Jugend Forscht - Mathematik / Informatik

Rhein-Main West

Schriftliche Ausarbeitung

18. Januar 2024

What a FEEling

Statistische Analyse und Prognose von Transaktionsgebühren im Bitcoin-Netzwerk

Teilnehmer:
Oscar Stach (18.01.2006, 18 Jahre)
Heinrich-von-Kleist-Schule
Eschborn, Hessen, Deutschland
oscarstachmail@gmail.com

Betreuer:
Rene Pickhardt
Open-Source Bitcoin-Entwickler
Gjøvik, Norwegen
r.pickhardt@gmail.com

Zusammenfassung

Bitcoin-Nutzer zahlen häufig entweder zu hohe Transaktionsgebühren oder warten andernfalls länger als 10 Minuten darauf, dass ihre Transaktion durch die Bitcoin-Miner bestätigt wird. Eine potentielle Ursache dafür ist die hohe Volatilität der Transaktionsgebühren und die resultierende Diskrepanz zwischen den verwendeten und den tatsächlich notwendigen Gebührenraten. Diese Hypothese wollen wir mit statistischen Methoden untersuchen. Dafür extrahieren wir historische Transaktionsdaten und Gebühren aus der öffentlich zugänglichen Bitcoin-Blockchain und reichern diese mit historischen Mempooldaten an. Mithilfe der Analyse wollen wir Trends, Muster und Anomalien in den teilweise stark schwankenden Marktgebühren für Transaktionen beschreiben. Das Ziel ist es, die gewonnenen Erkenntnisse in ein neues Modell zur Vorhersage der Gebühren in der Bitcoin-Software einfließen zu lassen.

Inhaltsverzeichnis

1 Motivation **1**

1.1 Fragestellung 1

2 Wie funktioniert Bitcoin? **2**

2.1 Wallets und Adressen 2

2.2 Peer-to-Peer-Netzwerk und die Blockchain 2

2.2.1 Mempool 2

2.2.2 Blockchain 2

2.3 Sicherung der Blockchain 3

2.3.1 Blockheader und Hashfunktionen 3

2.3.2 Proof-of-Work als Sicherung der Blockchain 3

2.4 Blockbelohnung: Blocksubsidy und Transaktionsgebühren 4

2.5 Transaktionen 4

2.5.1 Erstellung einer Transaktion 4

2.5.2 Marktmechanismus im Mempool 5

2.6 Gebührenprognose / „fee esitmation“ 5

2.7 Transaktionen in der Blockchain 6

3 Materialien, Methodik und Vorgehensweise **6**

3.1 Erstellung der Datensätze 6

3.1.1 Bitcoin-Iterate 6

3.1.2 Extraktion der Mempooldaten 7

3.2 Weiterverarbeitung der Daten 7

3.2.1 Bereinigung der Mempoolverweildauern 7

3.2.2 Median der Transaktionsbestätigungszeiten 8

3.2.3 Kumulative Summe der Gebührendifferenzen 8

3.2.4 Moving-Average der Gebühren 9

3.3 Auswertung der Daten 9

3.3.1 Python und Jupyther-Notebooks 9

3.3.2 Erstellung der Plots 9

4 Ergebnisse **10**

4.1 Korrelation der Transaktionsbestätigungszeiten zur Veränderung der durchschnittlichen Transaktionsgebühren auf Blockebene 10

4.2 Allgemeine Beobachtungen im Datensatz 12

4.2.1 Anzahl steigende sinkende Blöcke 13

4.2.2 Anzahl von Trendwechseln 13

4.3 erste Schritte zur Entwicklung eines eigenen Models für Prognose der Transaktionsgebühren 13

4.3.1 Kolmogorov-Smirnov-Distanz 13

4.3.2 Test der Hyptotehse im Blockintervall [650000;656000] 14

5 Ergebnisdiskussion und Ausblick **14**

5.1 Ergebnisdiskussion 14

5.1.1 Einfluss der Varianz auf die Blockerstellungszeiten 14

5.1.2 Reduzierung auf Blockintervall 15

5.1.3 Child-Pays-For-Parent[5] 15

5.2 Ausblick 15

6 Danksagungen **15**

7 Quellen, Literaturverzeichnis und verwendete Programme **16**

A Anhang **16**

A.1 Eigener Parser Algorithmus 16

A.2 Kumulierte Verteilung der Transaktionsbestätigungszeiten von Block 653932 bis 654112 . 17

1 Motivation[1]

In den letzten Jahren haben Kryptowährungen allgemein und vor allem Bitcoin beziehungsweise das Bitcoin-Netzwerk sehr stark an Popularität gewonnen. Nicht nur der technologische Aspekt der Blockchain-Technologie gewann immer mehr an Interesse, sondern auch das wirtschaftliche Interesse an Bitcoin wuchs. Aktuell hat Bitcoin eine Marktkapitalisierung von rund 764 Milliarden Euro[2] und Bitcoin-Spot-ETF's wurden von der US-Finanzbehörde[9] zugelassen. Dementsprechend ist Bitcoin mittlerweile ein ernstzunehmender Bestandteil des globalen Finanzmarktes geworden. Innerhalb des Bitcoin-Netzwerkes stellen die Transaktionsgebühren einen zentralen Faktor zur Funktionsfähigkeit des Netzwerkes dar - sie sorgen unteranderem für die Sicherheit des Netzwerks. Somit spielen sie langfristig für die Zukunftsfähigkeit des Bitcoins eine tragende Rolle. Bisher wurde die Sicherheit des Netzwerkes primär durch die Ausgabe von Coins als Blockbelohnung ermöglicht. Allerdings geht die Schöpfung dieser neuen Coins langfristig gegen 0, denn die Anzahl an Bitcoins ist auf knapp 21 Millionen limitiert. Dadurch werden nach dem Mining des letzten Blockes mit Blockbelohnung keine neuen Bitcoins mehr als Belohnung für den Miner erstellt werden können und die Transaktionsgebühren werden den einzigen Anreiz für den Miner darstellen das Proof-of-Work-Rätsel zu lösen und die Unveränderlichkeit der Blockchain zu garantieren. Darüber hinaus sind die Gebühren ein elementarer Bestandteil der Nutzungserfahrung der Teilnehmer. Über die Höhe der Transaktionsgebühren wird indirekt bestimmt, wie lange auf die Bestätigung einer Transaktion (Aufnahme in die Bitcoin-Blockchain) gewartet werden muss und natürlich wie teuer das Versenden von Bitcoins ist.

1.1 Fragestellung

Im Bitcoin-Netzwerke haben die Transaktionsgebührenraten eine hohe Volatilität (siehe Abbildung 1). Auf der Abbildung ist zu erkennen, dass es zwar ein vermehrtes Aufkommen an Gebührenraten im

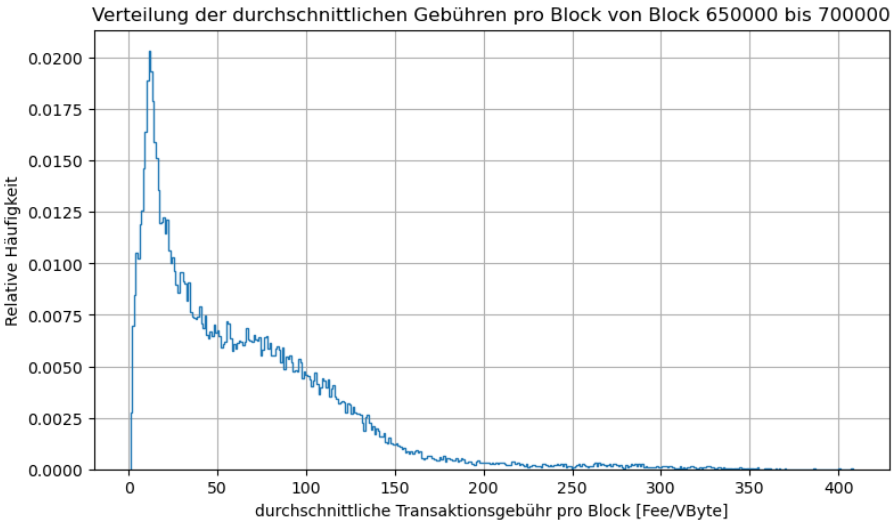

Abbildung 1: Häufgikeitsverteilung der durchschnittlichen Transaktionsgebühren von Block 650000 bis 700000

unteren Bereich gibt, allerdings liegen signifikante Schwankungen und dementsprechend Anteile in dem höheren Bereich der Gebührenraten vor. Durch die starken Schwankungen der Gebührenraten können

[1]Meine initiale Konfrontation mit Bitcoin beziehungsweise dem Bitcoin-Netzwerk hatte ich im Sommer 2023 im Zuge der Deutschen Schüler Akademie. Dort hatte ich an dem Kurs „Wie funktioniert eigentlich der Bitcoin?“ unter der Leitung von René Pickhardt und Janine Römer teilgenommen. Während der Akademie haben wir uns vor allem mit dem konzeptionellen Hintergrund von Bitcoin beschäftigt, dennoch wurde mein Interesse an den technischen Hintergründen des Netzwerkes entfacht. Im Zuge dieser Vertiefung ist diese Jugend-Forscht-Arbeit mithilfe der Unterstützung von René Pickhardt entstanden.

die Nutzer des Bitcoin-Netzwerkes stark beeinflusst werden. Ein Nutzer möchte die Gebührenrate für seine Transaktion möglichst exakt an die Marktgebührenrate anpassen, um eine zeitnahe Bestätigung zu gewährleisten, aber er möchte ebenfalls nicht zu viel bezahlen. Es stellt sich die Frage, wie sich die Gebührenraten bestimmen lassen und welche Einflüsse die Schwankungen auf die Gebührenprognose haben.

2 Wie funktioniert Bitcoin?

Um zu verstehen, warum die Gebühren schwanken, müssen wir begreifen, welche Funktion die Transaktionsgebühren im Bitcoin-Netzwerk haben. Dafür gucken wir uns die grundlegende Funktionsweise des Netzwerkes an.

2.1 Wallets und Adressen

Die Teilnahme am Bitcoin-Netzwerk ist für jede Person zugänglich und jede Person hat grundsätzlich die gleichen Rechte und Privilegien. Als Teilnehmer des Bitcoin-Netzwerkes erstellt man sich zunächst ein eigene Adresse (mit eigenem und privatem Key). Eine solche Adresse fungiert im Grunde analog, wie die Kontonummer bei einer Bank, nur das keine Bank oder sonst jemand weiß, zu wem diese Adresse gehört. Mithilfe dieser Adresse kann man Bitcoin empfangen und versenden. Ein Teilnehmer kann sich beliebig viele Adressen generieren und diese in einem Wallet zusammenführen - ein Wallet ist wie eine digitale Geldbörse.

2.2 Peer-to-Peer-Netzwerk und die Blockchain

Das Konzept des Bitcoin-Netzwerks wurde erstmals 2008 von Satoshi Nakamoto im Bitcoin-Whitepaper[4] veröffentlicht. Bei dem Bitcoin-Netzwerk handelt es sich um ein dezentrales Netzwerk. Es gibt keine zentrale Instanz und jegliche Informationen werden im Peer-to-Peer-Prinzip verwaltet und versendet. Das zentrale Element ist eine redundante Datenbank, die jeder individuell verwaltet. Sie wird als Blockchain bezeichnet. Neben der Blockchain unterhält jeder Nutzer einen eigenen Mempool.

2.2.1 Mempool

Der Begriff Mempool („Memory-Pool“) bezeichnet den individuellen Speicher von jedem Teilnehmer im Netzwerk in dem die Anwärtertransaktionen für den nächsten Block gespeichert werden. Sobald eine Node eine Transaktion empfangen und geprüft hat, fügt sie diese ihrem Mempool hinzu. Wenn später eine Transaktion bestätigt wurde, also in der Blockchain aufgenommen wurde, dann wird sie aus dem Mempool entfernt.

2.2.2 Blockchain

Die Blockchain ist das Register aller bestätigten Bitcointransaktionen, die das Bitcoin-Netzwerk akzeptiert hat. Sie kodiert damit insbesondere die Eigentumsverhältnisse der existierenden Bitcoins. Die einzelnen Transaktionen werden immer in Blöcken gespeichert, die durchschnittlich alle 10 Minuten gemined beziehungsweise erstellt werden und die maximal etwa 4MB groß sein dürfen, weshalb etwa 4000 Transaktionen in einen Block passen. Jeder Teilnehmer führt seine eigene Kopie der Blockchain und kann theoretisch alles an ihr verändern. Im Peer-to-Peer-Prinzip tauschen sich die Nodes (Rechner im Netzwerk) aus und vergleichen und adaptieren ihre Blockchains. Konsens über Änderungen an der Blockchain, wie das Hinzufügen von neuen Transaktionen, wird durch das Proof-of-Work-Verfahren ermöglicht.

2.3 Sicherung der Blockchain

Der elementare Bestandteil der Sicherung der Blockchain ist die Verkettung der Blöcke. Dies wird über die Blockheader und die Eigenschaften von Hashfunktionen ermöglicht.

2.3.1 Blockheader und Hashfunktionen

Ein Block beinhaltet neben den Transaktionen auch noch einen Blockheader. In ihm sind die zentralen Informationen des Blocks zusammengefasst und er beinhaltet die Versionsnummer der Software, den Hash des vorigen Blocks, den Hash des Merkle-Roots[10], den Zeitstempel, die Dificulty des Blocks und eine Zufallszahl, auch Nonce genannt.

Ein Hash ist in diesem Fall der Ausgabewert der SHA-256-Hashfunktion[11]. Der zentrale Aspekt dabei ist, dass ein bestimmter Eingabewert immer nur einen einzigartigen Ausgabewert erzeugt und man keine Rückschlüsse auf den Eingabewert ziehen kann, wenn man nur den Ausgabewert kennt. Allerdings lässt sich sehr schnell nachprüfen, ob Eingabewert und Ausgabewert zusammenpassen, wenn man beide Informationen hat. Diese Eigenschaft wird sich hier zu Nutze gemacht. Der Eingabewert für den Merkleroot sind alle Hashes der einzelnen Transaktionen, die sich wiederum als Ausgabewerte aller jeweiligen Informationen der Transaktionen ergeben.Der Merkleroot repräsentiert also alle Transaktionen. Diese 6 Informationen des Blockheaders werden wiederum zusammengehasht und ergeben den Blockhash. Wie

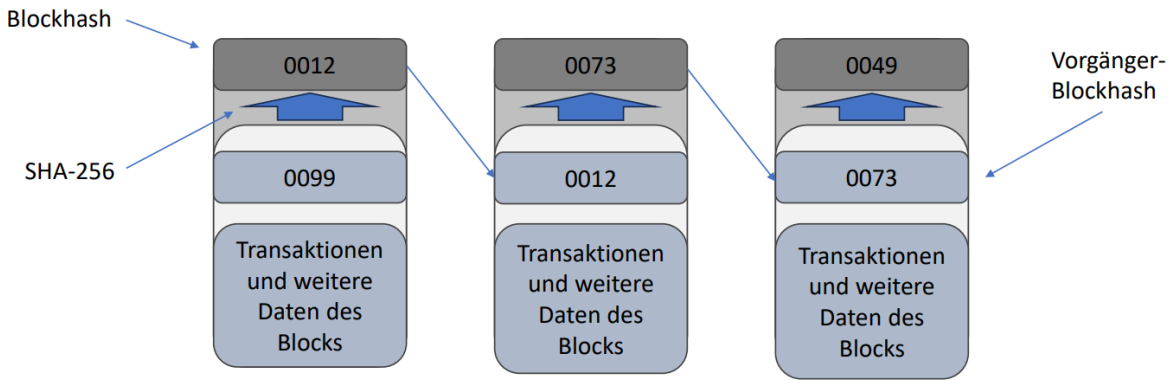

Abbildung 2: Konzeptioneller Aufbau der Blockchain

bereits beschrieben ist ein Element bei der Erstellung eines neuen Blockes der Blockhash des vorangegangenen Blockes. Somit sind die Blöcke aneinander gekettet und diese Kette ist die oben beschrieben Blockchain. Dadurch hat eine Veränderung der Blockchain, oder nur eine Veränderung einer Transaktion in den zurückliegenden Blöcken zur Folge, dass sich alle folgenden Blöcke (genauer: deren Blockhashes) ändern würden und deshalb die Verkettungen und alle Blöcke seit der Veränderung neu berechnet werden müssen. Damit das nicht so einfach geht, vertraut das Netzwerk nur der Blockchain mit dem meisten Proof-of-Work.

2.3.2 Proof-of-Work als Sicherung der Blockchain

Um zu gewährleisten, dass man nicht beliebige Daten in der Blockchain verändern kann, gibt es das Proof-of-Work-Verfahren. Proof-of-Work bedeutet übersetzt: Arbeitsnachweis. Das heißt, man muss Rechenleistung aufbringen und diese nachweisen.

Bei der Erstellung beziehungsweise beim Mining eines neuen Blockes muss der Block eine bestimmte Voraussetzung erfüllen: Er benötigt eine bestimmte Anzahl an führenden Nullen im Blockhash, welche durch die Diffuculty angegeben wird. Wie in Unterunterabschnitt 2.3.1 beschrieben gibt es die zufällige

Nonce als Element in der Generierung des Blockhashes. Durch das Verändern dieser Nonce verändern sich die Ausgabewerte der Hashfunktion. Mittels Bruteforcing lässt sich eine Nonce finden, für die der Blockhash genügend führende Nullen aufweist. Im Durchschnitt wird alle 10 Minuten eine solche Zahl gefunden und ein Block erstellt. Dadurch ist gewährleistet, dass, wenn man etwas an der etablierten Blockchain verändern würde, man alle Folgeblöcke neu berechnen müsste, um das Netzwerk zu überzeugen. Selbst wenn man es schafft temporär neue Blöcke schneller zu berechnen, wird langfristig das Netzwerk gewinnen, so lange es mehr als 50% der Rechenleistung aller Teilnehmenden besitzt. Dabei wird davon ausgegangen, dass die Mehrheit der Nodes, genauer gesagt der Rechenleistung, ehrlich ist und an der Echtheit der Blockchain interessiert ist. Um dies zu incentivieren erhält eine Node, die einen Gültigen Block findet, die Blocksubsidy (Teil der Blockbelohnung, bei der neue Bitcoins entstehen) und die Transaktionsgebühren.

2.4 Blockbelohnung: Blocksubsidy und Transaktionsgebühren

Wenn ein Miner es schafft die richtige Zufallszahl zu finden, dann publiziert er diesen Block mitsamt aller Transaktionen und dem Blockheader.Die Teilnehmenden des Netzwerkes können nun prüfen, ob dieser richtig erstellt wurde und ihn anschließend in ihre individuellen Blockchains aufnehmen. Der Miner erhält dafür das Recht, alle Transaktionsgebühren der in diesem Block bestätigten Transaktionenan sich selbst zu schicken und in der ersten Transaktion des Blockes einen bestimmten Betrag an Bitcoin auszugeben beziehungsweise an sich selbst zu senden ohne diese jemals besessen zu haben. Dadurch generiert er neue Bitcoins. Jedoch wird diese Belohnung in Zukunft immer kleiner durch die sogenanten Halvings[1].

2.5 Transaktionen

Grundsätzlich beginnt das Versenden von Bitcoins[2] mit dem Erstellen einer Transaktion. Nachdem eine gültige Transaktion erstellt und mit Hilfe des privaten Schlüssels signiert wurde, wird diese im Netzwerk veröffentlicht. Die anderen Teilnehmer prüfen, ob die Transaktion zum einen korrekt formatiert ist und durch Überprüfung der Signatur, ob die Coins rechtmäßigerweise ausgegeben wurden. Falls diese Voraussetzungen erfüllt sind, wird sie im Peer-to-Peer Netzwerk an die anderen Nodes weitergeleitet. Zunächst ist die Transaktion dann im (jeweils individuellen) Mempool (siehe Unterunterabschnitt 2.2.1 zu finden und sobald die Transaktion in einem Block gemined wurde, gilt sie als bestätigt.

2.5.1 Erstellung einer Transaktion

Der Ersteller gibt an, welche „UTXO‘s“[3] ausgegeben und an welche Teilnehmer beziehungsweise Adressen diese geschickt werden sollen. Zusätzlich muss man bei der Erstellung der Transaktion die Gebühren

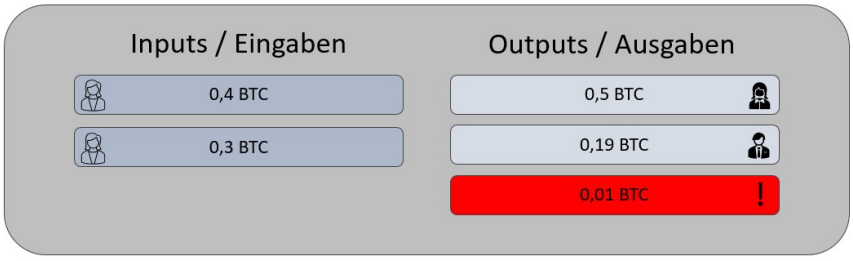

Abbildung 3: Grundlegender Aufbau einer Bitcoin Transaktion

beachten. Diese sind zwar grundsätzlich nicht explizit im Bitcoin-Protokoll festgeschrieben, allerdings

[2] genauer genommen Satoshis
[3] Unspent Transaction Output, siehe Kapitel 9 in [4]

nimmt kein wirtschaftlich kluger Miner eine fremde Transaktion ohne Gebühren in seinen Block auf. Diese werden nicht speziell definiert, sondern ergeben sich aus der Differenz der Inputs und Outputs. Die Gebühren sind immer größer oder gleich 0, da eine Transaktion[4] nur gültig ist, wenn die Summer der Outputs größer ist als die Summe der konsumierten Inputs.

Rechnung zur Abbildung 3 als Beispiel:

$$Transaktionsgebühr = \underbrace{(0,4+0,3)}_{\text{inputs}} - \underbrace{(0,5+0,19)}_{\text{outputs}}$$

$$\Rightarrow Transaktionsgebühr = 0,01 \ \ BTC$$

In diesem Beispiel beträgt die Gebühr also $0,01$ BTC. Will man eine Bitcoin Transaktion durchführen, stellt sich die Frage, wie hoch die Gebühren zu wählen seien sollten? Man möchte nicht zu viel bezahlen, allerdings gleichzeitig auch nicht zu lange auf die Bestätigung der Transaktion warten.

2.5.2 Marktmechanismus im Mempool

Der Mempool ist im Grunde in freier Markt. Die Auswahl, welche Transaktion es in den Block schafft folgt im dem klassischen Marktmechanismus von „Angebot und Nachfrage“. Dabei handelt sich beim Angebot um den Platz beziehungsweise die Speicherkapazitäten im aktuellen Block und bei der Nachfrage um die Anzahl der Transaktionen mit der Eigenschaft der individuellen Transaktionsgebühren. Aus wirtschaftstheoretischer Sicht ist das Angebot stets konstant, da die Blockgröße auf 1 MB limitiert ist und die Nachfrage ist bedingt elastisch.

Man möchte nun annehmen, dass solcher Miner simplerweise die obersten X Transaktionen mit den höchsten Gebühren nehmen könnte und den Block so auffüllt. Allerdings herrscht in diesem Markt eine asymmetrische Informationsverteilung und die einzelnen Transaktionen sind unterschiedlich groß oder können bereits an eine weitere Transaktion gekoppelt sein. Das bedeutet, nicht jeder Teilnehmer des Netzwerkes sieht die gleichen Transaktionen oder Koppelungen. Dadurch hat jeder Teilnehmer einen eigenen Mempool, sprich einen individuellen temporären Speicher der Anwärtertransaktionen für den nächsten Block. Ein wirtschaftlich kluger Miner wird die Transaktionen so wählen, dass die gesamten Gebühren, die er sich in dem Block auszahlen darf maximiert wird. Daher gelten prinzipiell die Grundannahmen:

1. Je höher die Gebührenrate der Transaktion, desto eher wird sie bestätigt.
2. Je mehr Teilnehmer gleichzeitig Transaktionen bestätigt haben wollen, desto höher wird die durchschnittliche Gebühr für den nächsten Block.

Zur besseren Vergleichbarkeit der Transaktionsgebühren wird die relative Einheit Satoshi[5]/VByte[6] verwendet, da Transaktionen durch verschiedene Formate und Anzahlen an In- und Outputs unterschiedlich groß sein können und unterschiedlich viel Platz im Block einnehmen.

2.6 Gebührenprognose / „fee esitmation“

Aktuell gibt es bereits einen Schätzer für Transaktionsgebühren in der Standard-Software Bitcoin Core[3]. Dieser erstellt sogenannte „fee-rate buckets“. Das bedeutet, dass Transaktionen, mit Gebührenraten in einem bestimmten Bereich, in Gruppen eingeteilt werden. Anschließend zählt Bitcoin Core wie lange es für die einzelnen Transaktionen der Gruppen i der kürzeren Vergangenheit dauert, gemined beziehungsweise bestätigt zu werden und erneuert diesen Datensatz stets mit den letzten Transaktionen, um ihn aktuell

[4] Außer wie oben beschrieben die Coinbasetransaktion

[5] Untereinheit von Bitcoin 100000000:1

[6] virtual byte: Bitcoin gewichtet die verschiedenen Transaktionsformate unterschiedlich, wodurch diese neue Einheit entsteht[6].

zu halten. Beispielsweise, wenn man eine Transaktion erstellen möchte, die in maximal 6 Blöcken (entspricht ungefähr 60 Minuten) bestätigt werden soll, dann wird geprüft, ob, beginnend bei der Gruppe mit den höchsten Gebührenraten, der Anteil der in maximal 6 Blöcken bestätigten Transaktionen über einer bestimmten Grenze liegen. Falls dies der Fall ist, wird das Verfahren mit der nächst-niedrigeren Gruppe wiederholt bis die Grenze unterschritten wird. Die Gebührenrate dieser Gruppe ist dann ein Indikator dafür, welche Gebührenrate einen relevant hohe Wahrscheinlichkeit zur Bestätigung in der gewünschten Zeitdauer von 6 Blöcken hat.
Neben diesem Algorithmus zur Gebühreprognose gibt es auch private Anbieter, wie beispielsweise „mempool.space“ oder „whatthefee.io“.

2.7 Transaktionen in der Blockchain

Nach der Erstellung der Transaktion wird diese im Peer-to-Peer-Netzwerk veröffentlicht und in den Mempool der anderen Nodes (und Miner) aufgenommen und bei adäquater Gebührenrate in den nächsten Blöcken bestätigt. Wenn die Transaktion nun in die Blockchain aufgenommen wurde, dann sind die Outputs dieser Transaktion zukünftig als Inputs für neue Transaktionen möglich und wieder versendbar.

3 Materialien, Methodik und Vorgehensweise

Grundsätzlich benötigen wir die Daten aller Transaktionen in der Bitcoin-Blockchain, um die Gebühren auf Block- und Transaktionsebene berechnen zu können. Aus der Blockchain extrahieren wir außerdem die Zeitstempel der Blockerstellungen beziehungsweise der Transaktionsbesätigungen. Zusätzlich benötigen wir die Mempooldaten der letzen Jahre[7], um die Zeitstempel der ersten Sichtungen der einzelnen Transaktionen im Mempool zu extrahieren.

Bei den Daten der gesamten Transaktionen sprechen wir über riesige Datengrößen. Hier ist ein kleiner Überblick der Datengrößen mit denen wir konfrontiert und herausgefordert waren:

Bitcoin Blockchain	620 GB
Mempool-Daten von C. Decker	60 GB
Dictionary der Mempool-Transaktionen ab Blockhöhe 650000	10 GB
Datensatz der Tranasktionen Blockhöhe 650000 bis 660000	17 GB

3.1 Erstellung der Datensätze

Grundsätzlich ist das Ziel mit einer minimalen Anzahl an Calls an die Bitcoin-API der lokalen Node die benötigten Informationen zu bekommen und die Daten zur einfacheren Weiterverarbeitung in einer CSV-Datei zu speichern. Zu Beginn haben wir einen eigenen Crawler geschrieben, um die durchschnittlichen Transaktionsgebührenraten pro Block zu berechnen und zu speichern. Eine schematische Darstellung des Algorithmus ist im Anhang zu finden in Unterabschnitt A.1.

Dennoch können die API-Calls teilweise Sekunden beanspruchen, weshalb wir für eine bessere Performance auf das Programm Bitcoin-Iterate zurückgegriffen haben.

3.1.1 Bitcoin-Iterate

Das Programm „Bitcoin-Iterate[7]“ , welches von Rusty Russel entwickelt wurde, ermöglicht es einem keine API-Calls mehr zu verwenden, sondern die Blockdateteien der lokalen Node direkt zu iterieren und ermöglicht somit eine deutlich schnellere Laufzeit. Ein von uns verwendeter Befehl ist beispielsweise:

[7]Wir bedanken uns bei Christian Decker für die Zuverfügungsstellung der historischen Mempool-Daten seiner Node.

bitcoin-iterate –start=650000 –end=660000 –tx='%th %bs %bN > output.csv'

Dieser Befehl über die Kommandozeile gibt uns dieser alle Transaktionshashes, den jeweiligen Timestamp und Blockhöhe aus. Diese Daten werden erneut in einer CSV-Datei gespeichert und später weiter verarbeitet und nutzbar gemacht.

3.1.2 Extraktion der Mempooldaten

Von Christian Decker haben wir die seit etwa 2015 aufgezeichneten Mempooldaten zur Verfügung gestellt bekommen. Dieser beinhaltet den Transaktionshash und den Zeitpunkt der ersten Sichtung der Transaktion im Mempool, aber auch systematische Fehler. Zum einen war die Node nicht durchgängig online, wodurch nicht alle Transaktionen aufgezeichnent wurden und zum anderen ist nicht gewährleistet, dass die Node die Transaktionen unmittelbar nach der Veröffentlichung erhalten hat. Mithilfe der Python-Library „bzcat"kann man auf die Datei mit 750.000.000 Einträgen zugreifen und durch sie iterieren.

3.2 Weiterverarbeitung der Daten

Wir haben nun jeweils zwei Werte zu jeder Transaktion. Zum einen, wann die Transaktion zuerst im Mempool gesichtet wurde und wann sie in der Blockchain aufgenommen wurde, sprich bestätigt wurde. [8]Durch das Bilden der Differenz der beiden Werte lässt sich berechnen, wie groß die Verweildauer der Transaktion im Mempool ist und dieser neue Wert der jeweiligen Transaktion zuweisen.

3.2.1 Bereinigung der Mempoolverweildauern

Die Erstellungszeiten der einzelnen Blöcke können stark variieren. Grundsätzlich folgt die Verteilung der Erstellungszeiten einer Poissonverteilung.

Es zeigt sich das mehr als 60% der Blöcke in unter 10 Minuten erstellt werden und 13,5% der Blöcke

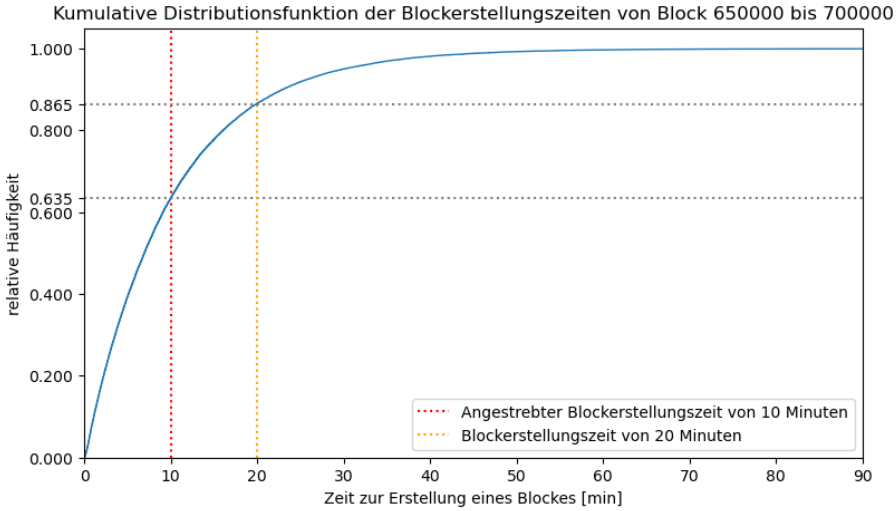

Abbildung 4: Kumulative Distributionsfunktion der Blockerstellungszeiten

werden erst nach mehr als 20 Minuten erstellt beziehungsweise das Proof-of-Work-Rätsel wird erst nach 20 Minuten gelöst. Bei diesen starken Schwankungen schleicht sich ein systematischer Fehler ein. Beispielsweise, wenn das Proof-of-Work-Rätsel zufälligerweise sehr lange nicht gelöst wird, dann steigen die

[8]Allerdings sind beide Datensätze unsortiert und enthalten teilweise Transaktionen, die nicht im anderen Datensatz zu finden sind. Teilweise hat C. Decker's Node die Transaktion nie erhalten obwohl die Transaktion später bestätigt wurde und manche gesichteten Transaktionen wurden nie in der Blockchain bestätigt. Entsprechend haben wir den Datensatz bereinigt, in dem wir uns in der folgenden Analyse nur auf die Transaktionen beschränken, die Sowohl in Christians Mempool als auch in der Blockchain vorkamen.

Mempoolverweildauern ebenfalls sehr stark an und verzerren den Datensatz. Denn dann können beispielsweise unabhängig von einem Anstieg der Gebühren die Mempoolverweildauern stark ansteigen und somit die Feststellung ihrer Korelation behindern indem ein steigender Block dennoch sehr lange Mempoolverweildauern der Transaktionen hätte.

Demnach haben wir von den Mempoolverweildauern jeweils ihre Blockerstellungszeit subtrahiert, um diesen systematischen Fehler auszugleichen. Dadurch können negative Transaktionsbestätitungszeiten (bereinigte Mempoolverweildauern) entstehen. Beispiel:

$$\underbrace{8\,\text{min}}_{\text{Mempoolverweildauer}} - \underbrace{12\,\text{min}}_{\text{Blockerstellungszeit}} = bereiningte\ \ Transaktionsbestätigungszeit$$

Diese neue Eigenschaft der Transaktionsbestätigungszeiten gibt uns die Möglichkeit eine neue Erkenntnis aus den Daten zu gewinnen. Wenn eine Transaktion eine negative Transaktionsbestätigungszeit hat, dann können wir davon ableiten, dass sie erst nach der Veröffentlichung des letzten Blockes erstellt wurde und die Transaktion es in den ersten möglichen Block geschafft hat.

3.2.2 Median der Transaktionsbestätigungszeiten

Die Transaktionsbestätigungszeiten, die wir in Unterabschnitt 3.2 berechnet haben, verbinden wir mit ihren korrespondierendem Block. Um für spätere Berechnungen ein vergleichbareren Wert zu haben, bilden wir den Median der Transaktionsbestätigungszeiten. Dieser stellt insofern einen besseren Wert zum Vergleichen dar, da die Bestätigungszeiten innerhalb eines Blockes teilweise sehr starke Schwankungen haben. Der Median glättet und wirkt dem Rauschen der Daten entgegen, da er nicht so anfällig ist für vereinzelte Ausschläge.

3.2.3 Kumulative Summe der Gebührendifferenzen

Wir wollen, um die Daten später besser auswerten zu können, Spikes, Anstiege / Abstiege und Anomalien im Datensatz der Gebühren zuverlässig erkennen. Dafür benutzten wir die kumulative Summe der Differenzen der einzelnen zeitlich chronologischen Gebührenwerte.
Um dies zu tun, müssen wir einiges definieren:

$$\begin{aligned}
\Delta b &= Blockintervall \\
S_n &= Gebührenveränderung \\
K_n &= Schwankungsrate \\
n &= aktuelle\ \ Blockhöhe \\
k &= Schwellenwert \\
K_n &= \sum_{i=0}^{\Delta b} |\text{ø}Fee_{Block_{n-i}} - \text{ø}Fee_{Block_{n-i-1}}| \\
S_n &= \text{ø}Fee_{Block_n} - \text{ø}Fee_{Block_{n-\Delta b}}
\end{aligned}$$

Wenn $S_n > k$, dann ist der Aktuelle Trend der Gebühren stark steigend.
Wenn $S_n < -k$, dann ist der Aktuelle Trend der Gebühren stark fallend.
Mit diesen Berechnungen lässt sich für jeden Block der aktuelle Trend der Gebühren errechnen. Wenn K_n einen großen Wert erreicht, dann bedeutet das im Sachzusammenhang, dass die Gebühren im zurückliegenden Blockintervall stark geschwankt haben, selbst wenn S_n nah bei null liegt. Wenn nun in einem Δb die Gebührenraten stark ansteigen, dann wird S_n dementsprechend groß. Grundsätzlich bedeutet dies, dass vom Beginn des Blockintervalls Δb bis zum aktuellen Block die Gebühren sich um mindestens den Wert k verändert haben.

3.2.4 Moving-Average der Gebühren

Beim Betrachten der rohen Daten fällt auf, dass es extrem starke Schwankungen gibt. Um den allgemeinen Verlauf besser zu visualisieren und zu glätten, nehmen wir den Moving-Average der durchschnittlichen Gebühren des Blockintervalls der letzten Blöcke. Damit wir einen menschlichen Zeitraum haben, entschieden wir uns für das Intervall von 144 Blöcken beziehungsweise 1008, was etwa einem Tag beziehungsweise einer Woche entspricht. Moving-Average bedeutet, dass man ausgehend vom betrachteten Block den Durchschnitt der durchschnittlichen Gebühren der Vorgängerblöcke im Intervall bildet. Dieser bewegt sich, da wir immer nur den Durchschnitt der letzten Δb Blöcke betrachten. Die Intervallgröße ist freiwählbar und anpassbar und im Grunde in unserer Forschung arbiträr mit 144 gewählt worden.

3.3 Auswertung der Daten

Die gesamten geschrieben Python-Skripte, sowie Jupyther-Notebooks sind im Github-Repository einsehbar[8]

3.3.1 Python und Jupyther-Notebooks

Um die Daten auszuwerten haben wir die Programmiersprache Python benutzt, da diese sich sehr gut durch ihre Libraries für Data-Science eignet. Primär benutzten wir die Librarys: Pandas, Numpy und Scipy.
Bei den meisten Untersuchungen waren wir mit minutenlangen Wartezeiten konfrontiert, da die Daten immer wieder neu in den Arbeitsspeicher geladen werden mussten. Deshalb haben wir die Daten zusätzlich in Jupyther-Notebooks ausgewertet, um die Variablen und Dataframes konstant im Arbeitsspeicher zu halten und somit Zeit zu sparen und die Auswertung effektiver zu gestalten. Durch das Gruppieren und Selektieren einzelner Spalten und Zeilen, das Zusammenführen von einzelnen Dataframes in Pandas lassen sich die Daten neu interpretieren und besser visualisieren.

3.3.2 Erstellung der Plots

Alle Diagramme und Graphiken in dieser Jugend-Forscht-Arbeit haben wir selber basierend auf den gegebenen Daten erstellt. Hauptsächlich haben wir dafür die Matplotlib.Pyplot-Library verwendet. Diese gab die Möglichkeit die großen Datenmengen besser zu visualisieren und zu verstehen.

4 Ergebnisse

4.1 Korrelation der Transaktionsbestätigungszeiten zur Veränderung der durchschnittlichen Transaktionsgebühren auf Blockebene

Wenn es einen Spike oder Anstieg der Transaktionsgebühren über einen bestimmten Zeitraum gibt, dann resultiert das in temporär kürzeren Wartezeiten, aber langfristig, nach einer Normalisierung der Gebühren, folgen hohe Wartezeiten durch die bereits vor dem Spike oder Anstieg erstellten Transaktionen, die nun „abgearbeitet“ werden. Um diese Hypothese zu prüfen, erkennen wir zunächst die Anstiege im Datensatz der Transaktionsgebühren .
Um das Verfahren zu vereinfachen und zu beschleunigen, betrachten wir zunächst ausschließlich das Intervall [650000;700000] der Blöcke (siehe Abbildung 5). Zunächst erkennt man wenig. Lediglich eine

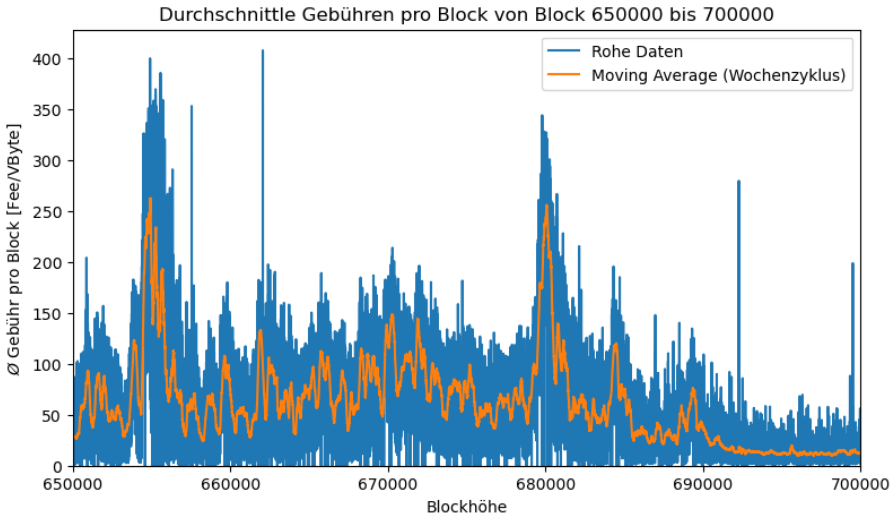

Abbildung 5: Durchschnittliche Gebühr pro Block von Block 650000 bis 700000

überwältigende Menge an Daten. Deshalb verkleinern wir unseren Scope weiter und wählen arbiträr ein kleineres Intervall.

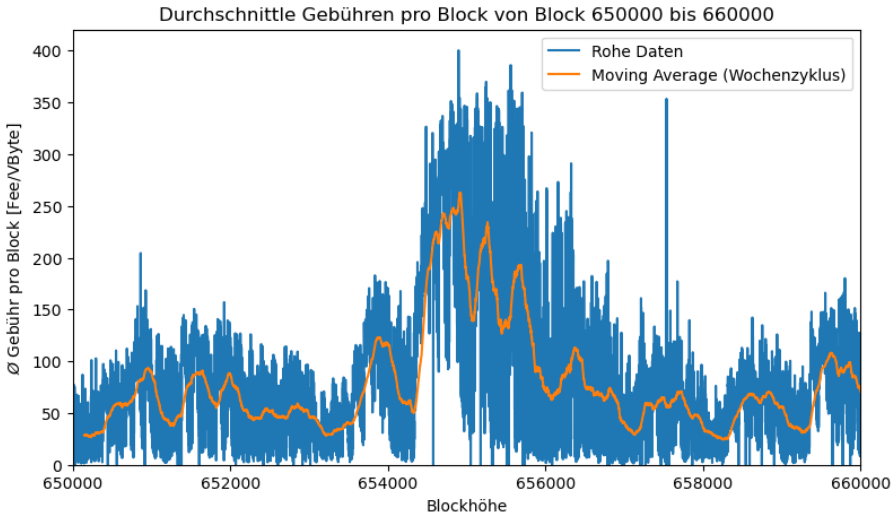

Abbildung 6: Durchschnittliche Gebühr pro Block von Block 650000 bis 660000

Wir sehen im Blockintervall [653500:654250] einen klaren starken Anstieg und wieder einen klaren starken Abstieg der Transaktionsgebühren pro Block und richten unseren Fokus zunächst auf dieses

Intervall. Zwar lässt sich mit dem bloßen Auge ein Anstieg und Abstieg der Gebühren beobachten, dennoch untersuchen wir dieses Intervall noch einmal algorithmisch.

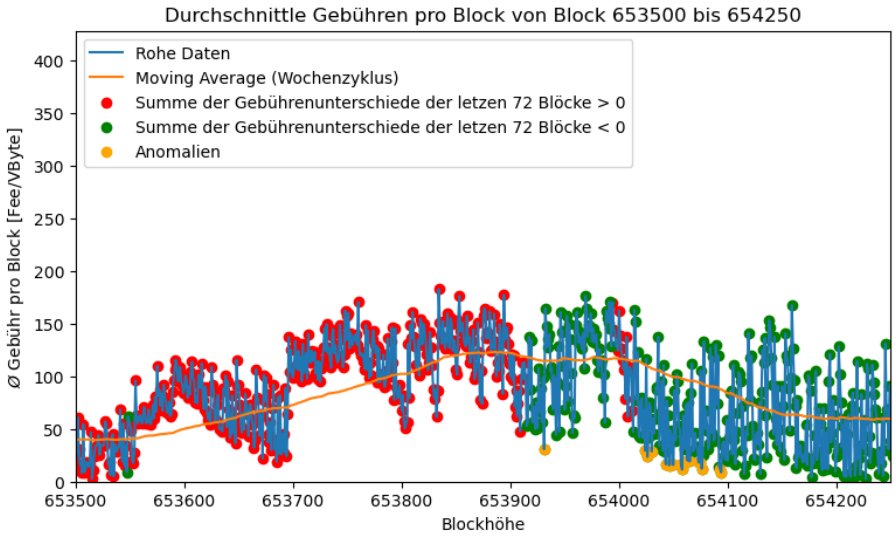

Abbildung 7: Durchschnittliche Gebühren pro Block von Block 652500 bis 654250 mit Indikatoren

Für diese Abbildung wurden die Parameter $\Delta b = 72$ und $k = 0$ gewählt und die Achsenskalierung in 144er-Schritten (≈ 1 Tag) gewählt. Es lässt sich klar ein klarer starker Anstieg, markiert durch die roten Punkte, erkennen, und ein darauf folgendes Sinken, markiert durch die grünen Punkte, und somit ein Wendepunkt bei einer Blockhöhe von etwa 654000 erkennen.
Deshalb betrachten wir nun die Verteilungen der Verweildauern im Mempool der einzelnen Transaktionen für die jeweiligen Folgeblöcke. Den Datensatz zu den jeweiligen Bestätigungszeiten der Transaktionen und zu den Erstellungszeiten der Blöcke haben wir in Abschnitt 3 bereits erstellt. Wir bereinigten den Datensatz um den systematischen Fehler der unterschiedlichen Erstellungszeiten der Blöcke, indem wir diese von den Bestätigungszeiten der jeweiligen Transaktionen subtrahieren (siehe Unterunterabschnitt 3.2.1). Wenn wir nun die Mediane der Transaktionsbesätigungszeiten der Folgeblöcke betrachten (siehe Abbildung 8), lässt sich eine klare Beobachtung machen. Wenn die Gebühren ansteigen, man sich also in einem steigendem Block befindet, dann sind die Zeiten bis zur Transaktionsbestätigungszeiten minimal - wenn nicht sogar negativ. Die negativen Werte entstehen durch die beschriebene Bereinigung und bedeuten,

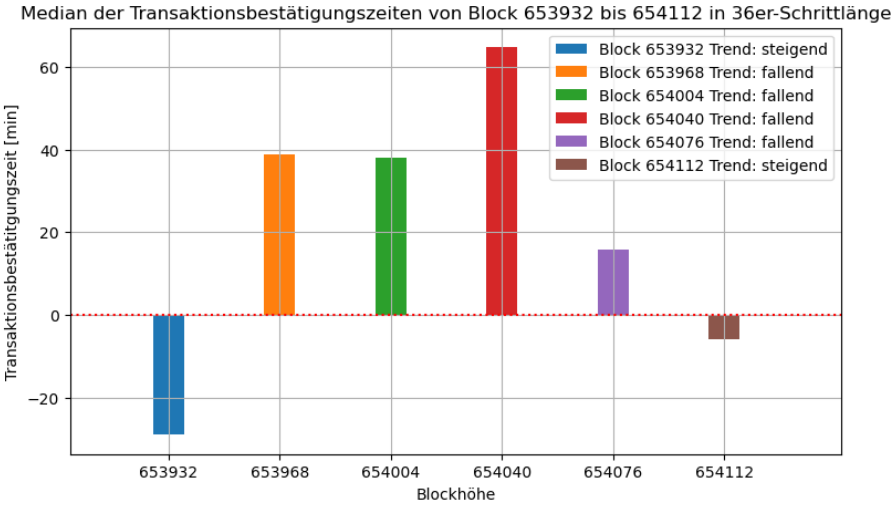

Abbildung 8: Balkendiagramm der Mediane der Transaktionsbestätigungszeiten von Block 653932 bis 654112 in 36er-Schrittlänge

die Transaktionen wurden erste seit der letzten Blockerstellung erstellt und somit neu sind. Für fallende Blöcke sind die Mediane der Transaktionsbestätigungszeiten positiv. Das bedeuet, sie wurden bereits vor der Erstellung des letzten Blockes im Mempool veröffentlicht und warteten seitdem auf ihre Bestätigung. Da wir die Mediane als repräsentatives Medium gewählt haben, lässt sich davon ableiten, dass mindestens 50% der Transaktionen der Blöcke mit steigendem Trend neu erstellt wurden und es sofort in den nächsten Block geschafft haben.

Bei Betrachtung des gesamten Intervalls [650000;656000] (siehe Abbildung 9) lässt sich das gleiche Er-

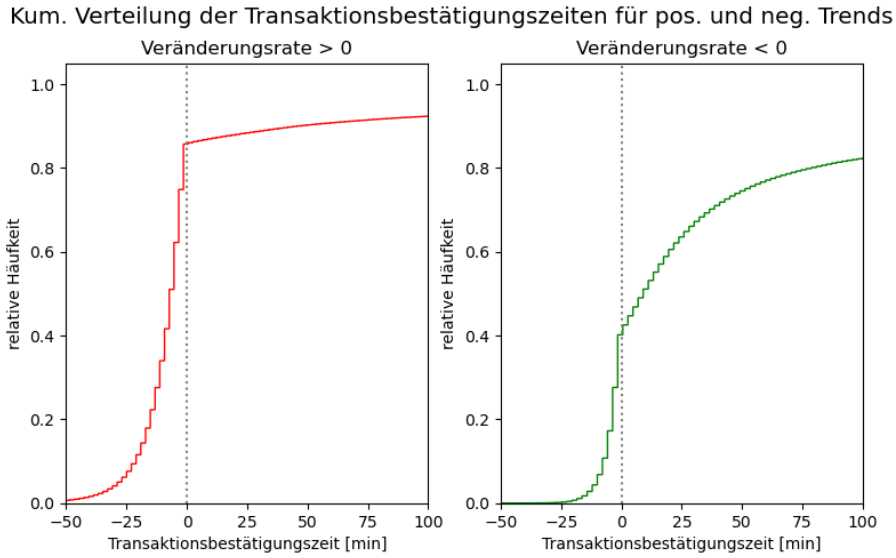

Abbildung 9: Kumulative Verteilung der Transaktionsbestätigungszeiten für positive und negative Gebührentrends

gebnis feststellen. Wenn die Gebühren steigen, dann sind die Transaktionsbestätigungszeiten zu über 80% negativ und wenn sie fallen, dann sind sie das nur zu knapp 40%.

4.2 Allgemeine Beobachtungen im Datensatz

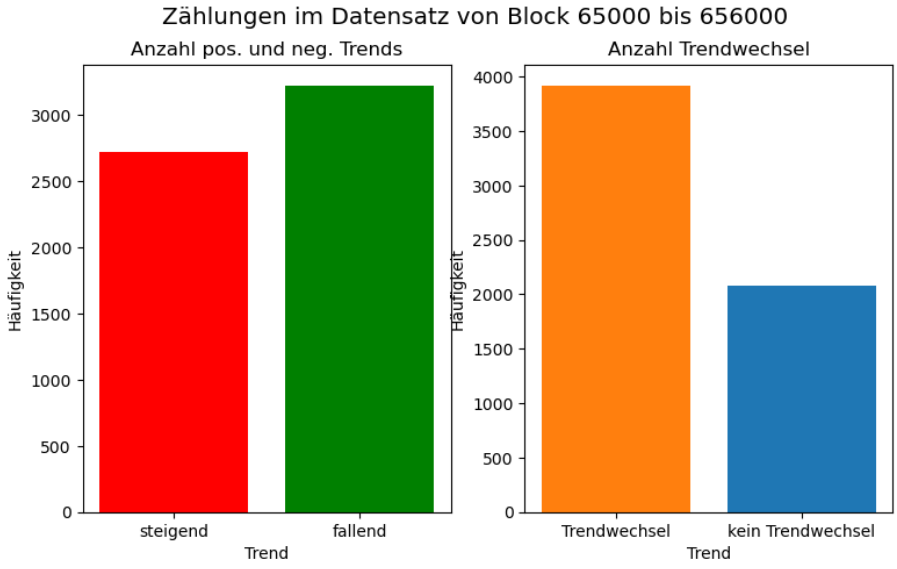

Abbildung 10: Zählungen von Trends und Trendwechseln

4.2.1 Anzahl steigende sinkende Blöcke

Im betrachteten Blockintervall von [650000;656000] haben wir einfach mal gezählt. Insgesamt sind die Gebühren etwa 2800 mal angestiegen und etwa 3200 mal gesunken. Das heißt mit einer Prognose von immer „sinkend“ wäre man in $\frac{3200}{6000} \approx 0,533$, also etwa 53% der Fälle richtig. Dieser Wert ist also die Mindestquote, die es zu erreichen gilt für unseren Schätzer.

4.2.2 Anzahl von Trendwechseln

Als Trend definieren wir erneut die entweder positive oder negative Änderung der Gebühren im Vergleich zum Vorgänger Block. Im betrachteten Blockintervall von [650000;656000] gibt es insgesamt 3916 Trendwechsel auf insgesamt 6000 Übergänge. Das bedeutet in $\frac{3916}{6000} \approx 0,653$, also etwa in 65,3 % der Fälle ändert sich der Trend des Blockes. Demnach könnte man einfach immer das Gegenteil des Blockes prognostizieren und man läge in 65,3 % der Fälle richtig. Unsere neue Mindestquote für den Schätzer ist also dieser neue Wert.

4.3 erste Schritte zur Entwicklung eines eigenen Models für Prognose der Transaktionsgebühren

4.3.1 Kolmogorov-Smirnov-Distanz

Mit den Ergebnissen aus Unterabschnitt 4.1 sind wir nur in der Lage eine einfache Prognose unter Berücksichtigung nur dieser einzelnen Eigenschaft zu machen.

Wir nehmen den Median des Vergangenen Blocks und prüfen ob dieser unter oder über einem kritischen Schwellenwert liegt. Den Schwellenwert bestimmen wir wie folgt: Mithilfe eines Kolmogorow-

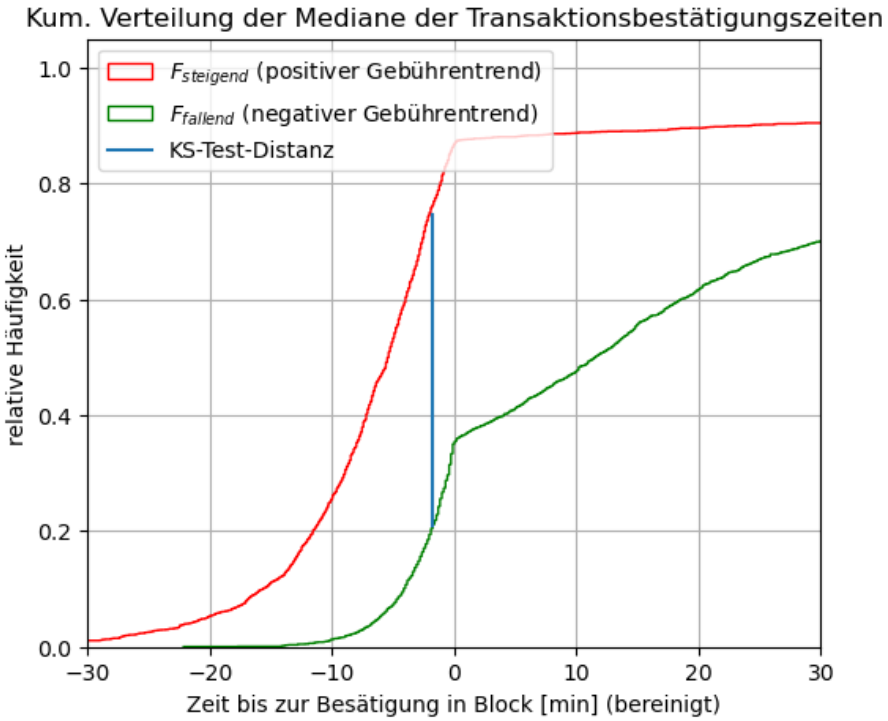

Abbildung 11: Kumulierte Verteilung der Mediane der Transaktionsbestätigungszeiten mit k = 0

Smirnow-Test bestimmen wir ob sich die Verteilung der Mediane für steigende und fallende Blöcke unterscheiden oder ob sie zur gleichen Wahrscheinlichkeitsverteilung gehören. Dabei berechenen wir den x-Wert bei denen die Graphen der kummulierten Verteilung die höchste Differenz haben, sprich am weitesten

entfernt sind.

$$
\begin{aligned}
d_{max} &= max|F_{fallend} - F_{steigend}| \\
H_0 &: \textit{Die beiden Verteilungen gehören zur gleichen Variable} \\
\alpha &= 0.05 \\
d_{max} &\approx 0,56 \quad x \approx -1,79
\end{aligned}
$$

Dieser Punkt liegt bei $x \approx -1,79$ vor und ist demnach unser Schwellenwert k. Wir finden außerdem heraus, dass der Test einen p-Wert von $1,316 \times 10^{-321}$ annimmt. Demnach können wir die Nullhypothese H_0 selbst bei einem Signifikanzniveau von $\alpha = 0.05$ ablehnen und stellen fest, dass es sich um zwei klar verschiedene Verteilungen handelt.

4.3.2 Test der Hyptotehse im Blockintervall [650000;656000]

Basierend auf dieser Eigenschaft haben wir begonnen einen Prototyp eines eigenen Schätzer zu bauen und zu testen. Unter der Berücksichtigung nur dieser einzelnen Eigenschaften und einer Anpassung des Schwellenwertes für fallende Transaktionen lässt sich bereits eine Trefferquote von mehr als 60% erzielen. In der Zukunft, wenn die verschiedenen beobachteten Eigenschaften kombiniert werden, hoffen wir, dass sich diese Quote wesentlich erhöhen lassen wird.

5 Ergebnisdiskussion und Ausblick

5.1 Ergebnisdiskussion

Rein mit den Daten der Bitcoin-Blockchain und den Mempooldaten von C. Decker waren wir in der Lage verschiedene Eigenschaften der Gebpührenentwicklungen herauszuarbeiten. Allerdings sind diese teilweise kritisch einzuordnen.

5.1.1 Einfluss der Varianz auf die Blockerstellungszeiten

Wir haben zwar in Unterunterabschnitt 3.2.1 versucht den systematischen Fehler der variierenden Blockerstellungszeiten zu eliminieren, dennoch ist ein Einfluss dieser Varianz noch im Datensatz enthalten. In

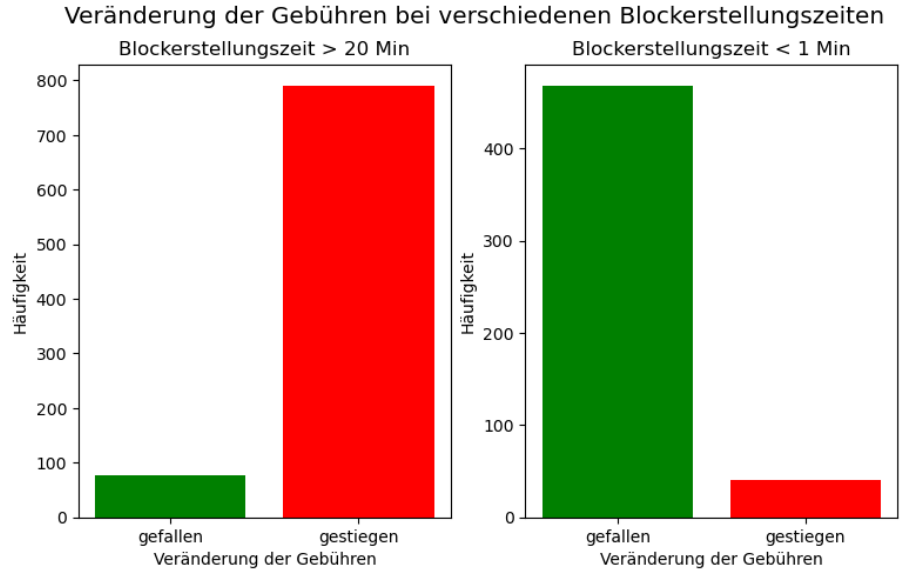

Abbildung 12: Veränderung der Gebühren bei extremen Blockerstellungszeiten

Abbildung 12 lässt sich der bestehende Einfluss erkennen. Wenn zufällig die Blockerstellungszeiten sehr hoch sind, dann wurden die hohen Transaktionen einer hohen Gebührenrate lang nicht abgebaut. Dies resultiert dann in einem temporär sehr starken Anstieg Transaktionsgebühren, obwohl sich in der eigentlichen Nachfrage oder in den Blöcken davor nichts verändert hat. Ähnlich verfährt es mit sehr schnell erstellten Blöcken. Dies hat dann zur Folge, dass die Gebühren kurzfristig sinken obwohl sich die Umstände eigentlich nicht geändert haben.

5.1.2 Reduzierung auf Blockintervall

In dieser Arbeit haben wir das Blockintervall [650000;700000] und vor allem das Subintervall [650000;656000] untersucht. Initial hatten wir diesen Zeitpunkt relativ arbiträr gewählt, um die zu untersuchenden Daten einzugrenzen. Allerdings war die Forschung sehr zeitintensiv und konnten das Intervall bisher noch nicht ausweiten. Unsere erhobenen Daten sollten allerdings durch die Größe von 6000 Blöcken dennoch ziemlich repräsentativ sein, aber zur Prüfung werden wir unseren Intervall zukünftig ausweiten.

5.1.3 Child-Pays-For-Parent[5]

Darüber hinaus wurde in dieser Auswertung nicht das Phänomen Child-Pays-for-Parent (CPFP) berücksichtigt. CPFP bedeutet, dass man einen Output einer Transaktion, die sich noch im Mempool befindet, in einer Transaktion mit sehr hohen Gebühren ausgibt. In diesem Sinne bezahlt das Kind (Child) die Gebühren für die Elterntransaktion (Parent) mit. Durch diesen Aufbau werden Miner incentiviert die Kind- und die Elterntransaktion mit in den Block aufzunehmen. Die Folge ist, dass es Transaktion mit einer unter Umständen extrem langen Transaktionsbesätigungszeit in den Block schaffen und somit den Median beziehungsweise den Durchschnitt dieser Zeiten verzerren können.

5.2 Ausblick

Das Ziel dieser Jugend-Forscht-Arbeit ist das Entwickeln eines eigenen Schätzers für zukünftige Transaktionsgebühren im Bitcoin-Netzwerk. Mit den gewonnenen Erkenntnissen werden wir nun versuchen einen eigenen Schätzer zu entwickeln und zusätzlich weitere Daten einbinden. Beispielsweise die aktuellen Mempool-Auslastung und -Daten oder der Bitcoin-Kurs sind mögliche Einflüsse auf die Gebühren, die es bei der Prognose der Gebühren zu berücksichtigen gilt.

Gleichzeitig werden wir unserer Analyse auf weiter Eigenschaften ausweiten und selbstverständlich den untersuchten Blockintervall vergrößern.

6 Danksagungen

Ein besonderes Dankeschön möchte ich an René Pickhardt aussprechen. Bereits während der Deutschen Schüler Akademie konnte ich sehr viel von ihm lernen und er inspirierte mich dazu, in die Welt des Bitcoins einzutauchen. Während der Erarbeitung dieser Forschungs-Arbeit stand er mir stets[9] mit Rat und Tat mit fachlicher und technischer Expertise zur Seite und unterstütze mich an jeglicher Front. Mir hat die Erarbeitung dieses Projektes zusammen mit ihm unglaublichen Spaß bereitet und ich blicke voller Euphorie auf die Vollendung dieses Projektes zu einem eigenen Schätzer.
Außerdem bedanke ich mich bei Christian Decker für die Zuverfügungsstellung des Mempool-Datensatzes seiner Bitcoin-Node und allen anderen Personen, die mich bei der Erarbeitung untersützt haben.

[9] wirklich zu jeder Uhrzeit :)

7 Quellen, Literaturverzeichnis und verwendete Programme

[1] Coinbase. *Bitcoin Halvings*. besucht am 18. Januar 2024. 2024. URL: https://www.coinbase.com/de/learn/crypto-basics/what-is-a-bitcoin-halving.

[2] Coinbase. *Marktkapitalisierung Bitcoin*. besucht am 18. Januar 2024. 2024. URL: https://www.coinbase.com/de/price/bitcoin/usd#:~:text=Die%20aktuelle%20Marktkapitalisierung%20von%20Bitcoin,Markt%20ein%20Asset%20hoch%20bewertet..

[3] Alex Morcos. *High level description Bitcoin Core's fee estimation algorithm*. besucht am 18. Januar 2024. 2023. URL: https://gist.github.com/morcos/d3637f015bc4e607e1fd10d8351e9f41.

[4] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: (2008).

[5] Bitcoin Optech. *Child Pays for Parent*. besucht am 18. Januar 2024. 2023. URL: https://bitcoinops.org/en/topics/cpfp/.

[6] River. *Virtuel Byte*. besucht am 18. Januar 2024. URL: https://river.com/learn/terms/v/vbyte/.

[7] Rusty Russell. *Bitcoin-Iterate*. besucht am 18. Januar 2024. 2015. URL: https://github.com/rustyrussell/bitcoin-iterate.

[8] Oscar Stach. *Jugend Forscht Fee Estimator*. besucht am 18. Januar 2024. URL: https://github.com/oscars181/Jugend-Forscht-Fee-Estimator.

[9] Tagesschau. *US-Börsenaufsicht genehmigt Bitcoin-Fonds*. besucht am 18. Januar 2024. 2024. URL: https://www.tagesschau.de/wirtschaft/weltwirtschaft/bitcoin-etf-sec-100.html.

[10] Wikiepedia. *Merkle Tree*. besucht am 18. Januar 2024. 2024. URL: https://de.wikipedia.org/wiki/Hash-Baum.

[11] Wikipedia. *SHA-2*. besucht am 18. Januar 2024. 2024. URL: https://de.wikipedia.org/wiki/SHA-2.

A Anhang

A.1 Eigener Parser Algorithmus

Eine schematische Beschreibung des anfänglichen Parser-Algorithmus für die durchschnittlichen Transaktionsgebührenrate pro Block:

1. *getblockhash block_height*
   Mit diesem Befehl lässt sich der Hash („Blockname ") von der API holen. Gleichzeitig berechnen wir basierend auf der Blockhöhe die Blocksubsidy für den Miner.

2. *getblock block_hash 2*
   Anschließend verarbeiten wir den Block Hash aus (1) weiter und holen uns die JSON-Datei vom Block in Verbosity-Level 2. Dies bietet uns die Möglichkeit

3. $\frac{Summe_{Outputs} - Blocksubsidy}{Gewicht_{Transaktionen}} = durchschnittliche\ Gebühr\ pro\ Block\ [Fee/VByte]$
   Nun können wir in einer Schleife die Outputs der Coinbase-Transaktion, die auszahlende Transaktion an den Miner, addieren und den gesamten Ouput dieser Transaktion berechnen. Die Differenz der Gesamtauszahlung und der Blocksubisdy sind die Gesamtgebühren des Blocks. Der Quotient dessen und des Gewichtes der Transaktionen aus (2) bildet nun die durchschnittlich Gebühr für diesen Block.

4. *Speicherung der gewonnenen Daten in einer CSV-Datei*
   Abschließend speichern wir die extrahierten Daten in einer CSV-Datei. Damit gewährleisten wir eine effiziente und skalierbare Weiterverarbeitung der Daten für beispielsweise Pandas-Dataframes.

A.2 Kumulierte Verteilung der Transaktionsbestätigungszeiten von Block 653932 bis 654112

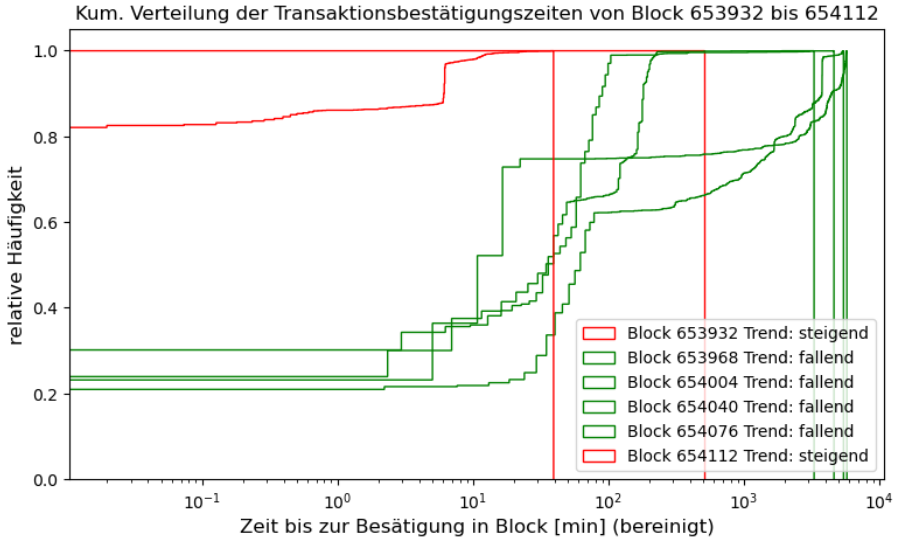

Abbildung 13: Kumulierte Verteilung der Transaktionsbestätigungszeiten von Block 653932 bis 654112