UMA PROPOSTA DE VIABILIZAÇÃO DA RELAÇÃO ENTRE O FINANCIADOR E O PRESTADOR DE SERVIÇOS AMBIENTAIS ATRAVÉS DE ORÁCULOS BLOCKCHAIN

Anderson Marinho Pontes de Barros

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2024

Anderson Marinho Pontes de Barros

UMA PROPOSTA DE VIABILIZAÇÃO DA RELAÇÃO ENTRE O FINANCIADOR E O PRESTADOR DE SERVIÇOS AMBIENTAIS ATRAVÉS DE ORÁCULOS BLOCKCHAIN

Dissertação apresentada ao Programa de Pós-Graduação em Informática, do Centro de Informática, da Universidade Federal da Paraíba (área de concentração: Computação Distribuída) como requisito obrigatório para o Título de Mestre em Informática.

Orientador: Guido Lemos de Souza Filho

Fevereiro de 2024

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Anderson Marinho Pontes de Barros, candidato ao título de Mestre em Informática na área de Sistemas de Computação, realizada em 29 de dezembro de 2023.

Aos vinte e nove dias do mês de dezembro do ano de dois mil e vinte e três, às 16h, no Centro de Informática da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho Final do discente Anderson Marinho Pontes de Barros, vinculado a esta Universidade sob a matrícula nº 20211000329, candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa "Sinais, Sistemas Digitais e Gráficos", do Programa de Pós-Graduação em Informática. A comissão examinadora foi composta pelos professores: Guido Lemos de Souza Filho, Orientador e Presidente; Tiago Maritan Ugulino de Araujo (PPGI/UFPB), Examinador Interno; e Cicero Inacio da Silva (UNIFESP), Examinador Externo à Instituição. Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes, comunicou a finalidade da reunião e passou a palavra ao candidato para que ele fizesse a exposição oral do trabalho de dissertação intitulado "**Uma Proposta de Viabilização da Relação Entre o Financiador e o Prestador de Serviços Ambientais Através de Oráculos Blockchain**". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "***aprovado***". Do ocorrido, eu, Iguatemi Eduardo da Fonseca, Vice-Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da Banca Examinadora. João Pessoa, 29 de dezembro de 2023.

gov.br Documento assinado digitalmente
IGUATEMI EDUARDO DA FONSECA
Data: 05/02/2024 09:19:26-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Iguatemi Eduardo da Fonseca

ICP Brasil Documento assinado digitalmente
GUIDO LEMOS DE SOUZA FILHO
Data: 30/01/2024 19:20:59-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Guido Lemos de Souza Filho
Orientador (PPGI-UFPB)

gov.br Documento assinado digitalmente
TIAGO MARITAN UGULINO DE ARAUJO
Data: 31/01/2024 15:09:43-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Tiago Maritan Ugulino de Araujo
Examinador Interno (PPGI-UFPB)

gov.br Documento assinado digitalmente
CICERO INACIO DA SILVA
Data: 31/01/2024 16:17:07-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Cicero Inacio da Silva
Examinador Externo à Instituição (UNIFESP)

Catalogação na publicação
Seção de Catalogação e Classificação

B277p Barros, Anderson Marinho Pontes de.
Uma proposta de viabilização da relação entre o financiador e o prestador de serviços ambientais através de oráculos blockchain / Anderson Marinho Pontes de Barros. - João Pessoa, 2024.
102 f. : il.

Orientação: Guido Lemos de Souza Filho.
Dissertação (Mestrado) - UFPB/CI.

1. Informática. 2. Processamento digital de imagem. 3. Oráculos blockchain. 4. Preservação ambiental. 5. Rede neural convolucional. I. Souza Filho, Guido Lemos de. II. Título.

UFPB/BC CDU 004(043)

Elaborado por Larissa Silva Oliveira de Mesquita - CRB-15/746

"Temos um grande problema se não entendermos o planeta que queremos salvar." (Carl Sagan)

AGRADECIMENTOS

Gostaria de expressar minha sincera gratidão a todas as pessoas que tornaram possível a realização deste trabalho. Esta jornada acadêmica foi enriquecedora e significativa, e muitas mãos foram fundamentais para o sucesso desta dissertação.

Primeiramente, quero agradecer ao meu orientador, Dr. Guido Lemos de Souza Filho, pela orientação dedicada, paciência incansável e valioso tempo investido. Sua *expertise* e *insights* foram cruciais para o desenvolvimento deste trabalho, proporcionando um ambiente propício para aprendizado e crescimento acadêmico.

À banca examinadora, composta pelos estimados professores Dr. Tiago Maritan Ugulino de Araujo e Dr. Cícero Inácio da Silva, agradeço pelas instruções valiosas e contribuições enriquecedoras durante a defesa desta dissertação. Cada um de vocês desempenhou um papel fundamental na avaliação e no aprimoramento deste estudo, e sou imensamente grato por suas valiosas sugestões.

À Professora Dra. Thais Gaudêncio do Rêgo, minha sincera gratidão pelas inúmeras vezes em que sua ajuda foi imprescindível. Suas orientações e apoio foram determinantes em momentos desafiadores, e agradeço por sua disponibilidade constante e generosidade em compartilhar conhecimento.

Ao Pesquisador Diego Ramon Bezerra da Silva, expresso meu reconhecimento pelos ensinamentos sólidos na área de aprendizagem de máquina e processamento digital de imagens. Suas orientações e contribuições foram fundamentais para alicerçar as bases teóricas desta pesquisa, proporcionando uma compreensão mais profunda e abrangente do tema.

A todos os que, de alguma forma, contribuíram para esta jornada acadêmica, meu muito obrigado. Este trabalho não seria possível sem o apoio e colaboração de cada um de vocês. Que este seja apenas o início de muitas conquistas e aprendizados compartilhados.

RESUMO

A necessidade de uma maior transparência e confiabilidade nas etapas de investimento e administração dos recursos dos projetos de preservação ambiental motiva a adoção de tecnologias que permitam o acompanhamento da utilização e aplicação desses recursos de forma transparente e auditável, e assegurem que esses recursos sejam liberados e utilizados pelas organizações e prestadores de serviços ambientais de forma responsável e correta. Este trabalho propõe uma solução utilizando o conceito de oráculos blockchain, que viabilize a operacionalização da Standing Forest Coin (SFC), um ativo digital que permite a coleta descentralizada de recursos a iniciativas que visam proteger o ecossistema da amazônia brasileira. Com a adoção de abordagens de aprendizagem de máquina e processamento digital de imagens, a solução apresenta o oráculo como a estrutura que garantirá credibilidade e segurança para os recursos aplicados em projetos dentro da SFC. A proposta será avaliada por um conjunto de experimentos, com intuito de analisar se o desempenho da arquitetura definida para o oráculo blockchain supre as demandas exigidas pela SFC em seu objetivo de viabilizar a relação entre o financiador e o prestado de serviços ambientais.

Palavras-chave: <Blockchain>, <Processamento Digital de Imagem>, <Oráculo>, <Preservação Ambiental>, <Rede Neural Convolucional>.

ABSTRACT

The need for greater transparency and reliability in the investment and resource management stages of environmental preservation projects motivates the adoption of technologies that allow the monitoring and application of these resources in a transparent and auditable way, and ensure that these resources are released and used by organizations and environmental service providers in a responsible and correct manner. This work proposes a solution using the concept of blockchain oracles, which enables the operationalization of the Standing Forest Coin (SFC), a digital asset that allows the decentralized collection of resources for initiatives that aim to protect the Brazilian Amazon ecosystem. With the adoption of machine learning and digital image processing, the solution presents the oracle as a structure that will guarantee credibility and security for the resources used in the projects. The proposal will be evaluated by a set of experiments, with the aim of analyzing whether the performance of the architecture defined for the blockchain oracle meets the demands required by the SFC in its objective of enabling the relationship between the financier and the provider of environmental services.

Keywords: <Blockchain>, <Digital Image Processing>, <Oracles>, <Environmental Preservation>, <Convolution Neural Networks>.

LISTA DE FIGURAS

1 Processo de encriptação e decriptação da criptografia simétrica 26

2 Envio de uma mensagem criptografada a uma chave pública 27

3 Procedimentos para executar uma função *hash* 27

4 Processo de certificação 28

5 Processo para gerar a assinatura digital de um documento 29

6 Processo para verificar a assinatura digital de um documento 29

7 Ilustração de um blockchain e seus ponteiros *hash*. 30

8 Encadeamento dos blocos 32

9 Blockchain com árvore binária Merkle 33

10 Processo de criação de um endereço Bitcoin 39

11 Formatação do conjunto UTXO 40

12 Lista encadeada de blocos 41

13 Interação com o *smart contract* 44

14 Ethereum *state function* 45

15 Ecossistema oráculo 46

16 A esquerda, uma imagem hipotética representada por um único canal com dimensões 5 x 5 que recebe a aplicação de um filtro 3 x 3. A direita, uma matriz ilustrando o somatório que fornece o resultado da convolução. . . . 51

17 Exemplo de um mapa de filtro 5x5 52

18 Exemplo mapa de filtro de curva 52

19 Exemplo de detecção de curva 53

20 Exemplo de função ReLU 53

21 Camada de Pooling 54

22 Exemplo de arquitetura de uma CNN 55

23 Arquitetura CNN com segmentação semântica 56

24 Exemplo do funcionamento de uma CNN de segmentação semântica 56

25 Rastreio parcial dos volumes de madeira 61

26 Arquitetura CEMVC 64

27 Plataforma de proteção ambiental baseada em blockchain 65
28 Mecanismo de interação entre o sensor e a rede blockchain 66
29 Monitoramento ambiental tridimensional 67
30 Caso de uso de compra e investimento em contrato 72
31 Caso de uso de compra e doação ao fundo de preservação 72
32 Diagrama de sequência dos fluxos presentes na implementação da SFC . . 73
33 Código-fonte do contrato inteligente da SFC 74
34 Funções de pagamento ao proprietário de terra e ao fundo de preservação . 74
35 Chamadas das funções de pagamento conforme a resposta do oráculo sobre a preservação da área 75
36 Verificação dos contratos com o EtherScan 76
37 Aquisição de SFC utilizando o MetaMask 76
38 Processamento do contrato do proprietário de terra 77
39 Resultado do processamento do contrato 78
40 Visão geral do experimento 80
41 Amostra de uma imagem em cores verdadeiras e sua máscara 85
42 Arquitetura U-Net 86
43 Matriz de confusão 90
44 Comparação das análises de imagens de satélite multitemporais: (a) imagem anterior e (b) imagem posterior 93
45 Polígonos de desmatamento (em vermelho) 93
46 Comparativo entre as cores verdadeiras (coluna 1), as máscaras refinadas manualmente (coluna 2) e o resultado do processamento da U-Net (coluna 3) 95

LISTA DE TABELAS

1 Estrutura do cabeçalho de um bloco da rede 32

2 Termos de busca relacionados à pesquisa 59

3 Resultado do processo de busca e seleção dos artigos 60

4 Avaliação dos principais artigos 68

5 Resultados do conjunto de testes 91

LISTA DE ABREVIATURAS

SFC – *Standing Forest Coin*

AC – Autoridade Certificadora

P2P – *Peer to Peer*

UTXO – *Unspent Transaction Outputs*

EOA – *Externally Owned Account*

EVM – *Ethereum Virtual Machine*

API – *Application Programming Interface*

IOT – *Internet of Things*

SPOF – *Single Point of Failure*

ONG – Organização não Governamental

FSC - *Forest Stewardship Council*

SDG - *Sustainable Development Goals*

ODS - Objetivos de Desenvolvimento Sustentável

CEMVC - *Commodity Ecology Mobile Virtual Community*

ICO - *Initial Coin Offering*

DLT - *Digital Line Tape*

CSV - *Comma Separated Values*

SVM - *Support Vector Machine*

CNN - *Convolutional Neural Network*

ReLU - *Rectified Linear Unit*

NIR - *Near-infrared*

NDVI - *Normalized Difference Vegetation Index*

Adam - *Adaptive Moment Estimation*

Sumário

1 INTRODUÇÃO **19**

1.1 Definição do Problema 20

1.2 Premissas e Hipóteses 21

1.3 Objetivo Geral 22

1.4 Objetivos Específicos 23

1.5 Estrutura da Dissertação 23

2 CONCEITOS GERAIS E REVISÃO DA LITERATURA **25**

2.1 Criptografia Simétrica 25

2.2 Criptografia Assimétrica 26

2.3 Funções *Hash* 27

2.4 Certificado Digital 28

2.5 Assinatura Digital 28

2.6 Blockchain 30

2.7 Estrutura Blockchain 31

2.8 Tipos de Blockchain 34

2.9 Protocolos de Consenso 34

2.10 Bitcoin 38

2.11 Carteira, Endereços e Criptografia de Curvas Elípticas 38

2.12 Transações 39

2.13 Mineração 40

2.14 Ethereum 42

2.15 Contratos Inteligentes 43

2.16 Oráculos 45

2.17 Aprendizagem de Máquina 48

2.18 Redes Neurais Convolucionais 50

2.19 Segmentação Semântica 55

2.20 Considerações 57

3 TRABALHOS RELACIONADOS **58**

3.1 Revisão Quasi-Sistemática da Literatura 58

3.1.1 Protocolo 58

3.1.2 Busca, Seleção e Classificação dos Artigos 60

3.2 Avaliação Qualitativa das Principais Contribuições 60

3.2.1 *LogLog: A Blockchain Solution for Tracking and Certifying Wood Volumes* 60

3.2.2 *Commodity Ecology: A Virtual Community Platform for Promoting Responsible Consumption and Production to Achieve SDG 12* . . . 62

3.2.3 *Environmental Monitoring System Based on Blockchain* 64

3.3 Considerações 68

4 METODOLOGIA **70**

4.1 Stading Forest Coin 71

4.2 Oráculo 78

4.3 Considerações 79

5 PLANEJAMENTO EXPERIMENTAL DO ORÁCULO **80**

5.1 Definição do Problema e Objetivos 82

5.2 Aquisição de Dados 83

5.3 Pré-Processamento de Dados 84

5.4 Divisão de Dados 85

5.5 Modelagem 85

5.6 Treinamento e Avaliação 87

5.7 Considerações 88

6 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS **89**

6.1 Discussão dos Resultados 89

6.2 Considerações 96

7 CONCLUSÕES E TRABALHOS FUTUROS **97**

REFERÊNCIAS **98**

1 INTRODUÇÃO

A região amazônica é um dos maiores desafios do mundo na busca pelo desenvolvimento de práticas sustentáveis de produção e consumo. Esse enorme desafio de desenvolvimento social, econômico e ambiental e a conservação desse bioma, inclui necessariamente a questão da produção sustentável, principalmente a criação de emprego e renda a partir da conservação e/ou exploração regrada dos recursos naturais [1].

Ainda segundo Lopes [1], é preciso deixar claro que o grande desafio de mudar o modelo de desenvolvimento regional é político e a ciência e a tecnologia, por si só, são insuficientes quando sozinhas. No entanto, a ciência e a tecnologia desempenham um papel fundamental no desenvolvimento sustentável da Amazônia, considerando a necessidade de novos conhecimentos para o pleno desenvolvimento das cadeias produtivas, valorizando a biodiversidade e os serviços ambientais dos ecossistemas.

A capacitação tecnológica provou ser uma ferramenta fundamental para manter as economias emergentes de importantes países em desenvolvimento. Nos últimos cinquenta anos, o Brasil tem sido capaz de criar ilhas de excelência em ciência e tecnologia, as quais são mais semelhantes às dos países desenvolvidos do que as de nações de baixa ou média renda [1].

No entanto, as desigualdades regionais históricas, especialmente as relacionadas à educação, criaram impedimentos, limitando drasticamente o uso intensivo da ciência e tecnologia para as economias e o desenvolvimento social das regiões mais pobres e menos favorecidas, incluindo a Amazônia e o Nordeste brasileiro [1].

A transformação trazida pelo desenvolvimento de novas tecnologias, pode ser um caminho eficaz para alinhar o desenvolvimento estratégico e sustentável à preservação ambiental. Um bom exemplo desse movimento é o conceito de Floresta 4.0 que consiste em uma ideia sobre como novas tecnologias estão impactando o setor de maneira direta e adjacente, o que deve ser visto com um olhar positivo sobre todo o sistema: desde as terras usadas para a produção até o uso dos produtos e serviços pelo consumidor final.

O conceito de Floresta 4.0 representa o uso de novas tecnologias para gestão dos ativos florestais: seu objetivo é trazer mais eficiência para o processo produtivo florestal em todas as suas etapas. Isto significa maior nível de automação dos equipamentos, maior geração de telemetrias (dados) para o monitoramento, controle e conectividade para uma tomada de decisão com maior agilidade e assertividade [2].

A ascensão da tecnologia blockchain representa um marco revolucionário no cenário global, oferecendo soluções inovadoras em diversos setores. No contexto da preservação da Amazônia, surge a perspectiva empolgante da aplicação da blockchain no conceito de Floresta 4.0. O conceito de descentralização apresentado inicialmente pelo Bitcoin [3] e

amplamente divulgado e desenvolvido por demais moedas digitais, como, por exemplo, o Ethereum [4], se demonstra cada vez mais um importante instrumento de transparência e inviolabilidade de dados para a sociedade.

A tecnologia proposta pelo blockchain não somente reduz a necessidade de intermediários que garantam a integridade de transações de um sistema proposto, mas também possui mecanismos que suprimem a perda de confiança e a possibilidade de fraude nas transações. O seu funcionamento reduz custos, tempo e burocracia e assimila transparência, segurança e rastreabilidade aos projetos.

Dentro desse contexto surge a Standing Forest Coin (SFC), uma proposta de moeda digital que promove inicialmente o financiamento e execução de projetos voltados à proteção do ecossistema amazônico, com mecanismos para assegurar o cumprimento desses projetos utilizando uma rede blockchain pública, com o propósito de tornar todas as transações e informações dos projetos e da moeda transparentes [5].

Um dos objetivos do projeto é permitir que as iniciativas de preservação de terra sejam apoiadas financeiramente de forma ágil, à medida que forem disponibilizados insumos comprovando a execução das etapas desses projetos e consequentemente a continuidade da preservação da floresta. Assim, frações da moeda serão liberadas para os projetos de conservação, conforme a conferência e análise das informações fornecidas por meio de um oráculo blockchain, aumentando a segurança e a credibilidade nos projetos e iniciativas fomentadas com a criptomoeda [5].

Esta pesquisa aborda uma solução que busca viabilizar o projeto da Standing Forest Coin e a execução de seus contratos inteligentes, através da aplicação de conceitos de oráculos blockchain. O objetivo é facilitar e potencializar a relação entre a instituição financiadora e o prestador de serviço ambiental, de tal maneira a gerar novas iniciativas e divulgar toda uma agenda de atividades que ajudem a proteger não só o ecossistema amazônico, mas também permita a abertura para o financiamento de atividades de conservação em outros ecossistemas brasileiros, como a Cerrado e a Caatinga.

1.1 Definição do Problema

As ONGs e instituições sem fins lucrativos têm dificuldade de arrecadar recursos mediante financiadores e doadores para execução de seus projetos, e essa dificuldade se dá pela falta de transparência com a maneira que esses recursos são aplicados em seus projetos. Assim, os interessados em investir perdem a confiança em doar para essas organizações, por não terem uma maneira de mensurar a efetividade da execução desses projetos [6].

Recentes escândalos de corrupção e mau uso de recursos levaram à diminuição da

confiança por parte de financiadores e possíveis doadores a projetos que visam a sustentabilidade ambiental. Dessa forma, os subsidiadores estão exigindo das instituições prestadoras de serviço ambiental maior transparência e responsabilidade, e essas organizações estão procurando formas de suprir essa necessidade.

É necessária a adoção de tecnologias que permitam que a sociedade acompanhe de forma transparente e acredite na inviolabilidade dos dados e nos projetos de financiamento, sem precisar confiar numa entidade centralizadora. Os projetos tendem a buscar estratégias que permitam, das mais variadas formas, aumentar a transparência no processo de gerenciamento de recursos de projetos, sejam eles públicos ou privados.

A tecnologia blockchain conceitualmente propõe uma garantia de confiança, transparência e rastreabilidade, e por via de mecanismos como contratos inteligentes e oráculos, permite a automação de regras de negócio, reduzindo a necessidade de intermediários e diminuindo consequentemente a possibilidade de fraudes nas transações.

A combinação do conceito de oráculos blockchain, aliada a abordagens e técnicas avançadas de aprendizagem de máquina e processamento digital de imagens, permitiram a elaboração de uma proposta de solução para monitoramento e análise das mudanças na cobertura florestal em uma das regiões mais críticas e sensíveis do planeta. Diante da necessidade de transparência nas etapas de investimento e administração dos recursos de organizações desses projetos que visem a sustentabilidade ambiental, se fez necessário a formulação das seguintes questões de pesquisa de (QP) para este trabalho:

- **QP1**: Como tornar mais transparente o processo de financiamento de projetos ambientais utilizando oráculos blockchain?
- **QP2**: Como tornar mais confiável a relação entre o financiador e o prestador de serviços ambientais através dos recursos fornecidos pelos oráculos blockchain?

A principal motivação deste trabalho é abordar as questões de pesquisa levantadas e propor soluções dentro do escopo da pesquisa abordado.

1.2 Premissas e Hipóteses

Baseado nas questões de pesquisa apresentadas no final da Seção 1.1, uma hipótese geral foi definida para o desenvolvimento do trabalho. Informalmente, ela pode ser definida da seguinte maneira:

- A partir da viabilização da Standing Forest Coin por meio de uma proposta de oráculo blockchain, capaz de monitorar a cobertura florestal de uma região alvo,

a relação entre o financiador e o prestador de serviços ambientais se tornará mais confiável e transparente;

A hipótese informal supracitada pode ser definida formalmente da seguinte forma:

- **Hipótese Nula H0**: A solução proposta não é capaz de identificar e distinguir de forma precisa e eficiente (Acurácia menor que 90 porcento) áreas preservadas e áreas desmatadas da região amazônica;
- **Hipótese Alternativa H1**: A solução proposta é capaz de identificar e distinguir de forma precisa e eficiente (Acurácia maior ou igual que 90 porcento) áreas preservadas e áreas desmatadas da região amazônica;

No Capítulo 5 será apresentado um projeto de experimentos desenvolvido para testar essa hipótese. Esse projeto inclui a definição, o planejamento e a execução do experimento para testar e avaliar a hipótese geral.

1.3 Objetivo Geral

O trabalho pretende potencializar a confiabilidade e a transparência durante o processo de financiamento de projetos, facilitando a relação entre o financiador e o prestador de serviços ambientais.

A segmentação semântica foi selecionada como a técnica central do processo de análise de imagens de satélite. Essa técnica permite separar áreas florestais das áreas desmatadas, possibilitando a identificação e quantificação das regiões afetadas pelo desmatamento com alta acurácia.

Consoante ao processamento das imagens, é imprescindível a utilização de redes neurais convolucionais como modelos computacionais para o processo de treinamento e aprendizado cujo viés é o reconhecimento de padrões complexos, identificando, monitorando e avaliando as mudanças na cobertura florestal em uma escala mais abrangente. Além disso, a adoção dos conceitos blockchain conferem um nível adicional de confiabilidade aos dados, assegurando sua integridade e imutabilidade.

Este trabalho propõe, dessa forma, uma solução de oráculo blockchain utilizando abordagens de processamento digital de imagens e aprendizagem de máquina, que permita o funcionamento da SFC, um ativo digital que proporciona a coleta descentralizada de recursos direcionados a iniciativas e atividades que visam proteger o ecossistema da Amazônia brasileira, garantindo a credibilidade, segurança e transparência para os financiadores e doadores de recursos investidos em projetos dentro da plataforma.

1.4 Objetivos Específicos

Para que a solução proposta tenha êxito, faz-se necessário, portanto, que os seguintes objetivos específicos sejam atingidos:

- Identificar aspectos relevantes para a análise de confiabilidade e transparência de soluções que utilizem conceitos blockchain no processo de investimentos em projetos sustentáveis;
- Verificar uso de técnicas de processamento digital de imagens, em especial, abordagens de segmentação semântica (binária) e arquiteturas de redes neurais convolucionais, que permitam a extração de características detalhadas das imagens de satélite e a identificação precisa das áreas desmatadas;
- Explorar a construção de um oráculo blockchain que viabilize o funcionamento da Standing Forest Coin (SFC), utilizando as estratégias técnicas citadas no item anterior;
- Realizar experimentos a fim de obter uma avaliação da arquitetura proposta para o oráculo;

1.5 Estrutura da Dissertação

A estrutura da dissertação segue o modelo recomendado pelo Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal da Paraíba (UFPB), sendo organizado em 7 capítulos correlacionados.

O Capítulo 1 apresenta uma introdução justificando a pesquisa, destacando a contextualização e a definição do problema. Nessa seção, também é apresentada as premissas e hipóteses elaboradas, assim como os objetivos da pesquisa.

O Capítulo 2 traz os termos, os fundamentos e os conceitos teóricos necessários para subsidiar a pesquisa.

O Capítulo 3 apresenta a revisão quasi-sistemática da literatura, na qual foram selecionados os trabalhos mais influentes consoante o tema desta pesquisa.

O Capítulo 4 descreve a solução proposta, com as estratégias que serão utilizadas para a realização do trabalho.

O Capítulo 5 mostra o planejamento dos experimentos a serem realizados, com o proposito de avaliar a solução proposta.

O capítulo 6 apresenta a execução do experimento conduzido, visando avaliar e apresentar os resultados obtidos, com as observações e discussões dos resultados.

O Capítulo 7 serão abordadas as conclusões deste trabalho.

2 CONCEITOS GERAIS E REVISÃO DA LITERATURA

Este capítulo apresenta os conceitos e termos fundamentais empregados nesta dissertação obtidos por via de uma revisão ad-hoc da literatura com a finalidade de dar sustentação teórica ao trabalho. Esses conceitos são essenciais para uma melhor compreensão dos resultados da pesquisa

2.1 Criptografia Simétrica

De acordo com Fraulkner [7], a criptografia simétrica é um algoritmo de encriptação de "dois caminhos" que utiliza a mesma chave para encriptar e descriptografar uma mensagem. Na teoria, tanto o emissor quanto o destinatário precisam conhecer a chave secreta, porém, na prática, sabe-se que é difícil compartilhar e manter a chave em segurança.

Para Oliveira [8], é o exemplo de criptografia mais antiga da ciência da computação, na qual ambas as partes devem manter uma chave em segredo. Essa chave pode ser chamada de senha, sendo usada pelo emissor para codificar a mensagem. O destinatário, utiliza a mesma senha para descriptografar a mensagem.

A lógica da criptografia simétrica retrata o seguinte: um emissor, chamado de A, criptografa um texto utilizando a chave secreta e um algoritmo específico. O receptor B utiliza o mesmo algoritmo e chave para descriptografar o texto com o objetivo de encontrar seu conteúdo original. O uso da chave privada em conjunto a encriptação faz-se necessário, pois, caso um intruso C saiba qual foi o algoritmo utilizado para encriptar o texto, a mensagem poderá ser manipulada. Conclui-se que, a segurança desse modelo de criptografia está na chave utilizada e não no algoritmo, desde que, os indivíduos A e B mantenham a chave em segredo. Toda mensagem enviada que utiliza essa técnica de criptografia é transmitida através de um canal seguro, que, por sua vez, possui uma determinada chave secreta [9].

Segundo Oliveira [8], a grande vantagem desta técnica é a simplicidade, uma vez que, é de fácil uso e os processos criptográficos são executados rapidamente. Destaca que quanto mais simples o algoritmo de encriptação e decriptação, maior é a facilidade de implementação e velocidade no processamento.

Na Figura 1 [10] o Emissor enviou uma mensagem encriptando-a com um algoritmo específico e com a sua chave secreta. O Receptor recebeu a mensagem e realizou o mesmo procedimento para obter o conteúdo original da mensagem. Um intruso somente conseguirá descriptografar a mensagem caso possua a mesma chave secreta do Emissor e Receptor, estabelecida pelo canal seguro.

Figura 1: Processo de encriptação e decriptação da criptografia simétrica

2.2 Criptografia Assimétrica

A criptografia assimétrica, também conhecida como criptografia de chaves públicas, permite que dois indivíduos troquem mensagens livremente entre si por meio de um par de chaves [11].

Neste conjunto de chaves privada e pública, a chave pública é gerada através da chave privada, portanto, fazer o caminho inverso e tentar alcançar a chave privada através da chave pública, é computacionalmente inviável. O processo para obter a chave pública é baseado em problemas matemáticos e atualmente não existe uma solução eficaz, visto o impasse do logaritmo discreto [12].

A chave privada deve ficar em posse de cada indivíduo, em segredo. Já a chave pública, deve ser disponibilizada na Internet. É através da chave pública que o indivíduo recebe a mensagem, e a partir de sua chave privada, a mensagem é descriptografada. Portanto, qualquer pessoa que possuir a chave privada de uma chave pública será capaz de ler a mensagem enviada ao endereço público [8].

De acordo com Oliveira [8], a grande vantagem em utilizar este tipo de criptografia é que qualquer pessoa pode enviar uma mensagem a um endereço público. A posse da chave privada é de responsabilidade do indivíduo, portanto, enquanto a chave privada estiver em segurança, a confidencialidade da mensagem é assegurada.

Na Figura 2 [10] o texto da mensagem é cifrado com a chave pública de Beto. Na sequência, Beto utiliza sua chave privada para descriptografar a mensagem. É importante lembrar que se outra entidade possuir a chave privada de Beto, a mensagem enviada também poderá ser descriptografada.

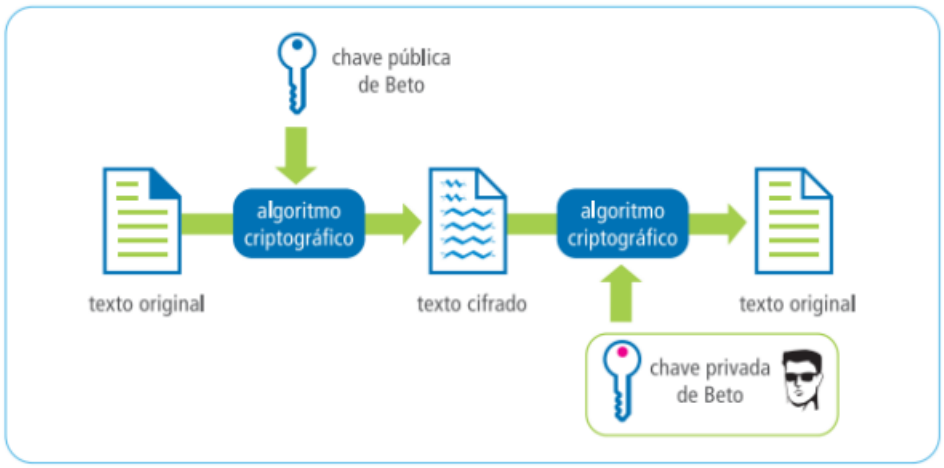

Figura 2: Envio de uma mensagem criptografada a uma chave pública

2.3 Funções *Hash*

Do inglês, *hash* ou *hashing*, é o ato de transformar um texto em uma sequência fixa de caracteres que representa o texto original. O processo de *hash* consiste em duas etapas. A primeira etapa é chamada de entrada e consiste em informar um texto na função *hash*, indiferente do tamanho ou quantidade de caracteres. Já a segunda etapa é chamada de resumo e é responsável pela criação da sequência fixa de caracteres conforme o texto original [13].

Uma das principais características da função *hash* é, por exemplo, determinar que um texto igual possui sempre o mesmo resumo/sequência de caracteres. Dado qualquer alteração no texto da entrada, ao executar novamente a função *hash*, o resumo obtido é totalmente diferente [11]. O resumo gerado através da função *hash* pode ser chamado de DNA, já que é único.

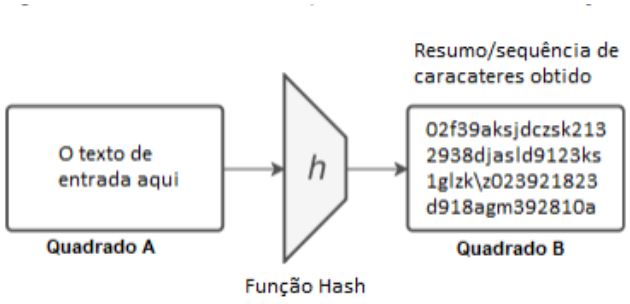

Figura 3: Procedimentos para executar uma função *hash*

Na Figura 3 [10] é ilustrado o processo de execução de uma função *hash* em um texto de entrada. O texto do Quadrado A gerou o resumo de caracteres do Quadrado B.

2.4 Certificado Digital

De acordo com Davis [14], um certificado digital é um pedaço de código responsável por identificar um indivíduo na Internet, criando uma espécie de "identidade digital".

Uma Autoridade Certificadora (AC) é responsável por atribuir uma chave pública a uma entidade ou pessoa, gerando um documento eletrônico com dados do titular e do órgão certificador emissor. A criptografia utilizada em certificados digitais é a criptografia assimétrica. Logo, um certificado digital contém os dados da AC emissora, o período de validade, política de utilização, nome e chave pública associada ao solicitante. No Brasil, o órgão responsável por gerenciar as chaves públicas é o ICP-Brasil [11].

Conforme a Figura 4 [10], o indivíduo Mario Rossi solicitou à autoridade certificadora um certificado digital. Uma chave pública foi concedida a "Mario Rossi" e a AC o certificou com uma data de validade.

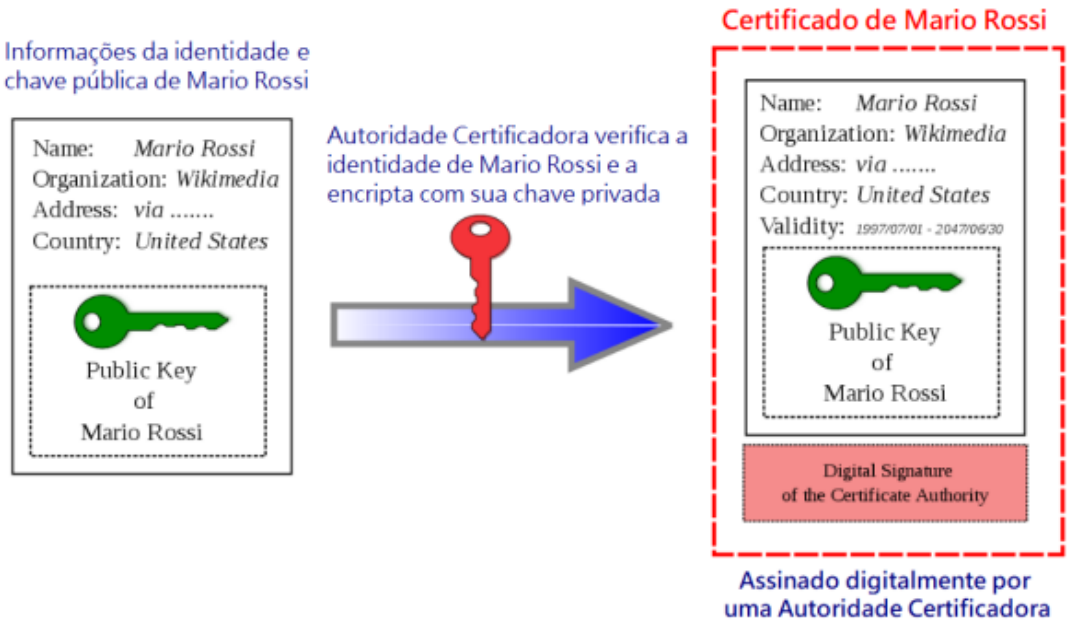

Figura 4: Processo de certificação

2.5 Assinatura Digital

Uma assinatura digital possui o mesmo propósito de uma assinatura física. Tem como objetivo assegurar que um documento ou mensagem foi de fato enviada pelo emissor, garantindo a integridade, autenticidade e principalmente a irretratabilidade do objeto em questão [15].

Para assinar digitalmente um documento, o indivíduo deve gerar o *hash* da mensagem através de uma função de *hash*. A chave privada do indivíduo é utilizada para encriptar o *hash* obtido do documento, gerando assim a assinatura digital. A assinatura

é anexada ao documento encaminhado, e a mensagem enviada às partes interessadas [12]. A Figura 5 [10], exemplifica este processo.

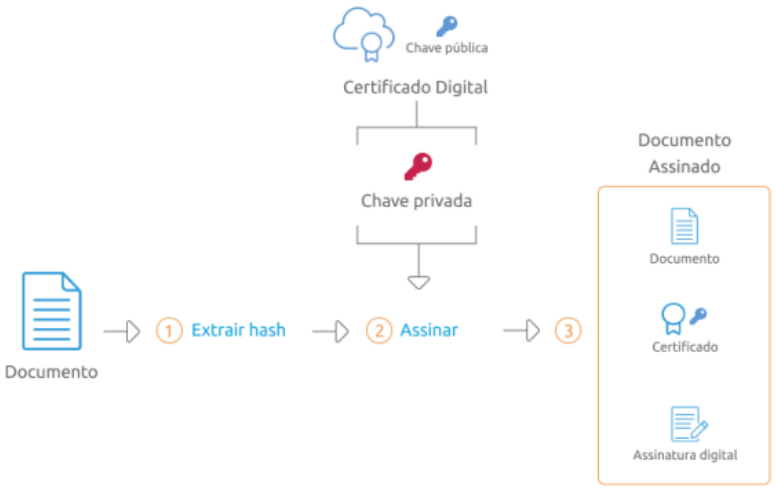

Figura 5: Processo para gerar a assinatura digital de um documento

Conforme a Figura 6 [12], as partes interessadas do documento podem verificar sua autenticidade encontrando o *hash* da mensagem, o *h1*. Para descriptografar a assinatura digital, é utilizada a chave pública do emissor da mensagem, o *h2*. Caso o *hash* do *h1* seja igual ao *hash* do *h2*, a mensagem é de fato do emissor e não houve nenhuma ação fraudulenta.

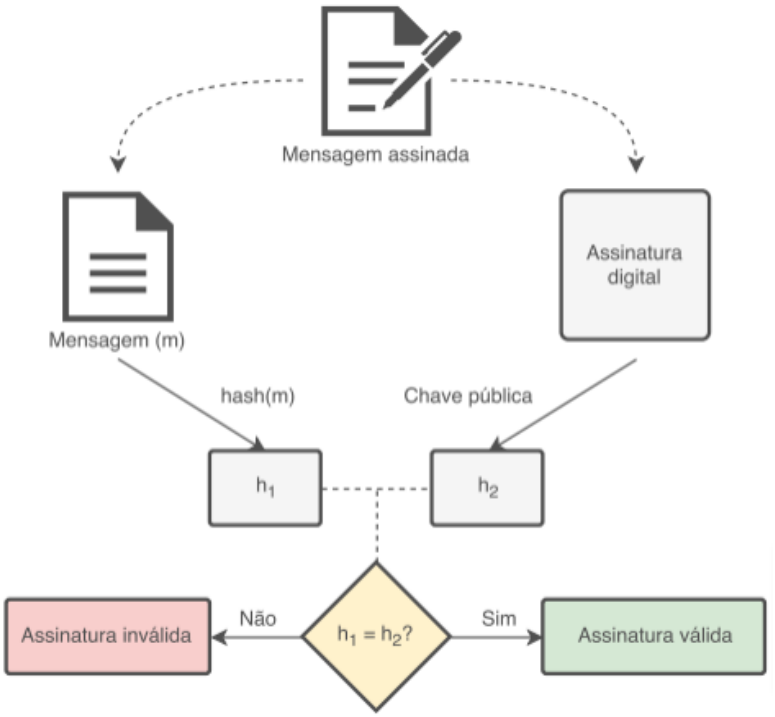

Figura 6: Processo para verificar a assinatura digital de um documento

2.6 Blockchain

Uma cadeia de blocos criptograficamente protegidos foi descrita pela primeira em um artigo de Haber e Stornetta [16]. O objetivo deles era implementar um sistema onde as estampas de tempo (*timestamps*) de documentos não pudessem ser adulteradas. No ano seguinte, eles adicionaram árvores de Merkle ao design do projeto, o que aumentou sua eficiência ao permitir que vários certificados para documentos fossem coletados em um bloco.

Tomando como base esse projeto, foi publicado em 2008, assinado sob o pseudônimo de Satoshi Nakamoto, o artigo Bitcoin [3]. Neste artigo é descrito o que se convencionou chamar de blockchain. As palavras *block* (bloco) e *chain* (cadeia) foram usadas separadamente no artigo original de Nakamoto, mas se popularizaram como uma única palavra.

Somente após a aplicação bem-sucedida no Bitcoin, o termo blockchain tornou-se amplamente popular, fazendo com que diversas criptomoedas alternativas surgissem, assim como muitas pesquisas relacionadas a sua utilização em outros cenários que vão muito além do mercado financeiro [17].

O conceito do termo blockchain é amplo e relativamente novo, sendo assim é possível encontrar diferentes definições para o mesmo. Analisando-o a partir de uma perspectiva técnica, Faour [18] o descreve como uma base de dados descentralizada funcionando sobre uma rede *peer to peer*. Já de um ponto de vista mais superficial, Swan [19] define a arquitetura, no contexto das criptomoedas, como um livro-razão que registra todas as transações ocorridas entre os participantes

De maneira geral, pode-se definir blockchain como um *ledger* (livro-razão) distribuído, *peer-to-peer* protegido criptograficamente, *append-only* (estritamente incremental), imutável, e atualizável apenas via consenso ou acordo entre os pares [20]. A Figura 7 [21] destaca a visualização da cadeia de blocos como uma lista encadeada de ponteiros *hash*.

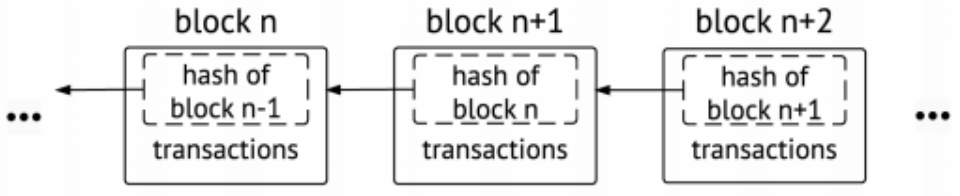

Figura 7: Ilustração de um blockchain e seus ponteiros *hash*.

O blockchain pode ser descrito como uma estrutura de dados imutável, onde novos registros podem apenas ser adicionados, e jamais removidos. Esta estrutura de dados é replicada e compartilhada entre os nós (também chamados *peers*) de uma rede, o que explica o termo rede blockchain [21]. A cadeia consiste em uma sequência de blocos de

informação onde o bloco *n* é criptograficamente ligado ao bloco *n - 1* por meio de uma função *hash*. Consequentemente, o bloco *n* não pode ser modificado sem que todos os blocos subsequentes *n + i, ..., n + k* também o sejam [3].

2.7 Estrutura Blockchain

Rede P2P Blockchain

Em uma rede P2P, os usuários são responsáveis por utilizar e fornecer recursos simultaneamente. Cada *peer*, conhecido como nó, disponibiliza uma parte do seu disco rígido (HD), poder de processamento e de Internet (download/upload) diretamente para outros nós, sem a necessidade de uma entidade central ou servidores estáveis [22]. Apesar dos nós da rede serem considerados iguais, há alguns que diferenciam-se por desempenhar funções adicionais [12].

Uma conexão P2P é completamente diferente do modelo tradicional de cliente-servidor, pois não há um ponto central de armazenamento. Outro diferencial em relação ao modelo tradicional, é que na rede P2P a informação é constantemente registrada e distribuída entre os nós participantes. Como a informação não é armazenada em um único local, este aspecto é um avanço para o armazenamento de dados na Internet, já que, um ataque malicioso precisaria ser realizado em todos os nós da rede. Caso a informação seja perdida em alguns nós, basta que um possua os dados do respectivo blockchain, assim, os nós afetados pelo ataque podem baixar novamente as informações para ter novamente a cópia original do blockchain [22].

O surgimento e o papel que a rede P2P desempenha em conjunto com a tecnologia blockchain pode ser visto como um novo sistema de comunicação, totalmente seguro e totalmente distribuído. Mesmo que os nós da rede fiquem "visíveis" durante os processos de transações ou mineração de um bloco, a identidade e toda a informação dos participantes são mantidas em sigilo através da criptografia [12].

Blocos

Um bloco é uma estrutura de dados que armazena um conjunto de transações e informações da rede blockchain. Em cada transação registrada no bloco, há informações sobre a data, o valor monetário referente a criptomoeda e o emissor que realizou a transferência. Possui, também, um identificador único, uma espécie de "impressão digital" que permite que todo bloco seja reconhecido pela rede [23].

O identificador único é armazenado no cabeçalho do bloco sendo definido pela execução dupla do algoritmo *Hash SHA-256* sobre os dados de entrada (*version, previous block hash, merkle root, timestamp, difficulty target, nonce*), formando o que é chamado de *block hash*. Além de salvar o seu identificador, o bloco atual salva o identificador do

seu bloco anterior, entendido como bloco pai, formando a cadeia de blocos "*block chain*" na qual se chega ao bloco gênese [12].

A Estrutura do cabeçalho de um bloco da rede é detalhada na Tabela 1 :

Tabela 1: Estrutura do cabeçalho de um bloco da rede

Version	Número usado para rastrear a versão do protocolo.
Previous Block Hash	Identificador do bloco anterior.
Merkle Root	Um resumo de todas as transações inseridas no bloco.
Timestamp	Hora aproximada em que o bloco foi criado.
Difficulty Target	Dificuldade estabelecida por um determinado algoritmo para um novo bloco ser minerado.
Nonce	Contador usado no cálculo da mineração de um bloco.

De acordo com Sousa [24], um bloco da rede Ethereum possui campos com outras nomenclaturas, mas que remetem ao mesmo funcionamento do Bitcoin. Há também campos adicionais, como, por exemplo, o *gasLimit* e *gasUsed*, onde ambos contabilizam informações referentes ao *gas* (detalhado nas subseções seguintes) utilizado na mineração do bloco.

A Figura 8 [12] exemplifica o encadeamento formado pelos blocos. O *hash* que é calculado duas vezes é representado no cabeçalho do bloco. Acima do cabeçalho, há um contador que identifica a posição em que o bloco foi criado (identificado como *block height*), porém, esse contador não faz parte do identificador único. O bloco gênese é o bloco de posição zero. O campo *previous block hash* faz referência ao identificador único do seu bloco anterior.

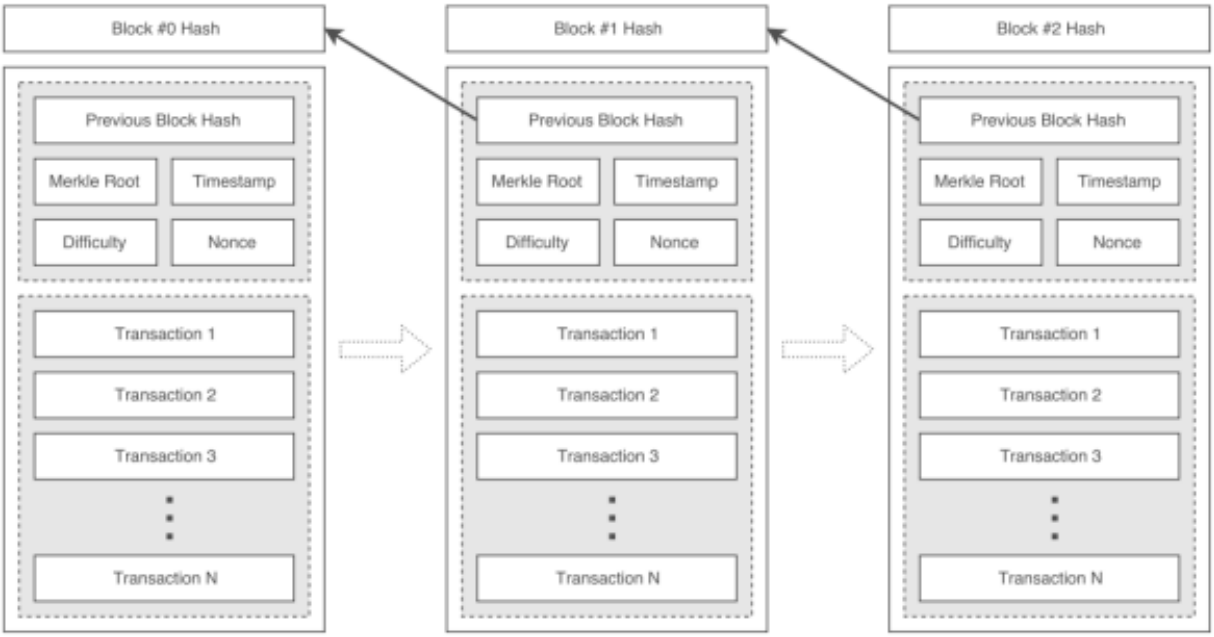

Figura 8: Encadeamento dos blocos

Árvores de Merkle

Denomina-se de bloco gênesis o primeiro bloco da lista. Cada bloco contém um conjunto de transações. Elas são organizadas numa estrutura de árvore binária de ponteiros *hash*, geralmente numa estrutura denominada árvore de Merkle (ou alguma variação dela).

As árvores de Merkle ou árvores de dispersão são árvores binárias utilizadas para agrupar e verificar de maneira eficaz a integridade de volumosos conjuntos de dados. No protocolo Bitcoin, as árvores são usadas para gerar um *hash* único, conhecido como *merkle root*. Na prática, cada transação do bloco é resumida pela função *Hash SHA-256*. O resumo, então, é concatenado a um resumo de outra transação, sempre em pares, no formato de árvore binária. Dado a estrutura criada se assemelhar a uma árvore, as transações iniciais são chamadas de "*leaves*" (folhas), as transações do meio de "*branches*" (galhos) e o resumo *hash* do topo, de "*root*" (raiz) [12]. Buterin [4] explica que o funcionamento das árvores no Ethereum segue os mesmos conceitos do Bitcoin.

Como mostra a Figura 9 [25], no último nível, estão as folhas que contêm os dados (especificamente, apontadores para as transações propriamente ditas). No penúltimo nível, os pais possuem apontadores *hash* para esses dados. Em seguida, em cada nível, os pais apontadores são agrupados dois a dois, até que se alcance a raiz da árvore. O *hash* da raiz da árvore é então armazenado de forma segura dentro do *header* do bloco, em conjunto com demais informações. Pela propriedade dos apontadores *hash*, se houver qualquer modificação nos dados da árvore, ela será identificada pela verificação dos *hashes* [26].

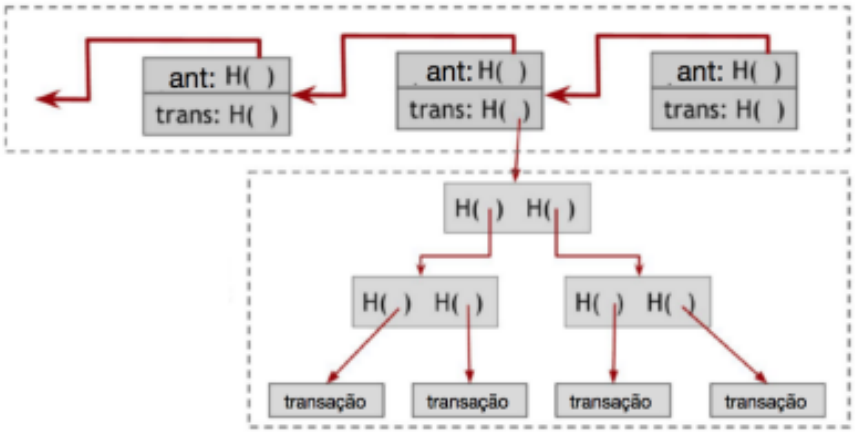

Figura 9: Blockchain com árvore binária Merkle

2.8 Tipos de Blockchain

Existem múltiplos tipos de blockchain, dependendo do modelo de negócio no qual a tecnologia é aplicada. Muitas vezes as definições não são claras ou até mesmo redundantes. Por isso serão descritos nesta subseção apenas os três tipos mais abrangentes.

- Blockchain Público - Esses blockchains estão abertos ao público e qualquer um pode participar enviando transações, assim como, contribuindo no processo de consenso [20]. Eles não possuem qualquer restrição de acesso. Nele ninguém é dono do *ledger* (livro-razão) e geralmente a rede atrai participantes oferecendo incentivos financeiros para aqueles que a mantiverem, tal como será descrito na subseção referente a protocolos de consenso a seguir.

- Blockchain Privado - Este tipo de blockchain exige permissão para acessá-lo e uma empresa detém o controle da rede, como o nome implica. Apenas pessoas convidadas pelos administradores da rede podem se juntar a ele [20]. Este tipo de blockchain é usado por empresas que querem aproveitar as vantagens do sistema distribuído, mas não se sentem confortáveis com o baixo nível de controle das redes públicas.

- Blockchain Federado ou de Consórcio - Este é um tipo de blockchain que exige permissão para participar, porém, não é controlado por uma entidade única [27]. Diferentes organizações compartilham a mesma rede e possuem igual poder de decisão. É considerada uma rede semi-descentralizada. Além disso, podem ser parcialmente abertos ao público, porém garantindo privilégios às organizações participantes do "consórcio".

2.9 Protocolos de Consenso

Consenso é um processo de acordo entre nós, sem confiança entre si, sobre o estado final dos dados. Para atingir o consenso, diferentes algoritmos podem ser usados. É fácil alcançar um acordo entre dois nós (por exemplo, em sistemas cliente-servidor) mas quando múltiplos nós estão participando em um sistema distribuído e eles precisam concordar sobre um certo valor, se torna muito difícil atingir um consenso. Este conceito de alcançar um consenso entre múltiplos nós é conhecido como consenso distribuído [20].

Para um bloco ser inserido no blockchain, todos os nós seguem regras de uma prova de consenso em comum. A prova estabelecida busca suprir os problemas de corretude, acordo e utilidade. Por meio do aspecto de corretude, os nós verificadores devem conseguir diferenciar uma transação maliciosa de uma transação legítima. Através do acordo, a cadeia de blocos deve-se manter confiável e única, e a aplicação da solução disponibilizada pelo blockchain deve ser útil, não demorando na verificação de transações e mineração

de blocos. Por exemplo, se um blockchain leva mais de um dia para concluir uma única transação, seu uso não é prático, portanto, a solução não é eficaz [28].

É a partir de uma prova de consenso, estabelecida como regra da rede, que os nós respondem à pergunta "Como garantimos com total certeza o que é verdadeiro e o que é falso?". O consenso estabelecido pelos nós da rede garante o mesmo estado de todo o blockchain, tornando-o um ecossistema independente de auditoria. As principais funções de uma prova de consenso são: manter a cadeia de blocos sempre atualizada, garantindo que todo novo bloco seja verdadeiro, enquanto, simultaneamente, incentiva os indivíduos da rede a realizarem tal ação; verificar se uma nova transação não foi forjada; impedir que uma única entidade controle ou prejudique toda a cadeia; considerar o novo bloco ou transação verdadeira, caso obtenha mais de 51 por cento de unanimidade, mesmo que alguns nós falhem em concordar com determinado consenso; evitar problemas de gasto duplo e dos generais bizantinos [3].

Segundo Martins [12], o problema dos generais bizantinos (*Byzantine Fault Tolerance*) foi desenvolvido para solucionar falhas em sistemas distribuídos, onde um dos problemas consiste em tentar fazer com que todos os atores obtenham sucesso na troca de informações em uma rede comprometida, ou seja, a informação distribuída deve ser legítima mesmo que na rede, exista atores tentando inserir informações forjadas. Por meio deste problema, pode-se usar como analogia uma rede P2P, na qual um nó malicioso tenta inserir milhares de blocos e transações com informações falsas, com o objetivo de prejudicar a rede ou realizar gastos duplos, duplicando sua quantidade de moedas indevidamente.

Pela característica de descentralização do blockchain, não é necessário haver uma autoridade central responsável por gerenciar as criptomoedas dos usuários da rede, papel esse atribuído aos nós. Todavia, um indivíduo pode tentar duplicar suas moedas realizando duas transações com a mesma quantia, neste caso, os nós responsáveis entram em um consenso para verificar qual das transações deve ser considerada válida e qual deve ser descartada. Atualmente existem vários algoritmos de consenso, como, por exemplo, o algoritmo conhecido como *Proof of Work* ou prova de trabalho, o primeiro a solucionar esta falha com eficácia. Posteriormente outras provas de consenso foram surgindo, buscando solucionar problemas evidenciados em consensos já existentes [29].

Os principais algoritmos de consenso são:

- *Proof of Work* (Prova de Trabalho) - Este é o algoritmo de consenso mais usado, sendo inaugurado no lançamento do Bitcoin. Ele é uma prova de que recursos computacionais suficientes foram gastos para se produzir um bloco válido. Neste modelo, os nós, chamados também de mineradores, competem para resolver um problema criptográfico. Este problema é encontrar um valor que, quando concate-

nado com os outros dados do bloco (transações, *timestamps*, *hash* do bloco anterior, etc.) na entrada de uma função *hash*, gere uma saída menor que um valor alvo estipulado pelo sistema [20]. Este valor é procurado a partir de um campo do bloco chamado *nonce*, que vai sendo incrementado até encontrar um valor que satisfaça essa condição. Quando a condição é satisfeita, um novo bloco é minerado e o nó que o minerou é recompensado. Transações são selecionadas, processadas e, enfim, armazenadas nesse novo bloco. Em seguida, o novo bloco é transmitido para todos os nós do blockchain.

O esforço computacional para a resolução do problema é exponencial em relação ao número de zeros à esquerda do alvo e pode ser verificado com uma única execução da função *hash* [3]. No caso do Bitcoin, a função é a *Hash SHA-256*. Para se obter uma taxa constante de geração de blocos, o valor do alvo (*target*) pode ser aumentado ou diminuído de acordo com o número de nós, tornando-se mais difícil quando há muitos mineradores e mais fácil quando há poucos.

Caso ocorra de haverem duas cadeias de blocos diferentes, a menor cadeia será substituída pela maior [3]. Isso pode acontecer quando dois nós mineram um mesmo bloco com pouca diferença de tempo entre os dois. Essas duas cadeias permanecerão em paralelo até que uma fique maior que a outra. Note que, como a criação de um valor de *hash* válido envolve os dados do bloco e o *hash* do bloco anterior, para modificar um bloco no meio da cadeia, um ataque malicioso teria que refazer o *Proof of Work* do bloco e de todos os blocos depois dele e ainda alcançar ou ultrapassar a cadeia com blocos honestos.

- *Proof of Authority* (Prova de Autoridade) - Em redes baseadas em *Proof of Authority*, transações e blocos são validados apenas por nós aprovados, conhecidos como validadores. Nesse processo, um nó validador é escolhido aleatoriamente para gerar um novo bloco onde ele coleta as transações que farão parte do bloco e as executa. Transações que são inválidas ou que não podem ser executadas são rejeitadas, como, por exemplo, uma transação que tenta gastar mais moedas do que o usuário tem. Em seguida, ele assina o bloco com a sua chave privada e envia o bloco para os outros validadores. Para validar o novo bloco, os outros validadores checam a assinatura do criador do bloco e executam as transações para verificarem se elas são válidas, ou seja, se elas são executadas com sucesso. Se a validação do bloco é bem-sucedida, ele é adicionado ao blockchain, caso contrário, ele é rejeitado e a reputação do criador do bloco é afetada negativamente.

  É necessário garantir que os nós participantes nesse processo de consenso não sejam maliciosos. Com o *Proof of Authority*, os indivíduos ganham o direito de se tornarem validadores, portanto, há um incentivo para manter a posição que conquistaram. Ao associar uma reputação à identidade, os validadores são incentivados a manter

o processo de transação, pois não desejam ter suas identidades associadas a uma reputação negativa, correndo o risco de serem banidos.

- *Proof of Stake* (Prova de Participação) - Em um sistema de *Proof of Stake*, um validador de transação (isto é, um nó na rede) deve provar que possui um certo ativo para participar da validação das transações [30]. No caso das criptomoedas, esse ativo é a quantidade de moedas, então, dependendo de quantas moedas o participante possui, ele terá uma chance maior ou menor de ser aquele que irá validar o próximo bloco. Cada nó tem uma chance proporcional à sua participação monetária no sistema. Dessa forma, um nó com 300 moedas terá 3 vezes mais chances de ser escolhido que um nó com 100 moedas. As partes envolvidas na transação pagam uma taxa ao validador da transação pelos seus serviços de validação. Existem ainda algumas variações deste algoritmo, onde o critério é a idade das moedas dos nós, quanto mais antigas, maiores as chances de ser escolhido para a validação.

- *Delegated Proof of Stake* (Prova de Participação Delegada) - É uma variação do *Proof of Stake* onde cada nó que possui ativos no sistema pode delegar a validação de uma transação para outros nós por meio de votação [20]. Ou seja, neste algoritmo os participantes não decidem diretamente se um bloco é válido ou não, eles votam em "delegados"que, por sua vez, fazem a validação.

- *Proof of Activity* (Prova de Atividade) - Este é um protocolo híbrido que combina *Proof of Work* com *Proof of Stake* [31]. Neste protocolo a mineração começa como um algoritmo *Proof of Work*, com mineradores disputando para resolver um quebra-cabeça criptográfico. Dependendo da implementação, os blocos extraídos não contêm nenhuma transação (funcionando como um *template*), portanto, o bloco vencedor conterá apenas um cabeçalho e o endereço de recompensa do minerador.

  A partir disso, o sistema muda para um algoritmo *Proof of Stake*. Com base nas informações do cabeçalho, um grupo aleatório de validadores é escolhido para assinar o novo bloco. Quanto mais moedas no sistema um validador possui, maior a probabilidade de ele ser escolhido. O novo bloco se torna um bloco completo assim que todos os validadores o assinam. Se alguns dos validadores selecionados não estiverem disponíveis para concluir a validação do bloco, o próximo bloco vencedor será selecionado, um novo grupo de validadores será escolhido e assim por diante, até que um bloco receba a quantidade correta de assinaturas. As taxas são divididas entre o minerador e os validadores que assinaram o bloco.

2.10 Bitcoin

As criptomoedas são ativos digitais que operam sob uma rede completamente descentralizada, independente de uma terceira parte confiável. Para verificar e garantir a segurança das transações, uma das principais técnicas utilizadas é a criptografia.

O Bitcoin, atualmente a mais conhecida das moedas digitais, introduziu um novo conceito de forma de transação comercial e propôs, em seus primeiros artigos, uma revolução na monetarização da sociedade mundial, causando potenciais impactos nas indústrias, finanças, governos, mídias e legislação, ou seja, no atual conceito de estrutura econômica e financeira na sociedade contemporânea.

O Bitcoin é uma moeda digital que utiliza um protocolo de comunicação descentralizado *open source*, que permite realizar transações financeiras. Vem crescendo desde o seu lançamento, logo após a crise econômica mundial de 2007-2008 e tem como principal objetivo ser um livro razão de transferências [32].

Segundo Corrêa [11], as transações são realizadas por meio de carteiras digitais. Dois indivíduos podem trocar criptomoedas livremente entre si com auxílio de suas chaves privadas e públicas. O Bitcoin é o primeiro protocolo criado totalmente descentralizado e fora do controle de uma instituição central ou de governo. Quem controla ou regula a criptomoeda são os nós espalhados pela rede P2P (*peer-to-peer*). A rede também garante o anonimato dos participantes e a transferência segura das informações.

2.11 Carteira, Endereços e Criptografia de Curvas Elípticas

Os Bitcoins (ativos da criptomoeda) de um indivíduo são gerenciados através de uma chave privada e pública, formando o que chama-se de carteira (*wallet*). As chaves são as "senhas" da carteira, portanto, quem possuir a chave privada da carteira poderá transferir Bitcoins livremente. Já a chave pública é utilizada para receber Bitcoins [12].

A criação de um endereço de Bitcoin dá-se a partir da sua chave privada que é composta por uma sequência de 256 bits. Por exemplo, pode ser um texto de 256 uns ou zeros (32 * 8 = 256), um texto binário, um texto de Base64, um texto qualquer, entre outros. A chave pública é gerada através da chave privada e utiliza o padrão Secp256k1. Secp256k1 é uma notação específica que descreve um conjunto de parâmetros para uma curva elíptica [33].

Para ter-se o endereço final utilizado como chave pública, alguns procedimentos são necessários. Primeiro a chave pública obtida através da chave privada é inserida e executada como entrada em duas funções *hashes*. A primeira é a função *Hash SHA-256* que gera uma sequência de 32 bytes e a segunda é a função RIPEMD-160, obtendo um endereço de 20 bytes em cima do conteúdo de 32 bytes. Como procedimento final, o

endereço de 20 bytes é codificado através do formato de codificação Base58Check, que é um subconjunto do Base64 e um *checksum* adicional, fornecendo proteção a erros de digitação [12]. O algoritmo de codificação Base58, derivado do subconjunto Base64, e compõem o Base58Check, elimina os caracteres "+", "/", "I" (letra "i" maiúscula), "O" (letra "o" maiúscula), "l" (letra "l" minúscula), e "0" (numero "0"), pois, possuem similaridade entre si [34].

A Figura 10 [10], exemplifica esse processo. Por meio de uma chave privada qualquer, a chave pública é obtida executando a multiplicação da curva elíptica. O resultado parcial então é codificado com os algoritmos *Hash SHA-256*, RIPEMD-160 e Base58Check. O resultado é a chave pública, que pode ser disponibilizada pela Internet para o indivíduo receber transações.

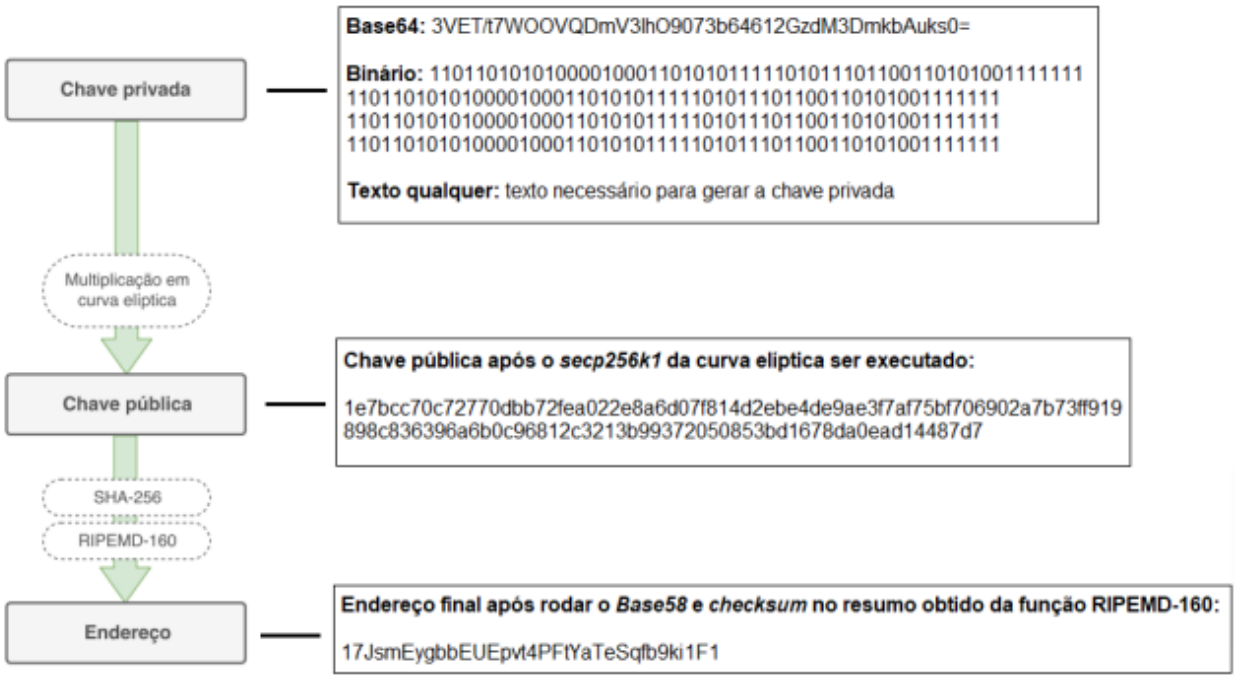

Figura 10: Processo de criação de um endereço Bitcoin

2.12 Transações

No Bitcoin, cada transação possui uma ou mais entradas, referente a origem dos fundos recebidos, e uma ou mais saídas, que especifica o destino para qual o valor será transferido. Novas transações são armazenadas em um conjunto chamado de UTXO, que significa "saída de transações não gastas" ou do inglês, "*Unspent Transaction Outputs*". O saldo de um indivíduo é somente visualizado na camada de aplicação, buscando facilitar a utilização do sistema. Na realidade, os fundos são salvos no conjunto UTXO, sendo que, a chave privada de cada usuário é a "senha" para desbloquear o saldo de determinada transação não gasta, habilitando a transferência das moedas disponíveis. Assim, o saldo de

uma carteira Bitcoin é definido pelo total de transações não gastas existentes no conjunto UTXO que respectivo endereço possui para gastar [12].

A Figura 11 [10] exemplifica a formação do conjunto UTXO. Para compor as entradas da Transação 12, a saída da Transação 1 e Transação 2 são utilizadas. O valor restante da Transação 1 e os fundos adicionados à Transação 12 podem ser utilizados para a entrada de outras transações futuras.

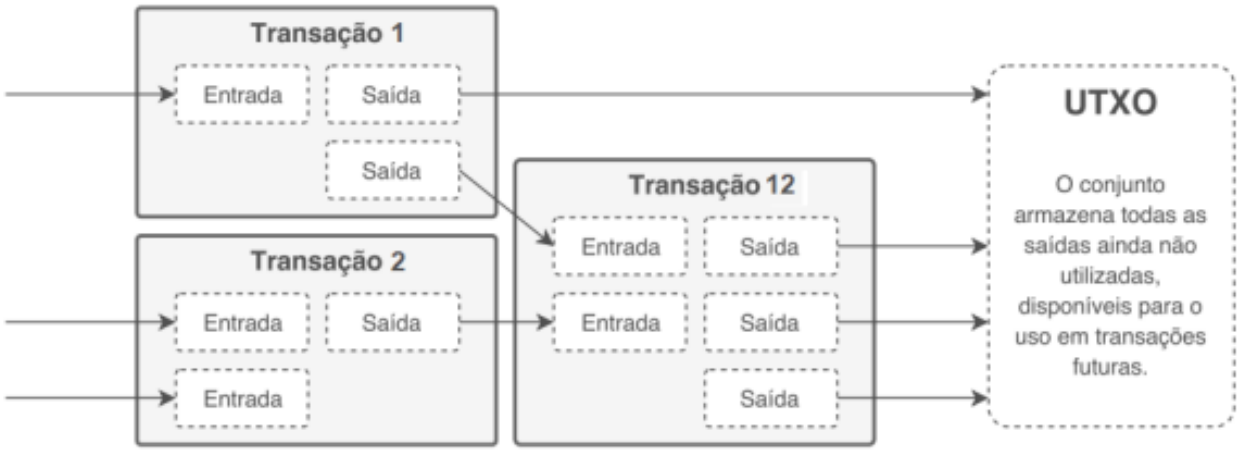

Figura 11: Formatação do conjunto UTXO

2.13 Mineração

Segundo Antonopoulos [29], a criação de cada novo bloco se dá pelo processo popularmente conhecido como mineração, onde os vários nós participantes da rede competem pelo direito de adicionar o novo bloco à cadeia e ganhar uma pequena bonificação por seu trabalho. Durante este processo, o minerador precisa utilizar um algoritmo de consenso, no caso do Bitcoin e do Ethereum, o algoritmo *Proof of Work*, para encontrar o *hash* de identificação do bloco e assim poder submetê-lo à aprovação dos outros nós através do mecanismo de consenso.

O processo de mineração e tudo que ele engloba é um dos pontos fundamentais para que o Bitcoin e o Ethereum sejam consideradas redes seguras, evitando transações fraudulentas ou inválidas. Segundo Swan [19], os mineradores fornecem seu poder de processamento computacional em troca da possibilidade de eventualmente serem recompensados com novas unidades das respectivas criptomoedas. Antonopoulos [29] menciona que esse mecanismo é chamado assim, pois as recompensas diminuem com o tempo, da mesma forma que geralmente acontece com a mineração de metais preciosos.

De acordo com Antonopoulos [29], sempre que uma transação é realizada entre duas carteiras, ela é transmitida para todos os nós mineradores da rede, que por sua vez a adicionam em um *pool* de transações pendentes. Cada minerador ordena estas transações

considerando prioridade e tempo de espera, para então agrupar um conjunto delas em um novo bloco. Uma vez que se tenha organizado as transações e gerado um bloco no formato esperado, o próximo passo é encontrar um *hash* de identificação para o bloco através do algoritmo de *Proof of Work*.

Uma função de *hash* criptográfica sempre gera a mesma saída para uma mesma entrada, portanto para que o *hash* gerado difira da tentativa anterior, um valor do *header* do bloco é alterado, e este valor é chamado de *nonce*. Ou seja, em sua maioria, o processo de mineração consiste em encontrar um *nonce* que resulte em um *hash* menor que o *target* (valor limite estipulado de caracteres alfanuméricos) definido pela rede, e como o resultado de uma função de *hash* criptográfica é imprevisível, a única forma existente é por via de tentativa e erro. Segundo Antonopoulos [29], a dificuldade para encontrar um valor de *nonce* que atenda o *target* estipulado é definida pela própria rede como forma de controlar o nível de dificuldade para geração de blocos.

O primeiro minerador que encontrar um *hash* válido segundo as especificações da rede e publicá-lo será o vencedor e como recompensa terá o direito de incluir no bloco uma transação conhecida como *generation transaction* ou *coinbase transaction*. A Figura 12 [35] exemplifica a estrutura de blocos minerados e concatenados através das funções de *hash*.

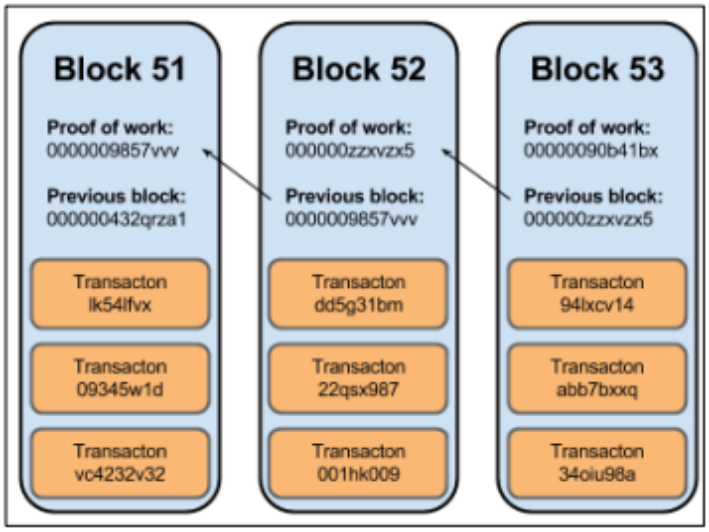

Figura 12: Lista encadeada de blocos

De acordo com Antonopoulos [29], quando um novo bloco é minerado, os outros nós realizam uma série de validações antes de propagá-lo para garantir que o bloco gerado é válido e atende a uma série de requisitos como: estrutura sintática do bloco está correta; o *hash* gerado é alfa numericamente menor que o *target* estipulado (*Proof of Work*); o tamanho do bloco está dentro do limite; a primeira transação é uma *generation transaction*, e somente a primeira; todas as transações incluídas no bloco são válidas.

2.14 Ethereum

Todo o furor causado pelo Bitcoin fez com que eventualmente a utilização do blockchain na solução de outros problemas fosse discutida. Porém, segundo Koç et al. [17], a rede criada por Nakamoto não foi projetada com o intuito de suportar outras aplicações, tendo assim poucos recursos para isso. Diante disto algumas alternativas de plataforma surgiram, dentre elas a que mais se destacou foi o Ethereum.

O Ethereum é uma plataforma de computação distribuída que usa blockchain para armazenar não somente o estado das contas de seus usuários, mas também código-fonte e seu estado associado [36]. Sendo proposta em 2013 por Vitalik Buterin, um antigo programador envolvido no projeto do Bitcoin, a plataforma visa permitir que qualquer um possa criar e executar aplicações distribuídas baseadas em blockchain, tendo assim acesso a todas as características inerentes a arquitetura, porém sem a complexidade de criar sua própria rede [37].

Segundo Schüpfer [36], no contexto do Ethereum, as contas são objetos que podem armazenar o balanço de um usuário ou o próprio estado de um contrato inteligente (*smart contract*), cuja estrutura será discutida no tópico a seguir. A primeira é chamada de *externally owned account* ou conta externa, pois pertence a um usuário externo, permitindo a ele enviar mensagens e realizar transações assinadas com sua chave privada. Já a segunda é chamada de *contract account* ou conta de contrato, e seu estado é controlado pelo código-fonte de um contrato. Sempre que uma conta contrato recebe uma mensagem, seu código-fonte é executado e tem permissão para enviar mensagens a outros contratos ou realizar transações com outras contas.

A unidade de moeda corrente no Ethereum é o Ether. Além de ser movimentada através das transações, ela também é utilizada como taxa de pagamento para a criação ou execução de algum contrato. Assim como o Bitcoin, o Ether pode ser obtido em casas de câmbio de criptomoedas, conhecidas como *cryptocurrency exchanges*, e pode ser dividido em partes menores, sendo a mais comum delas o Gwei [36].

Com intuito de evitar ataques de negação de serviço, assim como outros ataques de spam, o Ethereum utiliza um mecanismo chamado de *gas* (abreviação de gasolina em inglês). Essa unidade é usada internamente pela plataforma e não é propriamente uma moeda [37]. Ainda segundo o autor, cada unidade de *gas* representa uma computação realizada por um contrato, ou seja, quanto mais complexo seu código-fonte, maior será a quantidade de *gas* necessária para executá-lo.

Da mesma forma que no Bitcoin, as transações no Ethereum realizam a transferência de uma quantidade de Ether de uma conta para outra, mas, além disso, podem criar contratos ou executar funções de um já existente. Segundo Schüpfer [36], elas são um pacote de dados contendo informações do remetente, destinatário, quantidade de Ether,

limite e preço do *gas*, além de um campo opcional de dados. Existe também a possibilidade de que um contrato envie uma mensagem para outro invocando uma de suas funções, o que funciona de forma muito parecida com a chamada de uma função em uma linguagem de programação de alto nível.

2.15 Contratos Inteligentes

De acordo com Koç et al. [17] o Ethereum se distingue do Bitcoin principalmente por fazer uso do conceito de *smart contracts*. Estes contratos inteligentes são um tipo especial de conta, que além do balanço em Ether, armazena também código-fonte e seu estado associado. Segundo Schüpfer [36], outros usuários interagem com o *smart contract* enviando Ether através de transações ou executando alguma função.

Idealizado por Szabo [38], um contrato inteligente é um código determinístico executado por computadores distintos que concordam com o resultado da execução. Assim, os contratos inteligentes operam como atores autônomos, cujo comportamento é completamente previsível [21]. Os contratos inteligentes são usados para permitir interações entre partes que não confiam umas nas outras, realizando transações, sem depender de terceiros. Depois que as partes envolvidas aceitam o contrato inteligente, ele é executado pelos computadores. A partir daí, as partes envolvidas não podem alterar o contrato inteligente nem interferir na sua execução.

A tecnologia blockchain provê a infraestrutura sobre a qual rodam os contratos inteligentes [20]. Ela é usada para armazenar e endereçar contratos inteligentes na forma de *scripts* executáveis e imutáveis. Ao invés da identificação por chave pública, um contrato inteligente é geralmente identificado por um *hash*. Uma transação ligada a um contrato inteligente aciona sua execução pelos participantes da rede *peer to peer*. Enquanto está sendo executado, um contrato inteligente pode mudar seu estado e originar novas transações. O histórico de estados de um contrato inteligente, bem como as transações que ele recebeu e enviou, são permanentemente armazenados no blockchain.

Nem todas as redes blockchains foram projetadas para suportar contratos inteligentes, já o Ethereum fornece um ambiente de execução *Turing-Completo* chamado *Ethereum Virtual Machine* (EVM), que permite qualquer pessoa escrever um contrato inteligente com suas próprias regras arbitrárias e com impossibilidade de alteração após implementado e executado [39]. Para facilitar a elaboração de contratos inteligentes, linguagens de alto nível que compilam essas instruções específicas de *bytecode* podem ser usadas, como, por exemplo, o Solidity, uma linguagem orientada a objetos influenciada pelas linguagens Python, C++ e JavaScript e estruturada para ser compatível com a EVM. Os contratos inteligentes baseados em Solidity contêm funções e variáveis. Por exemplo, o construtor inicializa variáveis dentro do contrato inteligente e é executado uma vez que o contrato

inteligente é implantado, isto é, criado no blockchain.

Para proteger o blockchain do Ethereum de abusos e recompensar a rede pela execução de contratos inteligentes, são aplicadas taxas. O endereço que deseja ter um contrato inteligente executado deve pagar as taxas de execução. Essas taxas são definidas em termos de *gas*. O *gas* é a unidade de custo computacional do Ethereum [4]. As taxas de execução começam a partir de 2100 unidades de *gas*. Além disso, é preciso pagar por todas as instruções a serem executadas no contrato inteligente. Por exemplo, executar uma instrução de adição custa cinco (5) *gas*. Os usuários do Ethereum podem pagar com Ether e quanto mais Ether for oferecido pelo *gas* estipulado, mais rápido sua transação será processada pela rede.

Segundo Schüpfer [36], outros usuários interagem com o *smart contract* enviando Ether através de transações ou executando alguma função, como é possível observar na Figura 13 [36]. Informações como nome da função a ser executada e os parâmetros passados são enviados através do campo de dados da transação, mencionado anteriormente.

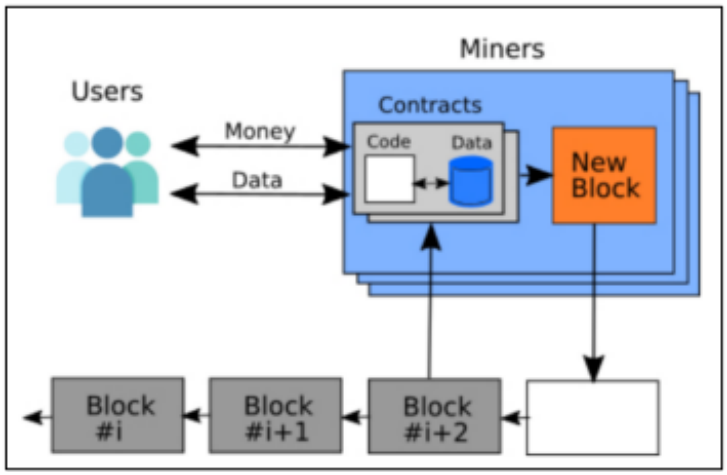

Figura 13: Interação com o *smart contract*

Uma conta Ethereum possui basicamente quatro campos. O primeiro é o *nonce* que caso a conta seja de propriedade externa (*externally owned account*), representa o número de transações enviadas do endereço da conta e caso a conta seja uma conta de contrato (*contract account*), o *nonce* é o número de contratos criados pela conta. O segundo é o atual saldo de Ether que existe na carteira. O terceiro seria o código do contrato, se existente. Por último, existe o armazenamento da conta, um *hash* de 256 bits do nó raiz de uma árvore Merkle que codifica o conteúdo de armazenamento da conta (um mapeamento entre valores inteiros de 256 bits). A árvore Merkle codifica o *hash* do conteúdo de armazenamento dessa conta, sendo vazia por padrão.

Os contratos têm a capacidade de mandar mensagens para outros contratos. Essa mensagem é composta basicamente por cinco itens. O primeiro é o endereço de quem mandou a mensagem, o segundo é o destinatário, o terceiro é a quantidade de Ether que

será transferido junto com a mensagem, o quarto é um campo de dados opcional e para finalizar, o quinto item se refere ao *startgas*, que inicializa um limite de quanto poder computacional de execução de código pode ser usado.

O Ethereum *state function* refere-se ao conjunto de informações que define o estado atual de todos os contratos inteligentes e contas de usuários no blockchain. Os contratos inteligentes no Ethereum possuem um estado interno que pode ser modificado por meio de funções específicas definidas no contrato. Sua estrutura é ilustrada na Figura 14 [4].

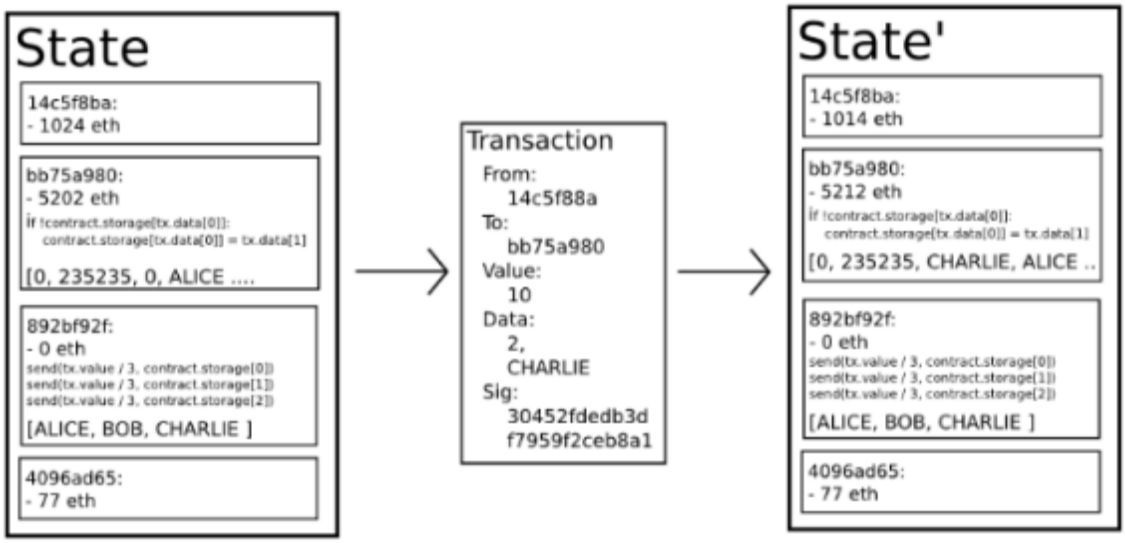

Figura 14: Ethereum *state function*

2.16 Oráculos

Os blockchains não acessam diretamente a informações fora da cadeia de blocos, não existindo assim, uma forma direta para validar as condições preestabelecidas nos contratos inteligentes. Desse modo, um oráculo atua como um intermediador da informação colhida no mundo exterior e a transporta para dentro do blockchain Para um contrato inteligente cujas condições de execução estão vinculadas a indicadores de tempo ou escritas no blockchain, a verificação será automática. Por outro lado, caso seja necessário verificar uma condição externa, é necessário recorrer a um terceiro de confiança, um oráculo, no Ethereum. O oráculo pode ser um terceiro indicado por ambas as partes, um instituto/associação de confiança ou um consenso de muitos terceiros [40].

No entanto, é importante observar que um contrato inteligente não espera que os dados de uma fonte externa fluam para o sistema. O contrato deve ser invocado, o que significa que é preciso gastar recursos de rede para chamar dados do mundo exterior. Isso induz custos de transação de rede, no caso do Ethereum, o *gas* [41]

Conforme mostrado na Figura 15 [42], o ecossistema do oráculo compreende geralmente as seguintes partes:

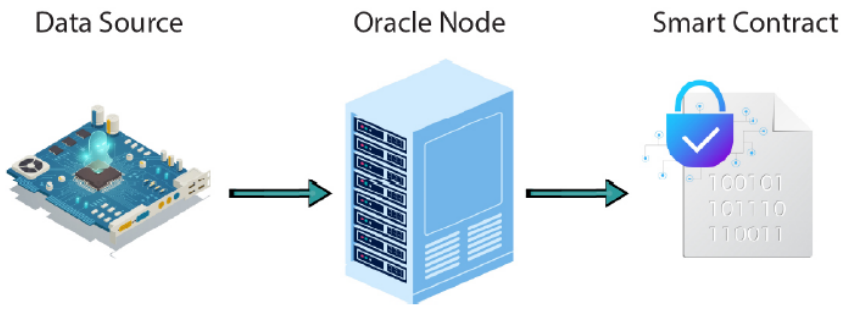

Figura 15: Ecossistema oráculo

- Fonte de dados: esta é a fonte no qual os dados são coletados e armazenados. Pode ou não ser utilizado por uma aplicação descentralizada. A fonte de dados pode ser uma *Application Programming Interface* (API), um sensor, ou um humano ciente de um conhecimento/evento;

- Canal de comunicação: é geralmente chamado de "nó". Ele coleta dados da fonte e os entrega a um contrato inteligente para que este possa ser executado.

- Contrato inteligente: contém o código que estabelece como os dados coletados podem ser gerenciados. Normalmente, tem critérios de qualidade pré-especificados para que os dados sejam aceitos ou rejeitados. Se necessário, também pode realizar cálculos para entregar os dados apropriados ao contrato;

Dependendo de como essas três partes estão organizadas e interagem entre si, vários tipos de oráculos podem ser projetados. Essas três partes de um oráculo nem sempre são separadas umas das outras, uma vez que a mesma entidade pode às vezes cobrir duas ou três funções simultaneamente.

Um ser humano, por exemplo, pode servir como fonte de dados e comunicar os dados diretamente a um contrato inteligente. Na realidade, ter mais de uma entidade que cobre o papel de fonte/nó é possível e desejável. Contar com múltiplas entidades é, de fato, crucial para garantir a execução de contratos inteligentes, especialmente quando uma ou mais fontes/nós de dados estão com defeito, ou *off-line*.

Existem diferentes tipos de oráculos:

- Oráculo de *software*: oráculos de *software* (também conhecidos como oráculos determinísticos) interagem com fontes de informação online e as transmitem para o blockchain. Essas informações podem vir de bancos de dados online, servidores,

sites, essencialmente, qualquer fonte de dados na Internet. O fato dos oráculos de *software* estarem conectados à Internet não apenas permite que eles forneçam informações a contratos inteligentes, mas também transmitam essas informações em tempo real. Isso os torna um dos tipos mais comuns de oráculos de blockchain [43].

- Oráculo de *hardware*: oráculos de *hardware* são projetados para obter informações do mundo físico e disponibilizá-las para os contratos inteligentes. Tais informações podem ser retransmitidas via sensores eletrônicos, IoT (*Internet of Things*), *scanners* de código de barras/QR-Codes, robôs e outros dispositivos de leitura de informações. Um oráculo de *hardware* essencialmente "traduz" eventos do mundo real em valores digitais que podem ser entendidos pelos contratos inteligentes [43].

- Oráculo humano: indivíduos com habilidades e conhecimentos especializados em um campo específico também podem servir como oráculos. Eles podem pesquisar e verificar a autenticidade das informações de várias fontes e traduzir essas informações aos contratos inteligentes. Como oráculos humanos podem verificar sua identidade usando criptografia, a possibilidade de um fraudador falsificar sua identidade e fornecer dados corrompidos é relativamente muito baixa. Os oráculos humanos não são apenas capazes de transmitir dados determinísticos, mas também de responder a perguntas arbitrárias, o que pode ser uma tarefa difícil para uma máquina [43].

- Oráculos computacionais: até agora, discutimos apenas oráculos no contexto de solicitação e entrega de dados (também conhecido como *Data Carrier Oracles* ou *Automated Oracles*). No entanto, os oráculos também podem ser usados para executar uma solução de computação "*off-chain*", uma função que pode ser especialmente útil devido ao limite de *gas* e ao custo computacional comparativamente caro de executar funções dentro do blockchain. Em vez de apenas retransmitir os resultados de uma consulta, os oráculos de computação podem ser usados para realizar a computação em um conjunto de entradas e retornar um resultado calculado que poderia ser inviável para calcular no blockchain. Por exemplo, pode-se usar um oráculo de computação para realizar um cálculo de regressão computacionalmente intensivo para estimar o rendimento de um contrato de títulos [43].

- Oráculo de entrada/saída: oráculos de entrada transmitem informações de fontes externas para os contratos inteligentes, enquanto os oráculos de saída enviam informações de contratos inteligentes para o mundo externo [43].

- Oráculo específico de contrato: oráculo específico de contrato é aquele projetado para ser usado por um único contrato inteligente. Isso significa que se alguém quiser implantar vários contratos inteligentes, um número proporcional de oráculos específicos do contrato deve ser desenvolvido. Esse tipo de oráculo é considerado

muito demorado e caro de manter. As empresas que desejam extrair dados de várias fontes podem achar essa abordagem impraticável. Por outro lado, como os oráculos específicos de contrato podem ser projetados do zero para atender a um caso de uso específico, os desenvolvedores têm alta flexibilidade para adaptá-los a requisitos específicos [43].

- Oráculos baseados em consenso: os dados são obtidos via mercados de consenso e de previsão. Usar apenas uma fonte de informação pode ser arriscado e não confiável. Para evitar manipulação do mercado, os mercados de previsão implementam um mecanismo de classificação de oráculos. Para maior segurança, um conjunto de oráculos distintos pode ser utilizado para validar uma movimentação blockchain [41].

Quanto à sua arquitetura, oráculos podem ser classificados em centralizados e descentralizados [44]. Uma arquitetura centralizada se baseia em entidades confiáveis que obtém informações do mundo externo quando requisitadas por um *smart contract*. Embora conceitualmente mais simples, arquiteturas centralizadas inevitavelmente introduzem pontos únicos de falha (*Single Point of Failure* - SPOF), que facilmente se tornam alvos de ataques de segurança. Uma arquitetura descentralizada, por sua vez, estabelece oráculos como entidades redundantes, com a função de prover informações e também validá-las, por meio de protocolos envolvendo voto e verificação. Isso elimina a existência de SPOFs, mas por outro introduz complexidade ao processo desempenhado pelos oráculos.

2.17 Aprendizagem de Máquina

A aprendizagem de máquina é uma subárea da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem aos sistemas aprenderem padrões e tomar decisões com base em dados. O objetivo é capacitar as máquinas a melhorar seu desempenho em tarefas específicas ao longo do tempo sem serem explicitamente programadas para cada situação.

A aprendizagem de máquina permite que um sistema (máquina) aprenda a partir de dados e experiência, ajustando-se automaticamente para melhorar o desempenho em uma tarefa específica. Seus componentes principais são:

- Dados: A aprendizagem de máquina depende de dados. Os algoritmos são alimentados com conjuntos de dados para aprender padrões e tomar decisões;
- Modelos: Um modelo de aprendizagem de máquina é uma representação matemática do comportamento esperado com base nos dados fornecidos. Esse modelo é treinado para fazer previsões ou tomar decisões;

- Treinamento: O treinamento do modelo envolve alimentar dados ao algoritmo para que ele possa ajustar seus parâmetros e aprender padrões. Durante esse processo, o modelo é ajustado para minimizar erros;
- Algoritmos: São os procedimentos matemáticos que os modelos de aprendizagem de máquina seguem para aprender padrões e realizar tarefas específicas;

Tipos de Aprendizagem de Máquina:

- Aprendizagem Supervisionada: O modelo é treinado em um conjunto de dados rotulado, onde a saída desejada já é conhecida. O objetivo é aprender a mapear entradas para saídas;
- Aprendizagem Não Supervisionada: O modelo é treinado em um conjunto de dados não rotulado, e o algoritmo tenta encontrar padrões e estruturas nos dados sem orientação sobre as saídas desejadas;
- Aprendizagem por Reforço: O modelo aprende tomando ações em um ambiente dinâmico e recebendo recompensas ou penalidades. O objetivo é aprender a sequência de ações que maximiza a recompensa ao longo do tempo;

As aplicações práticas da aprendizagem de máquina vão desde o reconhecimento e identificação de padrões de dados, como reconhecimento de voz, reconhecimento de imagem e processamento de linguagem natural, até a realização de análises e previsões com base em dados históricos, como previsões do tempo, preços de ações e demanda de produtos. Outras aplicações seriam o auxílio na tomada de decisões ao capacitar sistemas para tomar medidas de forma autossuficiente, como carros autônomos e robôs industriais e também a personalização de recomendação de produtos, conteúdo online e serviços com base no comportamento do usuário.

Existem alguns desafios e considerações éticas que devem ser levadas em consideração principalmente em relação à otimização de modelos para evitar a memorização de dados de treinamento (*overfitting*), onde o modelo aprende os detalhes específicos do conjunto de treinamento, mas não generaliza bem para novos dados. Nesses casos, é essencial a regularização, o aumento de dados e a escolha de modelos mais simples. Também deve ser levado em conta a otimização de modelos para evitar simplificações excessivas (*underfitting*), onde o modelo é muito simples e não captura bem a complexidade dos dados, sendo preciso a escolha de modelos mais complexos e ajustes dos hiperparâmetros.

É crucial compreender como e por que um modelo toma decisões, especialmente em aplicações críticas e é essencial garantir que os modelos não perpetuem ou ampliem um viés existente nos dados, ou seja, equilibrar a capacidade do modelo de se ajustar aos

dados de treinamento (baixo viés) com a capacidade de generalização para novos dados (baixa variância), promovendo a equidade e encontrando um equilíbrio que minimize erros tanto no conjunto de treinamento quanto de testes.

2.18 Redes Neurais Convolucionais

No contexto de inteligência artificial e aprendizado de máquina existem as Redes Neurais Convolucionais (CNNs), modelos computacionais inspirados no funcionamento do cérebro humano. Essas redes são compostas por unidades básicas chamadas neurônios, que estão organizadas em camadas e interconectadas por meio de conexões ponderadas. A ideia é simular a maneira como os neurônios no cérebro humano estão conectados e respondem a estímulos.

As Redes Neurais Convolucionais (CNNs) são uma abordagem específica dentro do amplo campo da aprendizagem de máquina, uma classe especial de redes neurais projetadas para processar dados que têm uma estrutura de grade, como dados de imagens. Elas são particularmente eficazes em tarefas relacionadas a visão computacional, como reconhecimento de imagem, classificação de objetos, detecção de objetos e segmentação de imagens.

As CNNs são treinadas usando técnicas de aprendizagem de máquina, especialmente aprendizagem supervisionada. Os pesos das conexões nas CNNs são ajustados durante a etapa de treinamento para otimizar o desempenho em tarefas específicas, como reconhecimento de imagem. Assim como em outros modelos de aprendizagem de máquina, o objetivo das CNNs é generalizar padrões aprendidos nos dados de treinamento para fazer previsões precisas em novos dados.

Técnicas como *transfer learning*, uma abordagem de aprendizagem de máquina no qual um modelo treinado em uma tarefa é reutilizado como ponto de partida para uma tarefa relacionada, são frequentemente aplicadas a CNNs, permitindo que modelos pré-treinados em grandes conjuntos de dados (como ImageNet) sejam reutilizados para tarefas específicas com conjuntos de dados menores. Isso é especialmente útil quando temos um conjunto de dados pequeno para a tarefa-alvo

As CNNs aprendem automaticamente hierarquias de características, começando com características simples nas camadas iniciais e progredindo para características mais complexas nas camadas posteriores. Além disso, as CNNs são eficazes em reconhecimento de padrões mesmo quando os padrões estão em diferentes partes da imagem, devido à propriedade de invariância de translação das camadas convolucionais, e possuem a capacidade de reconhecer padrões locais, permitindo capturar informações contextuais em regiões específicas da imagem. Algumas características das CNNs incluem:

Camadas Convolucionais

A operação de convolução é a pedra angular das CNNs. Ela envolve o uso de filtros (também chamados de *kernels*) que são passados sobre a imagem para extrair características locais, como bordas, texturas e padrões. Convolução é uma operação matemática entre duas funções f e g produzindo uma terceira função, que pode ser interpretada como uma função modificada de f. No processamento de imagens, onde a imagem é definida como uma função bidimensional, a convolução é útil para detecção de bordas, suavização de imagem, extração de atributos, entre outras aplicações [45].

A convolução de uma imagem pode ser interpretada como o somatório da multiplicação de cada elemento da imagem, com seus vizinhos locais, pelos elementos da matriz que representa o filtro de convolução. Esse cálculo é ilustrado na Figura 16 abaixo:

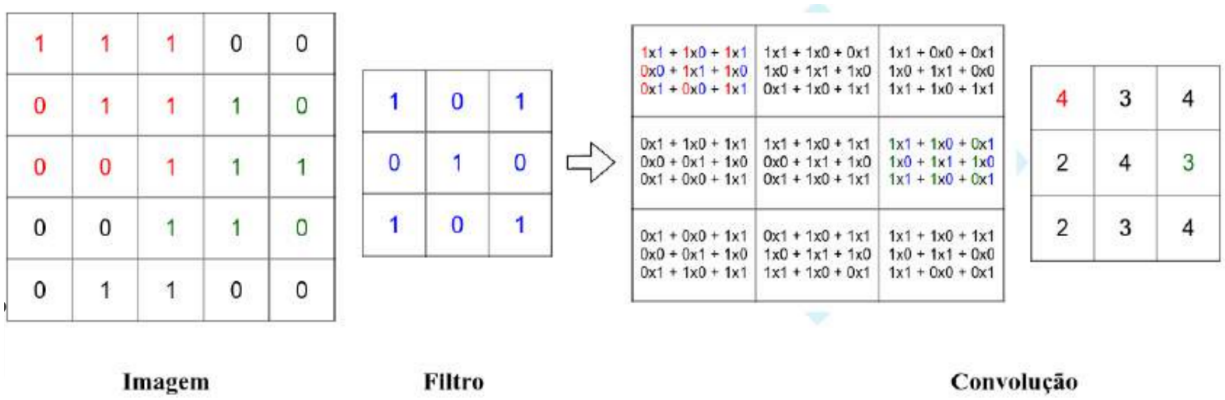

Figura 16: A esquerda, uma imagem hipotética representada por um único canal com dimensões 5 x 5 que recebe a aplicação de um filtro 3 x 3. A direita, uma matriz ilustrando o somatório que fornece o resultado da convolução.

Na abordagem utilizada nesse exemplo, após a aplicação do filtro de convolução a imagem original é reduzida em tamanho proporcional as dimensões do filtro utilizado. Todavia, existem outras abordagens de aplicação utilizando, por exemplo, criação de novas células adjacentes às bordas da imagem, fazendo com que a imagem resultante mantenha as mesmas dimensões da imagem original após a aplicação do filtro de convolução.

Para melhor entendimento do conceito de convolução, será dado um exemplo de uma imagem 32 x 32 e profundidade 3 (RGB). Imagine a luz de uma lanterna apontada no canto superior esquerdo da imagem, essa luz abrange o espaço de uma área de 5 x 5 deslizando por todas as partes da imagem. Essa lanterna em termos de aprendizagem de máquina é conhecida como filtro, e a região que a luz atinge é o campo receptivo. A medida que o filtro se deslocar em torno da imagem de entrada, ocorre a multiplicação de matrizes envolvendo os valores de píxeis originais da imagem, já que o filtro é uma matriz 5 x 5 x 3. Ao final é gerada uma matriz de 28 x 28 x 1 que é o mapa de ativação, isso quer dizer que existem 784 locais diferentes que um filtro 5 x 5 pode estar contido em uma imagem 32 x 32. Esses 784 números são mapeados para uma matriz de 28 x 28.

Exemplo na Figura 17 [46].

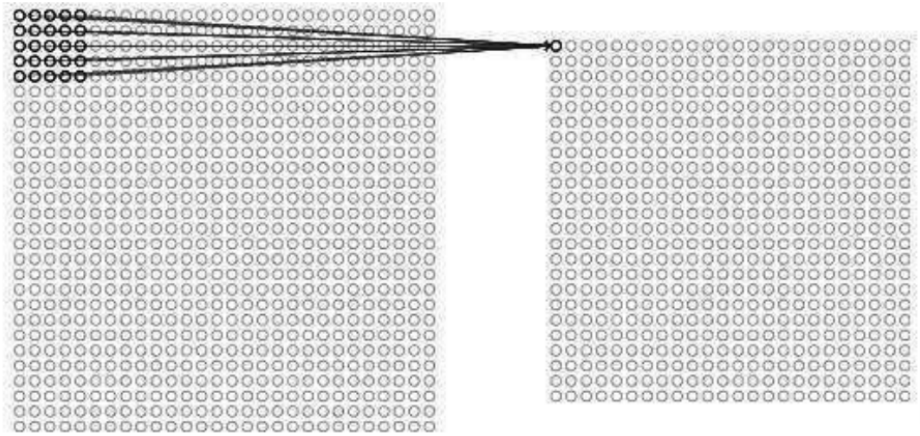

Figura 17: Exemplo de um mapa de filtro 5x5

Esses mapas de filtros buscam a detecção de características da imagem como: bordas, retas, cores simples e curvas [47]. Um exemplo pode ser visto na Figura 18 [47] em que se tem um mapa de filtro de curva com as dimensões 7 x 7 x 3.

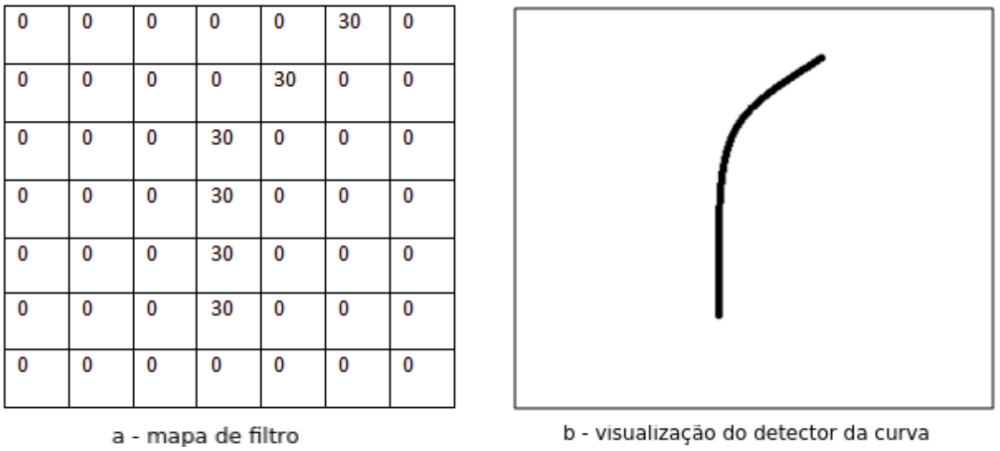

Figura 18: Exemplo mapa de filtro de curva

Esse filtro é passado em uma imagem em busca da detecção de uma curva compatível, observável no exemplo a seguir, na Figura 19 [47]. A matriz de píxeis da imagem original será multiplicada pela matriz de filtro: (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600. No caso em que o valor resultante da multiplicação de matrizes seja grande, significa que há a presença de uma curva compatível com o filtro. Caso o filtro passe em uma área da imagem sem a curva compatível, o valor resultante da multiplicação das matrizes será próximo a zero.

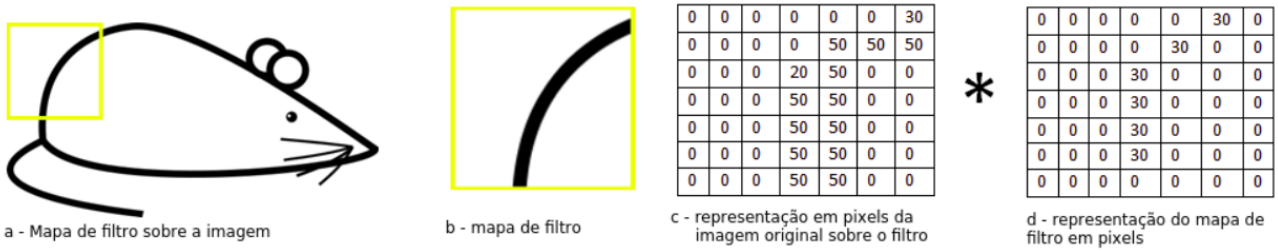

Figura 19: Exemplo de detecção de curva

Ao final dessa camada de convolução é gerado o mapa de ativação, esse mapa mostrará a chance de haver uma curva na imagem. Seguindo o exemplo apresentado, o mapa de ativação 26 x 26 x 1 terá o valor de 6600, o que mostra a presença da curva ativando o filtro. É possível ter outros filtros, sendo que quanto mais filtros, maior a profundidade do mapa de ativação e mais informação sobre a entrada [48].

Funções de Ativação

Funções de ativação não-lineares, como ReLU (*Rectified Linear Unit*), são comumente usadas para introduzir não-linearidade nas camadas, permitindo que a rede aprenda padrões mais complexos. Ela faz isso trocando valores de píxeis negativos da imagem por zero. Como visto na Figura 20 [46].

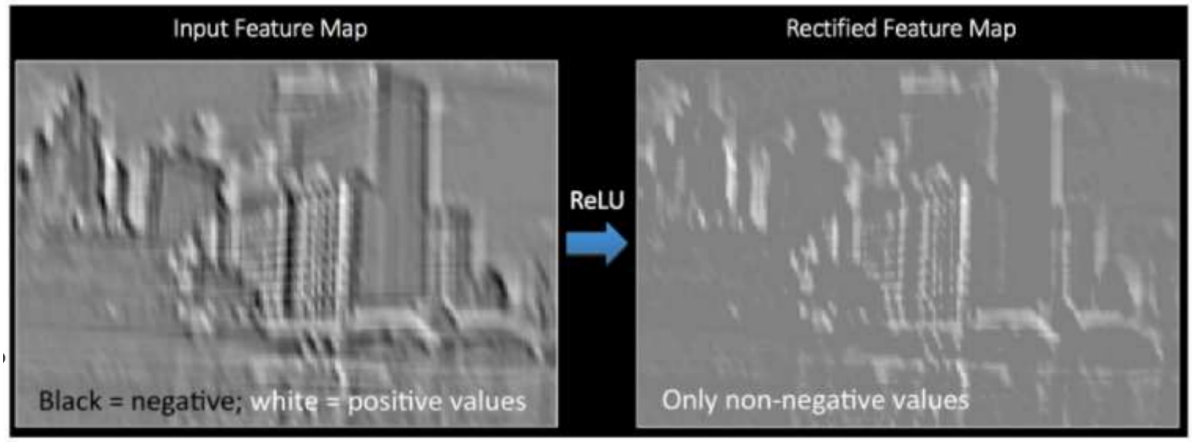

Figura 20: Exemplo de função ReLU

Camadas de Pooling

Em redes convolucionais é preciso o uso da camada de *pooling*, logo após uma cada de convolução. Essa técnica visa reduzir o tamanho espacial das matrizes resultantes da convolução, o que reduz a quantidade de parâmetros a serem aprendidos na rede[46]. Essa camada opera de forma independente em cada um dos canais do resultado da convolução. É necessário estabelecer o tamanho do filtro da camada de *pooling*, na Figura 21 [46] é utilizado um filtro 2 x 2 para criar uma camada de *maxpooling*, que consiste basicamente

uma operação com matrizes para extrair o maior elemento do quadrante aplicado pelo filtro.

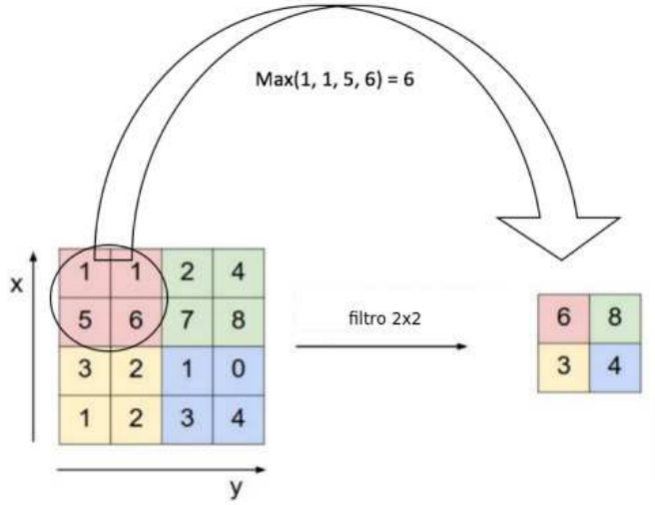

Figura 21: Camada de Pooling

Como mostrado, a primeira camada convolucional detecta características de baixo nível, como bordas e curvas. Essas características não são o suficiente para detectar objetos complexos [47], então são utilizadas outras camadas convolucionais ligadas em sequência. Basicamente a saída da primeira camada convolucional será a entrada da segunda camada convolucional, nessa nova camada a entrada é o mapa de ativação que resulta da primeira camada. Aplicando um conjunto de filtros na segunda camada, a saída será ativada com características de nível superior. Os tipos dessas características podem ser semicírculos, quadrados ou figuras complexas.

Camadas Totalmente Conectadas

A Camada Totalmente Conectada (*Fully Connected Layer*), também conhecida como camada densa, é uma parte crucial de uma Rede Neural Convolucional. Enquanto as camadas convolucionais e de *pooling* são responsáveis por extrair características espaciais da imagem, as camadas totalmente conectadas são responsáveis pela tomada de decisão final.

As camadas totalmente conectadas em uma CNN têm a função de integrar informações extraídas por camadas convolucionais para realizar tarefas finais como classificação ou regressão. Essa camada proporciona à rede a capacidade de aprender padrões mais complexos e relações entre características. A seguir temos a Figura 22 [46] detalhando a arquitetura completa de uma CNN.

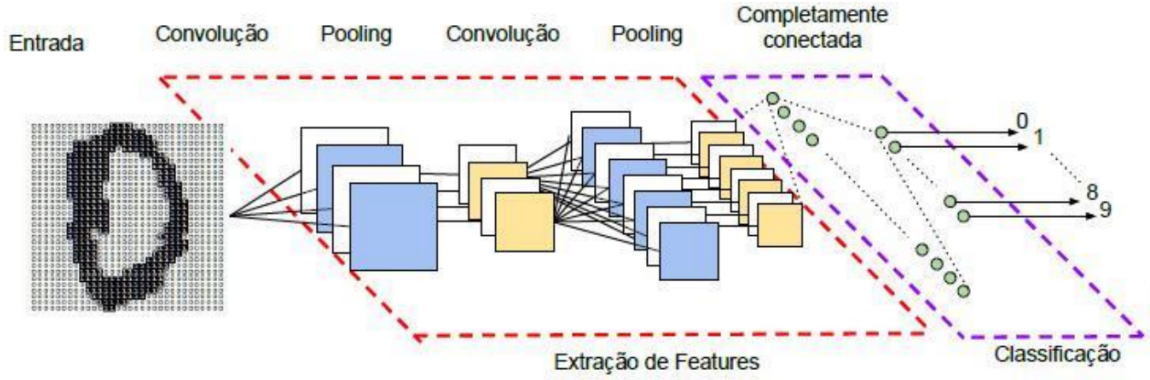

Figura 22: Exemplo de arquitetura de uma CNN

2.19 Segmentação Semântica

Na tarefa de segmentação semântica, cada píxel da imagem é classificado, possuindo sua própria distribuição de probabilidades: a rede é treinada para prever a qual classe cada píxel pertence. Isso permite não apenas que uma variedade de classes seja identificada numa mesma imagem, mas também determinar sua localização.

A segmentação semântica é uma técnica de classificação da imagem ao nível de píxel. Assim, cada píxel é classificado como pertencente a uma das classes definidas no problema de segmentação, que normalmente referem-se aos diferentes objetos presentes em uma imagem. Ao classificar os píxeis de um objeto como pertencentes a uma classe distinta, a segmentação semântica propicia, na verdade, a segmentação dos diversos objetos presentes em uma imagem. A primeira rede de segmentação semântica foi proposta por Long et al. [49]. A Figura 23 [49] mostra um exemplo de uma CNN com segmentação semântica, onde a primeira parte da rede, em que ocorre uma diminuição do tamanho dos mapas de ativação, é denominada de codificador (*encoder*). A segunda parte da rede, em que ocorre um aumento do tamanho dos mapas de ativação, é denominada de decodificador (*decoder*).

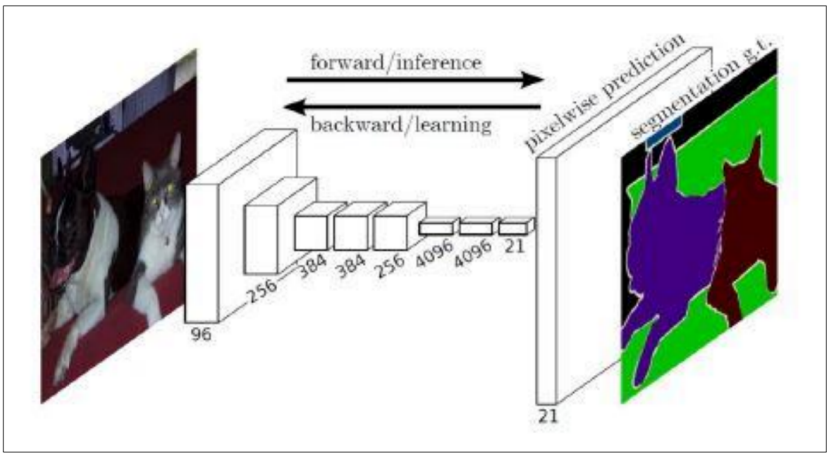

Figura 23: Arquitetura CNN com segmentação semântica

Uma imagem segmentada se assemelha a imagem original quanto a sua forma, e se difere quanto a cor. Na imagem segmentada, cada píxel é colorido com um número N de cores, onde N é o número de classes que se está segmentando. Ao segmentar uma imagem de satélite referente a uma região, por exemplo, isso pode ser simples como N = 2 ("floresta", ou "não-floresta"), como também pode capturar uma série de diferentes classes.

Uma arquitetura geral de segmentação semântica pode ser amplamente considerada como uma rede codificadora, seguida por uma rede decodificadora. Como visto na Figura 24 [50] o codificador é geralmente uma rede de classificação pré-treinada seguida por uma rede de decodificação [50]. A tarefa do decodificador é projetar semanticamente os recursos discriminativos aprendidos pelo codificador no espaço de píxel para obter uma classificação densa.

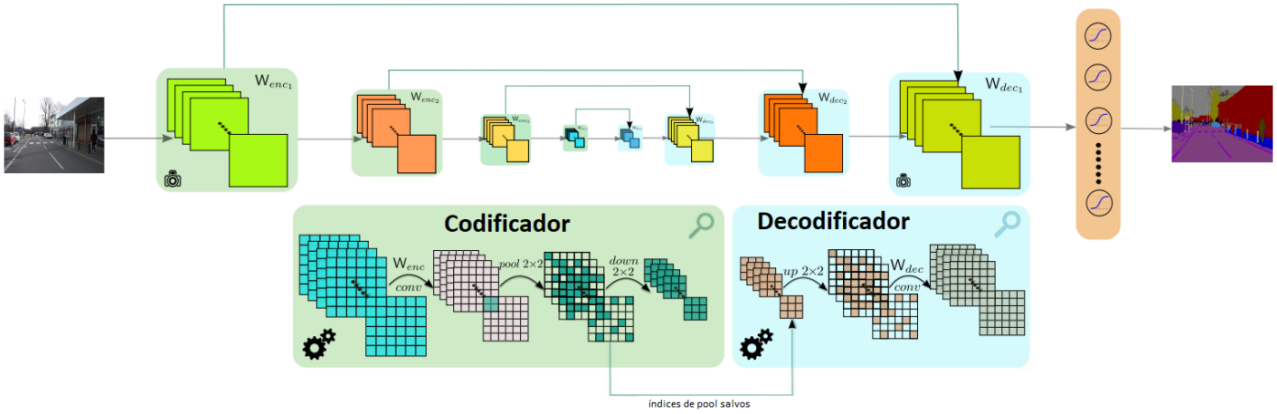

Figura 24: Exemplo do funcionamento de uma CNN de segmentação semântica

Com a popularidade do aprendizado profundo nos últimos anos, muitos problemas de segmentação semântica estão sendo resolvidos usando arquiteturas profundas, na mai-

oria das vezes por CNNs, que ultrapassam a eficácia de outras abordagens em precisão e eficiência.

A incorporação de abordagens de Redes Neurais Convolucionais (CNNs) e segmentação semântica aos oráculos blockchain pode proporcionar vários benefícios, principalmente quando o objetivo é a análise de imagens e identificação precisa de áreas específicas em uma cena. Os benefícios dessas abordagens aliadas a tecnologia blockchain fornecem uma análise mais detalhada e precisa de imagens, permitindo que o sistema tome decisões mais informadas e automatize ações com base nessa análise. Isso é particularmente valioso em situações em que a identificação detalhada de elementos em uma imagem e a interpretação de dados visuais é essencial para as operações do oráculo.

2.20 Considerações

Neste capítulo foram abordados desde os conceitos mais básicos até os fundamentos de maior abstração relacionados ao desenvolvimento desta pesquisa, com objetivo de contextualizar e fundamentar o referencial teórico do trabalho. Diante dos diversos conceitos, informações e pontos de intersecção presentes na literatura, este trabalho pretende aliar as melhores características da tecnologia blockchain em conjunto com as abordagens de inteligência artificial e processamento de imagens, de modo a fornecer uma proposta de solução que possa atingir os objetivos da pesquisa.

No próximo capítulo serão apresentados os estudos mais relevantes relacionados ao tema desta pesquisa selecionados a partir de um protocolo de revisão quasi-sistemática da literatura.

3 TRABALHOS RELACIONADOS

3.1 Revisão Quasi-Sistemática da Literatura

Com o intuito de identificar os principais trabalhos relacionados ao tema desta pesquisa, foi realizada uma revisão quasi-sistemática da literatura das soluções e propostas utilizando conceitos blockchain que visam o aumento da confiança e transparência nas etapas de investimento, administração e aplicação dos recursos de projetos promovam a sustentabilidade. Uma revisão quasi-sistemática da literatura consiste em localizar os estudos existentes, selecionar, avaliar e sintetizar suas contribuições, permitindo conclusões razoavelmente claras sobre o que é e o que não é conhecido na literatura [51].

3.1.1 Protocolo

Para realização do processo de revisão quasi-sistemática da literatura é necessário a definição de um protocolo de revisão, no qual é descrito um planejamento previamente estabelecido para a condução da revisão e descreve, qual é a questão da pesquisa, quais métodos e estratégias de buscas utilizados e quais os critérios de seleção das publicações que constituíram a amostra. Para o tema selecionado foram definidas as seguintes questões de pesquisa:

- **QP1**: Como tornar mais transparente o processo de financiamento de projetos ambientais utilizando oráculos blockchain?
- **QP2**: Como tornar mais confiável a relação entre o financiador e o prestador de serviços ambientais através dos recursos fornecidos pelos oráculos blockchain?

A estratégia de busca utilizada foi a busca automática sobre algumas bibliotecas digitais de artigos científicos. Foram selecionados os portais:

- IEEEXplore
- ACM Digital Library
- ScienceDirect
- Springer

Visando construir as *strings* de buscas foram selecionadas algumas palavras-chave, sinônimos e termos relacionados à pesquisa. Esses termos, apresentados na Tabela 2, juntamente com suas variantes foram então combinados, formando a seguinte *string* de busca:

Tabela 2: Termos de busca relacionados à pesquisa

Palavras-chave	**Sinônimos e Termos Relacionados**
Blockchain Sustainability	Smart Contract, Cryptocurrency, Environmental Preservation, Environmental Protection

("Blockchain" OR "Smart Contract" OR "Cryptocurrency") AND ("Sustainability" OR "Environmental Preservation" OR "Environmental Protection")

Após o processo de busca, para a seleção das publicações para este trabalho, foram selecionados alguns critérios de exclusão (E) e inclusão (I):

- **I1**: Incluir todos os artigos completos relacionados ao contexto da revisão, identificados por meio da leitura do título e do resumo e, para casos em que não se foi obtido informações relevantes e suficientes, da leitura completa do artigo;
- **I2**: Caso existam mais de 30 trabalhos não repetidos no resultado da busca em algum dos portais, incluir os 30 artigos mais citados;
- **I3**: Caso existam trabalhos semelhantes ou repetidos, deve-se incluir o mais recente;
- **E1**: Fora do escopo da pesquisa;
- **E2**: Trabalhos com acesso restrito;
- **E3**: Trabalhos repetidos, preservando apenas o mais recente;
- **E4**: Trabalhos que não pertencem à categoria de artigos científicos;

Após o processo de inclusão e exclusão com base nos critérios estabelecidos pelo protocolo definido, realizou-se a classificação a partir das seguintes regras:

1. **Plataforma blockchain**: identifica qual plataforma blockchain o autor define para a solução proposta;
2. **Oráculos blockchain**: identifica se a proposta do autor utiliza oráculos blockchain e caso, sim, qual tipo de oráculo é projetado;
3. **Padrões regulatórios via contratos inteligentes** : identifica se a proposta do autor implementa mecanismos reguladores via contratos inteligentes;
4. **Estratégia sustentável**: identifica qual estratégia utilizada pelo autor para alcançar o objetivo sustentável definido. Entre as principais estratégias estão:

(a) Rastreamento de cadeia de suprimentos;

(b) Monitoramento e cobertura de áreas;

(c) Comunidade/Marketplace digital;

(d) Gerenciamento de energia limpa;

(e) Mercado de crédito de carbono;

3.1.2 Busca, Seleção e Classificação dos Artigos

Segundo o protocolo definido, a partir da *string* de busca *("Blockchain" OR "Smart Contract" OR "Cryptocurrency") AND ("Sustainability" OR "Environmental Preservation" OR "Environmental Protection")* e dos portais definidos no protocolo foi possível aplicar o processo de busca de artigos científicos. A pesquisa foi realizada no dia 15 de junho de 2022 e nela foram encontradas 402 contribuições da literatura relacionadas à temática deste trabalho, como mostra a Tabela 3.

A partir dos resultados do processo de busca, foi aplicado o processo de filtragem. Esse processo foi efetuado em duas iterações. A primeira iteração filtrou os trabalhos selecionados seguindo os critérios de inclusão e exclusão, definidos anteriormente, considerando apenas o título e o *abstract* do artigo. Após a primeira iteração, foram selecionados 117 artigos que seguiam os critérios estabelecidos. Na segunda iteração, foi feita a leitura completa dos trabalhos considerados indeterminados ou que poderiam estar relacionados com a pesquisa. Com isso, ao término das duas iterações, foi possível filtrar 10 trabalhos dentre os 402 iniciais selecionados. A Tabela 3 apresenta os resultados do processo de busca e seleção dos artigos.

Tabela 3: Resultado do processo de busca e seleção dos artigos

	IEEEXplore	**ACM**	**ScienceDirect**	**Springer**	**Total**
Processo de Busca	33	212	130	27	402
Seleção (1ª Iteração)	30	30	30	27	117
Seleção (2ª Iteração)	7	1	1	1	10

3.2 Avaliação Qualitativa das Principais Contribuições

3.2.1 *LogLog: A Blockchain Solution for Tracking and Certifying Wood Volumes*

O comércio de madeira ilegal tem um tremendo impacto negativo nos mercados local e internacional, bem como no ecossistema compartilhado. Esse fenômeno impulsiona o desenvolvimento contínuo de padrões florestais e legislações sustentáveis. Uma das autoridades de certificação florestal mais importantes da América do Norte é o *Forest*

Stewardship Council (FSC), que supervisiona a gestão de 150 milhões de hectares de floresta e mais de 3.500 empresas nos EUA e no Canadá.

Neste artigo é apresentado uma solução denominada LogLog, que consiste em um sistema de auditoria para manter os padrões de certificação FSC e rastrear a madeira proveniente utilizando conceitos blockchain. O sistema proposto opera através da tokenização de volumes de madeira, juntamente com todos os dados exigidos pelo FSC (como propriedade e localização) relacionados a cada volume. As mudanças decorrentes desses volumes, como fusão, divisão ou qualquer outra forma de transformação, são inteiramente reguladas por contratos inteligentes. São utilizados organismos de certificação (ou seja, terceiros confiáveis) para avaliar a conformidade regulatória dos membros do FSC e das empresas candidatas, sendo esses tradicionalmente os únicos responsáveis por identificar anomalias e verificar se os requerentes estabeleceram controles eficazes. A Figura 25 ilustra parte deste rastreio:

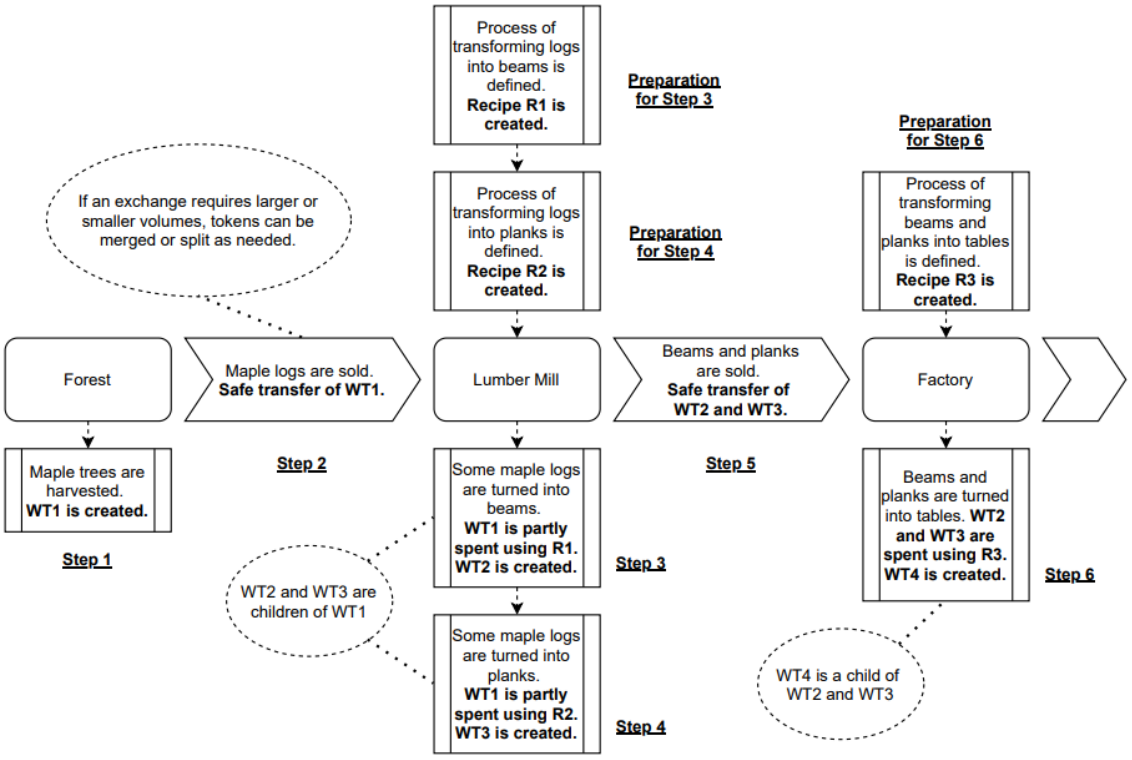

Figura 25: Rastreio parcial dos volumes de madeira

Através do uso de tecnologias de registros distribuídos, visa-se oferecer: digitalização de registros; maior rastreabilidade de mercadorias; auditoria transacional e maior transparência e acesso à informação para consumidores e autoridades. Assim, as contribuições deste estudo segundo os autores são:

- Propor uma solução que implemente os padrões regulatórios do FSC usando contratos executados inteiramente *on-chain*, demonstrando a capacidade dos DLTs para

expressar lógica de validação específica de domínio relacionado à indústria florestal;

- Fornecer duas implementações referentes ao mesmo design (uma usando o padrão de *token* ERC-721 e outra utilizando o padrão de *token* ERC1155) inteiramente escrito em Solidity e implantado em um ambiente Ethereum;
- Avaliar e apresentar resultados que demonstram a viabilidade da abordagem em termos de custos operacionais de *gas* em comparação com as tradicionais soluções de múltiplos bancos de dados;

O protótipo atual demonstra a adequação de uma solução totalmente *on-chain* para rastrear volumes de madeira e fazer cumprir os padrões do FSC, utilizando de organismos de certificação (oráculos humanos), para verificar a autenticidade das informações e verificar o controle do consumo sustentável de recursos por parte das empresas. Os resultados mostram que a tecnologia blockchain pode produzir sistemas competitivos de banco de dados único que satisfaçam as demandas industriais. Nesse contexto, os autores afirmam que a solução proposta possui os seguintes benefícios em relação aos sistemas tradicionais:

- Maior disponibilidade: a proposta utiliza um único banco de dados distribuído. As abordagens hibridizadas precisam manter pelo menos dois bancos de dados diferentes *on-line* o tempo todo, com pelo menos um deles sendo distribuído.
- Maior rastreabilidade: a solução rastreia todas as trocas e modificações dos volumes de madeira. As abordagens híbridas são limitadas em sua capacidade de capturar essas mudanças devido à lacuna informacional entre os dados *on-chain* e os dados *off-chain*.
- Maior integridade de dados: os dados de volumes rastreados pela LogLog têm um alto grau de imutabilidade, pois são totalmente *on-chain*. As abordagens tradicionais usam dados *off-chain* que são mais suscetíveis a adulterações.

3.2.2 *Commodity Ecology: A Virtual Community Platform for Promoting Responsible Consumption and Production to Achieve SDG 12*

Os Objetivos de Desenvolvimento Sustentável (ODS) foram estabelecidos pela Assembleia Geral das Nações Unidas em 2015 para o ano de 2030. Entre as diretrizes estabelecidas está a meta referente ao "Consumo e Produção Responsável" com objetivos de reduzir substancialmente a geração de resíduos por meio da prevenção, redução, reciclagem e reutilização de materiais. Neste artigo, os autores descrevem a plataforma *Commodity Ecology Mobile Virtual Community* (CEMVC) destinada a promover globalmente o consumo e produção responsável.

Segundo os autores, o CEMVC oferece quatro tipos de apoio aos membros da comunidade: apoio instrumental (negociação de produtos/resíduos dentro das ecorregiões; encontrar parceiros de negócios ou produtos), suporte informativo (compartilhamento de informações sobre produtos/resíduos e notícias de negócios/indústrias), apoio emocional (encontrar amigos e fazer networking com indivíduos, seja globalmente ou em uma ecorregião em particular) e apoio à avaliação (reconhecimento de boas ideias que constroem o desenvolvimento sustentável regional). O protótipo existente do CEMVC fornece os seguintes serviços aos membros de uma "região inteligente":

- Facilitar discussões sobre sustentabilidade de *commodities* entre pessoas na mesma ecorregião ou em diferentes regiões ecológicas;
- Criar postagens de um usuário na ecorregião ou globalmente selecionando uma, ou várias categorias de *commodities*;
- Visualizar, curtir, compartilhar e comentar postagens criadas por outros usuários na ecorregião ou globalmente;
- Inscrever membros como produtores ou consumidores em uma ecorregião particular com a criação de uma página pessoal para anunciar e encontrar outros produtores e consumidores;
- Permitir que os produtores exibam seus produtos disponíveis e seus resíduos úteis para que outros produtores e consumidores possam visualizá-los e comprá-los;
- Proporcionar aos produtores e consumidores a capacidade de se comunicar e partilhar produtos e resíduos;
- Incentivar trocas sustentáveis de materiais dos produtores aos consumidores no mercado (serviço de comércio de resíduos);
- Promover uma plataforma única útil para negócios/indústria com novidades e tendências sobre todos os materiais sustentáveis;

No CEMVC, a aplicação da tecnologia blockchain é empregada para facilitar a sustentabilidade durante a jornada do produto/material, rastreando origem e transferência de compras de produtos/resíduos e dando suporte e transparência aos processos. A jornada do produto/material conforme previsto na CEMVC é semelhante à jornada do produto/material na gestão de cadeias de suprimentos, onde materiais, informações e serviços fluem de um produtor para um consumidor. A Figura 26 mostra a aplicação da tecnologia blockchain no CEMVC para facilitar a viagem do produto/resíduo e permitir recomendações de produtos/resíduos para os usuários.

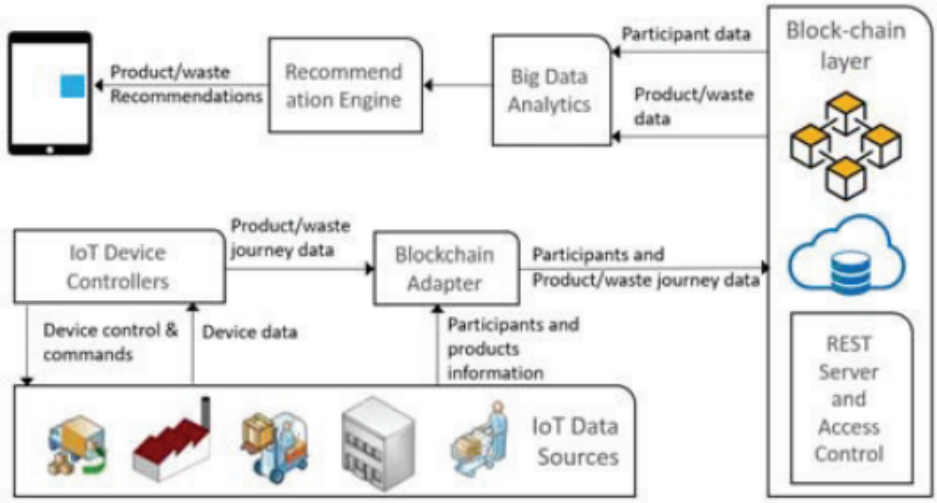

Figura 26: Arquitetura CEMVC

O sistema proposto realiza a rastreabilidade em toda a cadeia de abastecimento agroalimentar, reunindo, transferindo e compartilhando dados autenticados da produção, processamento, armazenamento, distribuição e venda. O blockchain também é projetado para dar suporte a transações financeiras e para auxiliar no gerenciamento da cadeia de suprimentos durante a entrega de bens e *commodities*. Na gestão da cadeia de suprimentos, a sustentabilidade é definida como um conceito de *triple-bottom-line* (tripé de sustentabilidade) que inclui um equilíbrio entre os aspectos ambientais, sociais e dimensões de negócios. É neste momento que a tecnologia blockchain se torna um impulso crescente para a construção de soluções sustentáveis.

De acordo com os autores, o CEMVC é uma solução que visa potencializar e viabilizar a relação entre os interessados no processo sustentável, através da criação de um ambiente virtual ecológico democrático em que as pessoas, profissionais e indústrias conversam com outros produtores e consumidores em "regiões inteligentes".

3.2.3 *Environmental Monitoring System Based on Blockchain*

O monitoramento ambiental é um elo importante na construção da civilização ecológica. Com base nos dados de monitoramento ambiental, pode-se objetivamente avaliar a qualidade ambiental, assim como também refletir a eficácia das estratégias ambientais. Além disso, a gestão e as tomadas de decisões também devem ser baseadas em dados de monitoramento ambiental. No entanto, a fraude de dados sobre poluentes fornecidos pelas empresas, a interferência em equipamentos de monitoramento ambiental por parte de governos e a inconsistência de dados ambientais em vários departamentos, contribuem para a redução da qualidade dos dados de monitoramento ambiental e afetam o nível de gestão ambiental. Neste artigo, é proposto um sistema de monitoramento ambiental

baseado em blockchain.

O sistema de proteção ambiental proposto pode ser visto na Figura 27. Na parte inferior estão os dispositivos de aquisição de dados ambientais, os sensores de qualidade do ar e da água. Os dados brutos coletados de milhares de dispositivos são reunidos diretamente na plataforma de *big data* e com base nesses dados, duas etapas são executadas. Em uma etapa são realizados os cálculos do *hash* dos dados brutos, estatísticas, indicadores e os dados de proteção ambiental revisados para serem inseridos na plataforma blockchain. Em outro momento é realizado o monitoramento inteligente, incluindo um padrão de proteção ambiental, monitoramento, análise de parâmetros dos dispositivos, aquisição de dados brutos, monitoramento de dados revisados, etc.

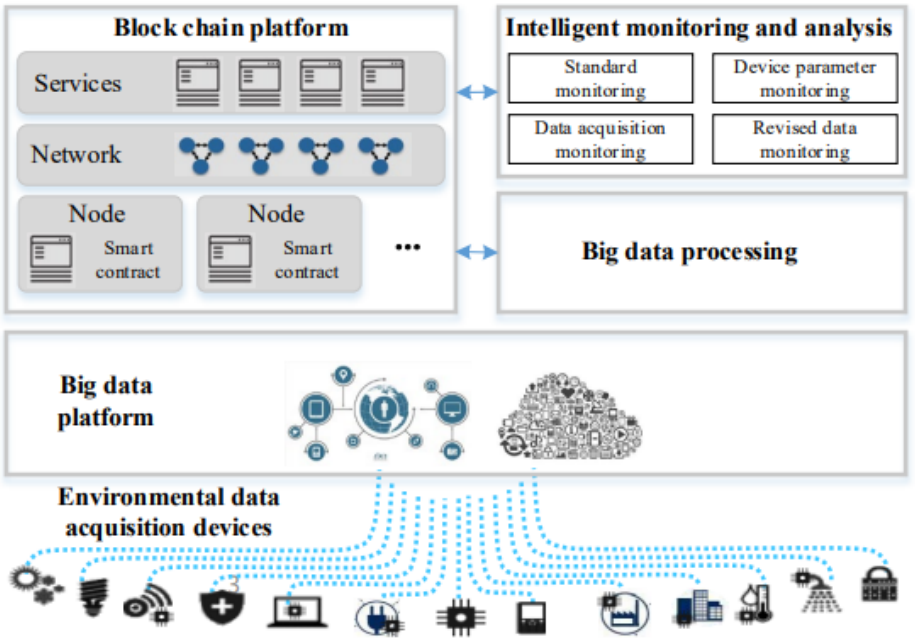

Figura 27: Plataforma de proteção ambiental baseada em blockchain

Segundo os autores, o valor da tecnologia blockchain na prevenção da falsificação dos dados ambientais podem ser listados da seguinte maneira:

- Oferecer dados ambientais unificados de toda a rede, para evitar a fraude de dados;
- Os dados originais são gravados diretamente no blockchain aumentando a confiabilidade dos dados;
- A modificação de parâmetros dos dispositivos e dados monitorados precisam de consenso, melhorando a supervisão e dificultando ações maliciosas;
- O processamento, o consenso e a publicação de dados exigem autenticação multipartidária e as informações dos registros não podem ser modificadas, tornando clara as responsabilidades e os poderes dentro do processo de monitoramento ambiental;

- Estabelecer um sistema de rastreabilidade de responsabilidades pela autenticidade dos dados originais e modificações de dados inseridos no sistema de monitoramento;
- Os registros de dados ambientais são abertos e transparentes com intuito de aumentar a credibilidade dos dados, mas também com o propósito de facilitar a supervisão e implementação;
- Fornecer registros de dados confiáveis para análise de *big data*, para um melhor apoio para análise do *status* de operação do dispositivo e ajudar identificar fraudes e dados maliciosos;

O mecanismo de interação entre os sensores de proteção ambiental e a rede blockchain é descrito na Figura 28.

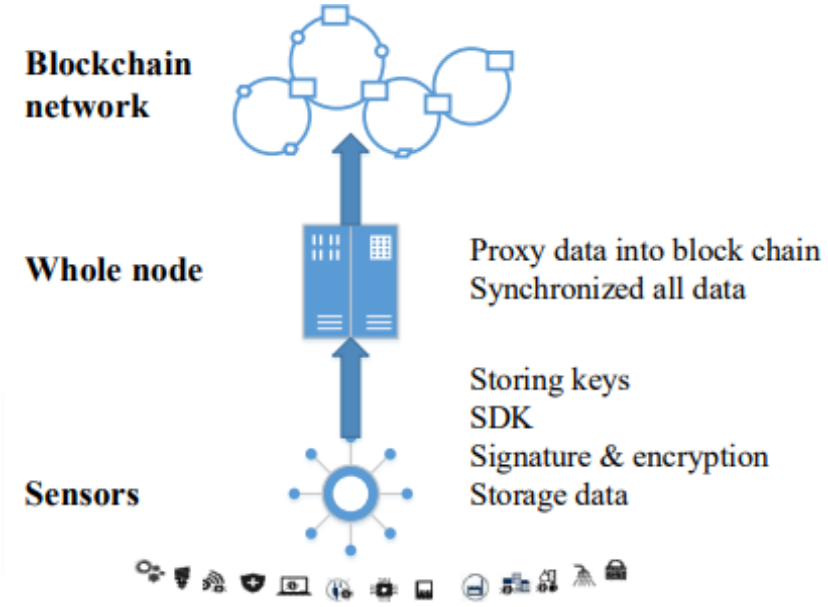

Figura 28: Mecanismo de interação entre o sensor e a rede blockchain

Os dados assinados e criptografados são transportados do sensor para o nó através da rede. Por meio de cada nó, os dados do sensor são enviados para o blockchain e todos os dados dentro da rede blockchain são sincronizados.

O sistema de monitoramento ambiental tridimensional proposto pelos autores consiste em quatro partes principais. A parte mais externa são os satélites no espaço, que podem realizar o monitoramento dinâmico da atmosfera, água, solo e ecologia. A segunda parte é o monitoramento da meteorologia, poluição e radiação no ar. A terceira parte está relacionada a atividades humanas, incluindo o monitoramento da indústria, agricultura, transporte e construção. A quarta e última parte monitora o *status* do solo, água e ecologia através da interconexão digital de objetos cotidianos com a Internet, IOT. A Figura 29 descreve a estrutura de monitoramento da proposta do autor:

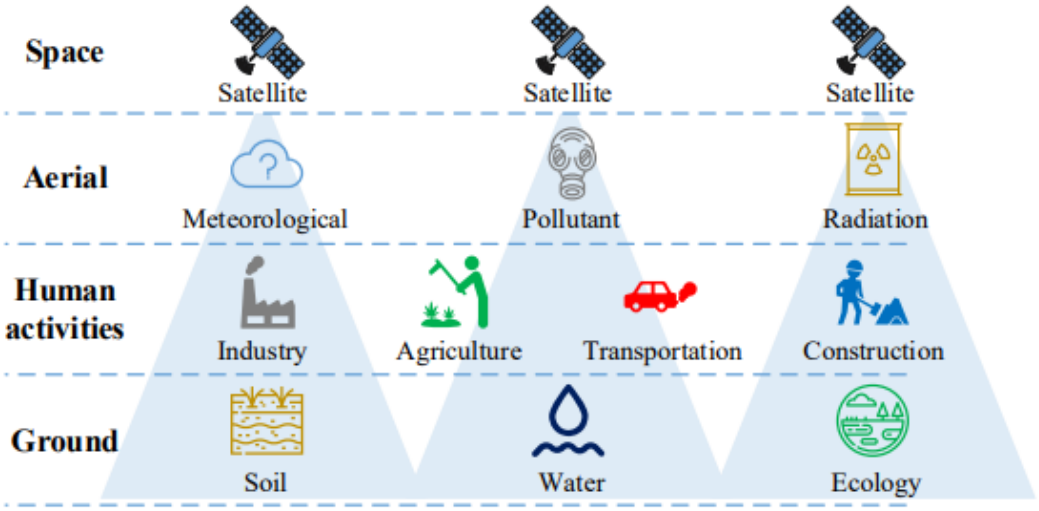

Figura 29: Monitoramento ambiental tridimensional

Embora o blockchain possa impedir a falsificação de dados em determinadas circunstâncias *on-chain*, não é capaz de garantir que os dados *off-chain*, ou seja, provenientes de fora da plataforma blockchain, são verdadeiros e eficazes. Com base na consideração acima, um dispositivo inteligente de confiança (oráculo) de proteção ambiental também é proposto pelos autores.

O oráculo de monitoramento ambiental tem as funções de percepção, visão, audição, análise, cálculo, aprendizado, gerenciamento, controle, etc. Múltiplos oráculos de monitoramento de proteção são implantados em áreas específicas conforme o escopo de cobertura definido. Eles estão interligados entre si e com a plataforma de proteção ambiental, formando uma rede de conhecimento para o monitoramento da proteção ambiental. Segundo os autores, as funções específicas do dispositivo são as seguintes:

- Capacidade de se conectarem entre si e com a plataforma de proteção ambiental no modo com ou sem fio;
- Poder coletar automaticamente os principais dados ambientais em áreas específicas, como parâmetros da atmosfera, água, solo, ruído, etc.;
- Monitorar as áreas específicas de proteção ambiental por câmera em tempo real.
- Controlar a operação mecânica inteligente da rede, via aprendizagem de máquina e análise dos dados;
- Analisar e diagnosticar o *status* de proteção de áreas específicas de forma sinérgica com os demais oráculos;

3.3 Considerações

Após aplicado o protocolo de revisão da literatura utilizado para nortear o processo de busca, seleção e classificação dos trabalhos, foram selecionados, segundo o método, os mais relevantes, que passaram por um processo de descrição e análise crítica.

Como demonstrado na Tabela 4, o primeiro aspecto avaliado foi identificar em qual plataforma blockchain as soluções propostas foram desenvolvidas. O segundo aspecto foi avaliar se as propostas utilizaram oráculos blockchain em suas soluções, e caso, sim, qual tipo de oráculo foi implementado. Outro fator importante levantado foi a verificação da existência de padrões regulatórios e legislações sustentáveis através da implementação de *smart contracts* e a análise de informações por órgãos reguladores certificados. Em relação às estratégias sustentáveis, os artigos foram classificados conforme como as soluções tratam as questões relacionadas à sustentabilidade. As principais estratégias foram definidas na Seção 3.1.1.

Tabela 4: Avaliação dos principais artigos

	Plataforma	Oráculo	Padrões Regulatórios via Contratos Inteligentes	Estratégia Sustentável
Munoz et al. 2021	Ethereum	Oráculo Humano	Sim	A
Whitaker et al. 2020	Não Informado	Não	Não	A, C
Yan et al. 2019	Não Informado	Oráculo de Hardware / Oráculo Computacional	Sim	B
Fu et al. 2018	Não Informado	Não	Não	E
Shahidehpour el al. 2020	Não Informado	Não	Não	D
Chen et al. 2019	Não Informado	Oráculo Computacional	Não	D
Song et al. 2020	Não Informado	Não	Não	A
Balanescu et al. 2020	Não Informado	Oráculo de Hardware	Não	B
Woo et al. 2021	Não Informado	Não	Sim	E
Shu et al. 2021	Ethereum	Não	Sim	E

Neste capítulo foi realizado um estudo secundário em busca de avaliar e identificar os resultados pertinentes ao tópico da pesquisa. O método de busca aplicado buscou evitar

a duplicidade de esforços e a repetição de erros cometidos previamente por outros autores. Para garantir que o processo fosse conduzido de forma apropriada, ele foi realizado com auxílio de um protocolo bem estabelecido, caso contrário, o viés dos pesquisadores poderia levar a resultados pouco confiáveis, pouco abrangentes e tendenciosos.

No próximo capítulo serão apresentados os métodos e procedimentos adotados no desenvolvimento da proposta, como a arquitetura, as tecnologias, o funcionamento e as ferramentas utilizadas para a implementação da solução.

4 METODOLOGIA

A solução proposta aborda a construção de uma plataforma que potencialize e viabilize o processo de financiamento de projetos ambientais. Tal plataforma exige um esforço multidisciplinar e envolve diversas etapas e artefatos desenvolvidos durante o desenvolvimento deste projeto.

A primeira etapa, consistiu no desenvolvimento de uma moeda digital, a Standing Forest Coin (SFC), que utilizando a plataforma blockchain Ethereum e a implementação de contratos inteligentes, foi capaz de realizar transações referentes a possíveis financiamentos de projetos voltados à proteção ambiental, com mecanismos para assegurar o cumprimento desses projetos de forma transparente e auditável. Mediante o uso de ferramentas de verificação foi possível analisar os fluxos das transações entre as carteiras digitais envolvendo os principais *stakeholders*, o financiador e o prestador de serviços ambientais (proprietário de terra). A arquitetura do projeto conta com o conceito de oráculo para informar aos contratos inteligentes como se comportar em relação aos ativos captados referentes a determinado projeto:

- Caso comprovada a preservação da área no qual o prestador de serviços ambientais se comprometeu a conservar, o contrato transfere o total investido no projeto para a carteira do prestador;
- Caso não seja comprovada a preservação da área no qual o prestador de serviços ambientais se comprometeu a conservar, o contrato transfere o total arrecadado no projeto para a carteira de um fundo de preservação que decidirá quais outros projetos financiar;

Nesse primeiro momento (ao longo da participação na disciplina de blockchain), que envolve a implementação da moeda digital, a presença do oráculo não era uma realidade desenvolvida; em vez disso, era simulado por meio de um objeto *mock*. Essa abordagem permitiu a execução simulada das condições definidas nos contratos inteligentes, mas ainda não representava o funcionamento completo da estrutura idealizada para a Standing Forest Coin.

A transição para o segundo momento, marcado pela etapa de concepção deste trabalho, é caracterizada pela elaboração de uma proposta para o oráculo previamente simulado. O oráculo propõe viabilizar de forma concreta e automatizada a execução dos contratos inteligentes implementados na Stading Forest Coin. Por meio da análise comparativa de imagens via satélite, verificando a área verde preservada, o oráculo levará as informações necessárias para a execução dos contratos inteligentes, trazendo dados "*off-chain*" para dentro do blockchain. Consequentemente, o oráculo é responsável por

determinar e informar aos contratos se as respectivas áreas dos projetos de preservação ambiental foram realmente conservadas e dessa forma beneficiar ou não os proprietários de terra que se comprometeram a proteger determinar área.

4.1 Stading Forest Coin

Para esse trabalho optou-se pelo uso de uma rede blockchain pública ao invés de um blockchain privado, visando tornar todas as transações e informações dos projetos e da moeda transparentes. Dentre os blockchains públicos, foi escolhido para este projeto o Ethereum, devido ao seu tempo de atividade e suas características intrínsecas que permitem a concepção desse projeto. O Ethereum é um blockchain de código aberto que permite a criação de contratos inteligentes e aplicações descentralizadas, que tem como moeda digital o Ether, que pode ser obtido por meio da mineração ou por transações [4].

Blockchains públicos garantem a visualização de toda e qualquer transação realizada, bastando para isso ter acesso ao livro-razão. Através de aplicativos de exploração de blocos, é possível visualizar em tempo real as transações e contratos executados na rede Ethereum. O *software* mais comum para a exploração de blocos da rede Ethereum é o Etherscan, que funciona direto do navegador de Internet. Isso permite que qualquer pessoa consiga visualizar de forma transparente as transações realizadas com a SFC.

A SFC tem como característica ser transacionável em mercados financeiros, sendo registrada em corretoras de câmbio internacionais. Assim, o caminho de troca da moeda é claro, sendo possível a troca da mesma por Ether (do blockchain Ethereum), que pode também ser trocada por diversas outras moedas, incluindo por moedas fiduciárias, como o dólar.

Por ser transacionável em mercados financeiros, a proposta da moeda é que o contribuinte da moeda teria um ativo, podendo eventualmente ter um ganho de capital decorrente desse ativo com o decorrer do tempo, tendo também a opção de financiar as atividades de valoração de ecossistemas ameaçados.

O diagrama de casos de uso da Figura 30 [5] representa o cenário no qual um investidor, seja ele pessoa física ou um fundo de investimentos, compra SFC e escolhe em qual projeto de preservação, envolvendo uma propriedade de terra, investirá. Ao término do contrato, o oráculo será consultado pelo contrato para que o proprietário de terra seja pago ou as criptomoedas sejam transferidas para gestão do fundo de preservação.

Na Figura 31 [5], o diagrama de casos de uso retrata outro caso proposto no qual um investidor compra uma certa quantidade de SFC e decide não investir em nenhum contrato específico. Como a criptomoeda proposta tem o propósito de ser uma ferramenta de doação, a cada ano é descontado 5% do valor total da carteira do investidor para ser

doado para fundo de preservação.

Dessa forma, o fundo assume a responsabilidade de gestão dos recursos e deve escolher como e em qual contrato inteligente o dinheiro do fundo deve ser investido da mesma forma como acontece no caso de uso da Figura 30.

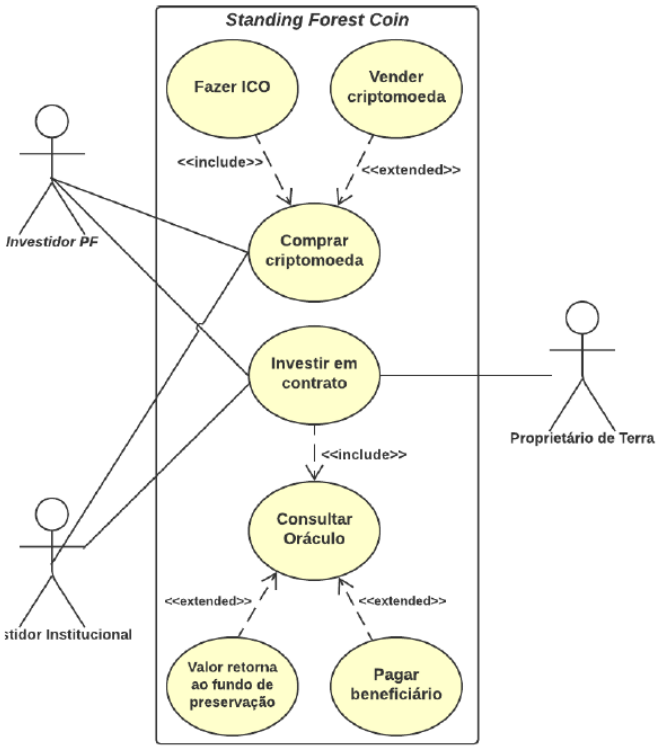

Figura 30: Caso de uso de compra e investimento em contrato

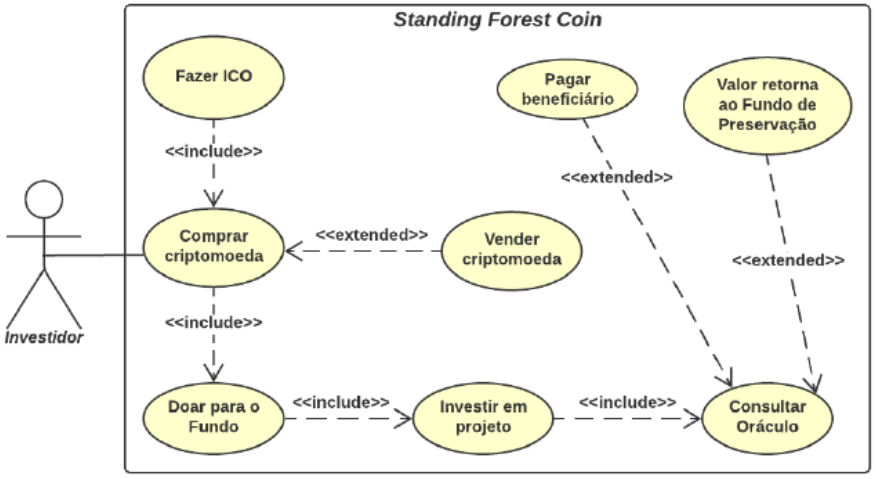

Figura 31: Caso de uso de compra e doação ao fundo de preservação

O diagrama de sequência apresentado na Figura 32 [5] representa os fluxos presentes na implementação da SFC:

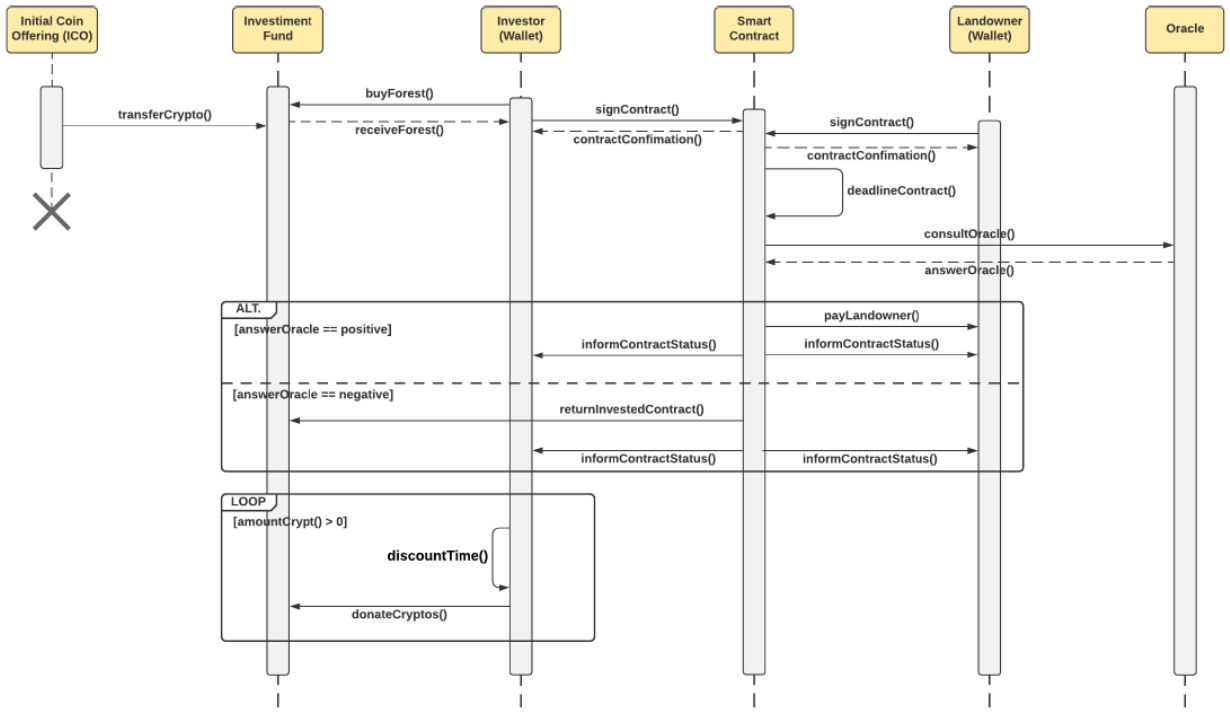

Figura 32: Diagrama de sequência dos fluxos presentes na implementação da SFC

O primeiro fluxo representa o ICO (*Initial Coin Offering*) da criptomoeda SFC. A partir do ICO o valor será transferido para a carteira do fundo de preservação que representa a carteira zero para a SFC. A partir da carteira "gênesis", qualquer investidor em qualquer lugar do mundo poderá comprar determinado valor em SFC. Com suas criptomoedas em sua carteira, ele decidirá se irá investir em um contrato de preservação específico ou se seus tokens serão doados anualmente ao fundo de preservação.

Quando um contrato de preservação é definido entre as duas partes, no caso o investidor e o guardião da terra, o valor definido pelo investidor é transferido para uma terceira carteira que representa o contrato inteligente. Após o vencimento do contrato, o oráculo será consultado para verificar se a preservação da terra foi ou não respeitada. Em caso positivo, o valor do contrato vai para a carteira do guardião da terra e em caso negativo para a carteira inicial do fundo de preservação.

Como passo inicial para implementação do projeto, foram escritos os contratos inteligentes da moeda SFC utilizando a IDE Remix, que permite desenvolver, compilar e depurar códigos na linguagem de programação Solidity.

Solidity é uma linguagem de programação de alto nível, influenciada por outras linguagens populares como Python e JavaScript, desenvolvida visando permitir a implementação de *smart contracts*, e tem como alvo a Máquina Virtual Ethereum (EVM), permitindo assim a construção de contratos para a rede Ethereum. Os contratos da SFC

foram desenvolvidos em Solidity, na versão 0.5.17 da linguagem.

Na Figura 33 é apresentada uma amostra do código do contrato inteligente da SFC escrito em Solidity utilizando a IDE Remix.

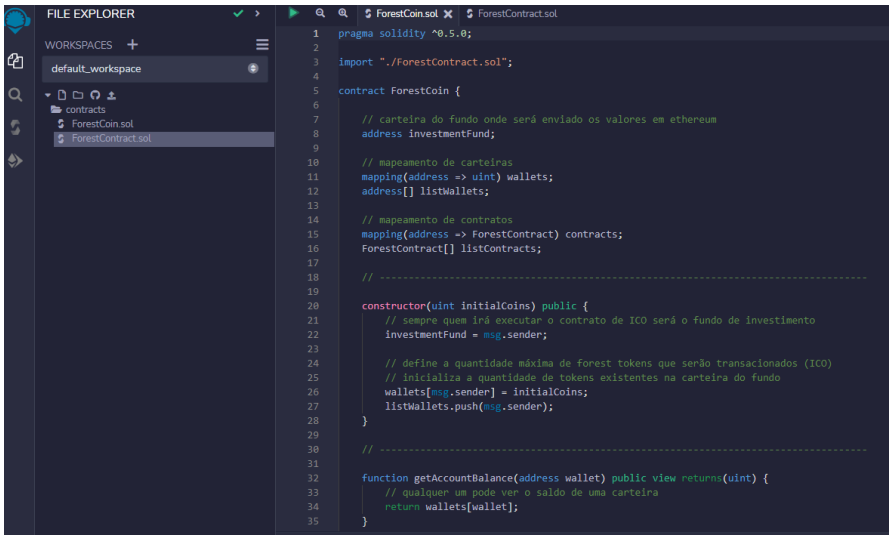

Figura 33: Código-fonte do contrato inteligente da SFC

As Figuras 34 e 35 demonstram os códigos-fonte do contrato inteligente relacionados as lógicas envolvendo os pagamentos aos proprietários de terra via consulta ao oráculo.

```
// ~> Pagar beneficiário
function payForestContractLandOwner(ForestContract _contract) private {
    // obtemos o saldo atual do contrato
    uint tokens = _contract.getContractBalance();

    // obtemos o endereço da carteira do dono de terra do contrato
    address _landowner = _contract.getContractLandOwner();

    // fazemos a retirada do valor
    _contract.withdrawalContractBalance(tokens);

    // adicionamos o valor à carteira do dono de terra
    wallets[_landowner] += tokens;
}

// ~> Retornar valor ao Fundo de Preservação
function returnForestContractInvestment(ForestContract _contract) private {
    // obtemos o saldo atual do contrato
    uint tokens = _contract.getContractBalance();

    // fazemos a retirada do valor
    _contract.withdrawalContractBalance(tokens);

    // retornamos o valor arrecadado à carteira do fundo
    wallets[investmentFund] += tokens;
}
```

Figura 34: Funções de pagamento ao proprietário de terra e ao fundo de preservação

```
// ~> Consultar Oráculo
function processForestContract(address contractAddress) public returns(bool) {
    // obtemos o contrato
    ForestContract _contract = contracts[contractAddress];

    // obtemos a carteira do dono de terra desse contrato
    address _landowner = _contract.getContractLandOwner();

    // essa função deve ser executada pelo dono de terra
    require(
        msg.sender == _landowner,
        "Apenas o dono do contrato pode processar um contrato"
    );
    require(
        _contract.isContractRunning(),
        "O contrato já foi finalizado. Não pode ser processado"
    );

    // consulta o oráculo
    _contract.mockConsultOracle();

    if (_contract.isContractAccomplished()) {
        // se conseguiu, então pagamos o dono do contrato
        payForestContractLandOwner(_contract);
    } else if (_contract.isContractCancelled()) {
        // se não conseguiu, retornamos o valor para o fundo
        returnForestContractInvestment(_contract);
    }

    // retorna se o contrato foi cumprido ou não
    return _contract.isContractAccomplished();
```

Figura 35: Chamadas das funções de pagamento conforme a resposta do oráculo sobre a preservação da área

Após a escrita dos contratos e a sua compilação usando a IDE, foi realizado o processo de *deployment* destes, na rede de testes do Ethereum. Esta etapa foi realizada na própria Remix IDE integrando-a com a carteira MetaMask.

O MetaMask é uma carteira digital que permite gerenciar várias criptomoedas, por meio de uma extensão nos principais navegadores de Internet. No MetaMask foi criada uma carteira para os testes do projeto utilizando a rede de testes Ropsten e então os contratos foram publicados com uma quantidade de 4.000 moedas para o ICO.

Com o contrato publicado na rede Ethereum, fez-se necessária a verificação dos contratos e dos arquivos-fontes do projeto utilizando o serviço EtherScan, como apresentado na Figura 36 [5]. O EtherScan é um mecanismo de busca que permite pesquisar transações que ocorreram na rede Ethereum e que possibilita a interação direta com os contratos.

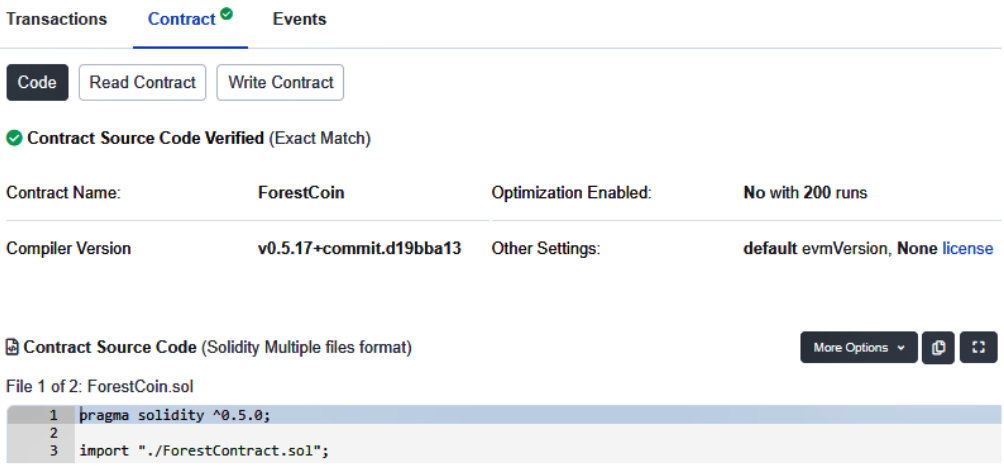

Figura 36: Verificação dos contratos com o EtherScan

A verificação de um *smart contract*, bem como dos arquivos-fonte, é uma etapa importante para garantir que o código dos contratos publicados são exatamente aqueles mesmos que foram desenvolvidos, e permitir que o contrato seja lido e auditado publicamente por qualquer indivíduo.

Após os contratos e o código-fonte verificados, torna-se possível a interação com as funções de leitura/escrita do contrato utilizando o EtherScan. Para isso é necessário, além de estar com o contrato aberto, também fazer a conexão ao EtherScan com um gerenciador de carteira digital, no nosso caso, o MetaMask. Na Figura 37 [5], é realizada a primeira interação com o contrato publicado, a compra de 120 tokens da carteira do Fundo de Investimentos diretamente para a carteira de um investidor.

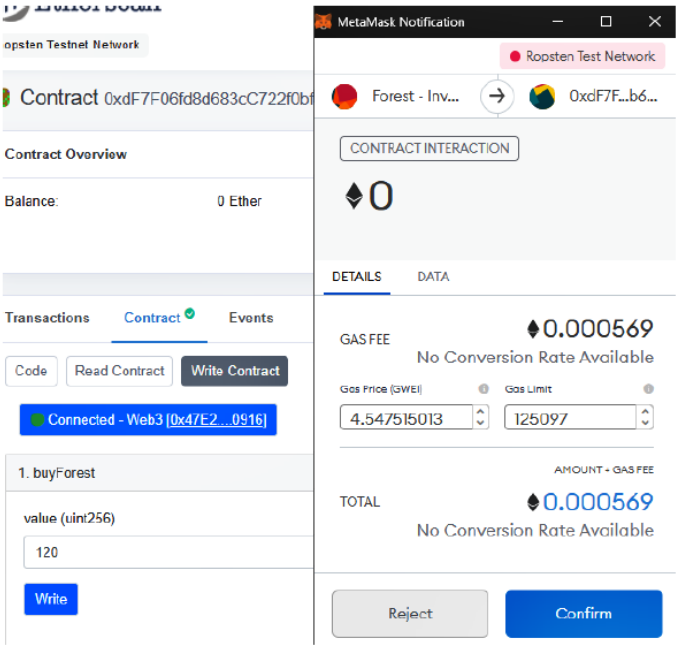

Figura 37: Aquisição de SFC utilizando o MetaMask

Com a compra dos tokens, foi realizada a simulação da criação de um contrato do proprietário de terra (contrato esse que receberá os financiamentos), tomando como entrada o endereço da carteira do proprietário de terra. Depois do contrato criado, o EtherScan exibe o endereço do contrato criado, que pode ser usado, por exemplo, pelo proprietário da terra para requerer investimentos. Uma vez gerado o contrato, ele permanece público na rede Ethereum.

Após a geração do contrato, foi realizada a transferência de 50 SFCs entre a carteira de um financiador e o contrato do prestador de serviços ambientais (proprietário de terra). Por meio do EtherScan, o investidor recebe uma confirmação da doação e pode acompanhar o estado do contrato usando o endereço do contrato do proprietário de terra.

A Figura 38 [5] apresenta o processamento do contrato do proprietário de terra. O contrato, via consulta ao oráculo, verifica se a área foi de fato preservada para liberar ou não o valor arrecadado no contrato. Oráculos, em blockchain, são serviços que fornecem informações externas aos *smart contracts*, atuando como uma "ponte"entre o blockchain e o mundo exterior. Neste primeiro momento, o oráculo foi "simulado", com intuito de demonstrar o funcionamento real do contrato.

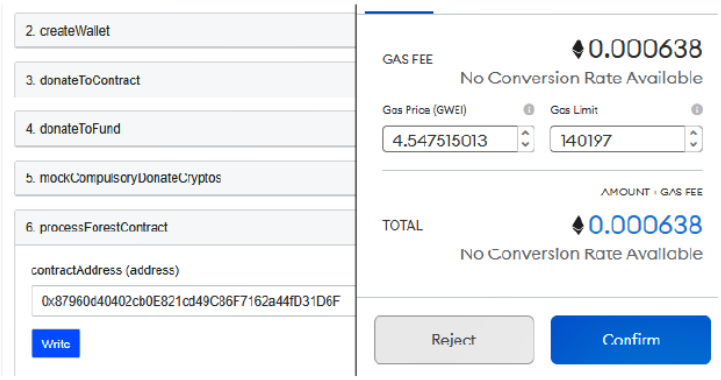

Figura 38: Processamento do contrato do proprietário de terra

Na Figura 39 [5], é exibido o resultado do processamento do contrato, onde o proprietário de terra conseguiu cumprir com o acordo de preservação segundo o oráculo e recebeu o valor investido no contrato, no caso 50 SFCs.

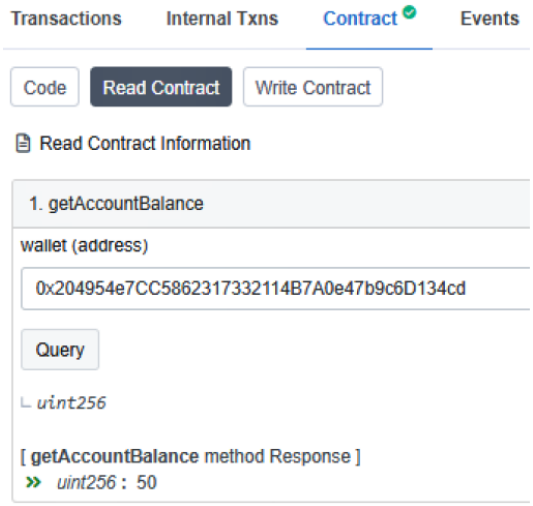

Figura 39: Resultado do processamento do contrato

O contrato do proprietário de terra, após a validação e movimentação dos tokens, se torna inativo. As transações realizadas com SFC (verificações, processamentos, consultas ao oráculo, etc.) são mantidas todas na rede Ethereum, podendo ser visualizadas por qualquer mecanismo de busca, como o EtherScan, mantendo assim a transparência nas operações realizadas.

4.2 Oráculo

A visão computacional é uma área da ciência da computação que busca analisar, interpretar e extrair informações relevantes de imagens e/ou vídeos para que decisões possam ser tomadas, ou para gerar dados relevantes para uma aplicação futura.

Uma etapa importante no processamento de imagens digitais para analisar o que está dentro dela é classificar seu conteúdo de píxeis em diferentes grupos. Esse processo é chamado de segmentação de imagem, cujo objetivo é transformar a representação de uma imagem em algo mais significativo e fácil de analisar.

Para o desenvolvimento da estrutura do oráculo blockchain que fosse capaz de realizar o mapeamento e monitoramento da cobertura florestal sugerido pela solução da SFC, foi escolhida a arquitetura U-Net, uma estrutura de rede neural profunda amplamente utilizada em tarefas de processamento digital de imagens, especialmente em tarefas de segmentação semântica.

A aplicação de um oráculo blockchain utilizando a abordagem U-Net para o mapeamento de cobertura florestal envolve a integração de duas tecnologias poderosas: a segmentação de imagens com a U-Net e a tecnologia blockchain para garantir a autenticidade e rastreabilidade dos dados. A criação de um oráculo blockchain utilizando a

abordagem U-Net para o mapeamento de cobertura florestal envolve uma série de etapas metodológicas para garantir a precisão, integridade e transparência dos dados.

Para isso se fez necessário a coleta de imagens de satélite de alta resolução da área de interesse e a criação manualmente de máscaras de cobertura florestal para um subconjunto das imagens selecionadas. Essas máscaras foram utilizadas como "*ground truth*"para treinar a U-Net.

Com os contratos inteligentes criados no blockchain, as funções que compõe esse contrato poderão consultar o oráculo quando assim for necessário para verificar e mapear a atual situação de cobertura vegetal da região de interesse. O oráculo ao ser chamado ira mapear a cobertura florestal, e ao realizar essa consulta periodicamente, o oráculo poderá identificar a preservação ou deterioração daquela região e informar ao contrato inteligente o resultado referente ao processamento dessas imagens, executando assim, as condicionais existentes codificadas no contrato e beneficiando ou não o prestador de serviços ambientais que se comprometeu a preservar aquela região.

Na sessão de planejamento experimental, a seguir, serão detalhados aspectos do oráculo referentes as abordagens utilizadas para o aprendizado de máquina, redes neurais convolucionais e a arquitetura de segmentação semântica (binária) que compõe a construção da solução.

4.3 Considerações

Nesta seção foram abordados os métodos e procedimentos adotados no desenvolvimento de artefatos de *software* resultantes da implementação da proposta de solução que visa promover e facilitar o processo de financiamento de projetos ambientais.

No próximo capítulo será descrito o projeto de experimentos condizidos para avaliar a solução proposta do oráculo idealizado para a Standing Forest Coin e testar a hipótese gerada no Capítulo 1. Esse projeto de experimentos inclui a definição, o planejamento e a execução do experimento para testar e avaliar a hipótese geral.

5 PLANEJAMENTO EXPERIMENTAL DO ORÁCULO

Este capítulo apresenta um planejamento de experimentos com o proposito de avaliar a solução idealizada para um oráculo blockchain que possa viabilizar a SFC. Experimentos, de acordo com Wohlin [52], são investigações sistemáticas, abrangentes e regulamentadas, cujo objetivo principal é testar teorias ou hipóteses. Em geral, eles são executados em ambientes controlados, como laboratórios, e manipulam uma ou mais variáveis nesse ambiente, mantendo todas as outras em um ponto fixo.

O planejamento experimental é uma abordagem sistemática para projetar e conduzir experimentos para obter resultados válidos e confiáveis. Quando aplicado ao contexto de aprendizado de máquina e redes neurais, o planejamento experimental envolve várias etapas-chave, desde a aquisição de dados até a avaliação do modelo. Visando estruturar os elementos dos experimentos, neste trabalho o processo foi definido da seguinte forma para realização dos estudos experimentais, como mostra a Figura 40:

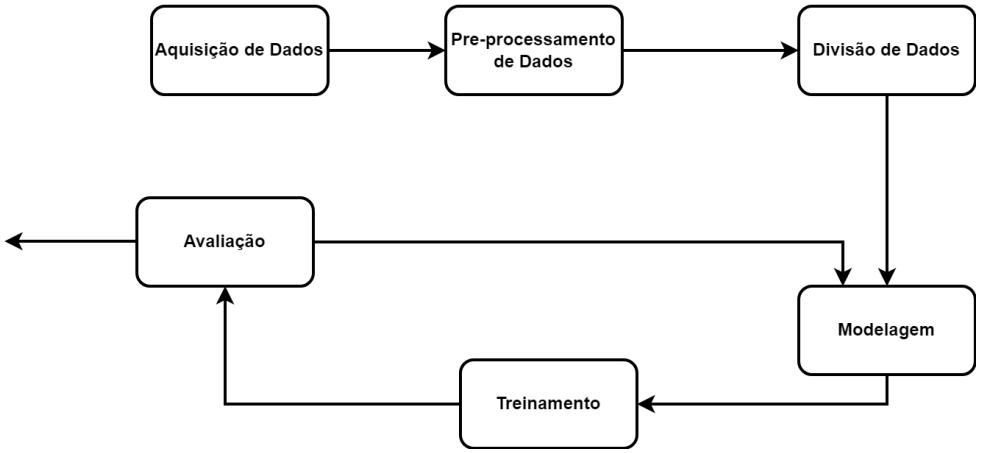

Figura 40: Visão geral do experimento

O planejamento experimental aplicado ao contexto de aprendizado de máquina e redes neurais envolve uma abordagem sistemática para projetar e conduzir experimentos visando obter resultados válidos, confiáveis e otimizados para a tarefa específica em questão. Esse processo pode ser dividido em várias etapas, cada uma desempenhando um papel crucial no desenvolvimento de um modelo eficiente. Estas etapas são descritas a seguir.

- **Aquisição de dados:** Nesta etapa, você deve obter os dados necessários para treinar e testar seu modelo. Isso pode envolver a coleta de dados brutos de fontes diversas, como bancos de dados, APIs, arquivos CSV, ou qualquer outra fonte que for relevante para o problema em questão. É importante garantir que os dados sejam de qualidade e representativos do problema que você deseja resolver;

- **Pré-processamento de dados:** Antes de alimentar os dados ao modelo, é necessário realizar o pré-processamento para torná-los adequados ao treinamento. Essa etapa pode incluir:
  - **Limpeza de dados:** Remover dados duplicados, ausentes ou inconsistentes;
  - **Transformação de dados:** Normalizar ou padronizar os dados para estarem em uma escala similar e facilitem o treinamento.
  - **Tratamento de *outliers*:** Lidar com valores extremos que podem afetar negativamente o modelo.
  - **Codificação de dados categóricos:** Converter variáveis categóricas em formatos numéricos;
- **Divisão de dados:** Após o pré-processamento, é fundamental dividir os dados em conjuntos de treinamento, validação e teste. Isso é feito para evitar o *overfitting*, garantindo que o modelo generalize bem para novos dados. A divisão típica pode ser 70-80 porcento para treinamento, 10-15 porcento para validação e 10-15 porcento para teste;
- **Modelagem** Nesta etapa, é escolhido o tipo de modelo que será usado para resolver o problema em questão. Isso pode incluir redes neurais, algoritmos de aprendizado de máquinas tradicionais (como regressão linear, SVM, etc.), ou até mesmo combinações de diferentes modelos (modelos *ensemble*). A escolha do modelo dependerá das características dos dados e da natureza do problema;
- **Treinamento:** Com os dados de treinamento preparados e o modelo selecionado, é hora de treinar o modelo. Isso envolve alimentar os dados de treinamento ao modelo e ajustar os parâmetros do modelo para minimizar a função de perda (ou erro). O treinamento geralmente envolve várias épocas, e cada época é composta por vários passos. A validação é uma parte crítica do processo de treinamento. Após cada época de treinamento, o modelo é avaliado em um conjunto de validação para verificar seu desempenho em dados não vistos durante o treinamento. Isso ajuda a monitorar o *overfitting* e ajustar hiperparâmetros do modelo, se necessário, para melhorar o desempenho do modelo;
- **Avaliação:** A avaliação é crucial para determinar o desempenho e a eficácia do modelo treinado. Esta etapa visa medir como o modelo se sai em dados que não foram usados durante o treinamento, fornecendo uma visão realista de sua capacidade de generalização. São utilizadas métricas específicas para avaliar o desempenho do modelo. As métricas variam conforme o tipo de problema;

O planejamento experimental no contexto de aprendizado de máquina e redes neurais é uma tarefa iterativa e requer experimentação cuidadosa para chegar ao melhor modelo possível para a tarefa específica em questão. Cada etapa deve ser bem planejada e executada para garantir resultados confiáveis e uma solução bem ajustada para o problema.

5.1 Definição do Problema e Objetivos

O problema que esse trabalho pretende resolver consiste em verificar e mapear áreas de desmatamento na região amazônica com alta precisão e eficiência, desenvolvendo um sistema de inteligência artificial que possa identificar automaticamente as áreas de desmatamento em imagens de satélite, permitindo uma detecção e monitoramento mais eficientes dessas atividades.

O objetivo do experimento é criar um modelo de segmentação binária usando técnicas de aprendizagem profunda (redes neurais convolucionais - CNNs) para classificar píxeis em imagens de satélite como pertencentes a áreas preservadas ou não preservadas. A segmentação binária é uma abordagem onde cada píxel na imagem é classificado como pertencente a uma das duas classes: "desmatamento"ou "não desmatamento".

O estudo pretende avaliar o desempenho da U-Net, arquitetura conhecida por suas vantagens de captura de informações contextuais, conexões de salto eficazes e desempenho sólido em tarefas de segmentação semântica, na identificação e segmentação da cobertura florestal e fragmentos florestais na região amazônica e avaliar sua capacidade de vigilância e monitoramento das mudanças na cobertura florestal de forma eficaz e escalável.

O resultado esperado é um modelo de segmentação binária treinado com alta acurácia, que pode ser usado para processar imagens de satélite e destacar as áreas de desmatamento na região amazônica. Esse modelo pode servir para monitorar continuamente o desmatamento, identificar áreas críticas, auxiliar governos e organizações ambientais em esforços de conservação e fornecer informações valiosas para estudos e pesquisas relacionadas ao impacto ambiental na região. A detecção precisa do desmatamento pode contribuir significativamente para a preservação da floresta amazônica e para ações de proteção ambiental.

Além disso, o experimento visa utilizar a tecnologia blockchain para validar e armazenar de forma segura as informações sobre as áreas de desmatamento identificadas pelo modelo. Neste contexto, o blockchain pode ser usado como um mecanismo para garantir a integridade dos dados sobre o desmatamento identificado pelo modelo e para fornecer uma fonte confiável e transparente para as informações geradas pelo sistema.

5.2 Aquisição de Dados

O presente trabalho utiliza o *dataset* previamente compilado pela pesquisadora Lucimara Bragagnolo (2021) [53], por meio de uma biblioteca Sentinelhub, disponível em Python, que permite o download das imagens do satélite Sentinel-2 em formato GeoTIFF Uint16. Este *dataset*, proporciona uma base sólida para o treinamento de modelos de visão computacional voltados para a segmentação semântica, contendo imagens do bioma brasileiro da Amazônia, utilizadas para treinar uma rede neural totalmente convolucional para a segmentação semântica de áreas florestais em imagens do satélite Sentinel-2 Nível 2A.

As imagens referem-se à composição das bandas 4 (Vermelho), 3 (Verde), 2 (Azul) e 8 (NIR). Cada banda foi convertida para um tipo de byte (0-255). As três bandas (4, 3 e 2) são parte do espectro visível e são frequentemente utilizadas em imagens coloridas para visualização. A combinação dessas bandas, conhecida como a composição colorida natural (RGB), permite a criação de imagens que se assemelham à visão humana, facilitando a interpretação visual das características do terreno. Essa combinação produz uma imagem colorida onde diferentes tipos de cobertura vegetal, corpos d'água e áreas urbanas podem ser facilmente distinguíveis

Já a banda NIR (*Near Infrared*) é uma faixa de comprimento de onda próxima à região do infravermelho, não visível aos olhos humanos, mas essencial para a detecção de características da vegetação, como saúde das plantas e estresse hídrico. A luz NIR é altamente refletida pelas folhas das plantas e é pouco absorvida pela clorofila. Portanto, quanto mais vigorosa e saudável estiver a vegetação, maior será a reflexão da luz NIR.

A combinação das bandas RGB com a banda NIR em análises de sensoriamento remoto permite extrair informações detalhadas sobre a superfície terrestre, possibilitando o monitoramento ambiental, agrícola e outros estudos importantes para a compreensão e preservação do nosso planeta. Através dessas bandas é possível obter o NDVI (*Normalized Difference Vegetation Index*), um indicador muito utilizado em sensoriamento remoto para avaliar a saúde e a quantidade de vegetação em uma determinada área. O NDVI é calculado a partir dos comprimentos de onda NIR e do vermelho (*Red*) de imagens de satélite ou de outros sensores.

A fórmula para calcular o NDVI é a seguinte:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{1}$$

Onde o valor do NDVI varia de -1 a +1, onde, valores próximos de -1 indicam uma área sem vegetação, como água ou solo exposto e valores próximos de +1 indicam uma área com vegetação muito densa e saudável.

O NDVI é uma ferramenta valiosa para analisar a vegetação de forma objetiva e quantitativa, possibilitando tomar decisões informadas em relação ao manejo de recursos naturais e ao monitoramento da saúde do ecossistema.

5.3 Pré-Processamento de Dados

Também segundo Bragagnolo (2021) [54], para compor os conjuntos de treinamento, validação e teste, foram utilizadas imagens com oclusão máxima de nuvens de 30 porcento ou menos, permitindo a criação de máscaras de forma automatizada através do uso de algoritmos e técnicas de aprendizagem não supervisionada, que permitem identificar padrões e estruturas relevantes nos dados sem a necessidade de rótulos ou supervisão humana, explorando a estrutura intrínseca dos dados sem a orientação externa. Cada cena foi subdividida em sub-imagens, blocos de 512 x 512 píxeis, para facilitar o processamento e permitir a aplicação de processamento paralelo. Para criação das máscaras através da aprendizagem não supervisionada, foram utilizadas as técnicas:

- **Segmentação por agrupamento (K-Means):** Nessa abordagem, os dados são agrupados em diferentes *clusters* com base em suas similaridades, de forma que os pontos dentro de cada grupo sejam mais semelhantes entre si e diferentes dos pontos em outros grupos. Cada cluster representa uma classe ou grupo sem que haja a necessidade de rótulos prévios. Na segmentação de imagens, cada cluster pode ser associado a uma região específica da imagem, formando uma máscara para cada grupo;
- **NDVI:** Nessa técnica há a segmentação de regiões de interesse em uma imagem, destacando áreas com vegetação saudável e diferenciando-as de outras áreas não vegetadas;
- **Modelo de Mistura Gaussiana:** Envolve a segmentação de dados em diferentes regiões usando uma combinação de distribuições gaussianas para modelar cada região;

As máscaras candidatas geradas pelos algoritmos citados foram analisadas e as mais representativas para cada imagem ao identificar as regiões de interesse foram manualmente refinadas. Dessa, forma, cada sub-imagem da base de dados tem uma máscara associada, que é uma matriz de 0s e 1s indicando a localização das porções de floresta em cada sub-imagem. As sub-imagens foram então rotuladas manualmente em duas classes de terra uso: florestal e não florestal. A classificação supervisionada foi realizada por meio da ferramenta GRASS-GIS 7.6.1. As imagens resultantes foram exportadas para o formato PNG para serem usadas como entrada no processo de treinamento da U-Net. As

ilustrações de uma amostra de imagem em cores verdadeiras e sua máscara são mostradas na Figura 41 [54].

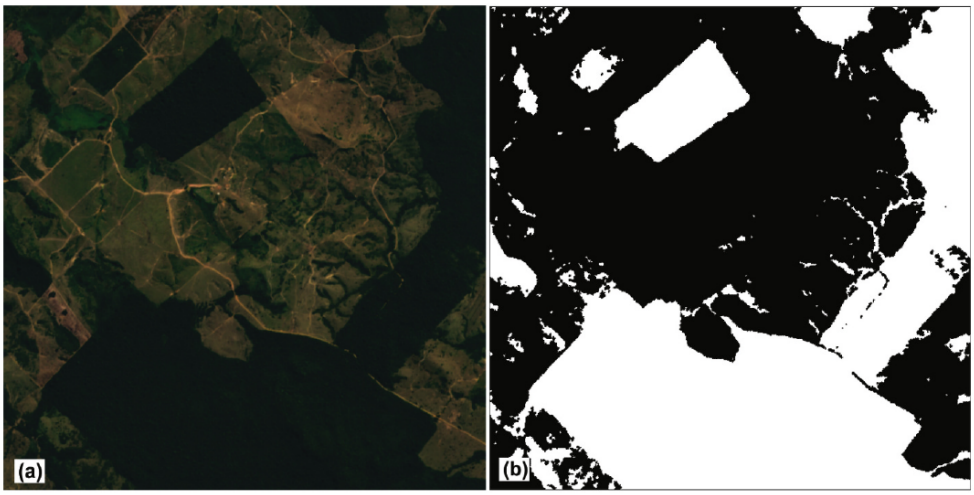

Figura 41: Amostra de uma imagem em cores verdadeiras e sua máscara

5.4 Divisão de Dados

O banco de dados de treinamento foi gerado a partir da subdivisão das cenas de satélite originais. Neste procedimento, sub-imagens foram geradas com dimensões de 512 × 512 píxeis contendo regiões florestais e as respectivas máscaras. Foram selecionadas 499 imagens utilizadas para treinar as CNNs e 100 para validação. Outras 20 imagens foram aplicadas na etapa de testes.

As 499 imagens do conjunto de dados de treinamento foram submetidas ao processo de ampliação de dados, de modo que novas imagens foram geradas por meio de rotações, mudanças em relação à altura, largura e ângulo de cisalhamento, alargamento, (*zoom*) e inversão das sub-imagens originais. Ainda, antes das imagens serem usadas como entrada no algoritmo de treinamento, elas passaram por um processo de normalização em que seus valores foram normalizados dentro do intervalo [0,1].

5.5 Modelagem

Conforme explicado por Bragagnolo (2021) [54], a escolha da modelagem utilizando a arquitetura U-Net visa atacar uma das principais limitações de se trabalhar com CNNs, a redução da resolução dos recursos extraídos devido às operações consecutivas de agrupamento. A arquitetura U-Net é uma estrutura de rede neural profunda amplamente utilizada em tarefas de processamento de imagens, especialmente em tarefas de segmentação semântica. Foi proposta pela primeira vez em 2015 por Olaf Ronneberger, Philipp Fischer e Thomas Brox na Universidade de Friburgo, Alemanha [55].

A U-Net recebe esse nome devido à sua forma simétrica em "U", que é uma de suas características distintivas. A arquitetura da U-Net, representada pela Figura 42 [54], consiste em duas partes principais:

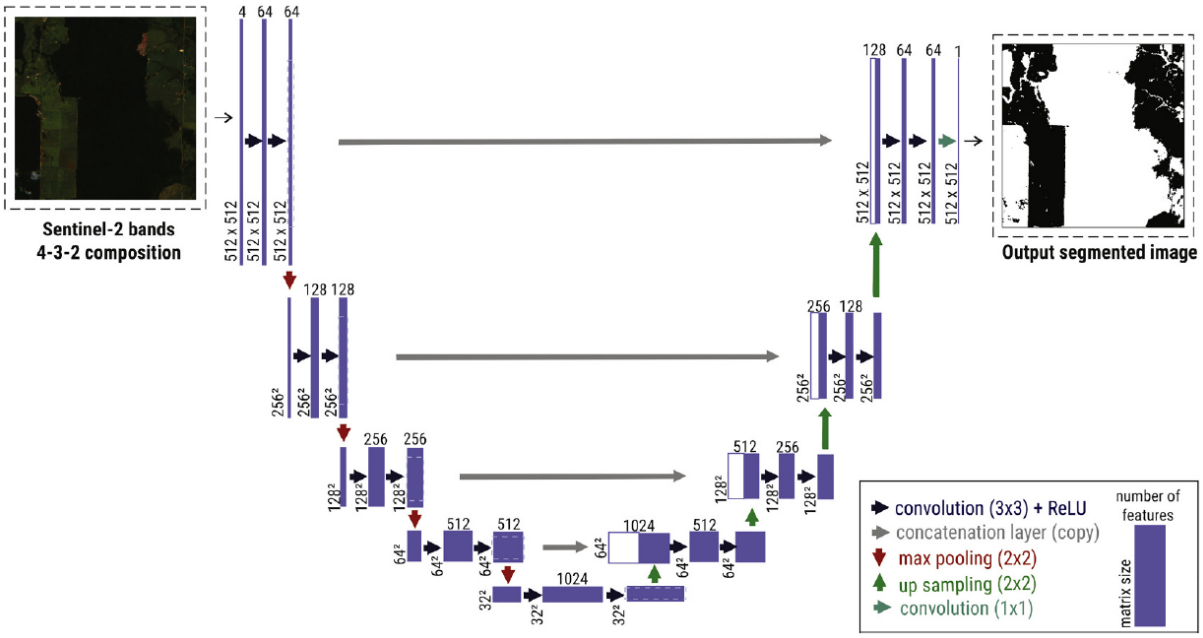

Figura 42: Arquitetura U-Net

- ***Encoder* (Codificador):** A primeira metade da rede, conhecida como codificador, é uma típica rede neural convolucional que realiza a tarefa de codificação. É composta por várias camadas convolucionais seguidas por camadas de *pooling* (geralmente *max pooling*). O objetivo do codificador é capturar características de alto nível da imagem e reduzir a resolução espacial ao longo da rede;

- ***Decoder* (Decodificador):** A segunda metade da rede, conhecida como decodificador, é responsável por reconstruir a segmentação da imagem a partir das características aprendidas pelo codificador. Ela é simétrica ao codificador, começando com camadas de *up-sampling* para aumentar a resolução espacial das características e, em seguida, utilizando camadas convolucionais para refinar a segmentação. As conexões *"skip"* são usadas para fundir informações das camadas do codificador diretamente com as camadas do decodificador, permitindo que detalhes espaciais sejam preservados durante o processo de decodificação;

As conexões *"skip"* são uma característica distintiva da arquitetura U-Net e são fundamentais para seu sucesso. Essas conexões permitem que informações de baixo nível, como detalhes finos e de borda, sejam transmitidas diretamente para camadas posteriores do decodificador, permitindo que o modelo preserve informações importantes em diferentes escalas de resolução.

A U-Net é frequentemente usada em tarefas de segmentação semântica, onde o objetivo é atribuir rótulos a cada píxel da imagem, destacando diferentes objetos ou regiões de interesse. Ela também pode ser adaptada para outras tarefas de processamento de imagens, como detecção de objetos, super-resolução e muitas outras aplicações em que a preservação de detalhes espaciais é importante.

5.6 Treinamento e Avaliação

O algoritmo Adaptive Moment Estimation (Adam), um algoritmo de otimização baseado em gradiente estocástico [56] foi usado no processo de treinamento com taxa de aprendizado definida como 0,0001. A abordagem utilizando a parada antecipada foi aplicada para evitar o *overfitting* dos parâmetros de rede. O *overfitting* ocorre quando um modelo de aprendizado de máquina se ajusta excessivamente aos dados de treinamento, memorizando os padrões específicos dos dados, em vez de aprender padrões gerais que podem ser aplicados a novos dados. Em outras palavras, o modelo se torna muito complexo para o conjunto de dados de treinamento e não consegue generalizar bem para dados não vistos, resultando em um desempenho ruim em conjuntos de teste.

Um limite de decisão de 0,5 foi aplicado para transformar as imagens em uma base binária de píxel por píxel. Assim, píxeis com refletância abaixo de 0,5 foram classificados como região não florestal e representados por 0, enquanto os píxeis com refletância igual ou superior a 0,5 foram classificados como áreas florestais e representados por 1. Este procedimento é obrigatório para permitir o cálculo das medidas de desempenho relativas as máscaras da imagem, permitindo assim o cálculo da classificação das métricas de desempenho. Para avaliar o desempenho da rede neural treinada, as métricas de acurácia, precisão, *recall* e F1-score foram calculadas individualmente para cada píxel nas imagens do conjunto de testes. Essas medidas são definidas conforme mostrado nas seguintes equações:

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{2}$$

$$Precision = \frac{Tp}{Tp + Fp} \tag{3}$$

$$Recall = \frac{Tp}{Tp + Fn} \tag{4}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

Onde Tp é o número de Verdadeiros Positivos, Tn é o número de Verdadeiros Negativos, Fp o número de Falsos Positivos e Fn o número de Falsos Negativos. Valores de precisão mais próximos de 1 indicam que a maioria das previsões foi correta. Por sua vez, alta precisão significa que cada objeto previsto como floresta é de fato uma região de floresta, independentemente de quantos áreas florestais não foram detectadas. Agora, um alto valor de *recall* significa que todas as regiões de floresta foram detectadas, independentemente de quantos objetos foram classificados erroneamente como floresta. O F1-score é a média harmônica entre precisão e revocação.

Os procedimentos de treinamento, validação e teste foram realizados por meio do uso do serviço de unidades computacionais do Google, o Notebook Colab Pro. As etapas do desenvolvimento da rede neural e seus cálculos foram realizados mediante recursos de computação com configuração máxima de 12,7 GB de RAM e 15,0 GB VRAM.

5.7 Considerações

Em síntese, este capítulo apresentou uma análise detalhada do planejamento experimental adotado para conduzir esta dissertação. A abordagem metodológica proposta é essencial para garantir a validade e a confiabilidade dos resultados obtidos na pesquisa.

É fundamental ressaltar a necessidade contínua de revisão e adaptação do plano experimental à medida que o projeto avança. As nuances que surgem durante a implementação podem exigir ajustes estratégicos para garantir a eficácia do design experimental. No próximo capítulo serão apresentados os resultados obtidos e a descrição da execução do experimento aqui detalhado.

6 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Este capítulo apresenta os resultados obtidos e descreve a execução do experimento cujo objetivo é mapear o desmatamento na região amazônica utilizando a arquitetura U-Net. A arquitetura U-Net, um modelo de aprendizado profundo amplamente reconhecido e utilizado em tarefas de segmentação de imagens, é a peça-chave do experimento realizado. Sua estrutura, que combina uma rede neural convolucional de codificação com um caminho de decodificação, permite a extração de características detalhadas das imagens de satélite e a identificação precisa das áreas desmatadas.

Ao longo deste capítulo, serão detalhados os aspectos metodológicos do experimento, incluindo detalhes das etapas de treinamento, validação e testes, e os hiperparâmetros utilizados. Serão apresentados os resultados quantitativos e qualitativos da segmentação das áreas desmatadas, evidenciando a eficácia da abordagem adotada.

Além disso, serão discutidas as vantagens e desafios encontrados ao utilizar a arquitetura U-Net nesse contexto específico, bem como as possíveis aplicações práticas dos resultados obtidos. A análise crítica dos resultados também oferecerá *insights* valiosos para o aprimoramento contínuo da metodologia e a expansão de suas aplicações em outras áreas de monitoramento ambiental.

6.1 Discussão dos Resultados

Para o treinamento da rede neural foram aplicadas 30 épocas com 1081 passos cada. O ponto de parada precoce (*early stopping*) ocorreu na época 23, prevenindo o *overfitting* e garantindo que o modelo mantivesse a capacidade de generalização e evitando treinar o modelo por um número excessivo de épocas, economizando tempo e recursos computacionais, levando a um treinamento mais eficiente.

Ao implementar a parada precoce, foi definido um "limiar de paciência"de valor igual a 20, que determina por quantas épocas consecutivas o desempenho no conjunto de validação pode não melhorar antes que o treinamento seja interrompido. Essa configuração depende do problema específico e da natureza dos dados.

Nesse momento da parada precoce, os valores médios de acurácia para cada píxel de 0,9930 e 0,9911 foram obtidos para os conjuntos de treinamento e validação, respectivamente. Outras métricas obtidas nesta sessão de treinamento foram a precisão de 0,9898, o *recall* de 0,9934 e uma pontuação de F1 de 0,9916, que representam resultados satisfatórios, pois os valores são próximos de 1 (um).

Os resultados destacam as arquiteturas U-Nets como uma abordagem robusta e

eficaz, capazes de produzir resultados de alta-fidelidade, com vastas aplicações potenciais para o monitoramento de mudanças na cobertura florestal. Visto que os satélites fornecem imagens de alta resolução que podem conter detalhes desnecessários para a identificação de polígonos de desmatamento em vastas áreas florestais, o processo de redução da resolução se apresenta como um processo vantajoso. Esse procedimento visa ajustar a resolução da imagem de modo a corresponder ao tamanho dos objetos de interesse, trazendo benefícios para a capacidade computacional dos recursos disponíveis na infraestrutura. Tal abordagem permite manter a carga computacional em níveis mais baixos, prevenindo filas e evitando sobrecargas severas.

No que se refere aos dados obtidos, visualizamos, na Figura 43, um alto valor de Verdadeiros Positivos Tp e Verdadeiros Negativos Tn em uma matriz de confusão gerada após o treinamento da rede, sendo, em geral, um indicativo positivo do bom desempenho do modelo. Ambos Tp e Tn são elementos cruciais para avaliar o desempenho de um modelo de detecção de cobertura florestal. Eles indicam a capacidade do modelo de acertar tanto nas regiões de preservação quanto nas áreas desmatadas. Ter valores altos para essas métricas sugere que o modelo está realizando uma boa discriminação entre as classes e tem uma forte capacidade de generalização para dados não vistos.

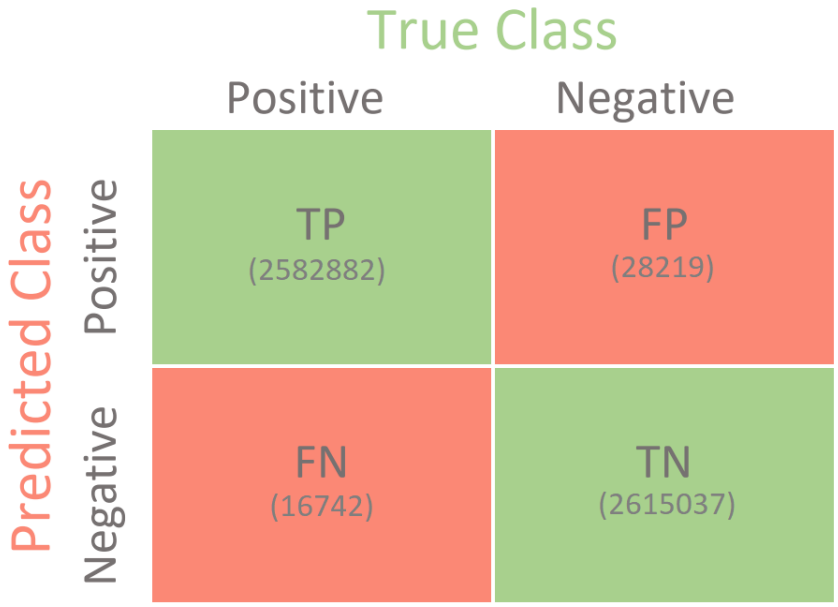

Figura 43: Matriz de confusão

Analogamente, o alance da minimização de Falsos Positivos Fp da CNN é um indicador importante porque representa a redução de alertas incorretos, evitando que áreas que não foram desmatadas sejam erroneamente classificadas como desmatadas. Reduzir Falsos Negativos Fn é igualmente significativo, pois indica que o modelo está identificando

eficientemente as regiões de desmatamento, evitando deixar de detectar áreas que foram afetadas. Ambos são indicadores positivos de um modelo de detecção de desmatamento com bom desempenho.

Foram aplicadas para cada píxel, métricas amplamente utilizadas para avaliar o desempenho do classificador treinado, incluindo acurácia (média de 0,99), precisão (média de 0,99), *recall* (média de 0,99) e pontuação F1 (média de 0,99), cujos valores se aproximaram de 1,0 (um) para o conjunto de teste. A inspeção visual das máscaras de classificação resultantes confirmaram a eficácia do desempenho. A Tabela 5 a seguir, expões os valores das métricas alcançadas durante a etapa de testes, com a avaliação do desempenho do classificador, tanto na tarefa de rotular pixels como floresta, como na tarefa de identificar pixels como não-floresta:

Tabela 5: Resultados do conjunto de testes

	Precisão	***Recall***	**F1-Score**	**Amostras**
Não-floresta	0,99	0,99	0,99	2611101
Floresta	0,99	0,99	0,99	2631779
Acurácia			0,99	5242880

- ***Acurácia:*** A acurácia próxima de 1 sugere que o modelo está acertando na classificação tanto das regiões preservadas quanto das áreas não preservadas. Em outras palavras, a maioria das predições do modelo é correta em relação ao total de instâncias;

- ***Precisão:*** Uma precisão próxima de 1 indica que o modelo faz poucos erros ao identificar regiões como preservada. Isso significa que a maioria das áreas identificadas como resguardadas pelo modelo foi realmente preservada, minimizando assim os falsos positivos;

- ***Recall:*** Um *recall* próximo de 1 significa que o modelo está identificando eficazmente a grande maioria das coberturas florestais presentes nas imagens de satélite. Em outras palavras, há poucos casos em que o modelo deixa de detectar florestas seguras (falsos negativos);

- ***Pontuação F1:*** A pontuação F1 próxima de 1 indica um equilíbrio entre precisão e *recall*. Isso significa que o modelo não está apenas fazendo previsões precisas, mas também está identificando a maioria das regiões preservadas. Essa métrica é especialmente útil quando há um desequilíbrio significativo entre as classes;

O desempenho das U-Nets na segmentação florestal, juntamente com a necessidade mínima de intervenção humana no período pós-treinamento, sugere que a arquitetura U-Net pode ser fundamental para o desenvolvimento de sistemas automatizados de

monitoramento de mudanças na cobertura florestal. De fato, boa parte do trabalho manual concentra-se no pós-processamento dos resultados, especialmente na validação dos polígonos de desmatamento. Durante essa fase, é essencial realizar a remoção de falsos positivos e falsos negativos para aprimorar a precisão dos polígonos de desmatamento candidatos, garantindo que correspondam à realidade subjacente.

A arquitetura U-Net treinada foi aplicada para identificar polígonos de desmatamento por meio de análise multi-temporal de imagens de satélites. Para isso, periodicamente, cenas espapaçadas do satélite Sentinel-2 foram selecionadas para a mesmo região de interesse e inseridas como entradas na rede neural, Figura 44. As matrizes de saída correspondentes às imagens de entrada foram então submetidas a subtração para gerar uma matriz de diferenças. A matriz de diferenças foi então aplicada para a identificação de regiões com provável desmatamento que foram representadas por meio de polígonos, Figura 45. Como forma de alcançar tal classificação, píxeis com valores iguais ou superiores de 0,5 foram classificados como áreas que não sofreram perda cobertura florestal, enquanto os píxeis com valores abaixo de 0,5 foram classificados como áreas de desmatamento. A partir dessa classificação, os polígonos de desmatamento foram produzidos e aplicados para avaliar o desempenho da arquitetura U-Net na tarefa de identificar e monitorar o desmatamento.

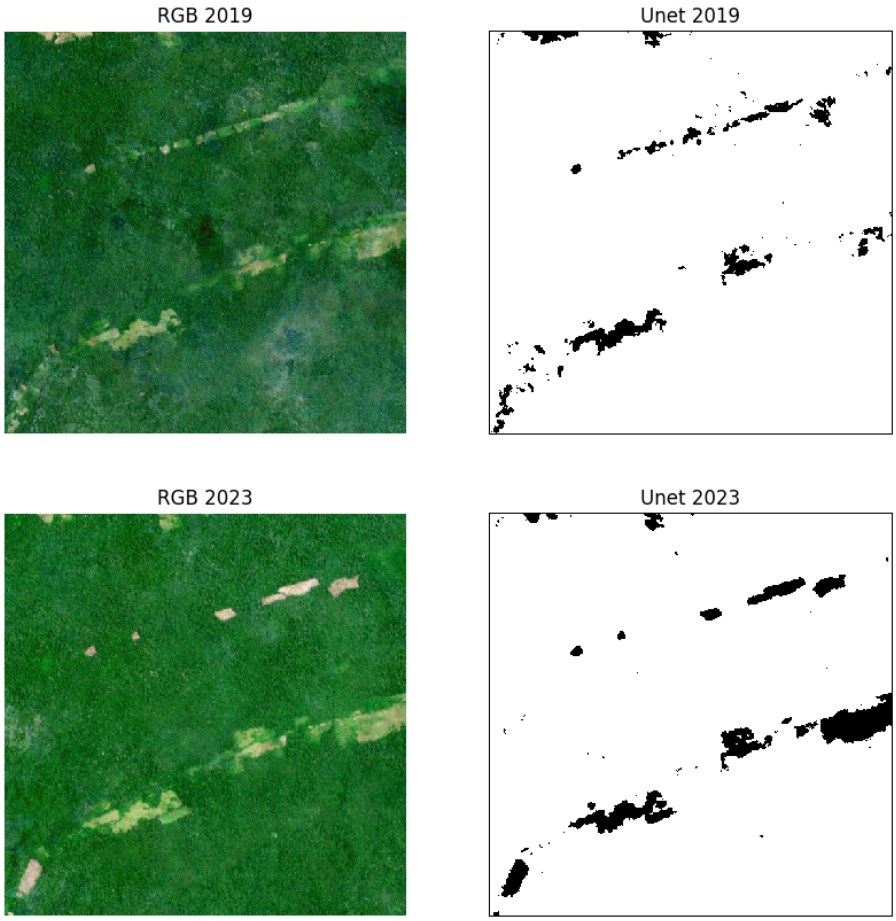

Figura 44: Comparação das análises de imagens de satélite multitemporais: (a) imagem anterior e (b) imagem posterior

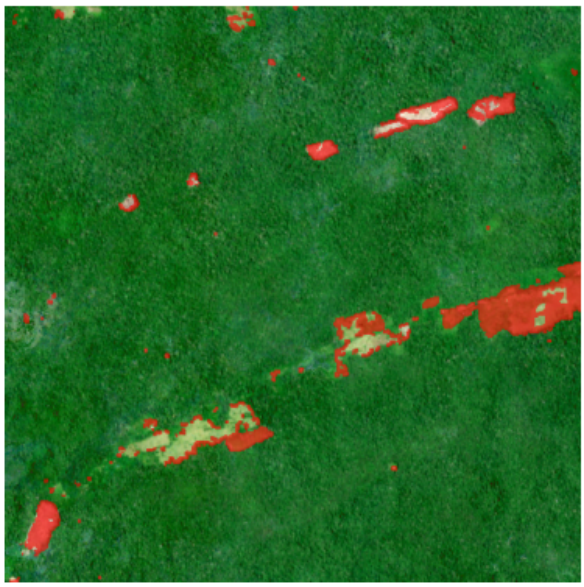

Figura 45: Polígonos de desmatamento (em vermelho)

Resumidamente, as matrizes de saída correspondentes às imagens temporais são

subtraídas para gerar uma matriz de diferenças que destaca as áreas que mudaram entre as duas datas, indicando potenciais de alteração de cobertura florestal. A análise temporal das imagens e polígonos permite observar padrões de mudança durante o intervalo do tempo entre as imagens, identificando tendências de desmatamento. Uma operação de limiarização é aplicada à matriz de diferenças para identificar áreas que excedem um determinado limite. Isso ajuda a distinguir regiões onde ocorreram mudanças significativas. Com base nos resultados da limiarização, algoritmos de identificação de regiões conectadas são aplicados para agrupar píxeis significativos em polígonos representativos das áreas de desmatamento. Cada polígono representa uma região onde a mudança foi detectada.

A Figura 46 mostra algumas das imagens classificadas pela U-Net treinada e as compara com as máscaras geradas manualmente e com as imagens originais. Os resultados indicam que as predições da U-Net estão alinhadas de forma significativa com as máscaras manuais e as imagens originais. Isso sugere que o modelo aprendeu a identificar e segmentar corretamente os padrões relevantes nas imagens.

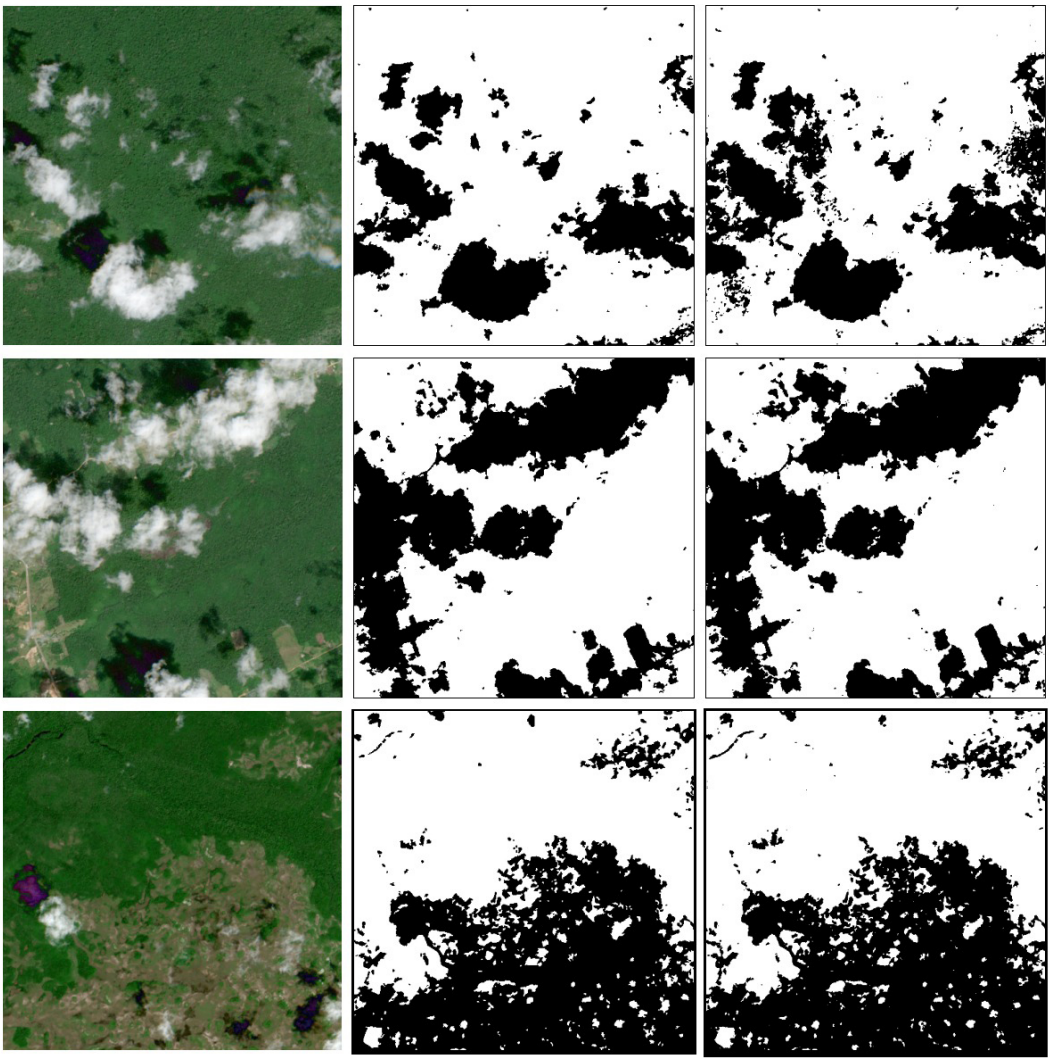

Figura 46: Comparativo entre as cores verdadeiras (coluna 1), as máscaras refinadas manualmente (coluna 2) e o resultado do processamento da U-Net (coluna 3)

Os resultados positivos também indicam que o modelo foi treinado eficientemente e é capaz de generalizar bem para novas amostras, ou seja, pode segmentar corretamente imagens não vistas durante o treinamento. Se as máscaras geradas pela U-Net estão em concordância com as máscaras manuais, isso sugere que o modelo pode ser útil em automatizar o processo de segmentação, economizando tempo e esforço humano.

Também é possível visualizar no comparativo a presença de obstruções nas imagens. A oclusão por nuvens ocorre quando nuvens presentes nas imagens de satélite cobrem áreas da superfície da Terra, obscurecendo informações valiosas sobre a paisagem. Isso pode criar grandes áreas de oclusão nas imagens, levando à perda de informações cruciais sobre mudanças na cobertura do solo ao longo do tempo. Isso é especialmente crítico em análises multi-temporais, onde a detecção de mudanças é fundamental.

A abordagem de análise multi-temporal envolvendo o uso de imagens de satélite

capturadas em diferentes momentos pode ser uma solução para essa problemática. A U-Net pode ser treinada para considerar informações temporais e aprender padrões de mudança ao longo do tempo, mesmo quando há oclusões temporárias devido a nuvens.

Dessa forma, cada imagem no conjunto de dados é pré-processada para realizar a segmentação de nuvens usando a U-Net treinada. Isso permite que a rede identifique e marque as regiões cobertas por nuvens. Em vez de usar uma única imagem para análise, várias imagens são consideradas para cada ponto. A mediana é calculada para as imagens segmentadas, proporcionando uma visão mais robusta das condições da terra ao longo do tempo. A escolha de um período mínimo é estratégica, esse intervalo de tempo permite uma comparação mais confiável das condições da terra, minimizando o impacto das flutuações naturais ou variações sazonais.

Há um significativo interesse em avançar na automação da segmentação semântica, pois as abordagens que vão desde métodos manuais até a segmentação semântica semi-automática e processos de classificação tendem a ser altamente onerosas e dispendiosas. Isso se deve ao grande volume de imagens a serem processadas, resultado das extensas áreas florestais a serem abrangidas e da alta frequência de captura de dados por satélites. Nesse cenário, mesmo técnicas semi-automáticas que apresentam um nível moderado de erros de classificação podem impor uma carga de trabalho excessiva.

Nesse contexto, o uso de uma U-Net treinada, como a utilizada neste estudo para a análise de imagens, possibilita o processamento e a classificação das cenas de forma imediata, logo após sua recepção. Isso viabiliza aplicações práticas para sistemas de monitoramento de mudanças na cobertura florestal, como o caso da SFC. Esses sistemas têm o potencial de auxiliar as partes interessadas a reagir de maneira rápida e eficaz diante as mudanças em curso da cobertura florestal. Além disso, o menor tempo de execução computacional necessário para as tarefas de classificação reforça ainda mais a sua viabilidade para aplicação prática em tais sistemas. Uma das principais vantagens associadas ao uso da U-Net como *core* para sistemas de monitoramento é sua estrutura de processamento, que demanda pouco em termos de pré e pós-processamento.

6.2 Considerações

Este capítulo forneceu uma visão abrangente dos resultados obtidos nesta pesquisa. As análises meticulosas revelaram *insights* significativos que contribuem para o entendimento do problema de pesquisa em questão.

Em resumo, a apresentação e a análise dos resultados corroboram a importância desta pesquisa e estabelecem uma sólida fundação para as conclusões finais desta dissertação, que serão apresentadas no próximo capítulo.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta de solução utilizando oráculos blockchain, aliada a estratégias de processamento de imagens e aprendizado de máquina, que permitisse a operacionalização da Standing Forest Coin (SFC). Por meio de uma arquitetura de Rede Neural Convolucional (CNN), denominada U-Net, o oráculo seria capaz de identificar áreas de cobertura florestal mediante imagens de satélite utilizando o conceito de segmentação semântica (binária).

Os resultados obtidos na fase de avaliação do modelo evidenciaram a capacidade da rede neural em discriminar com precisão áreas preservadas e áreas afetadas pelo desmatamento. As métricas de desempenho, incluindo acurácia, precisão, *recal*l e pontuação F1, alcançaram valores próximos à 1 (um), indicando uma robustez na capacidade do modelo de generalizar para dados não vistos.

A presente dissertação buscou responder às questões de pesquisa formuladas, explorando estratégias para tornar o processo de financiamento de projetos ambientais mais transparente e a relação entre financiadores e prestadores de serviços ambientais mais confiável, por meio da implementação de oráculos blockchain. Ao longo do estudo, foi possível validar a hipótese central, demonstrando que a solução proposta efetivamente identifica e distingue, de maneira precisa e eficiente, áreas preservadas e desmatadas na região amazônica, oferecendo uma base sólida para a tomada de decisões informadas.

Os resultados obtidos apontam que a utilização de oráculos blockchain contribui para a transparência no financiamento de projetos ambientais, proporcionando um registro imutável e acessível a todas as partes interessadas que participam do financiamento de projetos sustentáveis na Standing Forest Coin.

Os resultados aqui apresentados não apenas contribuem para o desfecho desta dissertação, mas também implicam em práticas significativas. A detecção precisa de regiões desmatadas é fundamental para a preservação ambiental, fornecendo informações valiosas para tomadores de decisão, pesquisadores e organizações de conservação. No entanto, reconhecemos que há desafios contínuos a serem abordados, e a evolução constante das tecnologias empregadas na solução, oferecerá oportunidades para melhorias futuras.

Em relação às dificuldades do estudo vale ressaltar a disponibilidade de dados de treinamento de qualidade para o modelo de aprendizagem de máquina. Garantir um conjunto de dados representativo e suficientemente grande para a tarefa específica de detecção de desmatamento na região amazônica em imagens de satélite é um ponto crucial. Além disso, a detecção de áreas preservadas e desmatadas pode ser uma tarefa complexa, pois pode haver variações significativas nos tipos de vegetação e características geográficas. O modelo deve ser robusto o suficiente para generalizar bem para diferentes

contextos.

Outro ponto a ser levado em consideração são os recursos computacionais consumidos, especialmente se o conjunto de dados for grande e as imagens de satélite tiverem alta resolução. Considerar os custos associados a esses recursos é crucial, especialmente em um contexto de blockchain, onde se somam taxas associadas à execução de transações e ao armazenamento de dados na cadeia de blocos. Um modelo que busque escalabilidade, implicará no potencial aumento do uso de recursos e serviços, elevando consequentemente os custos de manutenção da plataforma.

Em última análise, a confirmação de nossa hipótese reforça a promessa e o impacto positivo que a tecnologia pode ter na viabilização da relação entre o financiador e o prestador de serviços ambientais e consequentemente na preservação ambiental. Esta dissertação representa um passo significativo em direção a uma compreensão mais profunda e eficaz da detecção de desmatamento, pavimentando o caminho para aplicações práticas e soluções sustentáveis para um futuro mais preservado.

Fica como sugestão para futuros trabalhos, primeiramente, a expansão do modelo proposto para reconhecer áreas preservadas e desmatadas para outros biomas além da região amazônica, adaptando a solução para lidar com as características específicas de diferentes ecossistemas, como vegetação, topografia e padrões climáticos. Investigar a identificação e incorporação de características regionais específicas de cada bioma no modelo, aprimorando a capacidade de discriminação.

Outra sugestão seria a evolução do esquema de classificação binária, que agrupa todos os objetos não classificados como floresta na classe 'não-floresta', para uma abordagem que inclua mais classes de objetos, como áreas urbanas, corpos d'água, diferentes tipos de vegetação, proporcionando uma análise mais completa da região e permitindo uma tomada de decisão mais sofisticada em relação às diferentes categorias de cobertura terrestre.

REFERÊNCIAS

[1] "LOPES, Marina Stygar. Ciência e tecnologia para o desenvolvimento sustentável da amazônia, 2019. Disponível em: <https://www.matanativa.com.br/ciencia-tecnologia-desenvolvimento- sustentavel-amazonia>. Acesso em: 10 Jun. 2022."

[2] "SCHNEIDER, C. Floresta 4.0: COMO OS CONCEITOS DA INDÚSTRIA 4.0 ESTÃO REVOLUCIONANDO O SETOR FLORESTAL, Revista B. Forest, vol. 59, pp. 23–32, 2019. Disponível em: <https://revistabforest.com.br/2019/09/b-forest-a-revista-eletronica-do-setor-florestal-edicao-59-ano-06-n-08-2019/>. Acesso em: 10 Jun. 2022."

[3] "NAKAMOTO, Satoshi. A peer-to-peer electronic cash system, 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Acesso em: 10 Jun. 2022."

[4] "BUTERIN, Vitalik. Ethereum: A next-generation smart contract and decentra- lized application platform, 2022. Disponível em: <https://ethereum.org/en/whitepaper>. Acesso em: 10 Jun. 2022."

[5] "BORGES, Marcelo de A. et al. Standing Forest Coin (SFC), 2022. Disponível em:<https://arxiv.org/abs/2203.12600>. Acesso em: 10 Jun. 2022."

[6] "MAZET, R.; WOJCIECHOWSKI. J. Alice whitepaper, 2017. Disponível em:<https://github.com/alice-si/whitepaper>. Acesso em: 10 Jun. 2022."

[7] "FAULKNER, Jonh. **Getting Started with Cryptography in .NET**. BookRix, 2016 ."

[8] "OLIVEIRA, Ronielton Rezende. Criptografia simétrica e assimétrica: os principais algoritmos de cifragem, 2012. Disponível em: <http://www.ronielton.eti.br/publicacoes/artigorevistasegurancadigital2012.pdf>. Acesso em: 10 Jun. 2022."

[9] "VOITECHEN, Dainara Aparecida. ANÁLISE E COMPARAÇÃO DE ALGORITMOS PARA CRIPTOGRAFIA DE IMAGENS, 2015. Disponível em: <https://portaldeinformacao.utfpr.edu.br/Record/riut-1-16864>. Acesso em: 10 Jun. 2022."

[10] "SILVA, Rafael Rodrigues. UTILIZANDO A BLOCKCHAIN COMO MODELO DE CONFIANÇA PARA COMPROVAR A EXISTÊNCIA E IMUTABILIDADE DE UM ARQUIVO DIGITAL, 2019. Disponível em:<https://www.univates.br/bdu/bitstream/10737/2849/1/2010RafaelRodrigues.pdf>. Acesso em: 10 Jun. 2022."

[11] “CORRÊA, Otávio Augusto. ESTUDO DA APLICAÇÃO DE ESTRUTURA BLOCKCHAIN COM PROOF OF STAKE PARA ARQUIVAMENTO DE DOCUMENTOS COM REGISTRO NO TEMPO, 2017. Disponível em: <https://repositorio.ufsc.br/handle/123456789/187862>. Acesso em: 10 Jun. 2022.”

[12] “MARTINS, Thiago Fonseca. Prova de existência de arquivos digitais utilizando a tecnologia blockchain do protocolo Bitcoin, 2018. Disponível em: <https://lume.ufrgs.br/handle/10183/177585>. Acesso em: 10 Jun. 2022.”

[13] “SINGH, Mahima; GARG, Deepak. Choosing Best Hashing Strategies and Hash Functions, 2009. Disponível em: <https://ieeexplore.ieee.org/document/4808979>. Acesso em: 10 Jun. 2022.”

[14] “HART-DAVIS; Guys. **Mastering Microsoft VBA, 2nd Edition**. Wiley Publishing, 2005.”

[15] “MACCORMICK, John. **Nine Algorithms That Changed the Future: The Ingenious Ideas That Drive Today’s Computers**. Princeton University Press, 2012.”

[16] “HABER, S.; STORNETTA, W. S. How to time-stamp a digital document, 1991. Disponível em: <http://dx.doi.org/10.1007/BF00196791>. Acesso em: 10 Jun. 2022.”

[17] “KOÇ, A. K. et al. Towards secure e-voting using ethereum blockchain, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8355340>. Acesso em: 10 Jun. 2022.”

[18] “FAOUR, Nazim. Blockchain E-Voting Done Right: Privacy and Transparency with Public Blockchain, 2019. Disponível em: <https://arxiv.org/abs/1802.10134>. Acesso em: 10 Jun. 2022.”

[19] “SWAN, Melaine. **Blockchain Blueprint for a New Economy**. O’Reilly Media, 2015.”

[20] “BASHIR, I. **Mastering blockchain**. Packt Publishing Ltd, 2017.”

[21] “CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and Smart Contracts for the Internet of Things, 2019. Disponível em: <https://ieeexplore.ieee.org/document/9040770>. Acesso em: 10 Jun. 2022.”

[22] “GRIBBLE, Steven; HALEVY, Alon; IVES, Zahary; RODRIG, Maya; SUIU Dan. What Can Peer-to-Peer Do for Databases, and Vice Versa?, 2001. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.8638>. Acesso em: 10 Jun. 2022.”

[23] “ALVES, P.H. et al. Desmistificando Blockchain: Conceitos e Aplicações, 2018. Disponível em: <http://www-di.inf.puc-rio.br/ kalinowski/publications/AlvesLNRLK20.pdf>. Acesso em: 10 Jun. 2022.”

[24] “SOUSA, Suzane Carvalho. USO DA BLOCKCHAIN DO ETHEREUM PARA A REALIZAÇÃO DE LEILÕES DE ENERGIA ELÉTRICA ATRAVÉS DA CRIAÇÃO DE FERRAMENTA WEB, 2019. Disponível em: <https://repositorio.ufc.br/handle/riufc/49959?locale=en>. Acesso em: 10 Jun. 2022.”

[25] “OLIVEIRA, Walan Marcel Teles. Token Universitário por Meio de Blockchain, 2020. Disponível em:<https://ri.ufs.br/handle/riufs/12497>. Acesso em: 10 Jun. 2022.”

[26] “GREVE, F. et al. Blockchain e a revolução do consenso sob demanda, 2018. Disponível em: <http://143.54.25.88/index.php/sbrcminicursos/article/view/1770>. Acesso em: 10 Jun. 2022.”

[27] “HÖLBL, M. et al. An ecosystem for managing digital micro-credentials, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8534284>. Acesso em: 10 Jun. 2022.”

[28] “MENDANHA, Gabriel Oliveira. ASSEGURANDO A PROPRIEDADE E IMUTABILIDADE DE DOCUMENTOS DIGITAIS: UMA PROVA DE CONCEITO UTILIZANDO BLOCKCHAIN, 2017. Disponível em: <https://repositorio.ufc.br/handle/riufc/24738>. Acesso em: 10 Jun. 2022.”

[29] “ANTONOPOULOS, Andreas. **Mastering Bitcoin**. O’Reilly Media, 2014.”

[30] “HOUBEN, R.; SNYERS, A. **Cryptocurrencies and blockchain: legal context and implications for financial crime, money laundering and tax evasion**. European Parliament, 2018.”

[31] “BENTOV, I. et al. Proof of activity: Extending bitcoin’s proof of work via proof of stake, 2014. Disponível em: <https://dl.acm.org/doi/10.1145/2695533.2695545>. Acesso em: 10 Jun. 2022.”

[32] “ZOHAR, Aviv. Recent trends in decentralized cryptocurrencies (invited talk), 2017. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3055399.3079074>. Acesso em: 10 Jun. 2022.”

[33] “GARCÍA, Andrea Del Nido. Análisis e implementación del algoritmo de firma Digital de Curvas Eliptícas Múltiples (MECDSA) para blockchain, 2019. Disponível em: <https://oa.upm.es/55590/>. Acesso em: 10 Jun. 2022.”

[34] “ALFAIFI, Reem. Probabilistic cryptosystem with two pairs of private/public keys, 2017. Disponível em: <https://ieeexplore.ieee.org/document/8251927>. Acesso em: 10 Jun. 2022.”

[35] “FAGGART, Evan. Thoughts On Bitcoin Block Size Economics, 2015. Disponível em:<https://bitcoinist.com/thoughts-bitcoin-block-size-economics/>. Acesso em: 10 Jun. 2022.”

[36] “SCHÜPFER, Florian. **Design and Implementation of a Smart Contract Application**. 2017. Master Thesis (Communication Systems Group) – University of Zurich, Lucern, Suíça.”

[37] “FERREIRA, Frederico Lage. Blockchain e Ethereum: Aplicações e Vulnerabilidades, 2017. Disponível em: <https://linux.ime.usp.br/ fredlage/mac0499/Monografia.pdf>. Acesso em: 10 Jun. 2022.”

[38] “SZABO, N. Formalizing and securing relationships on public networks, 1997. Disponível em: <https://firstmonday.org/ojs/index.php/fm/article/view/548>. Acesso em: 10 Jun. 2022.”

[39] “WOOD, G. Ethereum: A secure decentralised generalised transaction ledger, 2022. Disponível em: <https://ethereum.github.io/yellowpaper/paper.pdf>. Acesso em: 10 Jun. 2022.”

[40] “LELOUP, L. **Blockchain: La révolution de la confiance**. Eyrolles, 2017.”

[41] “VOSHMGIR, S. Blockchain Oracles, 2019. Disponível em: <https://blockchainhub.net/blockchain-oracles>. Acesso em: 10 Jun. 2022.”

[42] “CALDARELLI, Giulio. Overview of Blockchain Oracle Research, 2022. Disponível em:<https://www.mdpi.com/1999-5903/14/6/175>. Acesso em: 10 Jun. 2022.”

[43] “BENIICHE, Abdeljalil. A Study of Blockchain Oracles, 2020. Disponível em: <https://arxiv.org/pdf/2004.07140.pdf>. Acesso em: 10 Jun. 2022.”

[44] “MA, L.; KANEKO, K.; SHARMA, S.; SAKURAI, K. Reliable decentralized oracle with mechanisms for verification and disputation, 2019. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8951647>. Acesso em: 10 Jun. 2022.”

[45] “PARKER, J. R. **Algorithms for image processing and computer vision**. John Wiley and Sons, 2010.”

[46] “VARGAS, A. C. G.; PAES, A.; VASCONCELOS, N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. 2015. Disponível em:<http://sibgrapi.sid.inpe.br/col/sid.inpe.br/sibgrapi/2016/09.12.15.44/doc/um-estudo-sobre.pdf/>. Acesso em: 01 Jun. 2023.”

[47] “DESHPANDE, A. A Beginner’s Guide To Understanding Convolutional Neural Networks. 2015. Disponível em:<https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner’sGuide-To-Understanding-Convolutional-Neural-Networks/>. Acesso em: 01 Jun. 2023.”

[48] “NUMPYDL. Convolution Neural Networks. 2015. Disponível em:<http://numpydl.readthedocs.io/en/0.2.0/tutorials/CNN Part1/?highlight=re>. Acesso em: 01 Jun. 2023.”

[49] “LONG, J.; SHELHAMER, E.; and Darrell, T. Fully Convolutional Networks for Semantic Segmentation. 2015. Disponível em:<https://arxiv.org/abs/1411.4038/>. Acesso em: 01 Jun. 2023.”

[50] “BADRINARAYANAN, Vijay; KENDALL, Alex; CIPOLLA, Roberto. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. 2017. Disponível em:<https://arxiv.org/abs/1505.07293/>. Acesso em: 01 Jun. 2023.”

[51] “DENYER, David; TRANFIELD, David. Producing a systematic review, 2009. Disponível em:<https://www.cebma.org/wp-content/uploads/Denyer-Tranfield-Producing-a-Systematic-Review.pdf>. Acesso em: 10 Jun. 2022.”

[52] “WOHLIN, C. et al. Experimentation in software engineering, Springer Science and Business Media, 2012. Disponível em:<https://link.springer.com/book/10.1007/978-3-642-29044-2>. Acesso em: 01 Fev. 2023.”

[53] “BRAGAGNOLO, L.; DA SILVA, R. V.; GRZYBOWSKI, J. M. V. Amazon and Atlantic Forest image datasets for semantic segmentation. 2021. Disponível em:<https://zenodo.org/records/4498086>. Acesso em: 01 Jun. 2023.”

[54] “BRAGAGNOLO, L.; DA SILVA, R. V.; GRZYBOWSKI, J. M. V. Amazon forest cover change mapping based on semantic segmentation by U-Nets. 2021. Disponível em:<https://www.sciencedirect.com/science/article/abs/pii/S1574954121000704>. Acesso em: 01 Jun. 2023.”

[55] “RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: convolutional networks for biomedical image segmentation. 2015. Disponível em:<https://arxiv.org/abs/1505.04597>. Acesso em: 01 Jan. 2023.”

[56] “KINGMA, D. P.; BA, J. Adam: a method for stochastic optimization. 2015. Disponível em:<https://arxiv.org/abs/1412.6980/>. Acesso em: 01 Jun. 2023.”

[57] “JAIN, Raj. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. John Wiley and Sons, 1990.”