers with a comprehensive collection of effective tools for forecasting Bitcoin transaction fees over the medium-term, enabling them to make more informed decisions. Going forward, Luxor will continue to refine these methods and make it easier for Hashrate Index Premium subscribers to access real time forecasts.

1

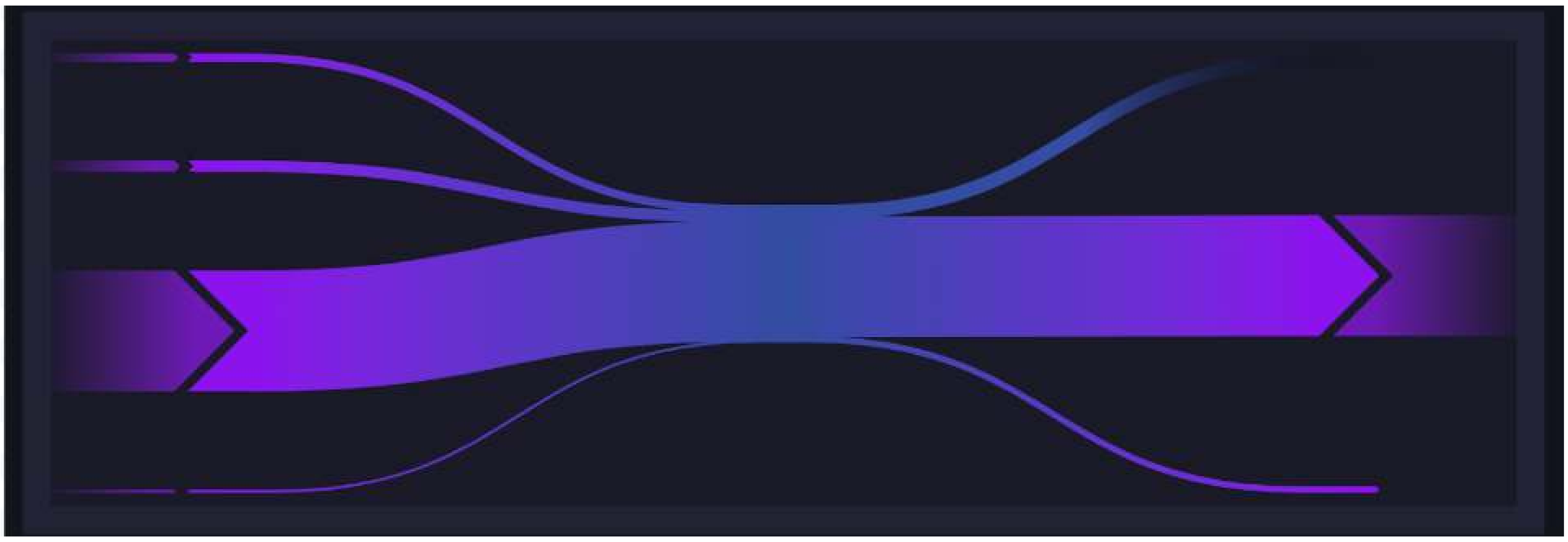
Bitcoin Transaction Fee Overview

What are Bitcoin Transaction Fees?

A Bitcoin transaction is sent, validated, and verified through a decentralized process across the nodes of the Bitcoin network. Anyone can initiate a Bitcoin transaction by providing all necessary information within the transaction data structure and broadcasting it to the Bitcoin network through a Bitcoin node. This involves adhering to the Bitcoin protocol's criteria for a valid transaction and signing the transaction with the required digital signature(s) to unlock the specified Bitcoin funds, known as unspent transaction outputs (UTXOs).

As part of the transaction, the user designates a recipient address for the funds and selects an amount of bitcoin to be sent. It's important to note that the output amount must be lower than, or equal to, the total bitcoin included in the transaction's inputs. **The discrepancy between inputs and outputs in a Bitcoin transaction is the transaction fee, which users incur for sending transactions on the Bitcoin blockchain.**

Simplified Bitcoin Transaction Example



Visualization of a Bitcoin transaction. Screenshot taken from mempool.space

# of Inputs: 4

Inputs = 3.50 BTC

Address 1: **0.10 BTC**
Address 2: **0.20 BTC**
Address 3: **3.00 BTC**
Address 4: **0.20 BTC**

Witness Data (i.e., signatures, etc.)

# of Outputs: 3

Outputs = 3.49 BTC

Address 4: **0.20 BTC**
Address 5: **3.00 BTC**
Address 6: **0.29 BTC**

Transaction Fee = Inputs - Outputs
Transaction Fee = 3.50 BTC - 3.49 BTC
Transaction Fee = 0.01 BTC

When a user wants to send Bitcoin to another address, their node broadcasts the transaction to the network. Other nodes in the network receive the transaction data and verify it against the rules of the Bitcoin protocol. If a transaction is valid, then a node will add it to their memory pool (a.k.a the mempool) and propagate the transaction to the rest of their peer nodes, a process known as gossiping.

Mining nodes, typically mining pools, aggregate transactions from the mempool into blocks and receive the fees associated with those transactions. A mining node can fill a block with up to four million weight units, which is referred to as 1 million vbytes and is equivalent to 4 MB of data. Any transactions remaining in the mempool after a block is filled waits for inclusion in the next block or blocks thereafter. As such, **miners are incentivized to prioritize transactions which pay the largest fee per weight unit in the blockchain.**

Term	Definition
Block Size	Refers to the physical storage capacity that a block or blockchain occupies, typically in MB or GB. This measure is most relevant for node operators.
Block Weight (a.k.a. Block Space)	Introduced after Segwit, this metric refers to the capacity for transactions within each block. It takes into account the impact of certain transaction data, such as witness data used in Segregated Witness (SegWit) transactions. Block weight is measured in weight units, and it can differ from the block size depending on the types of transactions included. This measure is most relevant for transaction fee markets and is the primary focus of this paper.

Prior to the BIP 141 soft fork in July 2017, titled “Segregated Witness (Consensus layer)” and commonly called Segwit, Bitcoin had a 1 MB block size limit. Though the reason was never stated publicly, this 1 MB limit was quietly introduced by Satoshi Nakamoto in late 2010. Some have speculated that it was intended as an anti-spam measure to prevent malicious miners from overloading the network with extremely large blocks full of artificial transactions, or as a temporary measure while the network grew. At the time, blocks were much smaller than 1 MB, so the limit was not a binding constraint.

The Bitcoin block size wars refer to the contentious debate within the Bitcoin community regarding the appropriate block size limit for the cryptocurrency's blockchain. The conflict primarily occurred between 2015 and 2017 as Bitcoin's popularity grew and transaction congestion became a concern. Advocates for a larger block size argued that increasing the block size would alleviate congestion and enable faster and cheaper transactions. On the other hand, supporters of the status quo emphasized the importance of decentralization and the centralization risks associated with larger blocks. The intense disagreement led to heated discussions and the emergence of competing coins via hardforking Bitcoin’s blockchain.

Introduced in 2015 and activated in 2017, Segwit was a compromise in a bid to address Bitcoin's scalability dilemma.[1] This involved splitting transaction data into two segments – the sender and receiver data would stay in their original section in the block, but the signature data (what's called the "witness") would be segregated to a different section of the block (hence Segregated Witness, or Segwit). Segwit introduced a 4 million weight unit limit whereby the original 1 MB data segment counted as 4 million weight units and witness data, limited to 4 MB, would be counted as one weight unit equaling one byte of data. In effect, this increased the block size limit from 1 MB to 4 MB. Block weight is also commonly measured in vbytes, whereby 1 vbyte is equivalent to 4 weight units.

The formulas below show the post-Segwit block weight limit.

$$4 \text{ million weight units} = 1 \text{ million vB}$$

$$4 \text{ million weight units} \geq \text{Block Weight}$$

$$4 \text{ million weight units} \geq MB_{\text{Original Data Section}} * 3 + MB_{\text{Original Data Section and Witness Data}}$$

$$4 \text{ million weight units} \geq MB_{\text{Original Data Section}} * 3 + (MB_{\text{Original Data Section}}) + (MB_{\text{Witness Data}})$$

$$4 \text{ million weight units} \geq MB_{\text{Original Data Section}} * 4 + MB_{\text{Witness Data}}$$

Taproot (BIP 340, 341, and 342), activated in November 2021, enhances data efficiency and privacy for certain Bitcoin transactions, such as multi-signature transactions. Taproot transactions replace Bitcoin's Elliptic Curve Digital Signature Algorithm (ECDSA) for signing transactions with Schnorr signatures; this enables, among other things, Bitcoin users to aggregate multiple public addresses under a single signature. This lowers the data load needed to execute multi-signature transactions (instead of having to store signatures for 5 addresses in a 5-of-7 multi-sig quorum, for example, now nodes only need to log 1 signature). Additionally, Taproot multi-sig and single-sig addresses use the same address format. Right now, Segwit-enabled and other multi-signature addresses are longer than their single-sig counterparts, so Taproot also gives multi-sig users a privacy boost.

Taproot could also pave the way for new smart contract implementations on the Bitcoin network. It provides a more flexible framework for executing complex spending conditions, allowing for the creation and execution of programmable transactions in a more data-efficient way than previously possible. This opens up possibilities for decentralized applications and innovative financial instruments, expanding the capabilities of the Bitcoin network.

A testament to how Taproot's data efficiency gains can usher in innovations, inscriptions blossomed in Q2 of 2023 largely thanks to Taproot's transaction improvement. Specifically, the inscription content costs much less to transact when stored within Taproot's scripts; even though inscriptions were technically feasible with Segwit, Taproot makes them more data efficient and thus cheaper.

[1] It was also a solution for transaction malleability, whereby unconfirmed transaction identifiers could be changed without invalidating the transaction.

From an economic perspective, transaction fees on the Bitcoin network serve multiple purposes. Firstly, users can adjust their fees accordingly to prioritize their transactions over others in the mempool and thus speed up transaction settlement. Secondly, transaction fees act as a deterrent against spam and denial-of-service attacks by imposing a cost barrier on users. This discourages malicious actors from flooding the network with unnecessary or harmful transactions. Lastly, as the block subsidy decreases, transaction fees become a crucial long-term funding source for miners for ensuring the security and sustainability of the Bitcoin network.

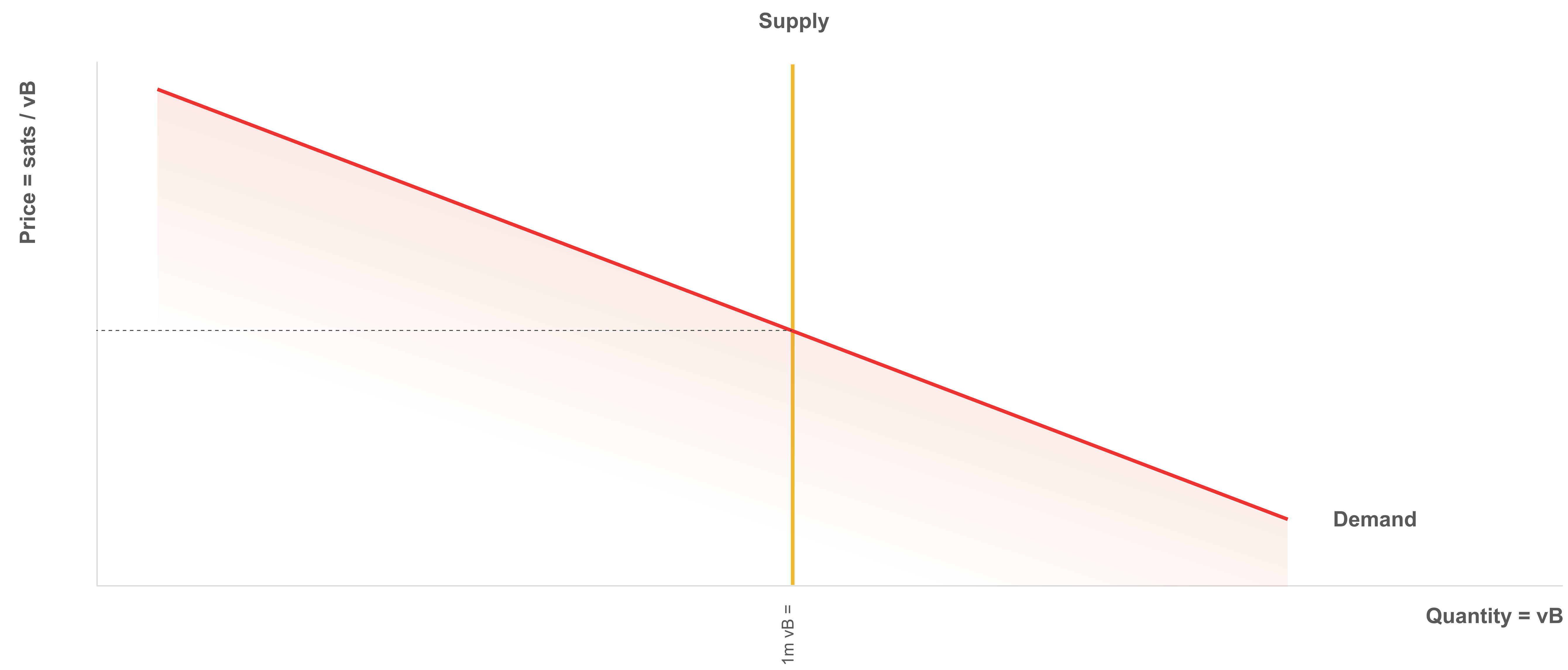
Transaction fees are an under-appreciated and increasingly important component of hashprice – the revenue miners earn for selling hashrate. Together with the block subsidy, they determine how much miners can earn from constructing blocks. Bitcoin's subsidy started at 50 BTC per block and is halved every 210,000 blocks (roughly 4 years), until block 6,930,000 (projected in 2140), **when the block subsidy becomes zero and mining revenue will come exclusively from transaction fees.**

As transaction fees become an increasingly important part of Bitcoin mining economics, understanding transaction fee dynamics and forecasting them into the future becomes critical for hashrate and hashprice market participants.

What Determines Transaction Fees on the Bitcoin Network?

Bitcoin's transaction fees are determined in the open market for block weight. As in any other open market, prices and quantities – in this case fees and vbytes – are determined by the economics of supply and demand. The determinants of Bitcoin transaction fees have been explored in some academic and professional literature.[2] For immediate and forward looking purposes, we focus on a basic supply and demand model to form an understanding of the current block weight market structure.

Economics of the Block Weight Market



Supply of Block Weight

The supply of block weight is determined by Bitcoin's consensus code. The block weight limit is represented by the vertical line at 1 million vbytes on the x-axis.

Assuming the mining market is competitive and transparent, if block weight is not at capacity, then miners have an incentive to include transactions with any positive fee in the next block. Conversely, there is no incentive for Bitcoin senders to pay a fee above the bare minimum if block weight is available.

With a fixed block weight limit, the marginal cost to miners with a full block is the opportunity cost of foregoing the lowest fee rate per vbyte. That is, miners are incentivized to place the highest bid per vbyte into their block.[3] If a user would like a transaction included in the next block, they must outbid other transactions to incentivize a miner to include their transaction in the blockchain.

[2] Literature on the determinants of transaction fees includes models and empirical evidence based on competitive and non-competitive block weight market dynamics, auction protocols, queuing theory, and social norms and convention arising from the default software settings of major wallet softwares or actions of large intermediaries in Bitcoin's early years. See for example, Houy (2014); Moser and Bohme (2015); Lopp (2016); Easley, O'Hara, and Basu (2017); Kasahara and Kawahara (2017); Huberman, Leshno, and Maollemi (2019); Li, Yuan, and Wang (2020); Tsang and Yang (2021); Iik, Shang, Fan, and Zhao (2021); Lehar and Parlour (2021); Fan, Ilk and Shang (2022); Burnett and Rochard (2022); Brown, Chiu and Koeppl (2022); Kim, Ryu and Webb (2023).

[3] Unless they are filtering for compliance or regulation.

In the short to medium term these supply parameters are fixed. However, in the longer term, hard forks and soft forks could change the supply of Bitcoin's block weight and minimum transaction fee. For example, a 0.01 BTC minimum transaction fee was implemented in 2010 as a deterrent to "spam" transactions, but was later removed as transaction volume and Bitcoin's price increased. Currently, a standard Bitcoin node will only relay transactions that have a fee rate greater than one sat/vbyte.

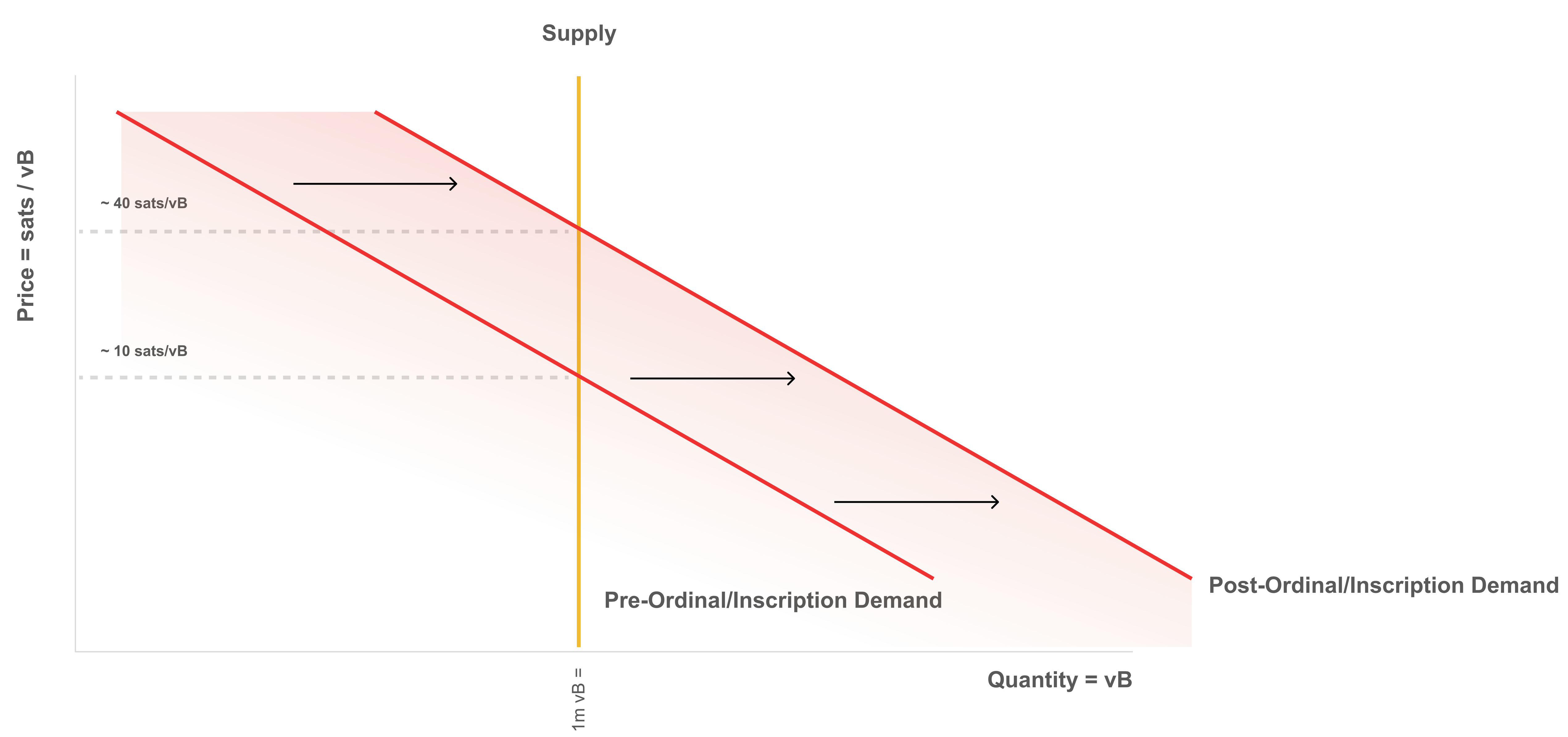
Demand for Block Weight

Demand for block weight on the Bitcoin network is influenced by the demand for transactions and the time users are willing to wait for confirmation. The level of time preference, whether low or high, plays a crucial role in determining transaction fees. Low time preference demand – where users are patient and willing to wait longer for confirmation – generally does not drive fees. By contrast, high time preference demand, where users require quick settlement, tends to be the primary factor driving block weight demand, particularly in the short term.

Historically, users have generated Bitcoin transactions predominantly for self-transfer of funds, payments, or trading. The more demand for transactions on the Bitcoin blockchain, the more demand for block weight all else being equal. Further, the more users who demand quick transaction settlement, the more demand for immediate block weight all else being equal.[4] At the exchange level, inflow and outflow volumes and transaction batching will have an impact on demand for block weight from trading activities, as do integrations of layer-2 technologies like the Lightning Network and Bitcoin sidechains.

Demand for block weight can also come from other sources. For example, discrete-log-contracts (DLCs) and DeFi impact the demand for block weight on the Bitcoin blockchain. More recently, ordinals and inscriptions opened up new demand for Bitcoin block weight and caused transaction fees to spike – most notably during the BRC-20 frenzy in early-to-mid-May this year. The diagram below illustrates how the introduction of ordinals/inscriptions impacted fee markets.

Economics of the Block Weight Market: Ordinal/Inscription Impact



[4] User need for quick settlement, for example, drove the BRC-20 frenzy in early- to mid-May 2023.

Soft forks like SegWit and Taproot can impact demand for block weight as well. For example, SegWit introduced a change in how the block weight was calculated, which reduced the weight of each transaction. This effectively results in reduced demand for block weight per transaction.

We realize this conclusion may strike some readers as counterintuitive, but our reasoning is: **SegWit reduces the amount of block weight a transaction requires. By reducing the effective weight of each transaction, that user's demand for block weight is reduced.**

While Segwit increased the total supply of block size in MB, which is important for node operators, in the block weight market where transaction fees bid for weight, it manifests as a reduction in demand for said weight. The supply of block weight was unchanged after the implementation of SegWit.

Before SegWit, the block size limit in Bitcoin was fixed at 1 megabyte (MB). This limited the number of transactions that could be included in each block, leading to congestion during periods of high demand and resulting in higher transaction fees. The fees were primarily determined by the size of the transaction in bytes.

With the implementation of SegWit, the block weight concept was introduced. While the block size calculation remained the same, the weight of transactions was calculated differently. SegWit transactions separated the signature data (witness) from the transaction data, and the weight of the witness data was discounted.

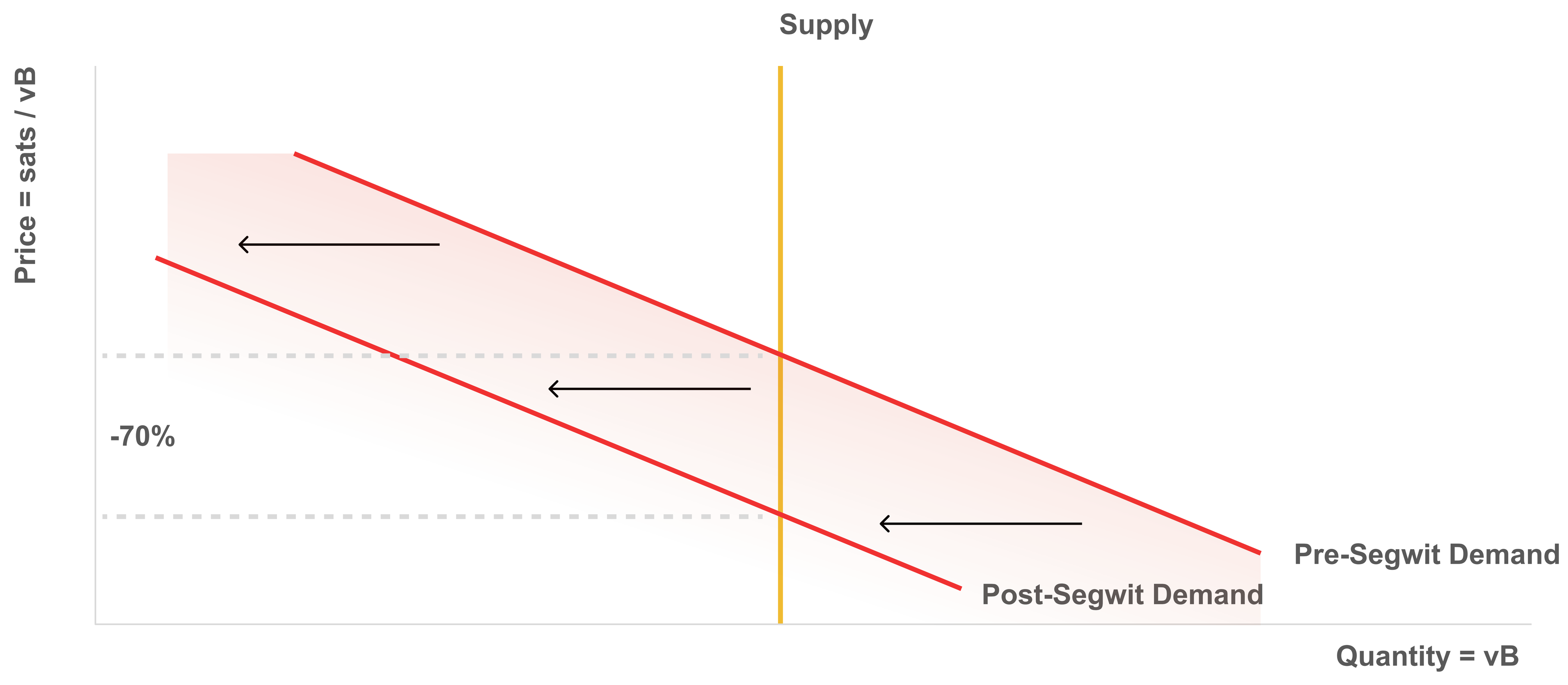
Reduction in Block Weight Demand for the Same Size (MB) Bitcoin Transaction

	Pre-Segwit	Segwit
Original Data in MB	1.00	0.33
Witness Data in MB	0.00	0.67
Size in MB	1.00	1.00
Weight in vMB	4.00	2.00

Where Weight = 4 x Original Data + Witness Data

The discounting of the witness data in the block weight calculation effectively reduced the demand for block weight. Users can fulfill the same demand for transactions using less block weight. This improved efficiency led to an increase in transaction capacity and a decrease in transaction fees for SegWit transactions.

Economics of the Block Weight Market: Segwit Impact



Brown, Chiu and Koeppl (2022) estimate that Segwit reduced transaction fees by ~70%

Given the market structure of block weight, with no minimum fee and a fixed supply cap, transaction fees fluctuate between periods of low activity and low fees, and periods of higher activity with volatility and spikes in fees. This occurs because when supply is perfectly inelastic, similar to real estate or collectibles markets, changes in price do not lead to changes in the quantity supplied. As a result, prices become more responsive to shifts in demand and cannot be stabilized by producers adjusting production levels.

In the following sections we observe this pattern in transaction fee data and develop forecasting techniques to take advantage of our knowledge of this market structure.

Summary of Supply and Demand Factors' Impact of Transaction Fees[5]

Variable	Impact of a Variable Increase on Transaction Fees
Supply of:	
Block Weight Limit	Decrease
Minimum Transaction Fee	Increase / No Change[6]
Demand for:	
Payments	Increase
Trading	Increase
Quicker Confirmation Times	Increase
Smart Contracts / DeFi	Increase
Ordinals / Inscriptions	Increase
Segwit / Taproot Adoption	Decrease
Layer 2 Adoption	Decrease

[5] The impact on transaction fees to a change in each variable is evaluated ceteris paribus.
[6] Depends if the minimum transaction fee is above or below the market rate.

A Note on Maximal or Miner Extractable Value (MEV) on Bitcoin

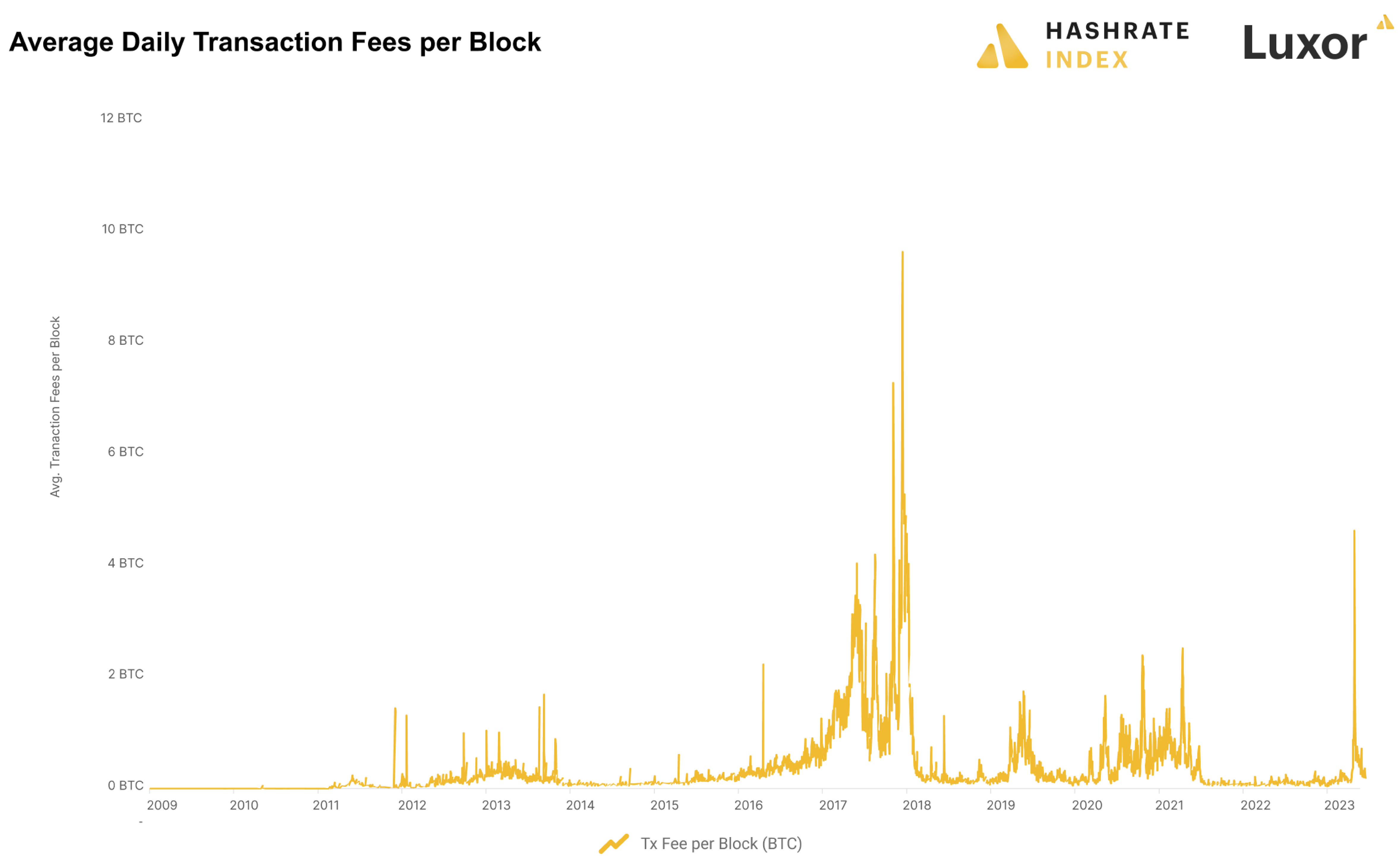
Maximal or Miner Extractable Value (MEV) refers to the potential profits that miners can extract from reordering, front-running, censoring, or otherwise strategically manipulating transactions within a blockchain network. Borrowing a real estate analogy from Sreeni and Zhang (2022), "being neighbors with a suddenly attractive piece of real estate or auction may be extremely valuable." That is, not all block space is valued equally.[7]

MEV played a crucial role in mining revenue on the Ethereum network prior to the merge. It encompassed various activities such as decentralized exchange arbitrage, liquidations, sandwich trading, and more. The presence of MEV opportunities largely depended on decentralized finance applications built on the blockchain's underlying layer.

It is important for participants in the hashrate market to monitor developments in MEV on the Bitcoin network. The emergence of DLC (Discreet Log Contracts) and token standards like BRC-20 has led to instances of MEV activity, although it remains relatively insignificant at present. An example of this was observed when F2Pool filtered transactions from other Stacks miners intending to send BTC to STX staker addresses, replacing a low-value transaction with their own BTC bid. This case, covered by Blockworks in a thread, highlights just one instance of MEV in the Bitcoin ecosystem.

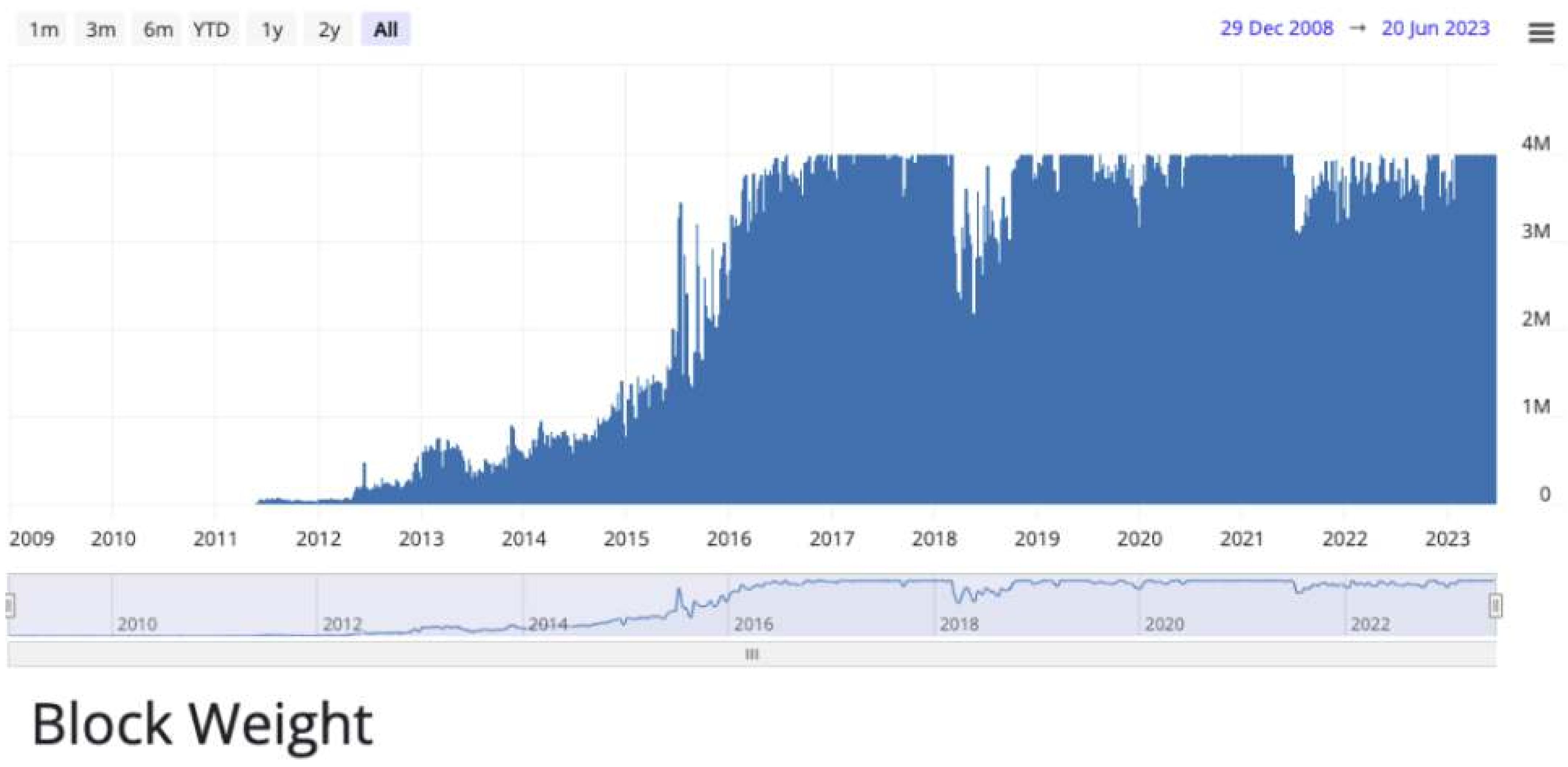
In a Miner Extractable Value (MEV) environment, block construction can be decoupled from mining, allowing for the different entities to perform either task. Transaction ordering relays like Flashbots in Ethereum act as intermediaries between miners and MEV traders (sometimes called "searchers"), facilitating MEV-related activities. This MEV environment presents new opportunities for miners to extract additional value from transactions and can become a significant source of revenue.

How Have Transaction Fees Behaved Historically?



[7] We highly encourage readers to check out Sreeni and Zhang (2022) and Zhang and Konstantopoulos (2021) to learn more about MEV economics in the market for block space.

The previous chart shows the history of transaction fees on the Bitcoin blockchain, from inception on January 9, 2009 to May 31, 2023.



Source: Bitcoinvisuals.com

Pre-mid 2016, as we can see in the block weight chart above, Bitcoin's block weight (block size at the time) was not at capacity and fees were miniscule. As documented by Moser and Bohmer (2015) and Lopp (2016), fees were positive and fluctuated not due to market congestion, but social norms and convention arising from the default settings of major wallet softwares or actions of large intermediaries.

Since mid-2016, Bitcoin transaction fees have experienced periods of low activity and low fees, as well as periods of higher activity with volatility and spikes in fees. These spikes occurred during 2017-2018, 2020-2021, and May 2023. The volatility can be explained by the following factors:

- **Bitcoin price boom and bust cycles** are closely tied to changes in demand for Bitcoin transactions. As transaction volume surpasses the network's capacity, users are required to pay higher fees in order to prioritize their transactions. We can observe this phenomenon during significant market upswings, such as the boom in 2017 and the pandemic period from 2020 to 2021, when the demand for Bitcoin was particularly high. During these times, the surge in demand pushed transaction fees upwards at a rapid pace. Conversely, when monetary policy shifted after these periods, leading to a decrease in demand for Bitcoin, transaction fees also decreased accordingly.
- **Protocol updates, technological advances, and technical standards** that increase transaction processing capacity can lower fees. For example, Segregated Witness (SegWit) separates transaction data from signature data, allowing more transactions in each block. Conversely, technical standards like the ordinals sequencing system, inscriptions, and BRC-20 tokens can precipitate activity that increases fees.

Below is a table describing daily average transaction fees per block by year, days spent below and above the average, and other relevant statistics. The data ranges from June 15, 2016, to May 11, 2023, focusing on recent periods of bitcoin transaction activity.

Year	Mean	Standard Deviation	Minimum	Maximum	Range	Number of days above or equal to mean (for that year)	Number of days below mean (for that year)
2016	0.502	0.126	0.147	2.237	2.09	83	117
2017	1.834	1.240	0.417	9.651	9.234	119	246
2018	0.449	0.873	0.079	4.905	4.826	52	313
2019	0.367	0.309	0.075	1.752	1.677	115	250
2020	0.515	0.414	0.057	2.400	2.343	149	217
2021	0.423	0.424	0.047	2.532	2.385	151	214
2022	0.103	0.046	0.032	0.315	0.283	145	220
2023	0.288	0.567	0.044	4.641	4.597	18	113

We note a few trends from the table. In 2017, transaction fees were significantly higher due to various catalysts, making it an outlier year. The average transaction fee in 2017 was almost 3.6 times larger than any other year. The extreme volatility during that year, reflected in the standard deviation and range, also exceeded other years. Conversely, 2022 had the lowest volatility and the lowest mean transaction fee.

Additionally, there is a consistent trend across the years, where the number of days with transaction fees below the mean is much higher than the days above the mean. This pattern holds for each year, with some years (like 2018 and 2023) showing a significant difference, where the number of days below the mean is more than six times greater than the days above the mean.

2

Forecasting Bitcoin Transaction Fees

At Luxor, we recognize that successful forecasting entails more than simply selecting and refining a single model. It involves employing a diverse set of models to gain a comprehensive understanding of the forecasted variable and the potential range of outcomes. The objective of this report is to provide readers with a comprehensive collection of the most effective tools for forecasting Bitcoin transaction fees, enabling them to make informed and accurate predictions.

This report concentrates on predicting average Bitcoin transaction fees for monthly and quarterly durations, aligning with the medium to long-term planning and decision-making processes of Bitcoin miners, hosters, investors, and hashrate traders. Although there is limited literature specifically dedicated to forecasting fees over these periods, it is worth noting the abundance of online tools and features within Bitcoin wallets that facilitate short-term fee predictions. These tools predominantly utilize mempool data to aid users when they send transactions.

In our paper earlier this year on forecasting network difficulty, we described three general methods commonly used in forecasting. They are:

1) Qualitative Techniques

2) Time Series Models (Univariate Models)

3) Causal Models (Multivariate Models)

We encourage readers to refer to Harvard Business Review's article on selecting optimal forecasting techniques for a deeper understanding of the techniques discussed in this section.

Qualitative Techniques

Qualitative techniques are an approach to forecasting typically utilized in situations where data is limited or when quantitative methods are hindered by resource constraints. These methods rely on expert opinions, human judgment, and insights regarding significant events to transform qualitative information into quantitative estimates.

The qualitative method provided in this paper is a case study where we will look at the development of transaction fees on the Ethereum network. The case study can be found in Appendix 1.1-1.2 while the conclusions of the case study are outlined below.

Conclusions from Ethereum Network Case Study

From our case study, we draw some general conclusions:

1. As with DeFi/NFTs on Ethereum, ordinals and inscriptions open up possibilities for Bitcoin beyond current use cases, many of which are yet to be discovered.
2. There will be periods of low activity and low volatility, where transaction fees are more easily forecastable.
3. There will be temporary periods of high fees and high volatility that are event driven. For example, hype around one random meme-coin, NFT release, or project might cause very short-lived spikes in transaction fees (e.g., BRC-20). More value-driven uses of the crypto network such as Defi create the possibility of more sustained but less volatile periods of high fees.

The points above illustrate the difficulty with forecasting transaction fees. Although transaction fees can be low and non-volatile a majority of the time, which makes them easier to predict during such conditions, it's very difficult to predict when they will spike dramatically. Luckily for Hashrate Index Premium subscribers, our forecasting methods and partnership with the Ordinal Hub team may help catch unexpected surges in transaction fees emanating from upcoming projects.

But how can we forecast Bitcoin transaction fees using quantitative techniques? In the following sections we evaluate univariate time series and multivariate causal forecasting techniques. We hypothesize:

1. It is possible to accurately forecast average Bitcoin transaction fees for the next month or quarter during periods of low volatility and stability; and
2. It is possible to develop signals to catch upcoming spikes in transaction fees using more advanced multivariate causal models and/or volatility based forecasting methods.

Quantitative Techniques: Univariate Time Series Forecasting

One of the most simple but fundamental phenomena in the field of forecasting is that variables tend to be correlated with past values. This phenomenon is called autocorrelation. The direction or pattern of a variable's past values (better known in the field as "lags") can help predict its future values.[8]

Autocorrelation gave birth to Autoregressive and Autoregressive Moving Average models – more commonly referred to as AR and ARMA models. Below is a brief explanation of these models, with an appendix section for our more interested readers which provides further details about the models, including tests, assumptions, and lag selection methods. Univariate models only include an analysis of the variable of interest itself and its past values (i.e., with no other variables included).

[8] From this point forward, our paper will refer interested readers to a detailed appendix that provides further information on the equations and assumption tests conducted for these models (including trends, stationarity, seasonality, etc.).

AR and ARMA Models

Autoregressive (AR) models rely on forecasting a variable based on a linear combination of its previous values. This method assumes current and future values are linearly dependent on past values and if the relationship is strong enough, past data on transaction fees can be used to forecast future transaction fees. The general equation for an Autoregressive model of order p, denoted as AR(p),[9] can be expressed as:

$$TXF_t = c + \Sigma(\alpha_i * TXF_{t-i}) + \varepsilon_t$$

Where,

- ***TXF_t represents transaction fees at time t,***
- ***c is a constant term***
- ***α_i represents the coefficients corresponding to the previous values TXF_{t-i},***
- ***ε_t is the error term at time t.***

Appendix 2.1 gives a further explanation of the equation above and AR models in general.

An Autoregressive Moving Average (ARMA) model can enrich the simpler AR model by including a Moving Average (MA) element which combines the linear dependence of past values (AR element) with the influence of past error terms (MA element). The error term, also known as the residual term, represents the part of the observed data that is not explained by the autoregressive component of the model, measuring the discrepancy between the predicted values of the model and the actual observed values.

By including the error term in the model, ARMA models can capture the random or unpredictable nature of a time series, potentially providing a more accurate representation of the data. The general equation for an ARMA model of order (p,q),[10] denoted as ARMA(p,q), can be expressed as:

$$TXF_t = c + \Sigma(\alpha_i * TXF_{t-i}) + \Sigma (\beta_j * \varepsilon_{t-j}) + \varepsilon_t$$

Where,

- ***β_j represents the coefficients corresponding to the past error terms ε_{t-j}***

Appendix 2.2 gives a further explanation of the equation above and ARMA models in general. A review of model selection methods and assumption tests are provided in Appendix 2.3

[9] The parameter "p" represents the order of the autoregressive model, indicating the number of lagged observations used to predict the current value.

[10] In addition to having the parameter "p" representing the order of the autoregressive model, ARMA models also include parameter "q" representing the order of the moving average component, indicating the number of lagged forecast errors used in the model.

Model Considerations and Evaluation

For the purposes of this paper, models and their coefficients will be based on a training set which spans January 2009 to September 2022 for the univariate time-series models and from June 2016 to September 2022 for the multivariate models.[11] All the models were evaluated over a testing set from October 2022 to May 2023.

This testing set from October 2022 to May 2023 was selected for two primary reasons. First, it is customary to use the most recent period as the testing set for models. Second, this timeframe encompasses both a period of low and stable transaction fees, as well as the sudden spike in fees that occurred in May 2023. This enables us to evaluate the performance of our models in different scenarios. The training and testing sets derived from this period are utilized in the subsequent analysis of multivariate causal models discussed later in the paper.

Mean absolute error (MAE) was used to determine forecast accuracy. It is calculated by taking the average of the absolute difference between actual transaction fees and a model's forecasted transaction fees. The mathematical equation for MAE is:

$$MAE = \frac{1}{N} \sum_{i=1}^{n} TXF_i - \widehat{TXF_i}$$

For exogenous variables, we included the most recent week's average transaction fee value in monthly forecasts and the most recent month's value in quarterly forecasts, and this improved the accuracy of the AR and ARMA models. More recent transaction fee values adjusted for weekly or other cyclical patterns have a stronger relationship with future fees.

To evaluate the accuracy of a model's forecast, one needs to compare its outputs with a baseline measure. Most economic literature uses a random walk or "naïve" model as a baseline, which uses last period's value to forecast the current period's value. For our purposes, we use the following naïve model methodology: this month or quarter's average transaction fee is the forecast for the next month or quarter average transaction fee. Mathematically the equation is the following:

$$TXF_{t+1} = TXF_t$$

In this paper, our time-series models were used to forecast one month and one quarter into the future. With this approach, we make predictions for each data point in the testing set based on the available information up to that point. The process involves iteratively forecasting one time step ahead and updating the model with the actual value for that time period. All these considerations and methods of evaluation mentioned in this section apply to the causal models in the later sections (unless otherwise specified).

[11] For reasons described above and the limited relevance and availability of data pre-June 2016.

AR and ARMA Model Results

Below are the results for three AR and five ARMA models tested for the forecast accuracy of monthly average transaction fees and three AR and three ARMA models for the forecast accuracy of quarterly average transaction fees.[12]

The forecast accuracy ratio is calculated by dividing the mean absolute error of a model by that of the baseline. If a model's accuracy measure is below 1, it was more accurate than the baseline and if it is greater than 1 it was less accurate. Highlighted in green are the best performing models during the testing period.

Monthly Forecast Results:

Model	Process	Forecast Accuracy Ratio (MAE)
Base Model	Naive (Last Month Avg)	Base
AR Models	AR (1)	0.9019
	AR (2)	0.8966
	AR (3)	0.8828
ARMA Models	ARMA (1,1)	0.9019
	ARMA (2,1)	0.9700
	ARMA (3,1)	**0.8798**
	ARMA (2,2)	0.9079
	ARMA (3,2)	0.9087

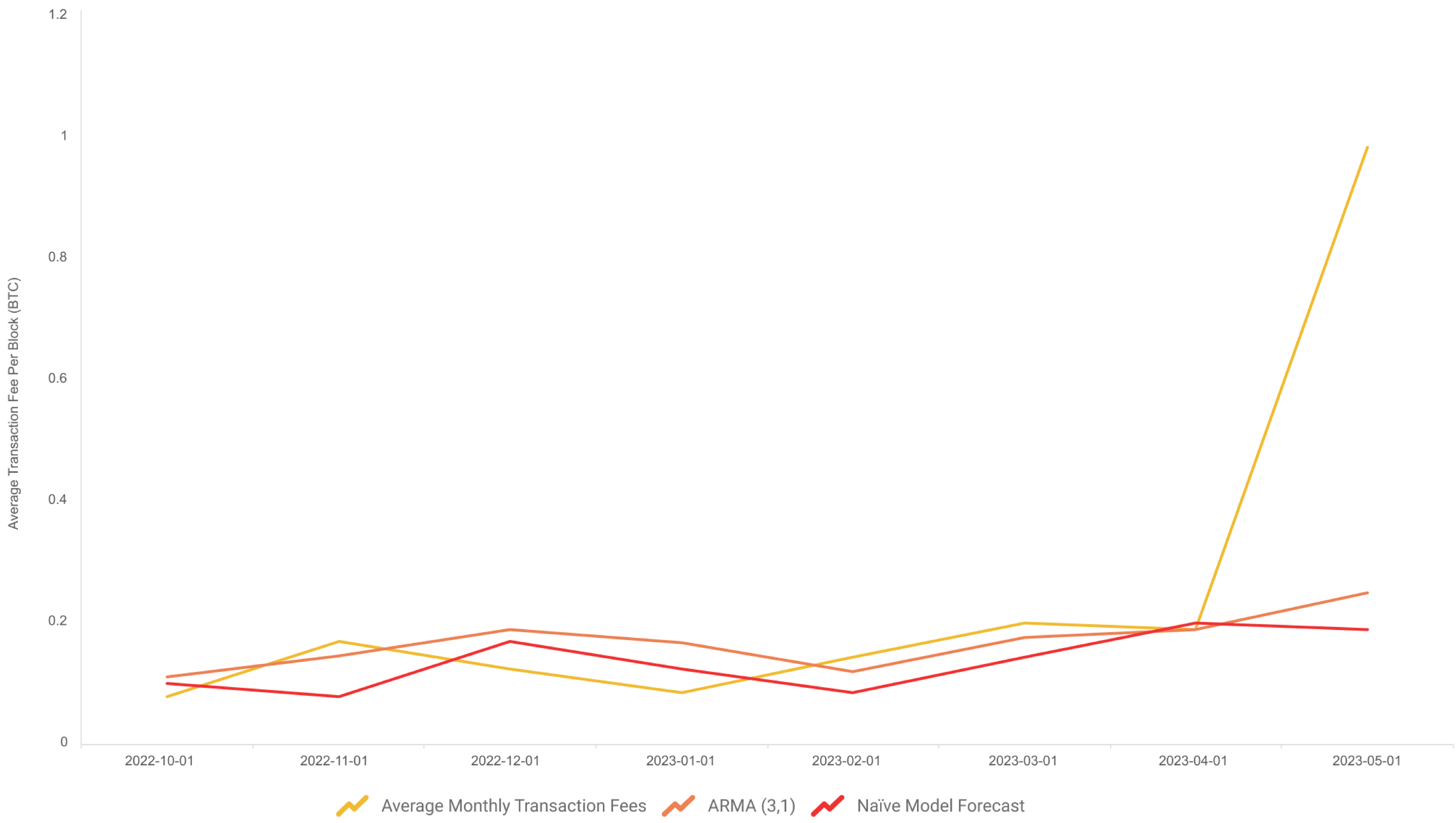
Quarterly Forecast Results:

Model	Process	Forecast Accuracy Ratio (MAE)
Base Model	Naive (Last Qtr Avg)	Base
AR Models	AR (1)	0.9568
	AR (2)	0.9463
	AR (3)	0.9452
ARMA Models	ARMA (1,1)	0.9611
	ARMA (2,1)	**0.9276**
	ARMA (3,1)	0.9450

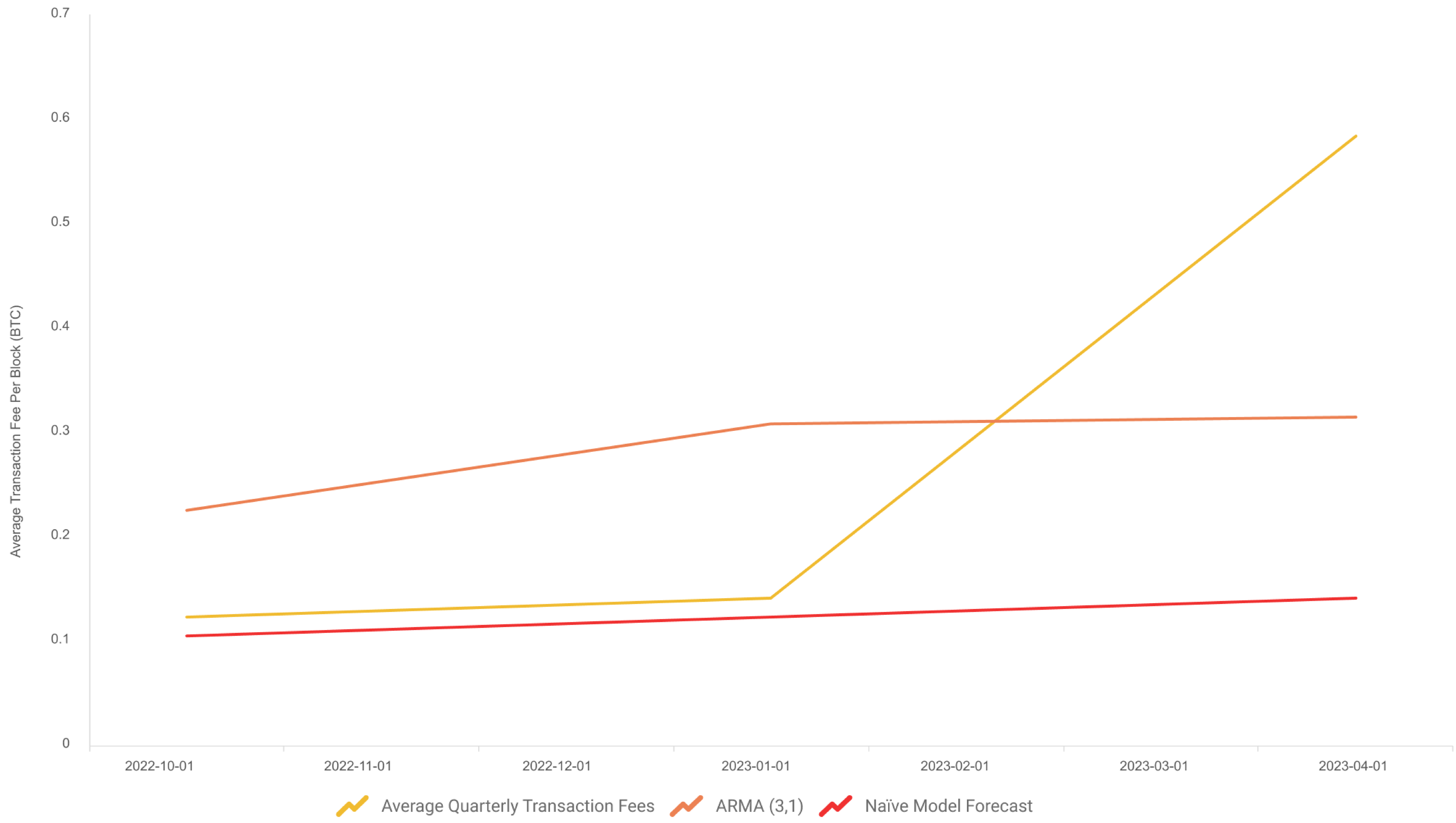
Below are charts comparing the best monthly and quarterly transaction fee forecasts with baseline and actual transaction fees for the period.

[12] The most recent week's average transaction fee value in monthly forecasts, and the most recent month's value in quarterly forecasts are included in the AR and ARMA models as exogenous variables.

Monthly Forecast for Average Transaction Fee Per Block (Testing Set) - ARMA (3,1)



Quarterly Forecast for Average Transaction Fee Per Block (Testing Set) - ARMA (2,1)



Our univariate time-series models marginally outperform our baseline model's forecasting performance. As with the baseline model, the limitation of these univariate models was their inability to catch the sudden spike in transaction fees in May 2023.

Volatility Forecasts: ARCH and GARCH Models

A prominent and growing area in forecasting literature has focused on forecasting volatility. Many things in life are too difficult to forecast using the techniques we talked about, such as asset prices, because their past values tell us almost nothing about their future values. However, there are promising techniques that use past volatility to predict future volatility. This has been the basis of many of the most popular options valuation models over the years such as the famous Black-Sholes model.

In finance and economics, models such as ARCH and GARCH are commonly used to forecast future volatility. These models are built upon the notion that volatility is not constant over time, but rather exhibits clustering and persistence. Our hope is that the volatility forecast can be incorporated as an exogenous variable in our time-series AR and ARMA models to forecast transaction fees. If that does not end up being effective, then we would at least uncover the relationship (or lack thereof) between volatility of transaction fees and transaction fee levels.

An Autoregressive Conditional Heteroscedasticity (ARCH) model is a type of time series model used to describe the volatility or variability of a sequence of data points over time. It is commonly used in financial modeling to capture the clustering of volatility in asset returns. The order of the ARCH(p) model depends on how many lags terms are included. The general equation of an ARCH model is:

$$\sigma_t^2 = c + \Sigma(\alpha_i * \varepsilon_{t-1}^2)$$

Where,

- ***σ_t^2 = represents the conditional variance of the time series at time t.***
- ***C = is a constant term that represents the long-term average variance.***
- ***α_i = are the ARCH parameters, where i denotes the order. These parameters capture the effect of past squared error terms on the current conditional variance.***
- ***ε_{t-1}^2 = refers to the squared error terms at different lags, which are residuals form previous time points.***

The Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model is an extension of the ARCH model that incorporates both past squared errors and past conditional variances to capture the volatility clustering in time series data. As with the ARCH model, the order of the GARCH(p) model depends on how many lags terms are included. The general equation of a GARCH model is:

$$\sigma_t^2 = c + \Sigma(\alpha_i * \varepsilon_{t-1}^2) + \Sigma(\beta_j * \sigma_{t-j}^2)$$

Where,

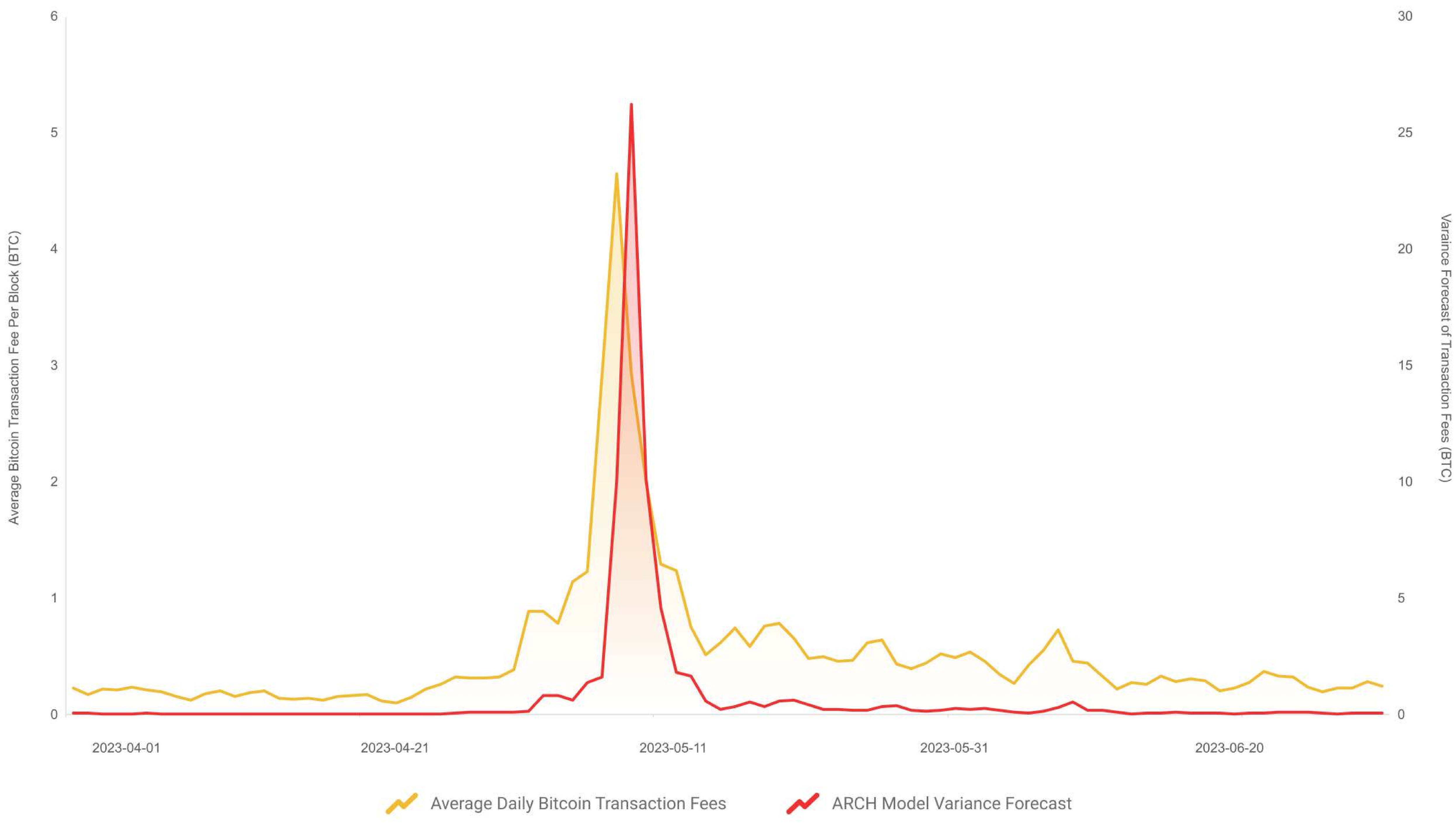
- ***β_j = are the GARCH parameters, where j denotes the lag order. These parameters capture the ffect of past conditional variances on the current conditional variance***
- ***σ_{t-j}^2 = represents the past conditional variances at different lags.***

ARCH and GARCH Model Results

For this model, we used daily data, as daily data feeds the models more of the volatility that a monthly average would smooth out. Below are graphs of average daily transaction fees and our volatility forecast from an ARCH(1) forecast and GARCH(1) from April 1, 2023 to June 29, 2023.

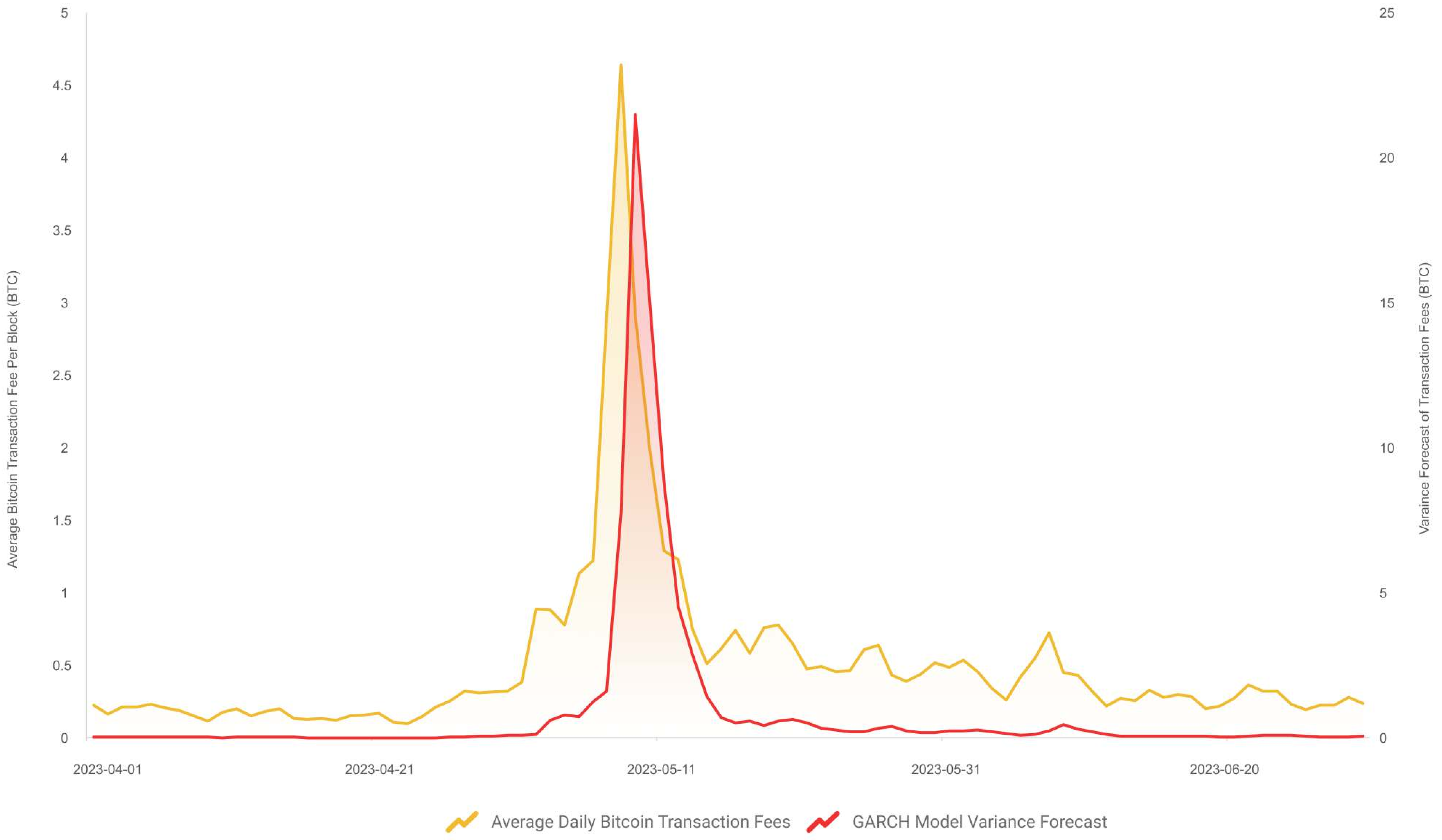
ARCH(1) Variance Forecast

HASHRATE INDEX Luxor



GARCH(1) Variance Forecast

HASHRATE INDEX Luxor



After producing a volatility forecast for daily average transaction fees, we attempted to use that forecast as an exogenous variable in our AR and ARMA model, which produced models with either similar or less accurate forecasts. However, the graphs above show us that volatility corresponds with upward shifts of transaction fees. In fact, during our testing period, there was a 71.1% correlation between our volatility forecast variable from the ARCH(1) variable and average daily transaction fees, while the GARCH(1) model had 70.9% correlation with average daily transaction fees. This indicates that although using volatility to forecast transaction fees was unsuccessful, volatility is a useful measure as an indicator for transaction fees. When volatility is high, transaction fees tend to soar upwards, and when volatility is low, fees stabilize.

Quantitative Techniques: Multivariate Causal Models

A model based solely on previous transaction fees is unlikely to be the earliest or most accurate detection mechanism for a sudden spike in transaction fees. The single time series cannot capture other more relevant external factors impacting block weight demand. In this section we explore more advanced multivariate causal models based on an assortment of independent variables to identify signals of upcoming transaction fee spikes. These multivariate models were trained and used to forecast solely based on monthly data.

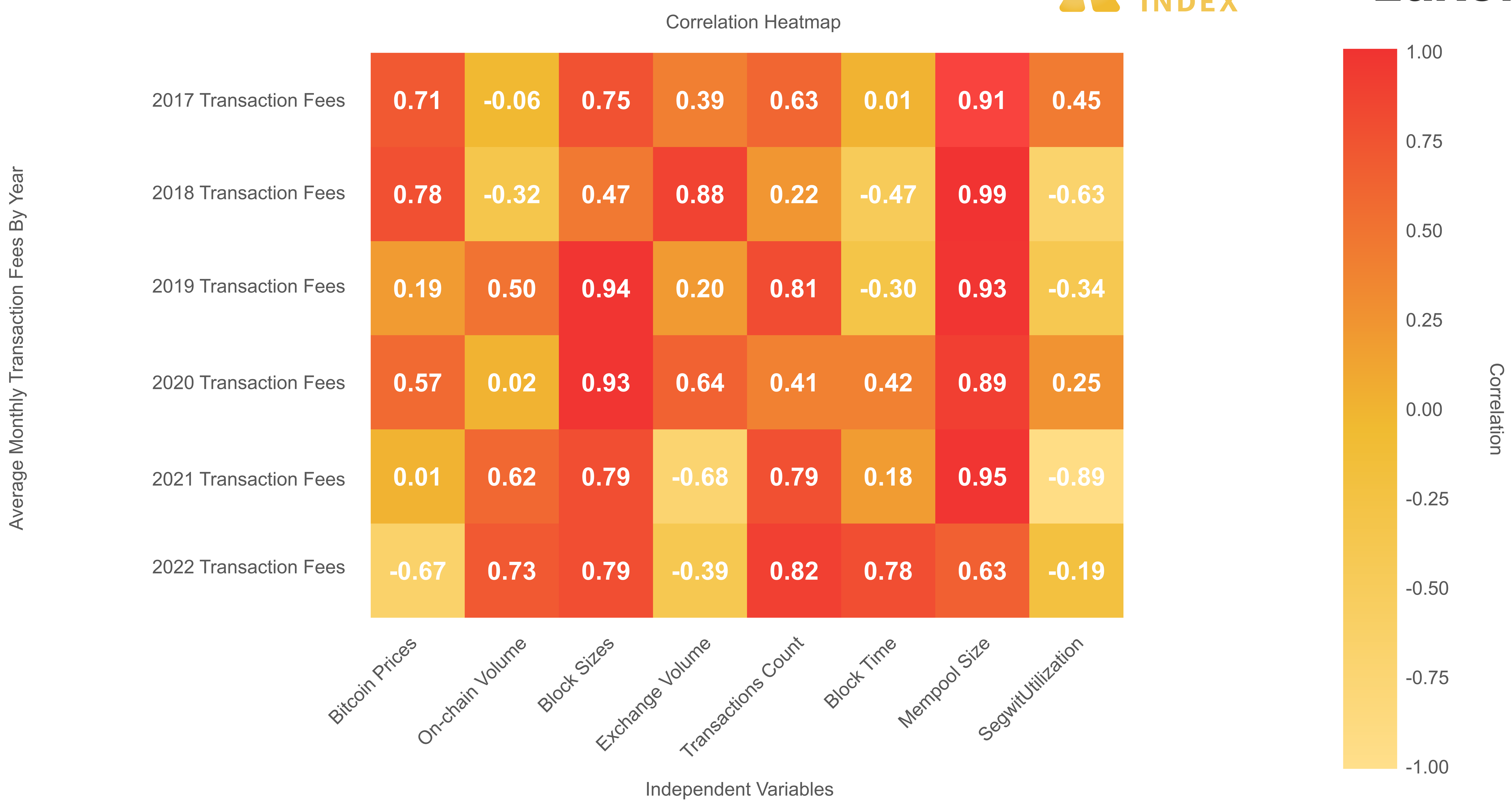
In Appendix 2.5, we provide a concise explanation for how to interpret the beta coefficients generated by these models. Understanding these coefficients is crucial for accurately interpreting and comprehending the results of these causal models.

For this paper, our multivariate causal models used the following variables:

1. Transaction fees per block
2. Bitcoin prices
3. On-chain transaction volumes
4. Block sizes
5. Number of transactions
6. Block times
7. Exchange volumes
8. Mempool size
9. Segwit utilization

Correlation Analysis

Before jumping into our multivariate models, let's explore the correlation between the monthly averages of bitcoin transaction fees and our independent variables from January 2017 to December 2022.



A look at the map above shows us that the most consistently correlated variables with transaction fees are number of transactions (transaction count), mempool size, and block sizes. This gives us an initial understanding of what variables could end up being highly predictive of transaction fees. However, to fully understand the drivers of our transaction fee variables, we should apply a regression methodology to control for all the other independent variables (and their lags). For example, transaction fees seems to be highly correlated with transaction count, but is that correlation still significant when the bitcoin price variable is taken into account? How do the lags of these variables play a role in predicting transaction fees? These are the types of questions our multivariate causal models can help answer.

Lower Bound Condition

Since transaction fees cannot be negative and are rarely close to zero, we introduced a lower bound into our multivariate models, whereby the baseline model was used for forecasts below 0.1 BTC per block. This lower bound condition greatly improved the accuracy of the models and we recommended this step to anyone attempting similar analysis.

Vector Autoregression (VAR) Model

Vector Autoregression (VAR) models capture the interdependencies and dynamics among multiple time series variables. It extends the autoregressive (AR) model to handle systems of variables rather than a single variable. In a VAR model, each variable in the system is regressed on its own lagged values as well as the lagged values of all other variables in the system. All variables are treated as endogenous unless specified otherwise, allowing a system examination of the variables.[13]

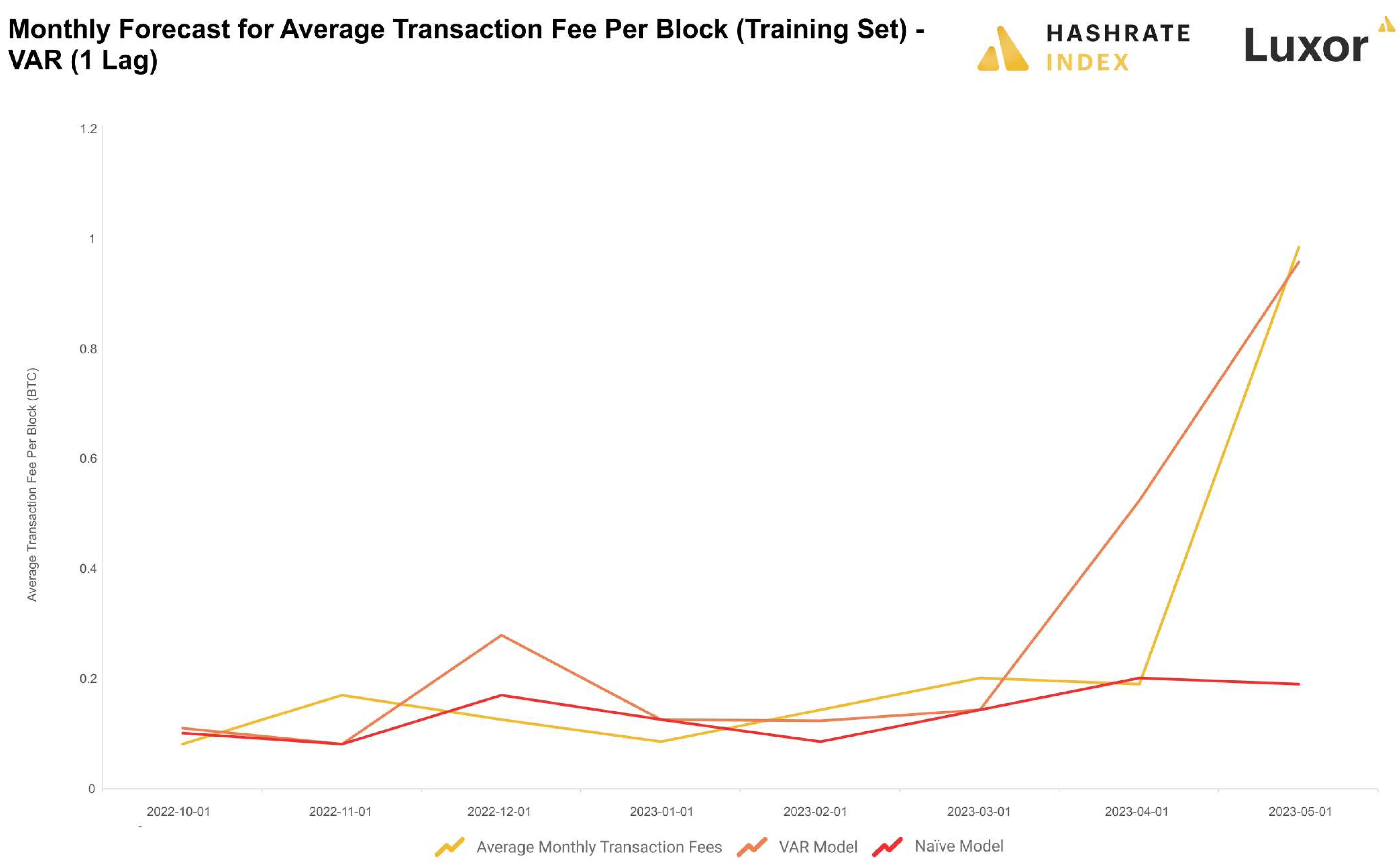
[13] Please note, we included the most recent week's value in monthly forecasts, and the most recent month's value in quarterly forecasts as an exogenous variable.

Appendix 2.6 gives a detailed explanation of our VAR model and its assumptions.[14]

VAR Results

As demonstrated in the table and chart below, our VAR model was more successful than our univariate models at forecasting transaction fees. Not only did the VAR model track actual transaction fees when fees were low, but **it predicted the dramatic rise in transaction fees in May 2023**. In addition to lagged transaction fees, the number of transactions, exchange volume, and block time were the most significant variables in our VAR model.

Model	Process	Forecast Accuracy Ratio (MAE)
Base Model	Naive (Last Qtr Avg)	Base
VAR Models	VAR with 1 Lag	**0.6722**
	VAR with 2 Lags	3.2058



Appendix 2.7 gives the coefficient table for the 1-lag VAR model.

Cross-Sectional Regression Models with Regularization

Cross-sectional regression models focus on examining the linear relationship between a dependent variable and independent variables, disregarding the time dynamics involved. In contrast to other discussed time-series models, these analyses treat the data as if it is from a single period of time. While it is evident that our data exhibits time dynamics (given that transaction fees are inherently a time-series variable), this type of analysis can still offer valuable insights by indicating the likelihood of future upward or downward spikes in fees based on identifying relationships between variables. We do incorporate a slight element of time-dynamics to these cross-sectional models by including up to three lags of each variable as independent variables within our model. Our hope is to identify variables to include in future models.

[14] This article by Analytics Vidhya provides a useful overview of VAR models and how they make use of Granger causality to predict how a variable changes based on changes on another variables' lagged values.

For this paper, we explore three types of cross-sectional regression models:

- A simple OLS model
- An OLS model with the Lasso regularization method, and
- An OLS model with the Ridge regularization method.

1. Multivariate OLS Regression

Multivariate Ordinary Least Squares (OLS) regression is a statistical technique used to model the relationship between multiple independent variables and a dependent variable. It estimates the coefficients that minimize the sum of squared differences between the observed and predicted values, allowing for identification of the strength and direction of the relationships between the variables.

2. Lasso and Ridge Multivariate OLS Regressions

OLS multivariate regressions risk "overfitting" when numerous variables are included. This refers to a situation where the model achieves a strong fit to the training data but performs poorly when applied to new data (out-of-sample fit) during forecasting. To address this issue, Ridge and Lasso regressions introduce a penalty term that regulates the magnitude of coefficients. By employing regularization, these methods act as a filter that eliminates independent variables that do not significantly contribute to predicting the dependent variable from the OLS equation. This ensures a more balanced and accurate forecasting performance.

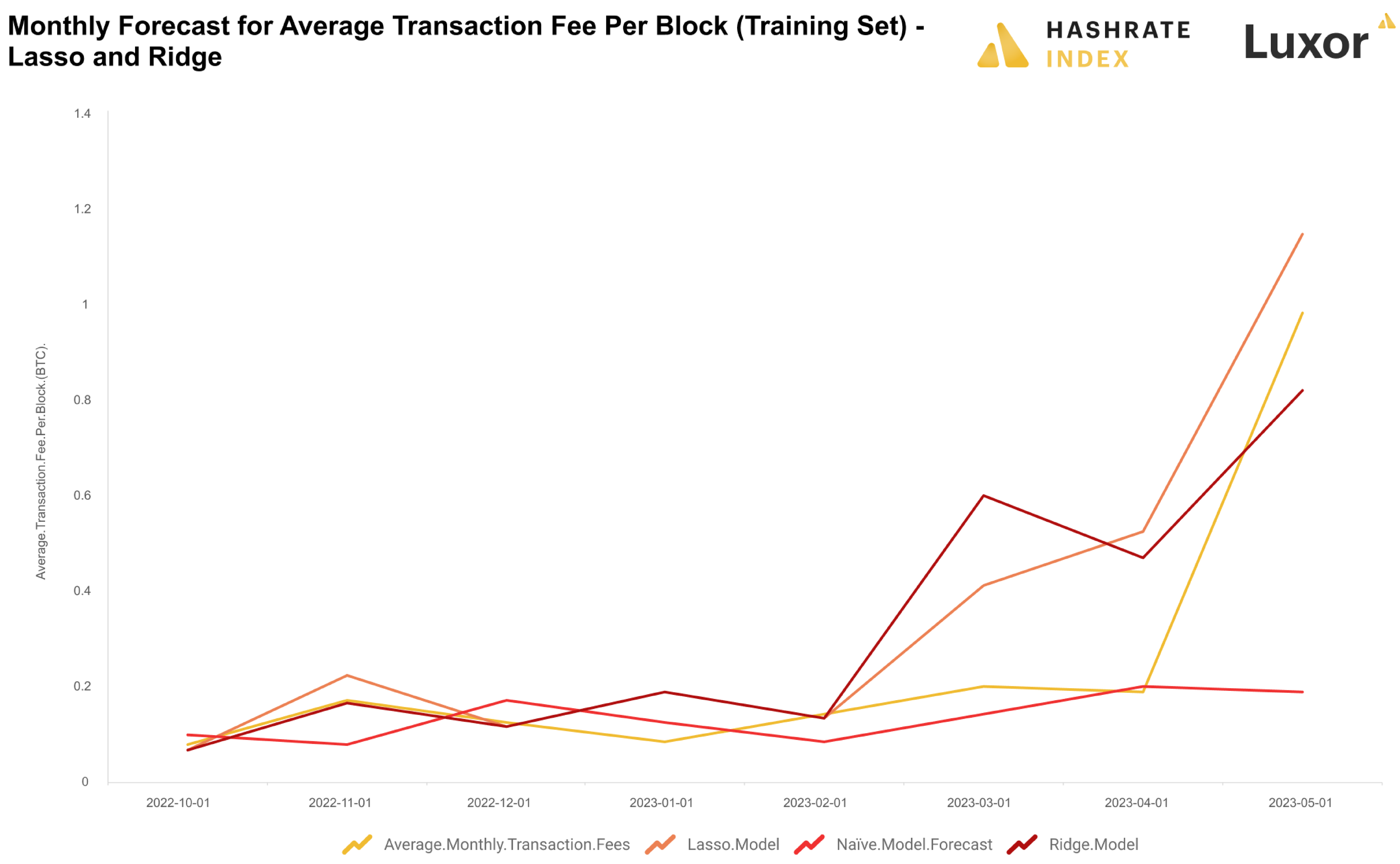
Further explanations about our OLS equation can be found in appendix 2.8. Appendix 2.9 explains how Lasso and Ridge regularization methods work in further detail, including all the factors we considered for these models.

Cross-Sectional Regression Results

Comparing cross-sectional regression models such as LASSO, OLS, and Ridge with a random walk model, which is based on time-series analysis, is not meaningful due to the fundamental differences in approach. While the baseline model compares trends over time, as mentioned, cross-sectional models analyze data as if they all belonged in a single time-period. For this reason, we use the OLS regression as the baseline model for the Lasso and Ridge models.

As demonstrated in the table and chart below, these models exhibited predictive power. Both the Ridge and Lasso regression models predicted rising transaction fees heading into the May 2023 spike.

Model	Forecast Accuracy Ratio (MAE)
OLS Model	Base
Lasso Model	0.4935
Ridge Model	**0.4440**



While these models do not offer direct time-series forecasts, the cross-sectional models offer valuable information that can help detect potential changes in fees and guide decision-making accordingly.[15]

Appendix 2.10 displays tables of the beta coefficients produced by the OLS, Lasso and Ridge regressions.

[15] Please note that we evaluated a time-series version of these cross-sectional models by only including lagged data on independent variables. These models did not perform as well, highlighting our recommendation that these types of models should only be considered for cross-sectional analysis and as a signaling indicator for potential upcoming spikes in fees.

Discussion on Forecasting Transaction Fees

The findings from the previous sections yield noteworthy conclusions regarding the forecasting of transaction fees.

First, we observed that **one-period ahead forecasts using Autoregressive (AR) and Autoregressive Moving Average (ARMA) models, both on a monthly and quarterly basis, slightly outperform baseline models**. While the improvements may not be substantial, they suggest that incorporating the time series characteristics of transaction fees can lead to slightly improved forecasts compared to relying solely on the most recent observation. It was also clear that although our volatility forecasts perform poorly as exogenous variables to forecast transaction fees, it did reveal clear positive correlation between fees and volatility.

Second, it is important to note that **Autoregressive (AR) and Autoregressive Moving Average (ARMA) models were unable to predict the significant spike in fees that occurred in May 2023.** This outcome is reasonable since the average monthly transaction data did not provide any indications of an impending spike. In our analysis, we determined that an ARMA(3,1) process yielded the most accurate forecasts for monthly data, while an ARMA(2,1) process performed best for quarterly data.

Third, our **VAR model with one lag demonstrated significant superiority over the baseline and univariate time-series models.** This result is particularly notable as the VAR model successfully predicted the spike in transaction fees that occurred in May due to the BRC-20 frenzy. Lagged transaction fees, the number of transactions, exchange volume, and block times were the most significant variables in our VAR model. This suggests that a VAR model has the potential to be a powerful forecasting tool for Bitcoin transaction fees. Its ability to capture the dynamics and relationships between variables provides valuable insights into medium-term fee fluctuations.

And finally, **common variables available to anyone without in-depth ordinal market knowledge carry information about upcoming transaction fees.** While mempool and Bitcoin price variables were insignificant factors in our VAR model, they were highly predictive components of the cross-sectional regressions. In conjunction with lagged transaction fees, the number of transactions, exchange volume, block times, and other variables,[16] multivariate models can help detect incoming and sudden rises in Bitcoin transaction fees.

[16] Some discussed, some yet to be explored.

3

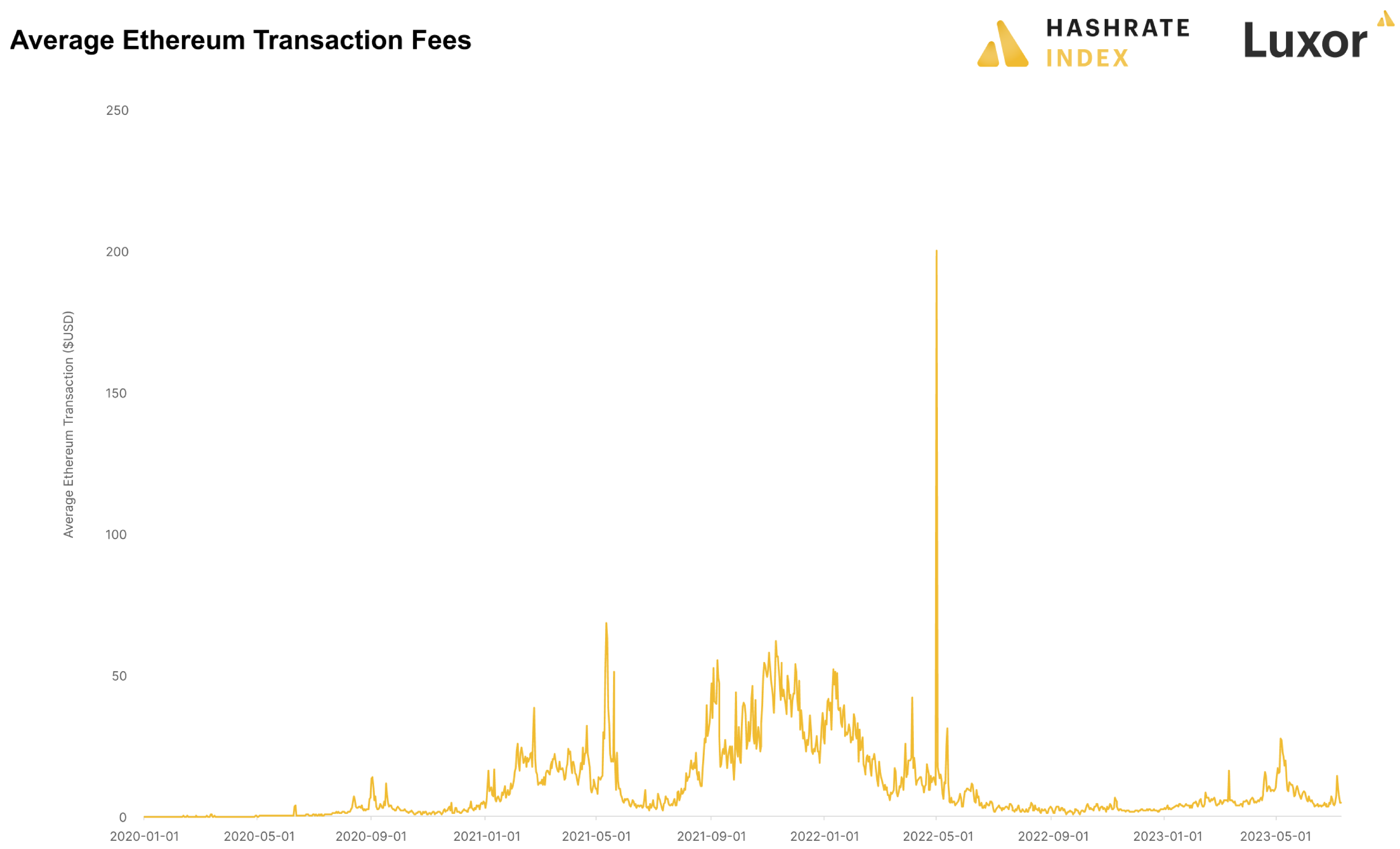
Appendix

Appendix

Appendix 1) Ethereum Case Study

Appendix 1.1) Ethereum Transaction Fee Case Study

The recent mania behind ordinals and inscriptions on Bitcoin has historical precedent. NFT markets on cryptocurrency networks such as Ethereum can be seen as an analog to ordinals and inscriptions. The development of inscriptions and ordinals opens up a Pandora's box of possibilities for use of blockspace. Studying the history of transaction fees on the Ethereum network (also known as "gas fees") may give us an idea of what is in store for Bitcoin transaction fees in this new post-ordinal environment.



The chart above illustrates average fee per transaction for Ethereum from January 2020 to May 2023, providing valuable insights that can be categorized into five distinct chronological chapters. These chapters can be understood by examining the contextual factors that influenced each specific time period.

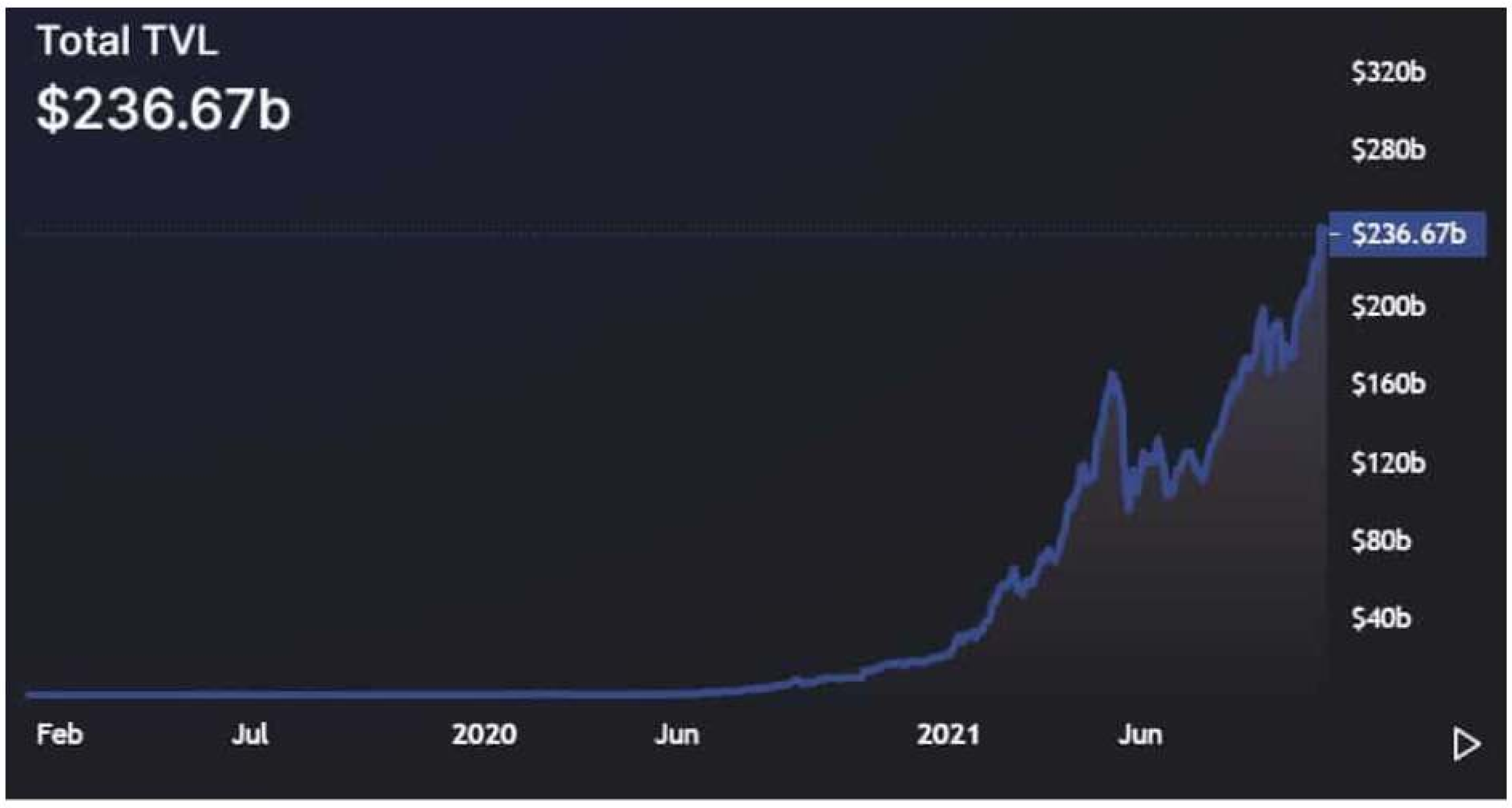
1. **The first (dramatic) rise in ETH transaction fees in Summer of 2020, then again from February 2021 to May 2021**

The first notable increase in transaction fees, as observed in our chart, occurred during the summer of 2020, followed by a more substantial surge from February 2021 to May 2021. In their March 2021 Ethereum Gas Report, the Coin Metrics highlights that the rise and fall of transaction fees during this period were closely linked to the emergence of DeFi (Decentralized Finance). According to the report, "Ethereum's gas price rise corresponds with the rise of DeFi, which is still likely the largest contributor to high gas prices." Furthermore, the report provides a case study focusing on Uniswap's UNI airdrop as an example, further substantiating its conclusions.

2. The second and long lasting rise in fees from late summer of 2021 to spring of 2022

Following Coin Metrics' report, there was another significant surge in transaction fees, but this time it was more sustained and enduring. Starting in August 2021, fees continued to rise and remained consistently high until April 2022, without dropping below the pre-August 2021 levels.

The sustained increase in transaction fees was fueled by the concurrent growth of DeFi and the NFT market. In October 2021, the Defi Total Value Locked metric surpassed $236 billion, indicating substantial exponential growth since 2020. Simultaneously, the NFT market experienced a significant surge, with average NFT sale prices exceeding $6,800. OpenSea, a leading NFT marketplace, reached a peak of $360 million in daily transactions in February 2022, although it subsequently experienced a sharp decline.



Source: Cryptopotato.com

3. Huge spike on May 1, 2022

May 1, 2022 is a day that deserves its own explanation. There was a significant and notable spike in daily average Ethereum transaction fees to as high as $200 per transaction. The cause behind this sudden surge was the release of Yuga Labs' land title collection called Otherdeed, which was part of the Otherside Metaverse collection. These NFTs could be minted using the project's APE coin but required ETH to cover network gas fees. The overwhelming demand for these NFTs inflated the average cost of an Ethereum transaction. Our very own company Luxor happened to mine these very Ethereum blocks which were yielding record high fees at the time, as can be seen here.

4. Period of very low fees post mid-2022

Starting in June 2022, the crypto markets, including the Defi and NFT sectors, entered a cool-down period. This decline in market activity was influenced by multiple factors, one of which was the aggressive targeting of inflation through interest rate increases in American monetary policy. The overall decrease in demand for crypto-related products and services resulted in a reduction in transaction volume, alleviating the previous strain on the network. Consequently, the period of high transaction fees came to an end as the crypto bubble burst in 2022.

5. Ethereum Moves to Proof of Stake

In September 2022, Ethereum moved to a Proof of Stake (POS) network fundamentally altering its block space and fee dynamics. We can end our case study here as this new design is too dissimilar to Bitcoin's proof-of-work design.

Appendix 1.2) Ethereum Network and Miner Extractable Value

An important topic that needs to be addressed as part of the Ethereum case study is the concept of Miner Extractable Value (MEV).[17] MEV refers to the amount of profit that miners can potentially extract from the order of transactions within a blockchain network, since miners have the authority to choose which transactions to include in a block and in what order. This power allows them to exploit certain transaction dependencies, front-running opportunities, and other types of arbitrage to potentially maximize their profits. At the beginning of 2021, the cumulative MEV extracted on Ethereum amounted to $78 million, while at the merge the total MEV extracted on Ethereum exceeded $675 million.

Source: explore.flashbots.net

[17] The article linked provides a well written piece further explaining MEV and it's inner workings.

Appendix 2) Forecasting Methodology

Appendix 2.1) Autoregressive Models

$TXF_t = c + \Sigma(\alpha_i * TXF_{t-i}) + \varepsilon_t$

The coefficient αi represents the weight assigned to the lagged term TXF_{t-i}, indicating its influence on the current value. The error term ε_t captures the residual variation in the model, accounting for any unexplained variation not captured by the auto-regressive component.[18] The constant term c represents the intercept of the dependent variable when all lagged terms and the error term are zero.

AR models estimate coefficients and predict future values based on past data, providing a simple and effective approach for time series forecasting. The choice of order depends on data and complexity needs. Higher orders capture long-term dependencies, while lower orders focus on short-term dynamics.

Appendix 2.2) Autoregressive Moving Average Models

$TXF_t = c + \Sigma(\alpha_i * TXF_{t-i}) + \Sigma(\beta_j * \varepsilon_{t-j}) + \varepsilon_t$

ARMA models add a new component to the AR equation, with β_j representing coefficients for past error terms (ε_{t-j}) in the moving average component. The ARMA model's order is defined by two parameters: p and q. Parameter p represents the order of the auto-regressive component, considering lagged terms of the variable of interest. Parameter q represents the order of the moving average component, including past error terms. ARMA, like AR, estimates coefficients and predicts future values based on past values and error terms, offering a slightly more complex but potentially more effective approach for time series forecasting.

Appendix 2.3) Assumptions and Model Selections

Assumptions

- **Stationarity:** Econometricians describe a stationary variable as having “consistent properties” over time. This means even if a variable diverges from its historic mean and variance in the short term, it will eventually return to those historic levels of mean and variance in the long-run.

- **Independence:** Observations should be independent of each other, without dependencies between specific time points.

- **No autocorrelation** (beyond what’s captured in the model): Autocorrelation should be accounted for in the model, leaving minimal residual autocorrelation.

- **Normality:** Residuals should follow a normal distribution, allowing for statistical tests and confidence intervals.

[18] Residual values refer to the difference between the observed value and the predicted value of a variable based on a model. In the context of time series analysis, the residual represents unexplained variation or discrepancy between the actual value of the variable and the value predicted by the model at a specific time point. It captures the part of the data that is not accounted for by the model's parameters.

Appendix 2.4 provides the results of the stationarity tests we ran on our monthly average bitcoin transaction fee variable. This test shows that the variable is stationary and there is no need to differentiate. Other tests that showed the rest of the assumptions held in our models were done routinely in constructing this paper.

Model Selection

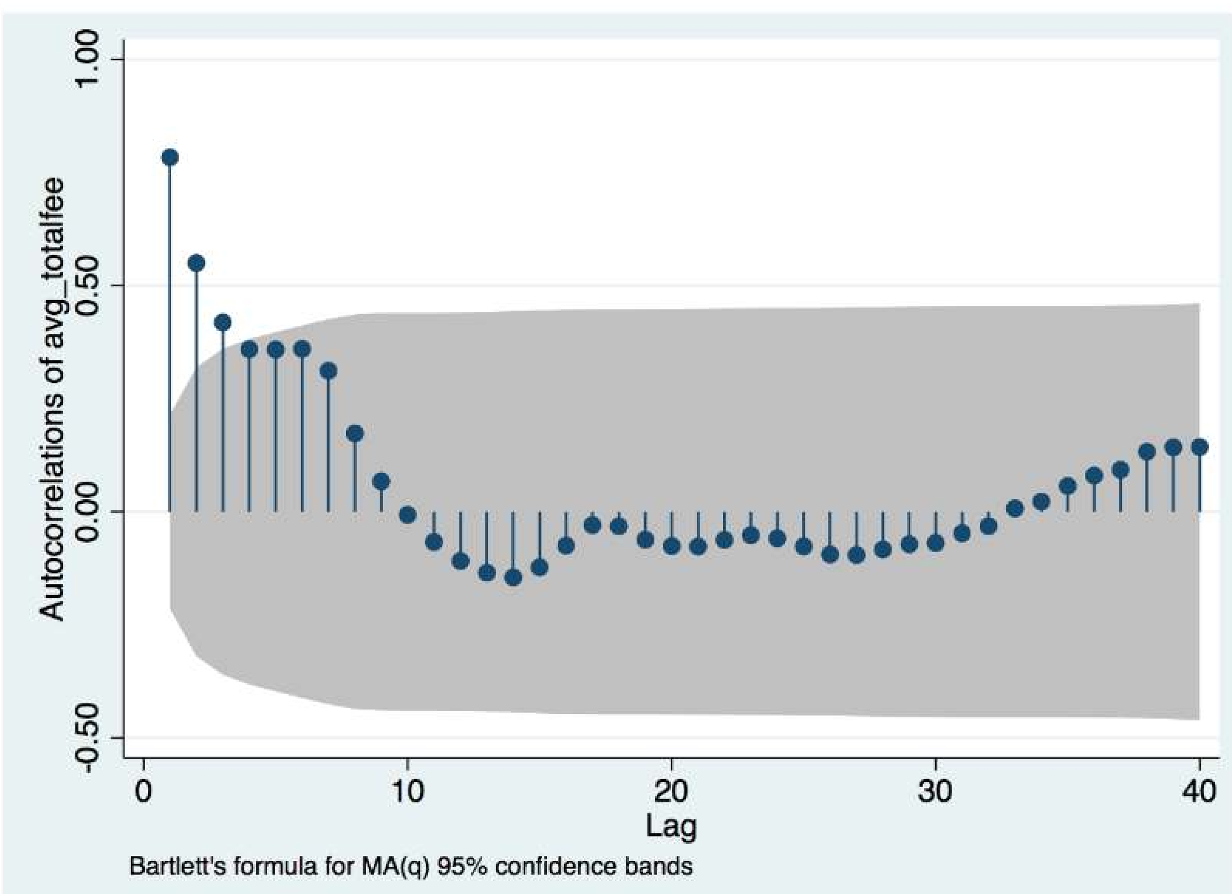
To choose the appropriate model for forecasting transaction fees, we analyze Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots. ACF measures the correlation between observations and their lagged values, while PACF accounts for intermediate lags. The ACF plot shows significant autocorrelation in the first, second lags and third lag with subsequent lags becoming statistically insignificant (reaches the gray in the plot). Similarly, the PACF plot reveals significant partial autocorrelation in the first lag, followed by insignificance in subsequent lags.

Below is a table showing a rule of thumb in terms of model selection depending on the tendencies of the ACF, PACF plots.

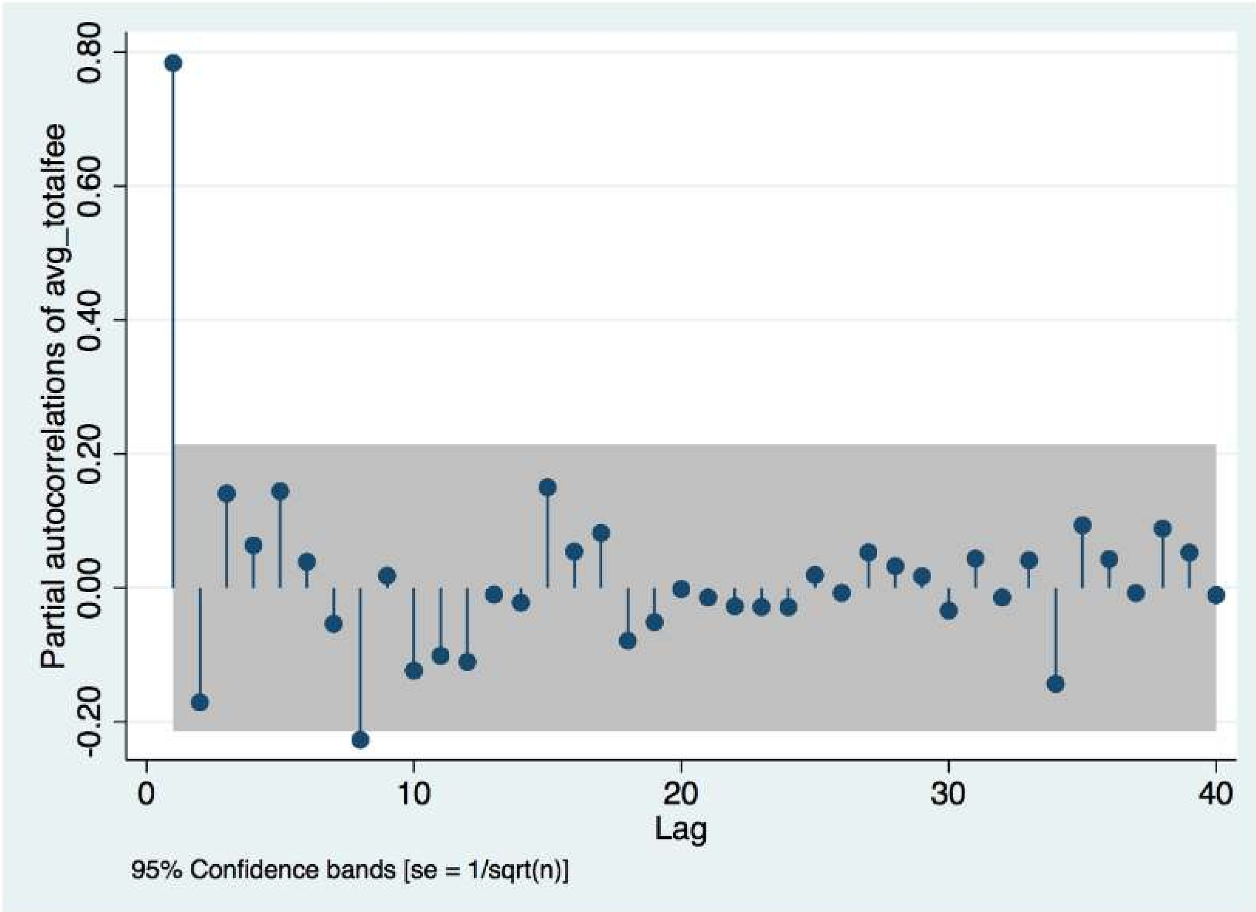
Model	ACF	PACF
AR (p)	Slow Decay	Drop off after lag p
MA (q)	Drop off after lag q	Slow Decay
ARMA (p, q)	Drop off after lag q	Drop off after lag p

Model and lag selections are an art rather than an exact science. The ACF and PACF suggest the possibility of AR or ARMA processes for average monthly transaction fees, ruling out MA due to the PACF's lack of slow decay. Results will demonstrate the performance of AR and ARMA models with different lag selections based on ACF and PACF. The most promising choice, according to the ACF and PACF plot, is an ARMA(3,1) model, with a drop in significance after 3 lags in ACF and 1 lag in PACF. Alternatively, statistical software can automatically select appropriate orders (p for AR and p and q for ARMA) using Information Criterion tests.

ACF Plot:



PACF Plot:



Appendix 2.4) Stationarity of Average Monthly Bitcoin Transaction Fees Variable

Stationarity of Bitcoin and Ethereum transaction fees is evident by observing their charts over time. Despite occasional surges, fees consistently revert to historical levels. To confirm stationarity, Dickey-Fuller and Phillips-Perron tests were conducted, and the results are presented below:

Dickey–Fuller test for unit root — Number of obs = **83**
Variable: **avg_totalfee** — Number of lags = **0**

H0: Random walk with or without drift

	Test statistic	Dickey–Fuller critical value 1%	5%	10%
Z(t)	**-3.624**	**-4.077**	**-3.467**	**-3.160**

MacKinnon approximate *p*-value for Z(t) = **0.0279.**

Phillips–Perron test for unit root — Number of obs = **83**
Variable: **avg_totalfee** — Newey–West lags = **3**

H0: Random walk with or without drift

	Test statistic	Dickey–Fuller critical value 1%	5%	10%
Z(rho)	**-23.531**	**-26.822**	**-20.394**	**-17.262**
Z(t)	**-3.660**	**-4.077**	**-3.467**	**-3.160**

MacKinnon approximate *p*-value for Z(t) = **0.0251.**

With both tests showing a p-value of less than 0.05, we can reject the null hypothesis that our transaction fee data is not stationary.

Appendix 2.5) Beta Coefficient Interpretation

Beta coefficients in the results section represent the predicted change in the dependent variable for a one-unit change in the corresponding independent variable, while controlling for all other variables. A positive beta coefficient signifies a positive relationship, meaning that as the independent variable increases, the dependent variable is expected to increase. Conversely, a negative beta coefficient indicates an inverse relationship, where an increase in the independent variable corresponds to a decrease in the dependent variable.

Appendix 2.6) VAR Model

Below would be an example of a two variable (Y and X) VAR model with 1 lag:

$$y_t = b_{10} - b_{12}x_t + \gamma_{11}y_{t-1} + \gamma_{12}x_{t-1} + \varepsilon_{yt}$$

$$x_t = b_{20} - b_{21}y_t + \gamma_{21}y_{t-1} + \gamma_{22}x_{t-1} + \varepsilon_{xt}$$

$$\begin{bmatrix} 1 & b_{12} \\ b_{21} & 1 \end{bmatrix}\begin{bmatrix} y_t \\ x_t \end{bmatrix} = \begin{bmatrix} b_{10} \\ b_{20} \end{bmatrix} + \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix}\begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{yt} \\ \varepsilon_{xt} \end{bmatrix}$$

By solving the system of equations, beta coefficients are obtained to forecast future values of Y_{t+1} and X_{t+1}. In our case, the system includes nine vectors from the nine endogenous variables we are modeling. We focus on the vector for average transaction fees for our forecasting purposes. Additionally, the final week's average transaction fees from the previous month are included as an exogenous variable outside the system of equations but within our overall model.

To meet the assumptions of the VAR model, steps were taken, including testing the stationarity of each variable and applying differencing if needed. The lag order selection was performed using statistical software's lag selection feature, which considered information criterion methods like AIC and BIC. The appropriate lag order according to those tests could be either one or two, thus we tested VAR models with both one and two lags.

Stationarity

Similar to univariate time-series analysis, stationarity is a concern in the VAR model. Tests conducted on variables such as average monthly bitcoin price, volume on exchange services, block sizes, transaction counts, and Segwit utilization revealed they were not stationary. To address this, differencing was applied to make these variables stationary. Consequently, when interpreting coefficients for these models, it should be noted that the mentioned variables are their differentiated versions.

Appendix 2.7) VAR Model Coefficient Table

Beta coefficients in the results section represent the predicted change in the dependent variable for a one-unit change in the corresponding independent variable, while keeping all other variables constant. A positive beta coefficient signifies a positive relationship, meaning that as the independent variable increases, the dependent variable is expected to increase. Conversely, a negative beta coefficient indicates an inverse relationship, where an increase in the independent variable corresponds to a decrease in the dependent variable.

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
vg_totalfee						
avg_totalfee						
L1.	.4548902	.2158237	2.11	0.035	.0318836	.8778968
avg_price						
LD.	-1.04e-06	.0000125	-0.08	0.934	-.0000255	.0000234
avg_volume						
L1.	4.65e-06	2.04e-06	2.28	0.023	6.48e-07	8.66e-06
avg_block_size						
LD.	-.7156681	1.000381	-0.72	0.474	-2.676379	1.245043
avg_tx_count						
LD.	.0010996	.0004765	2.31	0.021	.0001657	.0020334
avg_block_time						
L1.	-.209348	.1111859	-1.88	0.060	-.4272684	.0085723
avg_volume_btc						
LD.	.0000129	.0000127	1.01	0.311	-.0000121	.0000379
mempool_size_byte						
L1.	2.69e-09	3.36e-09	0.80	0.423	-3.89e-09	9.27e-09
segwit_utilization						
LD.	.5294702	2.635359	0.20	0.841	-4.635739	5.69468
last_week_avg	.1741632	.1671114	1.04	0.297	-.1533691	.5016955
_cons	2.053173	1.037817	1.98	0.048	.0190902	4.087257

For the variables average monthly bitcoin price, volume on exchange services, block size, transaction count and Segwit utilization, the coefficients are labeled with LD instead of L1. L1 represents the past (1) lag of the variable, while LD represents the past (1) lag of the differentiated variables. Differentiation was necessary to achieve stationarity, a crucial assumption for all variables in our model.

Appendix 2.8) OLS Equation

Our OLS regression equation would be the following:

$$TXF_t = \beta 0 + \beta 1 * TXF_{t-1} + \beta 2 * TXF_{t-2} + \beta 3 * TXF_{t-3} + \beta 4 * VOL_t + \beta 5 * VOL_{t-1} + \beta 6 * VOL_{t-2} + \beta 7 * VOL_{t-3} + \beta 8 * VOLEX_t + \beta 9 * VOLEX_{t-1} + \beta 10 * VOLEX_{-2} + \beta 11 * VOLEX_{t-3} + \beta 12 * TXC_t + \beta 13 * TXC_{t-1} + \beta 14 * TXC_{t-2} + \beta 15 * TXC_{t-3} + \beta 16 * BS_t + \beta 17 * BS_{t-1} + \beta 18 * BS_{t-2} + \beta 19 * BS_{t-3} + \beta 20 * BT_t + \beta 21 * BT_{t-1} + \beta 22 * BT_{t-2} + \beta 23 * BT_{t-3} + \beta 24 * BTC_t + \beta 25 * BTC_{t-1} + \beta 26 * BTC_{t-2} + \beta 27 * BTC_{t-3} + \beta 28 * MEM_t + \beta 29 * MEM_{t-1} + \beta 30 * MEM_{t-2} + \beta 31 * MEM_{t-3} + \beta 32\, TXFP_t + \beta 33 * SEG_t + \beta 34 * SEG_{t-1} + \beta 35 * SEG_{t-2} + \beta 36 * SEG_{t-3} + \varepsilon_t$$

Where β0 is the intercept, β1 to β36 are the regression coefficients for the respective independent variables, and ε is the error term. Note that the beta coefficients are solved through the classical beta equation:

$$\beta^{OLS} = (X^T * X)^{-1} * X^T * Y$$

The following explains the abbreviations in equation above:

- TFX = Average daily transaction fees
- TXFP = Average transaction fees of final week of previous month
- VOL = Average bitcoin volume on-chain
- VOLEX = Average bitcoin volume on exchanges
- TXC = Transactions count
- BS = Average Block size
- BT = Average Block time
- BTC = Average bitcoin price
- MEM = Average mempool size
- SEG = Average Segwit Utalization

Appendix 2.9) Lasso and Ridge Regression Explanation

In our OLS regression, there are numerous independent variables (36 plus a constant and error term) in the equation, which can lead to overfitting. Lasso and Ridge regressions offer a solution through regularization methods, which act as a metaphorical filtering mechanism. Independent variables which are not contributing towards making the regression (and subsequent forecast) more accurate are filtered out through regularization.

Lasso regression employs L1 regularization, adding the sum of absolute values of coefficients multiplied by a tuning parameter (lambda). This penalty term effectively selects relevant predictors by shrinking the coefficients of less informative variables to zero.

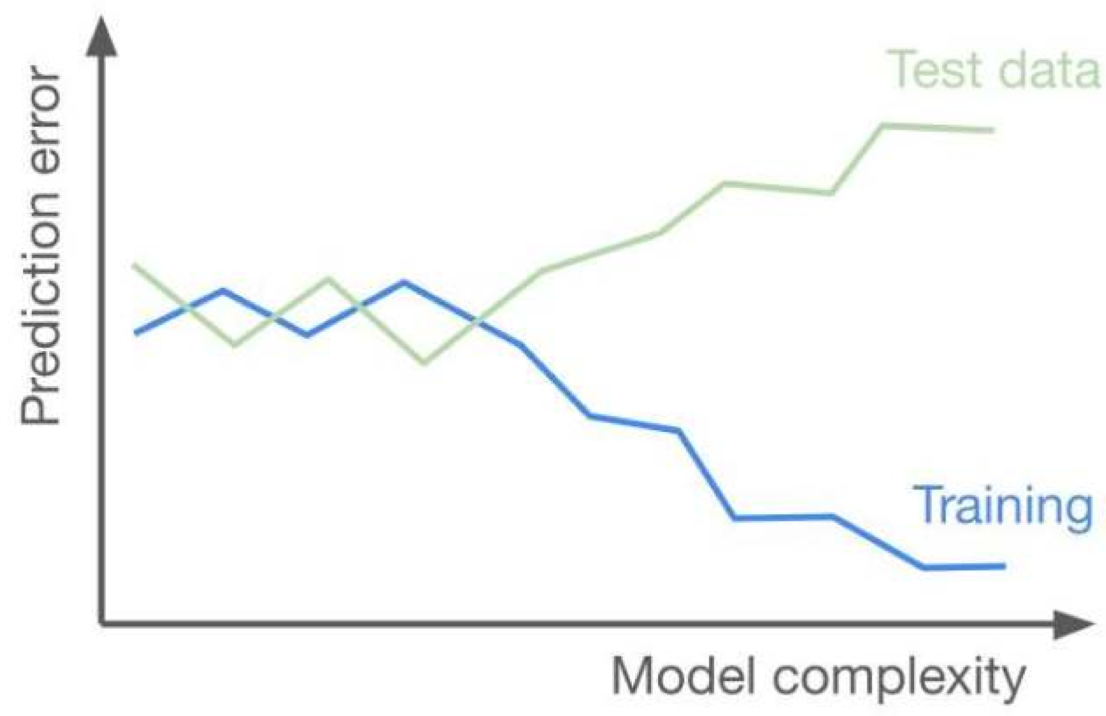
Ridge regression applies L2 regularization, summing the squared values of coefficients multiplied by a tuning parameter (lambda). The penalty term forces coefficients towards zero, but not exactly zero. Consequently, all variables are retained in the model with smaller coefficients.

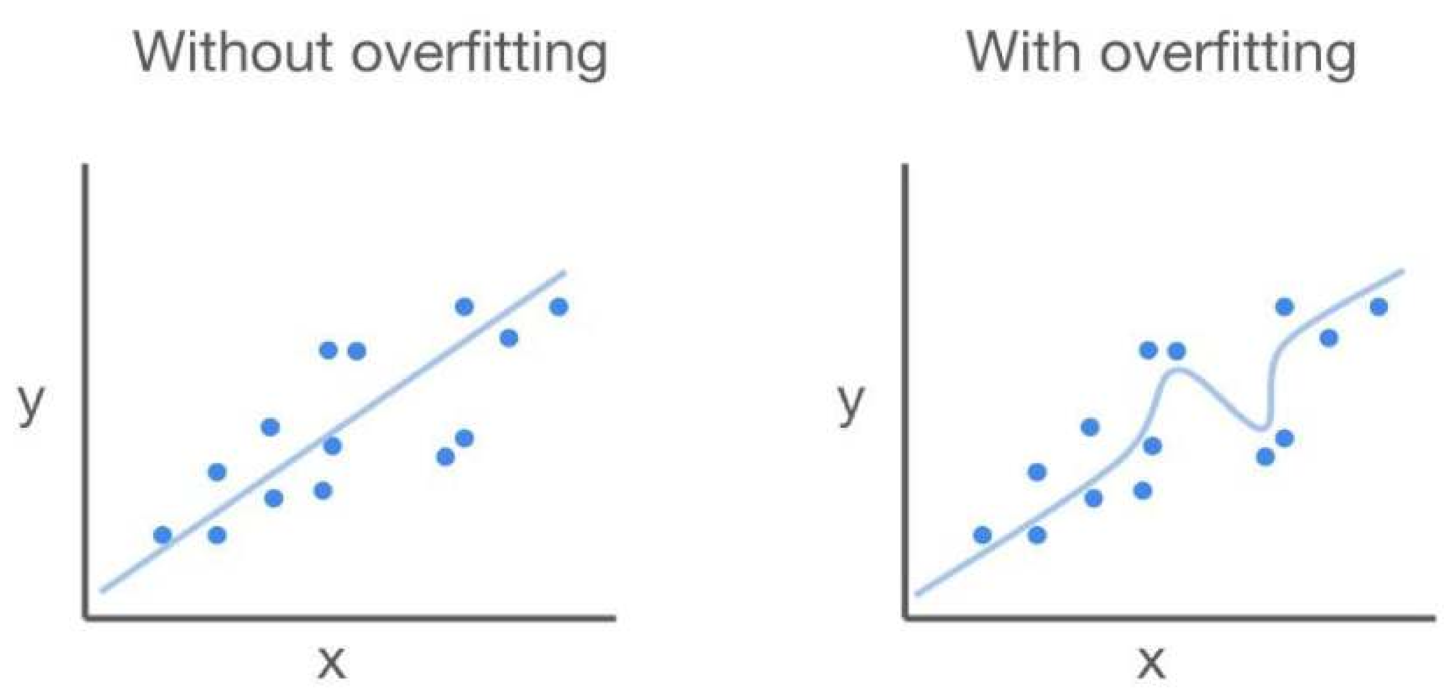
Mathematically, the OLS equation remains the same, but the beta equation transforms depending on whether Lasso or Ridge regression is employed:

$$\beta^{LASSO} = \begin{cases} y^t X_j + \lambda & y^t X_j \leq -\lambda \\ 0 & |y^t X_j| \leq \lambda \\ y^t X_j - \lambda & y^t X_j \geq \lambda \end{cases}$$

$$\beta^{RIDGE} = \frac{1}{1+\lambda} * X^t y = \frac{1}{1+\lambda} * \beta^{OLS}$$

Below is a graphical representation of the overfitting problem and in-sample vs out-of-sample trade-off. Complexity is a proxy for the number of independent variables included in the model.





Source: Crunchingthedata.com

Cross Validation Method and λ Selection

In the lasso and ridge equations, selecting how large of a value that lambda is plays a crucial role in the filtering of independent variables. Higher lambda values result in more variables being filtered out. This paper utilized cross-validation to determine the optimal lambda. Cross-validation involves dividing the model's training set into k folds, training on k-1 folds, and validating on the remaining fold. Lambda is selected based on the model's performance using a cross-validation function, the details of which are outlined in the following link.

Appendix 2.10) OLS, Lasso and Ridge Regression Coefficient Tables

	olsmodel	lassomodel	ridgemodel
lag3_segwit_utilization	-5.182651	-.1372027	-.0694052
segwit_utilization	2.624661		.0232375
lag2_segwit_utilization	2.248417		-.0251171
lag3_avg_block_size	-1.626035		-.0457009
lag1_avg_block_size	1.27779		-.0029171
_cons	1.057544	-3.33e-16	4.44e-16
lag1_segwit_utilization	.5696811		.0017827
avg_block_size	-.5521071		-.0207682
lag2_avg_block_size	.4474141		-.0115623
lag1_avg_totalfee	.3507949	.1987388	.1802626
lag1_avg_block_time	-.1711543		-.0127762
avg_block_time	-.0714736		-.0148678
lag3_avg_totalfee	.0444633	.0324349	.057818
lag3_avg_block_time	.0325792	-.0418153	-.0515691
last_week_avg	.0314311	.0727928	.1051114
lag2_avg_totalfee	.0076731		.0449129
lag2_avg_block_time	.0026886	.0098646	.0391574
avg_tx_count	.0007697	.1185504	.2030597
lag1_avg_tx_count	-.0001625		.0384358
lag2_avg_tx_count	-.000148	-.0153041	-.0602185
lag3_avg_tx_count	.0000948	-.0219834	-.0512699
lag3_avg_price	.0000254		.119848
lag1_avg_volume_btc	.0000189	.0167555	.0772473
avg_price	-.0000148	-.0690601	-.0813361
lag3_avg_volume_btc	-.0000116		-.0298028
lag2_avg_price	-.0000111		.0036463
lag2_avg_volume_btc	-6.46e-06		-.0432876
avg_volume_btc	5.29e-06	.0903375	.1022671
lag3_avg_volume	4.33e-06		.0154038
lag1_avg_price	-3.83e-06		-.084637
lag1_avg_volume	2.83e-06	.061896	.0758446
avg_volume	2.43e-06		.0260946
lag2_avg_volume	-1.56e-06		-.0469565
mempool_size_byte	1.76e-08	.3731761	.3484304
lag2_mempool_size_byte	-7.37e-09	-.0972175	-.1218515
lag3_mempool_size_byte	4.46e-09	.005976	.0721739
lag1_mempool_size_byte	1.53e-09		.0241778