UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eduardo Augusto de Medeiros Silva

Um método não paramétrico para identificação de anomalias na mineração do Bitcoin

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eduardo Augusto de Medeiros Silva

Um método não paramétrico para identificação de anomalias na mineração do Bitcoin

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Ivan da Silva Sendin

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

Eduardo Augusto de Medeiros Silva

Um método não paramétrico para identificação de anomalias na mineração do Bitcoin

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 24 de novembro de 2023:

Prof. Dr. Ivan da Silva Sendin
Orientador

Prof. Dr. Rodrigo Sanches Miani

Prof. Dr. Marcelo Zanchetta do Nascimento

Uberlândia, Brasil
2023

Dedico esse trabalho aos meus amigos e familiares, que estiveram comigos durante todos os contratempos nesses últimos anos e me incentivaram a continuar com empenho. Aos meus colegas de turma, por todo o companheirismo e momentos de felicidade e aprendizado compartilhados durante esse percurso.

Resumo

O objetivo deste trabalho é apresentar um método estatístico não-paramétrico que pode ser utilizado para identificar rastros deixados na Blockchain por mineradores egoístas. Inicialmente foi realizada uma revisão bibliográfica na literatura visando captar as publicações mais atuais acerca de estratégias para identificação de tal atividade. Em seguida, observando as limitações da literatura existente, observou-se a necessidade de elaboração de um método não paramétrico de modo a evitar a inferência sobre a amostra gerando resultados tendenciosos. O período de maio de 2021 até maio de 2022 foi selecionado, totalizando 12 meses de avaliação do total de blocos. Para analise e manipulação dos dados utilizou-se a linguagem de programação Python, a API da Blockchain e o software de banco de dados MongoDB. Com a utilização do algorítmo, alcançou-se o resultado de 18 mineradores suspeitos de praticarem a mineração egoísta. Apesar dos resultados obtidos serem comparáveis com os resultados presentes na literatura, acreditamos que a abordagem utilizada e o uso exclusivo de dados provenientes da Blockchain não sejam suficientes para identificar de forma segura os praticantes de mineração egoista.

Palavras-chave: Bitcoin, Blockchain, Mineração.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

SM	Selfish Mining (Mineração Egoísta)
P2P	Peer-to-peer (Par a par, ou ponto a ponto)
HTTP	Hypertext Transfer Protocol (Protocolo de comunicação utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos)
POW	Proof of Work (Prova de trabalho, protocolo utilizado pela Blockchain)
BaaS	Blockchain as a Service (Blockchain como serviço)

Sumário

1 Introdução

Posteriormente ao surgimento da Internet, o comércio intra rede era executado por instituições tradicionais baseadas na confiança em terceiros para um sistema de pagamentos eletrônicos. Como citado em Nakamoto (2009), era necessário a criação de um sistema de pagamentos baseado em criptografia, permitindo que duas pessoas executassem transações diretamente entre si. A tecnologia empregada nas transações objetiva a proteção dos usuários contra fraudes e a manutenção da integridade da rede.

No Bitcoin, bem como na maioria das criptomoedas existentes, os mineradores, como são chamados os usuários da rede, são responsáveis por manter a corretude da Blockchain. Ao validarem as transações e executarem a prova de trabalho, o Consenso de Sakamoto é obtido e a Blockchain pode ser considerada segura, ou seja, o seu conteúdo está correto e blocos anteriores não foram alterados.

A raiz da segurança está embasada em uma abordagem baseada na Teoria de Jogos onde a distribuição do poder computacional entre os participantes visa evitar que nenhuma parte isolada consiga fraudar o sistema. Como pagamento por este trabalho os mineradores recebem criptomoedas. Sendo um protocolo probabilístico, a longo prazo, cada minerador deve receber uma quantidade de moedas proporcional ao seu poder computacional.

Pouco após a popularização do uso do Bitcoin, Eyal e Sirer constataram que um desvio do comportamento proposto poderia trazer benefícios ao minerador (EYAL; SIRER, 2014) e eventualmente comprometer a segurança da Blockchain. Posteriormente, vários trabalhos que analisaram esta ação propuseram alterações no Bitcoin, em vista da aparição da estratégia de mineração egoísta (HEILMAN, 2020). Os usuários ou grupo de usuários que utilizam essa estratégia escolhem não divulgar um bloco descoberto, com o objetivo de conseguir uma vantagem de tempo na mineração do próximo bloco. Porém, apenas em Li, Yang e Tessone (2020a), Li, Yang e Tessone (2020b) foi apresentado um método estatístico que usa somente dados da Blockchain para detectar o comportamento egoísta de mineradores, esse baseia-se na avaliação do Z-Score e assume uma Distribuição Gaussiana dos dados.

Neste trabalho propomos uma alteração sobre o método de detecção de Mineração Egoista citado. Esta tem como principal vantagem ser independente da distribuição dos dados. Apresentamos também uma discussão sobre a viabilidade deste tipo de abordagem e propomos técnicas alternativas que podem ser usadas para este fim.

1.1 Objetivos

O presente trabalho tem como objetivo demonstrar um método de identificação da incidência do ataque de mineração egoísta, utilizando uma abordagem não paramétrica em relação aos dados presentes na blockchain do Bitcoin. Com a realização da contagem de pares de minerações consecutivas executadas por mineradores e embaralhamento da amostra original, pretende-se mostrar que o resultado obtido pelo minerador não é fruto do acaso.

Com base nisso, será possível identificar possíveis suspeitos de realização do ataque, sem que exista pressuposições sobre a amostra utilizada.

1.2 Justificativa

A literatura cientifica mais recente ilustra diversas anomalias durante o processo de mineração de criptomoedas. Entretanto, quando observamos o contexto recente de identificação de mineração egoísta no Bitcoin, percebe-se uma escassez de analises não paramétricas com essa finalidade. Desse modo, visando tornar mais robustas as evidências científicas atuais e contribuir para toda a comunidade de segurança da informação e criptomoedas, justifica-se a execução do presente estudo.

1.3 Organização do Trabalho

O presente trabalho está organizado em cinco capítulos que abordam de forma estruturada e separa os elementos fundamentais do trabalho realizado. No capítulo 2, Referencial Teórico, são apresentados os conceitos básicos relacionados ao Bitcoin, a tecnologia e a arquitetura, finalizando com uma seção sobre trabalhos correlatos que já abordaram a mineração egoísta. No seguinte capítulo, entitulado desenvolvimento, são descritos os procedimentos adotados na pesquisa, que abrangem uma abordagem não paramétrica para identificar a incidência da utilização da mineração egoísta. São apresentados os detalhes sobre a amostra, obtenção e organização dos dados, pseudo-código do algoritmo que foi utilizado nas análises. Em seguida, no quarto capítulo, são apresentados os resultados obtidos, gráficos utilizados para caracterização da amostra e tabelas para identificação de mineradores, terminando com uma tabela unindo informações completas sobre os mineradores suspeitos. Por fim, o último capítulo conclui o trabalho, destacando e considerando os resultados e além disso, são fornecidas recomendações para a continuidade do trabalho, com sugestões e possíveis direções para pesquisa e desenvolvimento futuro na área.

2 Referencial Teórico

Este Capítulo descreve conceitos essenciais para compreensão do trabalho: Blockchain, Bitcoin e a importância destas novas tecnologias. As duas últimas seções do Capítulo abordam o aparecimento da Mineração Egoísta, seu funcionamento, sua identificação e variações existentes.

2.1 Blockchain

Atualmente toda atividade que um usuário executa em uma rede gera informações, essas, por consequência, são armazenadas pelos donos da rede utilizada. Sites de compras online retem informações dos compradores, redes sociais possuem informações relacionadas aos interesses dos usuários, e há também sites que arquivam dados ainda mais sensíveis, como por exemplo, documentos e número de contato de quem utiliza seus serviços. Em todos esses casos, existe uma organização centralizadora dos dados, que detém o poder de utilização dos mesmos.

A suposição de confiabilidade dessas organizações que centralizam as informações se mostra perigosa, pois uma falta comprometimento em estabelecer medidas contra ataques põem em risco a segurança dos dados armazenados.

Visando corrigir esse problema, uma nova abordagem foi pensada usando uma arquitetura de base de dados distribuída chamada blockchain[1]. A eliminação da autoridade central de uma base de dados era um aspecto importante para essa abordagem, sendo um ponto favorável para essa arquitetura proposta. Apesar do conceito de blockchain ter sido popularizado em 2008 com o aparecimento da criptomoeda Bitcoin, a tecnologia já existia a mais de 10 anos como mostrado em Haber e Stornetta (1990), onde foi apresentada como uma cadeia de *hashes* para impedir modificações não autorizadas em documentos.

Partindo da definição descrita em Alves et al. (2018), "Blockchain é uma tecnologia que faz uso de uma arquitetura distribuída e descentralizada para registrar transações de maneira que um registro não possa ser alterado retroativamente, tornando este registro imutável". Como o nome sugere, é uma cadeia de blocos conectados por meio de uma função de *hashing* criptográfica (SENDIN, 1999), onde o bloco inicial dessa rede é chamado de Bloco Gênesis, a partir daí, o valor do *hash* que define cada bloco é calculado por uma função de *hashing* previamente definida (por exemplo, SHA-256) utilizando o conteúdo do próprio bloco e o *hash* do bloco anterior.

[1] Neste texto, adotamos a convenção de usar o termo blockchain, em minúsculo, para a estrutura de dados e Blockchain, com inicial maiúscula para a tecnologia específica usada pelo Bitcoin.

Na Figura 1 é mostrado um exemplo de uma blockchain com 3 blocos. O encadeamento provido pelo uso da função de *hashing* traz a imutabilidade dos dados dos blocos anteriores, isto é, o bloco n atesta a integridade do blocos $n-1$ e, indutivamente, até o bloco gênesis. Por exemplo, se o conteúdo do bloco 2 for alterado, o valor do hash do próprio bloco será alterado, pelas propriedades das funções de *hashing* criptográficas, e assim detectado por qualquer pessoa que tenha acesso ao bloco 3, que contém o valor de *hash* do bloco 2.

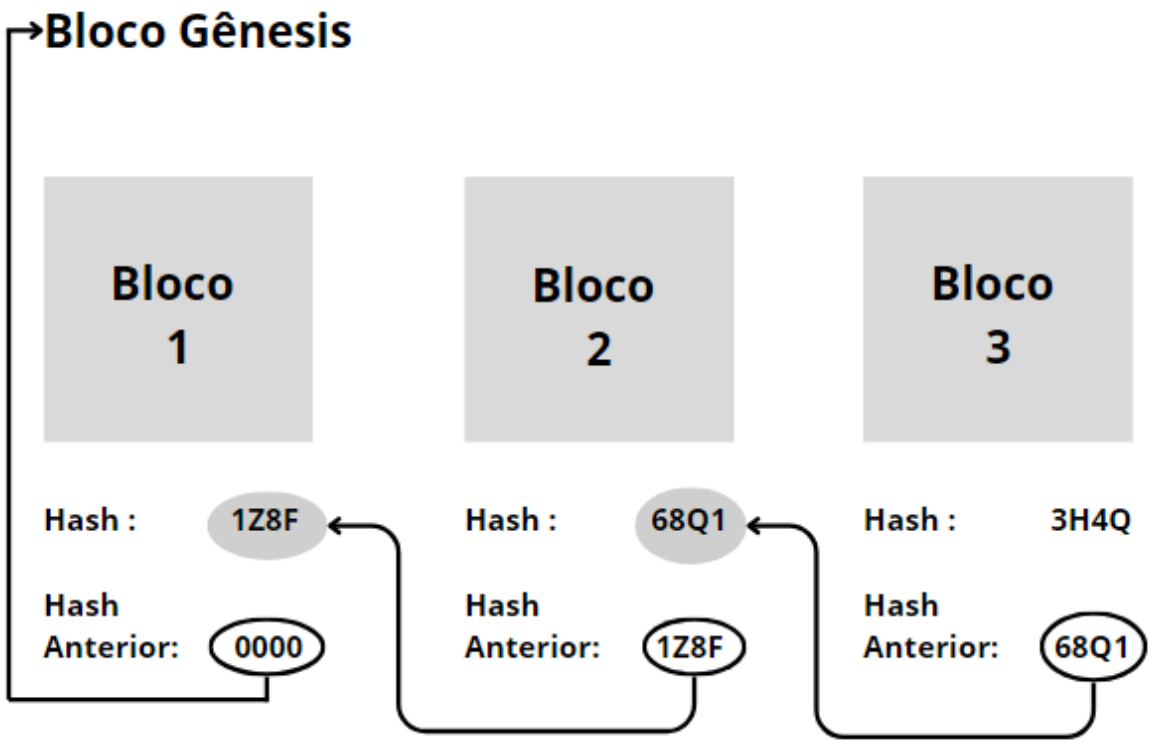

Figura 1 – Representação dos blocos dentro da Blockchain do Bitcoin, a partir do Gênesis, todo bloco subsequente contêm o *hash* do anterior considerado na geração do próprio *hash*.
Fonte: Elaborado pelo autor (2023).

2.2 O Bitcoin

O Bitcoin, apresentado ao mundo por Satoshi Nakamoto em Nakamoto (2009) pode ser visto como uma coleção de conceitos e tecnologias que formam a base do ecossistema monetário digital descentralizado. As unidades da moeda, também chamadas de bitcoins[2], são transferidas ou estocadas pelos usuários usando o protocolo próprio. O Bitcoin faz uso de uma rede P2P própria e aberta que roda em diversos tipos de aparelhos de computação, tornando a tecnologia extremamente acessível e resiliente. Uma particularidade do Bitcoin é que o seu autor não existe e o nome utilizado provavelmente é um

[2] A palavra Bitcoin, com a inicial maiúscula, é usada para se referir à tecnologia como um todo, e bitcoin, com a inicial em minúsculo, é usada para descrever a unidade monetária.

pseudônimo adotado por um grupo de pessoas. Além disso, o seu artigo seminal é sucinto e não esclarece diversas quastões técnicas, que foram discutidas em `maillist` e hoje estão concentradas no site <https://www.bitcoin.org>.

Assim como qualquer moeda convencional, os usuários podem utilizá-la para compra e venda, doações e investimentos, também podendo ser convertida em outras moedas seguindo uma taxa de conversão que depende do mercado financeiro. Possuindo potencial ilimitado, o Bitcoin pode ser considerado uma moeda muito viável para a Internet, por ser rápida e segura (ANTONOPOULOS, 2017).

Os usuários ou clientes dessa criptomoeda possuem endereços que são utilizados para realizar transações entre si. Os endereços do Bitcoin são essencialmente chaves públicas, cuja respectiva chave privada é usada para assinar as tranferências de Bitcoins, chamadas de transações. Essas transações, por sua vez, são salvas na Blockchain, que serve como um livro-razão público. Este, para a contabilidade, é um instrumento utilizado para anotar de maneira organizada, transações ou movimentações econômicas, de uma pessoa ou empresa, tal instrumento permite um controle individualizado de cada transação realizada e pela forma como que foi criado, impede a remoção de páginas.

A Blockchain é mantida pelos próprios usuários que executam tarefas como verificação das transações emitidas buscando consistência financeira e validade das assinaturas digitas, bem como a criação de novos blocos. Os usuários que optam por realizar estas tarefas são chamados de mineradores e são remunerados pelo próprio sistema do Bitcoin. Detalhes sobre o funcionamento do Bitcoin serão descritos nas próximas seções e o funcionamento geral do sistema de cripmoedas pode ser obtido em Antonopoulos (2017), Nakamoto (2009).

2.3 Transações

De modo geral, uma transação informa para a rede do Bitcoin que houve uma transferência de valores da moeda entre usuários. O antigo proprietário dos bitcoins os transfere para o novo, que pode gerar uma nova transação com as unidades recebidas, criando assim uma cadeia de transações dentro da rede (ANTONOPOULOS, 2017).

Como já dito anteriormente, a Blockchain pode ser vista como um livro-razão público, logo, podemos imaginar as transações como linhas dessa estrutura contábil. De um lado estão as entradas, podendo ser uma ou várias, que atuam como os débitos de uma transação, do outro lado estão as saídas, que funcionam como os créditos. Apesar de ser intuitivo, a soma dos débitos e créditos não são necessariamente iguais ao final, a diferença entre esses valores é chamada de taxa de transação (ANTONOPOULOS, 2017).

Dentro da cadeia de transações, a saída da última transação cria um valor em

bitcoins que só pode ser desbloqueado pela chave privada do dono. Assim que validado que o usuário é realmente o dono, essa saída poderá se tornar a entrada de uma próxima transação e assim por diante. Se assemelhando a vendas feitas com dinheiro em espécie, a entrada de uma transação não pode ser "dividida". Caso você queira comprar algo que custe cinco bitcoins, mas só possui uma entrada possível de 20 bitcoins, devido a esse ser o valor de saída de uma transação anterior, a nova transação terá uma saída referente ao pagamento de cinco bitcoins e outra referente ao "troco". Essa cadeia de transações está também explicada na Figura 2.

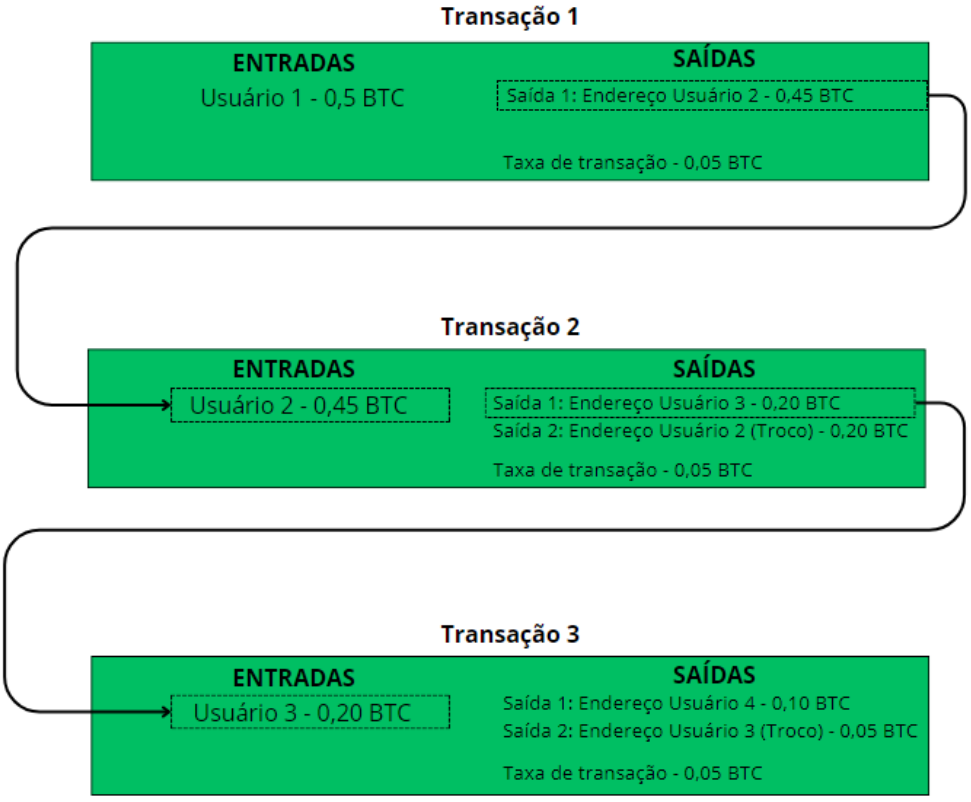

Figura 2 – Cadeia de transações, onde a saída de uma transação é a entrada da próxima.
Fonte: Elaborado pelo autor (2023).

Existe uma transação especial dentro do Bitcoin chamada de transação *coinbase*, é nela que a recompensa pela criação de um novo bloco é colocada. Uma de suas características é a falta de entrada, ou seja, não há "gasto" de moedas. Outra característica também importante é o *coinbase script*, uma mensagem que o dono do bloco pode inserir na transação, sendo muito utilizado para mensagens pessoais, assinaturas de identificação, comemoração de eventos e até referências a notícias atuais ao período de mineração do bloco.

2.4 Mineração e Consenso

Seguindo o caminho oposto dos bancos tradicionais, o Bitcoin, baseia-se em uma confiança descentralizada, ou seja, não existe uma autoridade que valida as transações que ocorrem nessa rede (ANTONOPOULOS, 2017). Em vez disso, essa validação é conquistada através de um consenso entre os participantes da Blockchain. As transações que ocorrem entre usuários são registradas em unidades de blocos, o detalhamento do bloco é mostrado na Figura 3.

Em Nakamoto (2009) são mostrados detalhes técnicos da rede e dos blocos, dentre as informações mais relevantes, o bloco possui:

***Hash* identificador** : Valor identificador do bloco, resultante de uma função *hash*.

***Hash* anterior** : Valor hash do bloco anterior.

Lista de transações : Lista contendo transações entre usuários registradas no bloco, o usuário criador mesmo recebe uma taxa para cara transação presente.

Hash* combinado de uma Árvore de Merkle ou *Merkle Root : Um valor *hash* resultado da combinação dos hashes de todas as transações registradas no bloco. A Árvore de Merkle (MERKLE, 1989), também chamada de árvore binária de *hashes* é uma estrutura de dados utilizada para sumarização e verificação de integridade eficiente em grandes quantidades de dados.


```
{
    "hash": "00000000000000010aa40340352cd5a62ad90e3946d3c6115d24105a27046c1",
    "ver": 2,
    "prev_block": "000000000000000014623ad94873ca2cc3faa7a8704b4e6eb06dd259d2ef3f63",
    "mrkl_root": "5f780c498b49934f27d54235617ce8b03394225403236987cfda24890f1dc863",
    "time": 1424704810,
    "bits": 404196666,
    "next_block": [
        "00000000000000000d799d7618f3350ef61ce3cb0f8fe2ee858f60a66c24106"
    ],
    "fee": 0,
    "nonce": 3163225836,
    "n_tx": 1,
    "size": 254,
    "block_index": 344822,
    "main_chain": true,
    "height": 344822,
    "weight": 1016,
    "tx": [···
    ]
}
```


Figura 3 – Informações de um bloco real dentro da Blockchain do Bitcoin obtido pela API da blockchain.com .
Fonte: Elaborado pelo autor (2023).

Esse processo de criação de um novo bloco na rede Blockchain é chamado de mineração. Ao minerar um novo bloco, o minerador publica a informação desse para os demais mineradores, para que eles possam trabalhar na mineração de outro bloco levando em consideração o endereço do "novo" bloco anterior. Em caso de mineração concorrente, ou seja, dois blocos são criados com o mesmo bloco anterior, a Blockchain é dividida em dois "galhos". Porém, o protocolo da rede se certifica de que os mineradores irão criar novos blocos a partir do galho com o maior número de blocos, ou em caso de ambos terem o mesmo tamanho, do galho que foi informado primeiramente pelo minerador.

Além da recompensa por criar um novo bloco, o minerador também recebe taxas para cada transação armazenada no bloco criado por ele. O envio da recompensa é registrado na primeira transação do bloco criado, a *coinbase transaction*, já explicada na seção anterior.

A validação das transações e da entrega de criptomoedas é executada pelos próprios usuários da rede, chamados de *full nodes*, eles possuem um software de cliente Bitcoin e uma cópia completa da Blockchain em um computador ou servidor. Geralmente possuindo grande poder de processamento e de armazenamento, realizam uma tarefa difícil, porém essencial na sustentação da rede.

O objetivo dos usuários é estabelecer um consenso global, e o mecanismo utilizado para tal é chamado de Prova de Trabalho, comumente citado utilizando a abreviação `PoW`, referenciando o termo em inglês *Proof of Work*. A prova de trabalho é um desafio computacional complexo dado aos mineradores durante a mineração, onde o mesmo tenta encontrar um valor nomeado como *nonce* que combinado à algumas informações do bloco, gere um *hash* que comece com um número específico de zeros à esquerda. Os mineradores modificam o valor do *nonce* e calculam o *hash* até encontrarem um resultado válido para a rede.

Em média, um novo bloco é minerado a cada dez minutos, isso se mantêm desde a criação do Bitcoin em 2009, porém, era previsto pelo seu criador que o poder computacional dos mineradores iria aumentar com o passar do tempo. Para contornar esse aumento e manter a taxa mineração dos blocos constante, a Blockchain ajusta a dificuldade de mineração a cada 2016 blocos, ou aproximadamente 1 semana de blocos minerados. Esse ajuste é feito automaticamente e de forma independente por cada *node* (usuários que podem possuir apenas uma parte da Blockchain), utilizando o tempo calculado para mineração dos últimos 2016 blocos e o valor esperado de 20160 minutos (valor alcançado se os blocos seguirem o tempo médio de mineração). Assim que o cálculo é terminado, a dificuldade do desafio da Prova de Trabalho é ajustado de acordo com o resultado, fica mais fácil se o tempo médio ficou maior que o esperado e mais difícil se o tempo ficou menor.

Como dito por Antonopoulos em Antonopoulos (2019), "o criador da Blockchain

original, Bitcoin, inventou o algoritmo conhecido como *Proof of Work*, e indiscutivelmente, o *PoW* é a invenção mais importante que sustenta o Bitcoin". Coloquialmente chamado de mineração, o objetivo desse algoritmo de consenso, diferente do conceito literal da palavra, não é criar uma nova moeda ou câmbio por meio de metais preciosos ou outros recursos, e sim manter a segurança da Blockchain sem perder a característica de um sistema descentralizado e difundido ao maior número de participantes (ANTONOPOULOS, 2017).

Utilizando a recompensa como um meio e a manutenção da segurança como objetivo, os mineradores são incentivados a participar da mineração de forma honesta, ao tentar utilizar fugir das regras durante esse processo, o minerador estará arriscando o recurso em eletricidade utilizado até então, forçando a maioria a ter um comportamento honesto. Ao longo do trabalho os termos "minerador" e "*pools* de mineradores" serão utilizados para representar quem de fato minera o novo bloco na Blockchain, na maioria dos casos estarão se referindo ao grupo de usuários que unem seu poder computacional para obterem maiores chances de mineração.

2.5 Aparição do Selfish Mining

Por muito tempo, acreditou-se que o protocolo da prova de trabalho do Bitcoin fosse compatível ao incentivo, isto é, seguir o protocolo é a melhor estratégia para maximizar os ganhos. Porém, um ataque documentado ao sistema de mineração chamado de mineração egoísta (SM, do termo em inglês *Selfish Mining*), mostrou que se desviar do protocolo pode gerar benefício maior que esperado dado o seu poder computacional, compromentendo a segurança da Blockchain.

De forma simplificada, caso um minerador consiga obter a prova de trabalho para um bloco de altura n, ao invés de divulgá-lo para rede P2P e todos os mineradores iniciarem a busca pelo bloco na altura $n+1$, este minerador egoista não divulga o bloco minerado e fica trabalhando na altura $n+1$ enquanto os demais mineradores ainda trabalham na altura n, ganhando uma vantagem na busca pelo bloco $n+1$. Uma consequencia direta da mineração egoísta é que o minerador que a pratica produz blocos em sequência, e esta informação pode, em geral, ser identificada na transação *coinbase*, onde o minerador recebe o seu pagamento.

Desde a sua aparição, a mineração egoísta chamou atenção da comunidade e alguns outros ataque ao mecanismo de consenso surgiram, tais como, o ataque Sybil (DOUCEUR, 2002) e o ataque de eclipse (HEILMAN et al., 2015). Entretanto, surgiram também propostas para aumentar a robustes da Blockchain para combater ataques deste tipo (SAAD et al., 2020; HEILMAN, 2020).

2.6 Testes Paramétricos x Não Paramétricos

Durante a execução de testes estatísticos, a escolha entre um teste paramétrico e um não paramétrico leva em conta a natureza da distribuição de dados (CAMPBELL; SWINSCOW, 2009). Os testes paraméticos exigem que seja pressuposto a distribuição dos dados da amostra, na maioria dos casos, a distribuição normal. Já os não paramétricos, não exigem a pressuposição de distribuição.

Como vantagem, caso a amostra siga uma distribuição normal, apesar de poder ser usado o teste não paramétrico, o mesmo apresenta um menor poder estatístico em relação ao paramétrico.

A escolha do teste utilizado no trabalho objetivou contrapor a literatura já existente sobre o assunto. A hipótese levantada durante o desenvolvimento não pressupõe que a amostra coletada da Blockchain siga uma distribuição normal, ou seja, um teste não paramétrico traria resultados mais robustos e com maior poder estatístico.

2.7 Trabalhos correlatos

Identificar comportamento dos mineradores egoístas é uma tarefa complexa: a estratégia de mineração egoísta é executada internamente ao minerador desonesto e pouca informação é deixada na Blockchain. Recentemente em Li, Yang e Tessone (2020a), Li, Yang e Tessone (2020b) é apresentado um método estatístico paramétrico aplicado à Blockchain que detecta blocos que foram obtidos por mineradores desonestos. Utilizando uma abordagem parecida com a que será mostrada nesse trabalho, o método também se utiliza da ocorrência de minerações consecutivas e do embaralhamento da amostra, gerando um valor final que leva em conta também o desvio padrão da amostra original. A conclusão desse trabalho mostra que a distribuição da recompensa pela mineração na criptomoeda Ethereum é desigual, acreditando que grupos de mineradores se utilizam da estratégia de mineração egoísta para obter sucesso ao executar minerações consecutivas.

Ainda em Li, Yang e Tessone (2020a), Li, Yang e Tessone (2020b) é mostrado que *pools* de mineradores da criptomoeda Bitcoin com menos de 25% de poder computacional não possuem incentivo para executar a estratégia de mineração egoísta, podendo até, aumentar o risco da mineração sem aumentar a recompensa correspondente.

Já em Negy, Rizun e Sirer (2020) é apresentada uma nova abordagem a fim de maximizar os ganhos do minerador egoísta. O trabalho expõe os principais problemas e impedimentos da estratégia original, tais como, a continuidade de aplicação da estratégia após a execução do ajuste de dificuldade da rede. Apresentando a `Mineração Egoísta Intermitente` como uma variante que contorna esse problema. Simplificadamente, o minerador egoísta voltaria a minerar honestamente logo após o ajuste de dificuldade. Por

fim, o artigo traz algumas informações sobre o poder computacional mínimo necessário para a execução rentável da nova variante da estratégia de mineração egoísta, 33% é apresentado como um valor razoável ao levar em consideração algumas métricas demonstradas nas análises, por exemplo, a proporção de mineradores honestos atuando em um bloco criado por mineração egoísta.

Na parte de detecção, Eyal e Sirer (2014) aponta duas possíveis abordagens para detecção do ataque de mineração egoísta, porém, ambas apresentam dificuldades de serem mensuradas em definitivo. Uma delas seria a detecção de cadeias de blocos órfãos, que não foram incorporadas na cadeia principal por conta da utilização de mineração egoísta, porém, as informações sobre essas cadeias são quase impossíveis de se conseguir por alguns motivos. O principal deriva do fato de que esses blocos não são do interesse da maioria dos mineradores comuns, pois não afetam as transações válidas, logo, concentram-se em blocos válidos pertencentes à cadeia principal, direcionando poucos recursos para o monitoramento dos blocos órfãos.

A outra abordagem liga-se à lacuna de tempo entre blocos sucessivos. A presença de mineração egoísta mostraria um desvio nesse tempo, pois o minerador precisa apresentar uma cadeia de blocos de comprimento $n + 1$ logo após um minerador honesto apresentar uma cadeia de blocos de comprimento n. O problema dessa segunda abordagem seria a detecção de apenas uma parte do comportamento do minerador egoísta, sendo necessária uma quantidade muito grande de dados para uma análise relevante.

3 Desenvolvimento

O método utilizado no trabalho tem uma abordagem não paramétrica em relação a amostra coletada na Blockchain. Realizando uma contagem de pares de mineração consecutivas realizadas pelo minerador, consequência do ataque de mineração egoísta, e comparando com uma recontagem na amostra embaralhada, o objetivo é descobrir suspeitos de executarem mineração egoísta. Este capítulo descreve as tecnologias utilizadas, o tratamento com os dados e os algoritmos responsáveis pela execução do método citado.

A Figura 4 ilustra as etapas de execução do método utilizado no trabalho. O p-value é gerado no final e serve como critério de egibilidade dos suspeitos.

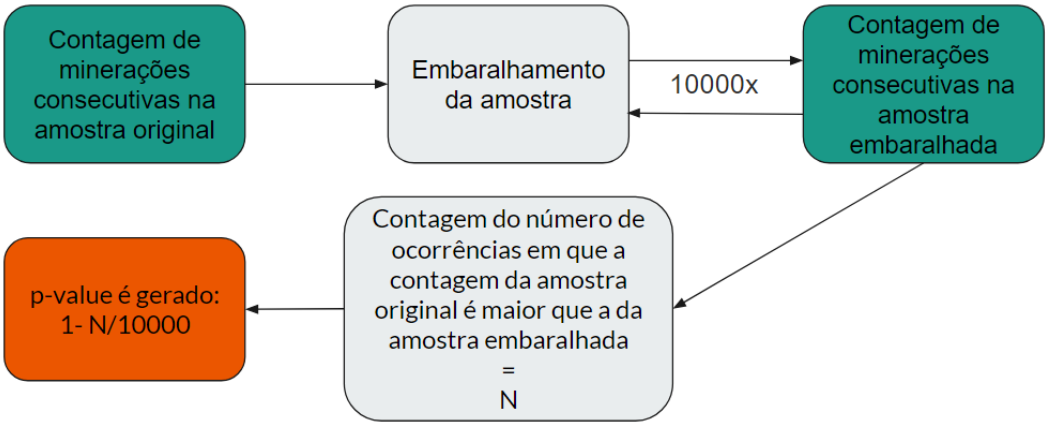

Figura 4 – Fluxo do método utilizado no trabalho.
Fonte: Elaborado pelo autor (2023).

3.1 Ferramentas e Tecnologias

Para a realização dos experimentos presentes no trabalho foram utilizadas várias ferramentas e tecnologias que tornaram mais fácil o armazenamento, análise e manipulação dos dados da Blockchain.

3.1.1 Python

Python é uma linguagem de programação de alto nível, ou seja, se aproxima da linguagem humana em sua sintaxe. A linguagem suporta multiplos paradigmas de programação, dentre eles, estão a Programação Orientada a Objetos (POO) e o paradigma funcional. Durante o desenvolvimento do trabalho foram utilizadas várias bibliotecas fornecidas pela mesma.

A escolha da linguagem ocorreu principalmente pela simplicidade na escrita, podendo realizar muitas tarefas com poucas linhas de código. A facilidade de depuração e o tratamento com grande volume de informações também foram relevantes na seleção. Em relação aos dados, o Python oferece várias bibliotecas avançadas para manipulação e tratamento dos mesmos, se extendendo a modelagens estatísticas e visualização. Todo o código utilizado para a realização do trabalho pode ser encontrado no repositório github.com/eduardoamedeiros22/blockChainAnalysis na plataforma GitHub.

NumPy : Biblioteca que suporta o processamento de estruturas de dados multi-dimensionais e possui grande coleção de funções matemáticas complexas e suas operações com essas estruturas;

Matplotlib : Biblioteca utilizada na criação de gráficos e visualizações de dados;

Random : Módulo utilizado para gerar números pseudo-aleatórios;

Requests : Biblioteca que permite a utilização de requisições HTTP com a linguagem Python.

3.1.2 Blockchain.com

Blockchain.com é uma empresa financeira de criptomoedas, sendo a primeira do ramo a oferecer um serviço de rastreio de transações dentro do Bitcoin. Esta fornece principalmente uma plataforma com opções para gerenciamento de ativos de criptomoedas e a busca de informações sobre transações, incluindo também, movimentações financeiras padrões, infraestrutura e dados de análises de mercado.

Além dos recursos utilizados neste trabalho, a plataforma fornece também o serviço de carteira digital possibilitando compra e venda de criptomoeda utilizando cartão de crédito e débito em conta. Gráficos de variação de valores, funções e APIs exclusivas para o gerenciamento tornam essa empresa uma grande fornecedora de *Blockchain as a Service* (BaaS).

Neste trabalho foi utilizado o serviço fornecido pela plataforma chamado Blockchain API, que utiliza o protocolo HTTP para expor *endpoints* de busca por informações sobre a Blockchain monitoradas pela empresa.

3.1.3 MongoDB

MongoDB é um software de banco de dados orientado a documentos, sendo esse livre, de código aberto e multiplataforma, escrito na linguagem C++.

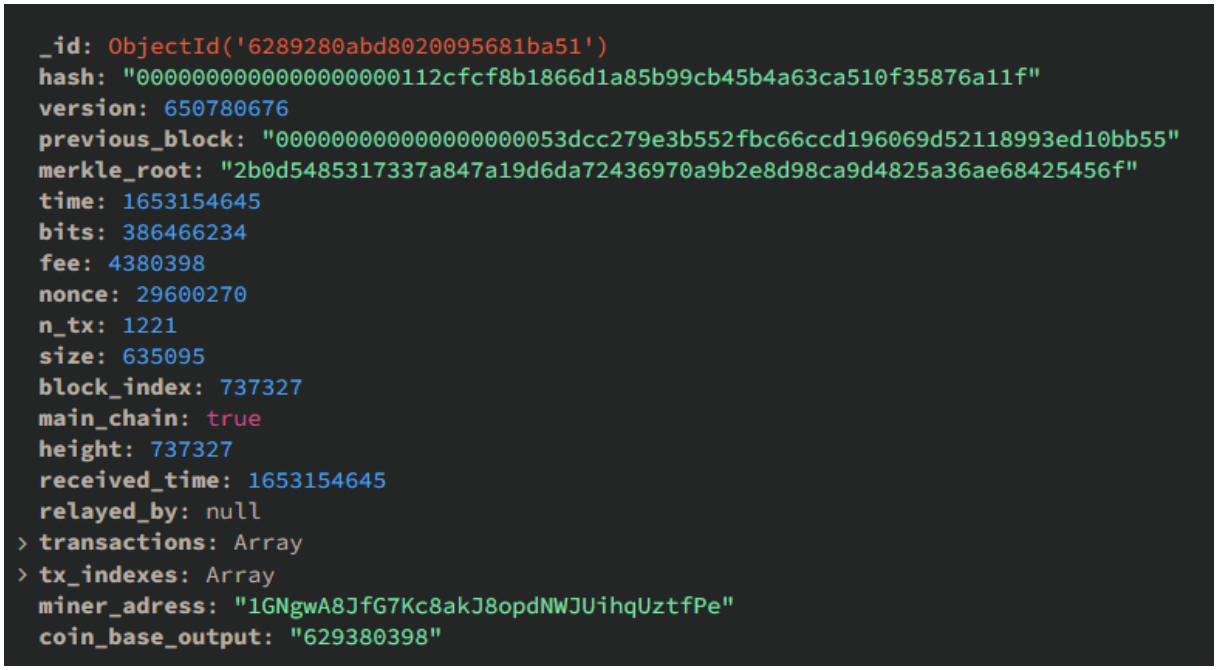

Figura 5 – Informações brutas do bloco salvas no MongoDB.
Fonte: Elaborado pelo autor (2023).

Devido a facilidade de utilização e flexibilidade o MongoDB foi escolhido para armazenar os dados brutos captados na API da Blockchain.com. Utilizando algoritmos em Python, as informações foram organizadas e novamente salvas no banco em estruturas mais fáceis de serem manipuladas durante a análise.

3.2 Obtenção dos dados

A API fornecida pela Blockchain.com já traz as informações mais relevantes sobre a Blockchain do Bitcoin, por exemplo, a quantidade total de blocos já minerados, a recompensa ao minerar um bloco (quantidade de Bitcoin recebida pelo minerador), dentre outros. Porém, as informações mais importantes para este trabalho, que também são fornecidas pela API, são as relacionadas aos blocos minerados.

Durante a captura de dados, a API fornece informações essenciais para armazenamento da sequência correta de blocos como, altura e *hash* do atual e do anterior. Para identificar o minerador de cada um deles, a transação *coinbase* (a primeira contendo a recompensa do minerador) está identificada dentro da lista de transações fornecida pela API. Dentro dessa, o campo *address* guarda o *hash* do minerador do bloco.

Aproximadamente um ano de blocos foram coletados, totalizando 56.008. Estes foram armazenados em uma instância local de uma banco NoSQL usando o software MongoDB. Na Tabela 1 estão a data, hora e altura do mais velho e mais novo bloco armazenado.

Tabela 1 – Informações sobre os blocos analisados

Data e Hora	Altura
19-05-2021 12:49:40	684192
21-05-2022 17:37:25	737327

Fonte: Elaborado pelo autor (2023).

O código criado para captura e armazenamento dos dados foi escrito na linguagem Python, também utilizada para o agrupamento e a criação de análises e geração de resultados.

3.3 Organização dos dados

Para que as análises e a busca por evidências de mineração egoísta fossem feitas, foi preciso realizar um pré-processamento dos dados de forma a facilitar o processo.

A lista de blocos armazenadas no banco local no formato apresentado na Figura 5 foi transformada em uma lista simples como na segunda coluna da Tabela 2 contendo apenas o *hash* do minerador como identificador do bloco, e essa foi salva em um arquivo com a extensão ".npy" utilizando a biblioteca NumPy da linguagem Python.

Com a criação dessa lista, houve um ganho de performance nas análises devido a um número menor de consultas ao banco de dados. Objetivando uma maior eficiência, foi gerada uma nova lista, mostrada na primeira coluna da Tabela 2 atribuindo números inteiros aos *hashes* dos mineradores, otimizando a identificação da mineração em sequência. Uma lista foi criada visando uma correspondência dos valores inteiros com os endereços dos mineradores.

Tabela 2 – Seção final de uma lista de blocos utilizando o endereço e o inteiro como identificador do minerador

Blocos (Lista Inteiros .npy)	Blocos (Lista .npy)
...	...
5	1KFHE7w8BhaENAswwryaoccDb6qcT6DbYY
18	19dENFt4wVwos6xtgwStA6n8bbA57WCS58
9	12dRugNcdxK39288NjcDV4GX7rMsKCGn6B
2	1Bf9sZvBHPFGVPX71WX2njhd1NXKv5y7v5
2	1Bf9sZvBHPFGVPX71WX2njhd1NXKv5y7v5
2	1Bf9sZvBHPFGVPX71WX2njhd1NXKv5y7v5
1	1GNgwA8JfG7Kc8akJ8opdNWJUihqUztfPe
4	1CK6KHY6MHgYvmRQ4PAafKYDrg1ejbH1cE
18	19dENFt4wVwos6xtgwStA6n8bbA57WCS58
18	19dENFt4wVwos6xtgwStA6n8bbA57WCS58

Fonte: Elaborado pelo autor (2023).

Com a lista finalizada, foi realizada uma divisão em doze listas menores, o que equivale ao período de 1 mês de blocos minerados.

3.4 Metodologia

Em Li, Yang e Tessone (2020a), Li, Yang e Tessone (2020b) os autores apresentam um método para identificar mineração egoísta baseado na contagem de blocos de um mesmo minerador em sequência. O método proposto consiste em utilizar o teste Z-Score (ou pontuação padrão) para identificar a que distância os dados da Blockchain estão de uma distribuição obtida por um conjunto de mineradores honestos. Quando este valor for superior ao limite 2, considera-se que os dados em questão são provenientes de mineração egoísta, com 95% de confiança. É possível observar que este método está fortemente ligado ao modelo de Distribuição Gaussiana, ou seja, ao apresentar desvios padrões maiores que 2 em relação a média, temos alta confiança de que este evento não é produto de mineração honesta.

O método desenvolvido nesse trabalho leva em conta a incidência de pares de blocos minerados consecutivamente dentro de um intervalo. Dado que essa distribuição não é necessariamente gaussiana, ou seja, não é especificado nenhum parâmetro para a mesma, faz-se necessária a utilização de outra abordagem para análise desses dados. Assim, propomos um teste não paramétrico que conta diretamente as minerações sequenciais presentes na Blockchain e compara esses valores com os dados esperados em um cenário sem mineração egoísta.

Algorithm 1: Análise de mineração consecutiva na amostra original

input : $BlockInterval \leftarrow$ *Lista de blocos a ser analisado*
$QtdMiners \leftarrow$ *Quantidade de mineradores diferentes no intervalo*

output: $LMCO$ (Lista da de mineração consecutiva na amostra original)

```
1 LMCO ← []
2 for i ← 0 to QtdMiners do
3     LMCO[i] ← 0
4 N ← length(BlockInterval)
5 for k ← 1 to N do
6     if BlockInterval[k] == BlockInterval[k − 1] then
7         blockMiner ← BlockInterval[k]
8         LMCO[blockMiner] + +
9 return LMCO
```

No Algoritmo 1 é mostrada a contagem das minerações consecutivas na sequência original de blocos, nota-se no código que há a inicialização de uma lista de tamanho

igual a quantidade de mineradores ativos no período analisado, para serem anotados os resultados da contagem. Criando essa lista, as minerações consecutivas podem ser contadas incrementando um valor a posição, utilizando o índice como o identificador do minerador, visto que a lista dos blocos como mostrada na Tabela 2, possui números inteiros de 0 a 88.

No Algoritmo 2 é realizada a mesma estratégia de contagem, porém, com a lista de blocos original embaralhada `N` vezes. O resultado é diferente do Algoritmo 1, pois se trata de uma lista de listas com a contagem das minerações consecutivas.

Algorithm 2: Análise de mineração consecutiva na amostra embaralhada

input : $Nperm \leftarrow Número\ de\ permutações$
$BlockInterval \leftarrow Lista\ de\ blocos\ a\ ser\ analisado$
$QtdMiners \leftarrow Quantidade\ de\ mineradores\ diferentes\ no\ intervalo$
output: $LLMC$ (Lista de listas das análises de mineração consecutiva)

```
1  while Nperm > 0 do
2      Shuffle(BlockInterval)
3      LMC (Lista de contagem de mineração consecutiva)← []
4      for i ← 0 to QtdMiners do
5          LMC[i] ← 0
6      N ← length(BlockInterval)
7      for k ← 1 to N do
8          if BlockInterval[k] == BlockInterval[k − 1] then
9              blockMiner ← BlockInterval[k]
10             LMC[blockMiner] + +
11     Nperm − −
12     LLMC.append(CMC)
13 return LLMC
```

Seguindo no Algoritmo 3, a lista contendo a contagem original retornada pelo Algoritmo 1 é comparada com as dez mil permutações, gerando uma lista com o número de vezes em que a contagem original é maior que a obtida por permutações.

Algorithm 3: Teste não paramétrico - Comparação da análise original com as análises dos blocos embaralhados

input : $LMCO \leftarrow$ *Lista das análises de mineração consecutiva da amostra original*
$LLMCO \leftarrow$ *Lista de lista das análises de mineração consecutiva da amostra embaralhada*
$QtdMiners \leftarrow$ *Quantidade de mineradores diferentes no intervalo*
$Nperm \leftarrow$ *Número de permutações*

output: AME (Lista de Tuplas (Id minerador , p-value) Análise de mineração egoísta utilizando o p-value de 0,05)

```
// AFC = Análise final das comparações amostra original x
   permutações
```

```
AFC ← []
AME ← []
for i ← 0 to QtdMiners do
    AFC[i] ← 0
for analisysList in LLMCO do
    for k ← 0 to QtdMiners do
        if LMCO[k] > analisysList[k] then
            AFC[k] + +
for analisys in AFC do
    if analisys >= (Nperm * 0.95) then
        // A função index retorna o index (posição) do elemento na
           lista
        TuplaResultado = (index(analisys), analisys/Nperm)
    AME.append(TuplaResultado)
return AME
```

Como mostrado no Algoritmo 3, uma ultima análise é feita com a lista gerada durante sua execução. Utilizando um *p-value* de 0,05, os mineradores suspeitos monstram uma contagem original de minerações consecutivas maior que 95% das permutações. Sendo também uma informação importante para a análise, junto aos mineradores suspeitos, um *script* foi criado para gerar dados do poder computacional do minerador no período.

4 Experimentos e Resultados

Neste Capítulo são apresentadas as análises e resultados gerados pela aplicação do método desenvolvido no Capítulo anterior. Identificação dos mineradores considerados suspeitos e outras características da amostra que contribuem para a conclusão. Na ultima Seção é colocada em destaque as discussões sobre o método utilizado e as características do próprio ecossistema.

4.1 Identificação da amostra

O primeiro resultado significativo obtido com a organização e processamento dos dados, foi a distribuição dos mineradores dentro do intervalo de blocos minerados. Essa distribuição é ilustrada pelo gráfico de dispersão da Figura 6.

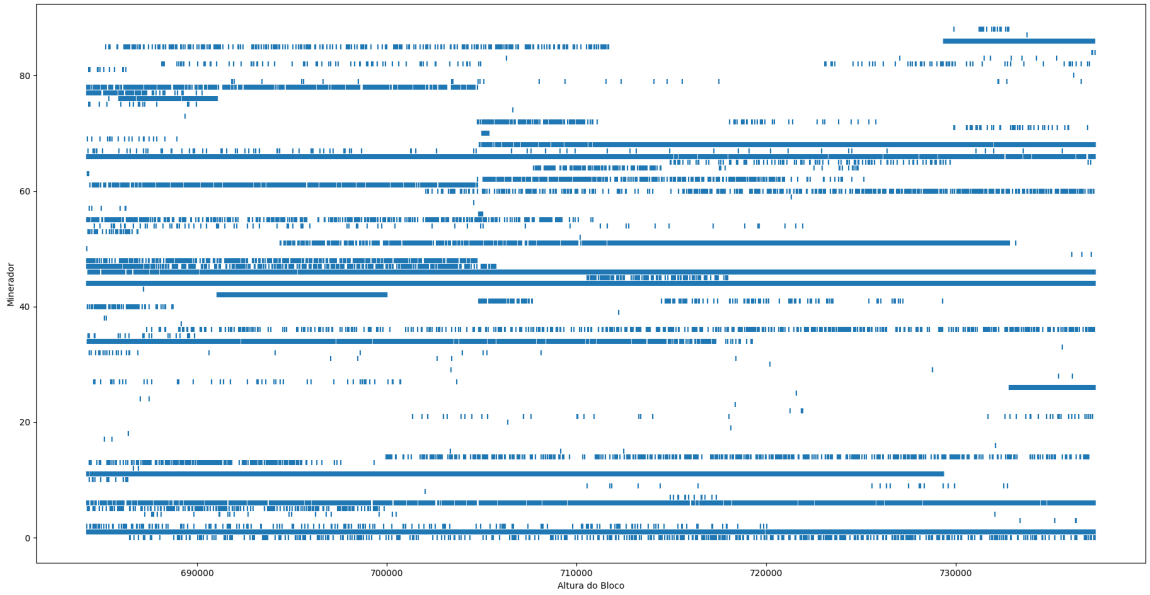

Figura 6 – Gráfico de Dispersão (Minerador x Altura do bloco).
Fonte: Elaborado pelo autor (2023).

É possível perceber no gráfico uma grande variação na participação dos mineradores, com alguns apresentando constância durante todo o período (mesmo os com baixa taxa de mineração), ou seja, se mantém ativos durante todo o período analisado. Outros possuem uma grande quantidade de blocos minerados em alguma faixa de tempo e poucos ou nenhum em outra faixa.

4.2 Identificação dos mineradores

A Tabela 3 mostra uma relação entre o nome, o *hash* do minerador e o número atribuído a ele para as análises ao seu nome cadastrado na blockchain.info, facilitando o entendimento dos resultados subsequentes que utilizam apenas o número ou o *hash*.

Utilizando o método proposto pelos algoritmos no capítulo anterior, 18 dos 89 mineradores do período analisado foram considerados suspeitos, estes foram numerados de 1 a 18 de acordo com a sua aparição dentro do intervalo de blocos. Portanto, apenas os 18 mineradores suspeitos foram utilizados nas discussões e análises finais do trabalho.

Tabela 3 – Relação entre o *hash* dos mineradores e o inteiro associado, originalmente existiam 89 mineradores presentes no período analisado, mas apenas 18 foram considerados suspeitos seguindo o método proposto, o número inteiro associado foi utilizado objetivando maior eficiência dos algoritmos utilizados no trabalho. Os nomes das *pools* de mineradores foram obtidos da plataforma blockchain.com.

Minerador	Endereço Minerador	Nome
1	1GNgwA8JfG7Kc8akJ8opdNWJUihqUztfPe	Poolin
2	1Bf9sZvBHPFGVPX71WX2njhd1NXKv5y7v5	BTC.com
3	bc1qadv9a92ln0l58hh6g0e9jeuz4y5cg0m0sjpf8x	SBI Crypto
4	1CK6KHY6MHgYvmRQ4PAafKYDrg1ejbH1cE	Braiins Pool
5	1KFHE7w8BhaENAswwryaoccDb6qcT6DbYY	F2Pool
6	38XnPvu9PmonFU9WouPXUjYbW91wa5MerL	Antpool
7	1JvXhnHCi6XqcanvrZJ5s2Qiv4tsmm2UMy	Binance Pool
8	12dRugNcdxK39288NjcDV4GX7rMsKCGn6B	Antpool
9	12cKiMNhCtBhZRUBCnYXo8A4WQzMUtYjmR	12cKi-tYjmR
10	bc1qx9t2l3pyny2spqpqlye8svce70nppwtaxwdrp4	Binance Pool
11	1EepjXgvWUoRyNvuLSAxjiqZ1QqKGDANLW	Huobi Pool
12	191sNkKTG8pzUsNgZYKo7DH2odg39XDAGo	191sN-XDAGo
13	bc1quyzwwznn97ydemv8l3kzk700gck2tyexd5sz0f	bc1qu-5sz0f
14	1JhAQ7UNYXvVvrp4nfNHLoTWTPgfuHGCDp	1JhAQ-HGCDp
15	35y82tEPDa2wm6tzkEacMG8GPPW7zbMj83	35y82-bMj83
16	bc1qf274x7penhcd8hsv3jcmwa5xxzjl2a6pa9pxwm	F2Pool
17	125m2H43pwKpSZjLhMQHneuTwTJN5qRyYu	125m2-qRyYu
18	19dENFt4wVwos6xtgwStA6n8bbA57WCS58	Foundry USA

Fonte: Elaborado pelo autor (2023).

4.3 Variação do poder de mineração

Nessa seção serão apresentados gráficos demonstrando a variação de poder computacional de alguns grupos de mineradores considerados suspeitos por atingirem um valor específico de poder durante uma fração de tempo. O poder computacional de um determinado minerador é calculado utilizando a razão entre número de blocos por ele minerado

e a quantidade total de blocos do período.

Na Tabela 4 são apresentadas informações detalhadas sobre os 18 mineradores suspeitos selecionados, a quantidade de blocos minerados em todo o período analisado e a coluna de pico de processamento, que representa o maior valor de poder computacional que o minerador atingiu em algum intervalo desse período.

Tabela 4 – Mineradores suspeitos participantes dos blocos minerados de 19-05-2021 até 21-05-2022.

Minerador	Nome	Blocos Minerados	Pico de Processamento
1	Poolin	3162	23%
2	BTC.com	4282	22%
3	SBI Crypto	489	6%
4	Braiins Pool	2996	16%
5	F2Pool	7718	29%
6	Antpool	1332	24%
7	Binance Pool	3078	22%
8	Antpool	7238	31%
9	12cKi-tYjmR	75	4%
10	Binance Pool	2200	22%
11	Huobi Pool	468	10%
12	191sN-XDAGo	581	13%
13	bc1qu-5sz0f	21	10%
14	1JhAQ-HGCDp	14	6%
15	35y82-bMj83	266	10%
16	F2Pool	329	13%
17	125m2-qRyYu	49	3%
18	Foundry USA	7657	32%

Fonte: Elaborado pelo autor (2023).

O pico de processamento foi calculado junto a geração do gráfico de variação do poder computacional, utilizando intervalos de 100 blocos e uma variação de dez blocos entre os intervalos, onde, dentro desse período era contabilizada a porcentagem minerada pelo usuário analisado em relação ao todo. Assim, a cada nova etapa ocorria uma comparação da porcentagem atual em relação a anterior. Desse modo, o pico de processamento refere-se a todo o período de 1 ano de blocos analisados.

É importante citar que esses valores de poder computacional podem não indicar um poder real de processamento dos grupos de mineração. Como o cálculo leva em conta os blocos minerados pelo grupo, são inclusos também aqueles obtidos por meio da estratégia de mineração egoísta.

As Figuras 7, 8 e 9 ilustram a variação de poder computacional de três grupos de

mineradores selecionados, utilizando como critério de inclusão, aqueles que atingiram ao menos 25% de poder de processamento em algum momento. Essas informações podem ser utilizadas junto à análise de mineração consecutiva, comparando os períodos de atividade suspeita com os intervalos onde houve um maior poder de processamento registrado.

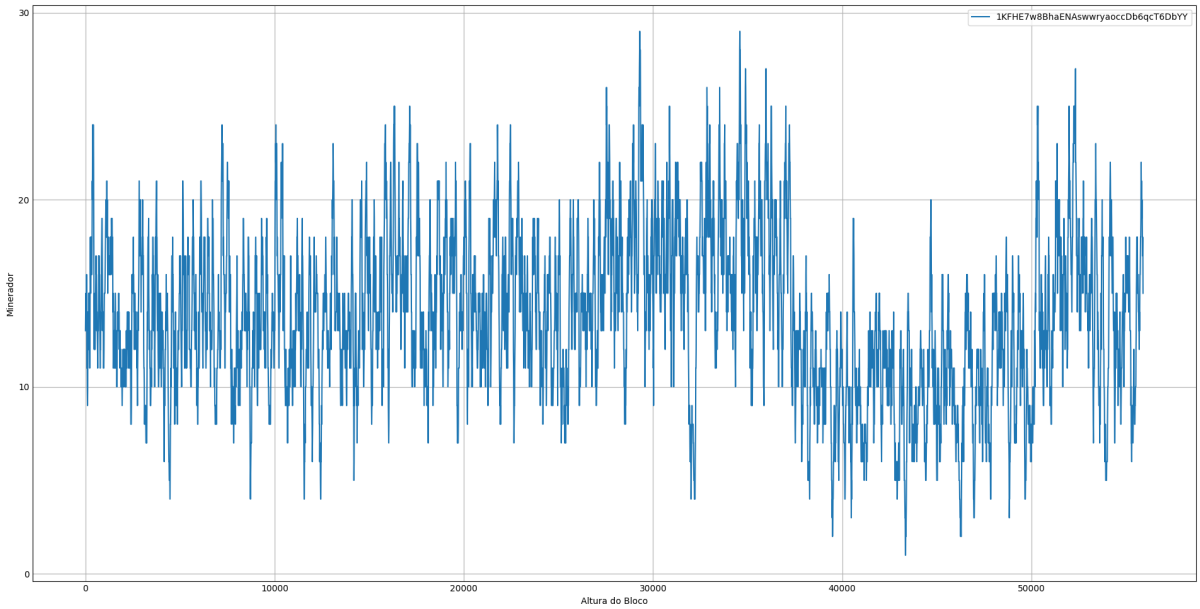

Figura 7 – Variação do poder computacional de F2Pool ao longo de todo o período analisado. Altura do bloco x Poder computacional.
Fonte: Elaborado pelo autor (2023).

A Figura 7 mostra a variação do poder computacional de um minerador (ou pool de mineradores) que não atinge 30%, mas em uma fração do período, mantém valores entre 25% e 30%.

As Figuras 8 e 9 mostram também a variação do poder computacional de 2 mineradores (ou *pools* de mineradores), porém, diferente da Figura 7, ambos atingem picos maiores que 30%.

4.4 Resultado das análises de mineração consecutiva

A Tabela 5 mostra o resultado da análise de mineração consecutiva explicada no capítulo anterior, o objetivo do trabalho era analisar mineradores que atingem o valor do p-value menor que 0,05, tornando-os suspeitos de executarem mineração egoísta. A coluna mês possui valores que representam um dos doze períodos analisados e a informação do poder computacional foi adicionada a tabela para complementar a análise de mineração consecutiva. É possível observar que mesmo mineradores com baixo poder computacional são indicados como suspeitos.

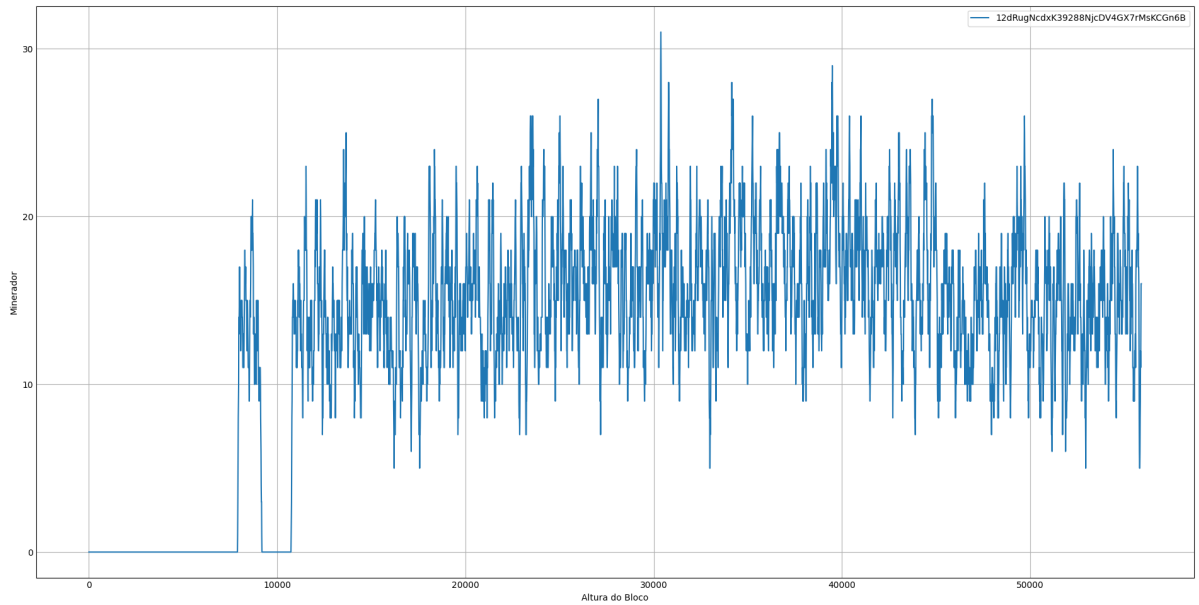

Figura 8 – Variação do poder computacional de Antpool ao longo de todo o período analisado. Altura do bloco x Poder computacional.
Fonte: Elaborado pelo autor (2023).

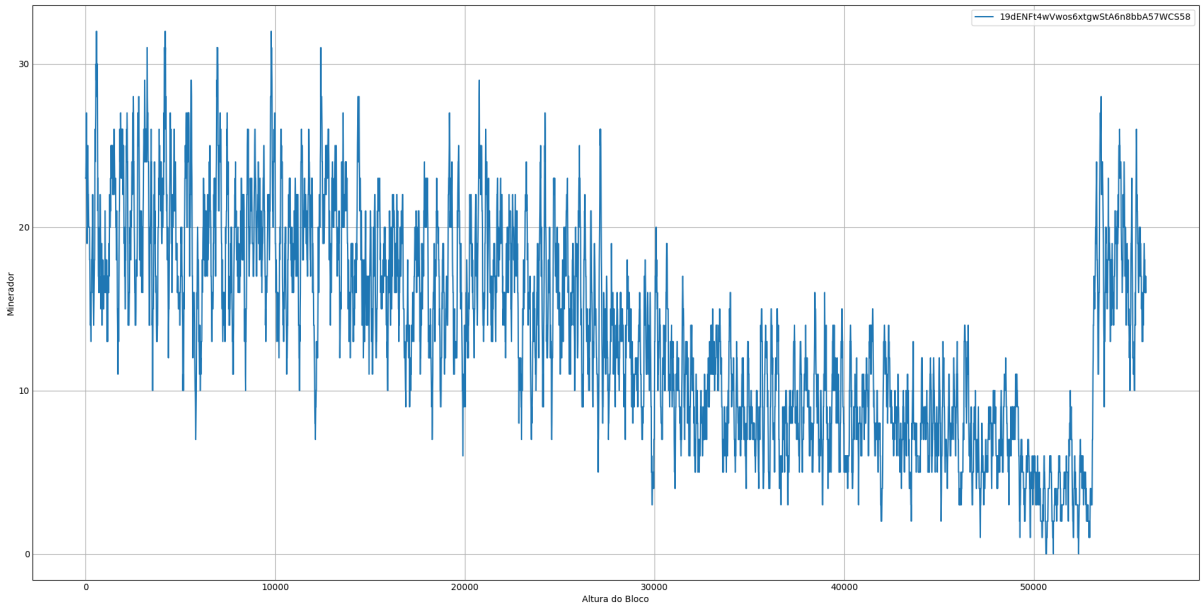

Figura 9 – Variação do poder computacional de Foundry USA ao longo de todo o intervalo de tempo analisado. Altura do bloco x Poder computacional.
Fonte: Elaborado pelo autor (2023).

Tabela 5 – Análise da probabilidade do minerador ter conseguido a mineração de blocos consecutivos considerando seu poder computacional e um processo de mineração aleatório ao permutar a lista de blocos.
PC - Poder Computacional Médio na fração do período
MC - Mineração consecutiva
p-value - Distribuição Original MC / Permutações (foram selecionados apenas os que possuíam valores menores ou iguais a 0,05)

Mês	Minerador Suspeito	MC (Distribuição Original)	PC	p-value	Blocos Minerados no Período
1	7	2	0,407%	0.0018	19
2	8	20	3,578%	0.0001	167

É mostrado na quinta coluna a aplicação do Algoritmo 3, ou seja, uma comparação entre a análise original do número de minerações consecutivas no período e a análise da massa embaralhada dez mil vezes. Assim, o valor resultante é o complemento da divisão entre o número de vezes que a análise da distribuição original apresentou um valor maior quando comparado com a amostra embaralhada e o número de permutações, no caso, 10.000.

4.5 Discussões

Ao considerar a distribuição de minerações em sequência como mecanismo de detecção de Mineração Egoísta, alguns fatores são importantes para a análise:

Problemas com o p-value : Dividindo todos os dados mês a mês, método que é necessário para considerar o poder computacional de cada minerador como estável, é formado um dado estatístico que pode criar p-values relevantes numericamente, porém sem nenhuma significância estatística. Tal situação é descrita em Head et al. (2015) e conhecida como *p-hacking*, ou seja, uma manipulação do p-value. Também notado em Ioannidis (2005), deve existir uma preocupação acerca da quantidade de dados de uma hipótese e distorções de julgamento causadas pelo envolvimento do pesquisador com a amostra.

O Poder Computacional real : Um resultado direto da aplicação de Mineração Egoísta é a produção de blocos de maneira desproporcional ao poder computacional. Paradoxalmente, quando determinamos que um minerador com uma proporção de blocos α está praticando Mineração Egoísta devido a geração de mais blocos em sequência do que outro com esse mesmo α produziria, pode-se supor que esse minerador tenha um poder computacional menor do que o α calculado.

O Ecossistema dos Mineradores : Os acordos e arranjos entre mineradores são fatores complexos com a `locação` de poder computacional e grupos de mineradores que podem mudar de um minerador chefe para outro. Essas situações mascaram a frequência de blocos, fazendo com que a abordagem apresentada nesse trabalho seja ainda menos efetiva na detecção do ataque.

Levando em conta todos esses fatores, deve ser considerada uma união com outras técnicas de detecção para que dados realmente relevantes sejam gerados. Como as abordagens já citadas em Eyal e Sirer (2014) de lacunas de tempo entre blocos e cadeias orfãs, em Davidson e Diamond (2020) reforçando a tática de retenção para driblar os algoritmos de ajuste de dificuldade.

5 Conclusão

Descoberta pouco depois da implementação do Bitcoin, o SM é uma prática que pode ser adota pelos mineradores, de modo a não deixar muita informação na Blockchain. Apesar de não ser um ataque clássico a um sistema financeiro, em relação há perdas explicitas para os usuários da criptmoeda, pode causar danos à confiança do sistema.

Existem poucas formas de se tentar identificar e combater o SM. Uma maneira desenvolvida para tal, é a contagem de minerações consecutivas e a utilização do Z-Score para determinar se uma sequência específica de identificadores de mineradores é compatível com o comportamento honesto.

Neste trabalho, foi apresentado um aprimoramento sobre esta abordagem valendo-se de método não-paramétrico para a identificação da mineração egoísta utilizando apenas dados da própria Blockchain do Bitcoin. É esperado que uma abordagem não paramétrica ofereça uma conclusão mais robusta do que uma baseada em uma distribuição específica, nesse caso, a Distribuição Gaussiana ou normal da mineração para o mesmo minerador em uma sequência de blocos.

Aproximadamente um ano de dados da Blockchain foram analisados e após a aplicação da técnica proposta, 18 mineradores foram considerados suspeitos de praticar mineração egoísta nesse período. Mesmo com essa evidência, o problema ainda está em trabalhar com outras abordagens para maior robustez dos resultados, visto que a análise utilizada pode deturpar o p-value calculado. As dificuldades em detectar a mineração egoísta possuem características que o estudo atual não consegue compreender.

Da mesma forma, considerando os problemas de segurança que envolvem a mineração egoísta e a complexidade do sistema de mineração, outras fontes de dados sobre a Blockchain devem ser consideradas para melhorar a qualidade dos resultados. Um dos exemplos seriam as informações da rede P2P armazenadas pelos full nodes. Esses programas validam as transações de outros full nodes executadas na Blockchain e mantém uma cópia completa dela.

Referências

ALVES, P. H. et al. Desmistificando blockchain: Conceitos e aplicações. In: _____. [S.l.: s.n.], 2018. p. 1–24. Citado na página 11.

ANTONOPOULOS, A. M. Book. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies (2nd ed.), O'Reilly Media, Inc.* [S.l.]: O'Reilly Media - EUA, 2017. ISBN 978-1-491-95438-6. Citado 3 vezes nas páginas 13, 15 e 17.

ANTONOPOULOS, A. M. Book. *Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, Inc.* [S.l.]: O'Reilly Media - EUA, 2019. ISBN 978-1491971949. Citado na página 16.

CAMPBELL, M.; SWINSCOW, T. *Statistics at Square One*. Wiley, 2009. ISBN 9781405191005. Disponível em: <https://books.google.com.br/books?id=O37YwAEACAAJ>. Citado na página 18.

DAVIDSON, M.; DIAMOND, T. *On the Profitability of Selfish Mining Against Multiple Difficulty Adjustment Algorithms*. 2020. Cryptology ePrint Archive, Paper 2020/094. <https://eprint.iacr.org/2020/094>. Disponível em: <https://eprint.iacr.org/2020/094>. Citado na página 32.

DOUCEUR, J. R. The sybil attack. In: DRUSCHEL, P.; KAASHOEK, F.; ROWSTRON, A. (Ed.). *Peer-to-Peer Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 251–260. ISBN 978-3-540-45748-0. Citado na página 17.

EYAL, I.; SIRER, E. G. Majority is not enough: Bitcoin mining is vulnerable. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 8437, p. 436–454, 2014. ISSN 16113349. Citado 3 vezes nas páginas 9, 19 e 32.

HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. *Journal of Cryptology*, v. 3, p. 99–111, 1990. Citado na página 11.

HEAD, M. et al. The extent and consequences of p-hacking in science. *PLoS biology*, v. 13, p. e1002106, 03 2015. Citado na página 32.

HEILMAN, E. *One Weird Trick to Stop Selfish Miners: Fresh Bitcoins, A Solution for the Honest Miner*. 2020. Citado 2 vezes nas páginas 9 e 17.

HEILMAN, E. et al. Eclipse attacks on Bitcoin's Peer-to-Peer network. In: *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, 2015. p. 129–144. ISBN 978-1-939133-11-3. Disponível em: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman>. Citado na página 17.

IOANNIDIS, J. Why most published research findings are false. *PLoS medicine*, v. 2, p. e124, 09 2005. Citado na página 32.

LI, S.-N.; YANG, Z.; TESSONE, C. J. Mining blocks in a row: A statistical study of fairness in bitcoin mining. In: *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. [S.l.: s.n.], 2020. p. 1–4. Citado 3 vezes nas páginas 9, 18 e 24.

LI, S.-N.; YANG, Z.; TESSONE, C. J. Proof-of-work cryptocurrency mining: a statistical approach to fairness. In: *2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*. [S.l.: s.n.], 2020. p. 156–161. Citado 3 vezes nas páginas 9, 18 e 24.

MERKLE, R. A certified digital signature. In: . [S.l.: s.n.], 1989. v. 435, p. 218–238. ISBN 978-0-387-97317-3. Citado na página 15.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. maio 2009. Disponível em: <http://www.bitcoin.org/bitcoin.pdf>. Citado 4 vezes nas páginas 9, 12, 13 e 15.

NEGY, K.; RIZUN, P.; SIRER, E. Selfish mining re-examined. In: _____. [S.l.: s.n.], 2020. p. 61–78. ISBN 978-3-030-51279-8. Citado na página 18.

SAAD, M. et al. *Countering Selfish Mining in Blockchains*. 2020. Citado na página 17.

SENDIN, I. da S. *Funções de Hashing Criptográficas*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 1999. Citado na página 11.