MESTRADO EM
MÉTODOS QUANTITATIVOS PARA A DECISÃO ECONÓMICA E EMPRESARIAL

TRABALHO FINAL DE MESTRADO
DISSERTAÇÃO

CLASSIFICAÇÃO DAS CARTEIRAS NA *BLOCKCHAIN ETHEREUM* USANDO *MACHINE LEARNING*

YIQING ZHU

ORIENTAÇÃO:

ALEXANDRA BUGALHO DE MOURA
RICARDO FERNANDO LOPES FONTES MARTINS

OUTUBRO-2023

Agradecimentos

Os meus agradecimentos a todos que contribuíram, de forma direta ou indiretamente, na construção desta dissertação.

A Professora Alexandra Moura e Ricardo Martins, os meus orientadores, agradeço os seus conselhos e orientações valiosos, apoio constante e paciência demonstrada ao longo deste processo.

A YiKui Liu agradeço pela disponibilidade e esclarecimento das minhas dúvidas.

Por fim, um agradecimento especial a todos os que me ajudaram e encorajaram a fazer esta dissertação: aos meus amigos e família que sempre me apoiaram, os meus profundos agradecimentos.

Resumo

O interesse em compreender o tipo de carteiras por trás das transações registadas na *blockchain Ethereum* tem crescido consideravelmente. Isso deve-se ao facto desta análise permitir perceber os comportamentos das transações e obter informações antecipadas sobre os movimentos dos grandes detentores da *Ether,* fornecendo uma visão valiosa do comportamento do mercado e tornando-se uma fonte de informações estratégicas cruciais para investidores e observadores do ecossistema de criptomoedas.

O objetivo deste trabalho é aplicar métodos de análise de dados e *Machine Learning* que permitam classificar o tipo de carteiras através das características das transações. Sendo uma área recente, a maioria da literatura concentra-se na deteção dos endereços de anomalia. A análise de tipo de carteiras por grau de segurança é uma área de pesquisa académia limitada, pois a própria definição das carteiras é subjetiva. Assim, com esta dissertação pretende-se realizar uma classificação das carteiras, caraterizando os grupos de carteiras com rótulos publicamente estabelecidos e os grupos de carteiras com rótulos definidas através das características essenciais recorrendo a técnicas de análise de dados.

A análise dos dados passa pela extração dos dados brutos até à aplicação de algoritmos de *Machine Learning*. Assim, foram considerados vários modelos para fazer a classificação de tipos de carteiras, como Regressão Logística, *Random Forest, AdaBoost e GradientBoosting*. A validação e comparação dos modelos elaborados foi feita de acordo com várias medidas como *accuracy*, precisão, sensibilidade, especificidade, *F_Score, e AUC*. A validação cruzada é o método escolhido para a avaliação dos modelos.

Dos resultados obtidos para dados de transações de *Ethereum* entre 2016 e 2023, conclui-se que as metodologias aqui proposta constituem uma ferramenta importante na classificação de carteiras *Ethereum.*

Palavras-Chaves: Criptomoedas, *Ethereum*, Análise de grupo, *Machine Learning, Cold Wallet*

Abstract

The interest in comprehending the various wallet types associated with transactions on the Blockchain Ethereum has seen substantial growth. This is primarily because such analysis yields invaluable insights into transaction patterns, furnishing early data on the actions of prominent Ether holders. Consequently, it provides a vantage point into market behavior, establishing itself as a pivotal source of strategic information for both investors and observers within the cryptocurrency ecosystem.

The objective of this work is to apply data analysis and Machine Learning methods to classify wallet types based on transaction characteristics. In the relatively recent field of cryptocurrency analysis, most research efforts have focused on anomaly address detection, while the analysis of wallet types by security level remains a limited area of academic research. This is partly due to the inherent subjectivity in defining wallet types.

In this study, the goal is to classify wallets from various perspectives, characterizing groups of wallets with publicly established labels and groups of wallets with labels defined through essential characteristics using data analysis techniques.

For the data analysis various tasks are considered, from data extraction to the use of Machine Learning algorithms for predictions. Several models are used to predict wallet types, including Logistic Regression, Random Forest, AdaBoost, and GradientBoosting. The validation and comparison of models are conducted utilizing a diverse set of metrics, including accuracy, precision, recall, specificity, F-value, and AUC. The chosen approach for model evaluation was cross-validation.

From de results, obtained using data from 2016 to 2023, we conclude that the methodologies proposed in this work constitute on important tool in classifying Ethereum wallets.

Key Words: Cryptocurrencies, Ethereum, Group Analysis, Machine Learning and Cold Wallet

Índice

Glossário

Altcoins: *Alternative digital assets to Bitcoin*

API: *Application Programming Interface*

CBDC: *Central Bank Digital Currency*

DAO: *Decentralized Autonomous Organization*

DApps: *Decentralized application*

DeFi: *Decentralized Finance*

DLT: *Distributed Ledger Technology*

ERC-20: *Ethereum Request for Comment 20*

ERC-721: *Ethereum Request for Comment 721*

EVM: *Ethereum Virtual Machine*

FTX: *Future Exchange*

LR: *Regressão Logística*

P2P: Rede ponto a ponto

PCC: Percentagem de casos Corretamente Classificados

PoS: *Proof-of-Stake*

PoW: *Proof-of-Work*

RF: *Random Forest*

ROC: *Receiver Operating Characteristic*

SPV: *Simple Payment Verification*

SQL: *Structured query language*

Índice de Tabelas

Índice de Figuras

Introdução

Com a introdução da *Bitcoin* em 2009, a primeira criptomoeda descentralizada, surgiram subsequentemente outras criptomoedas, levando a uma disseminação global destes ativos financeiros. A *Bitcoin*, além de ser uma criptomoeda, também é um projeto de código aberto que serviu de inspiração para muitos outros projetos, especialmente para as *alternative digital assets to Bitcoin (altcoins*), que utilizam os mesmos princípios básicos para implementar moedas digitais descentralizadas [1].

A inovação da tecnologia *Blockchain*, evidenciada pela *Bitcoin*, reside na capacidade de realizar trocas de valor diretas entre duas partes, sem necessidade de intermediários, usando a tecnologia para assegurar a fiabilidade e a segurança das transações [30].

A *Blockchain Ethereum*, devido às suas características únicas, ocupa uma posição de destaque como a segunda criptomoeda mais reconhecida. Ao contrário da *Bitcoin*, a *Ethereum* não se limita apenas a ser uma moeda de transação, sendo também uma plataforma que permite a criação de contratos inteligentes e aplicações descentralizados, conhecidos como *Decentralized Applications* (*DApps*) [9].

No contexto das transações e da segurança destes cripto ativos, as carteiras digitais desempenham um papel fundamental. Estas carteiras são aplicações ou dispositivos usados pelos utilizadores da rede para gerir com segurança os seus ativos. É importante observar que, devido à natureza relativamente recente do mercado de criptomoedas, existem uma variedades de carteiras disponíveis [22,23,40]. No entanto, os conceitos associados aos diferentes tipos de carteiras podem apresentar subjetividade e variações. Existem desde carteiras de *hardware*, físicas e altamente seguras, até carteiras de *software* para dispositivos móveis e *desktop*.

O objetivo desta Tese é estudar o comportamento das transações na rede *Ethereum* para obter informação valiosa que possa contribuir para uma melhor compreensão do ecossistema e para suportar a tomada de decisões informadas. Nesta análise, serão aplicados vários algoritmos de *Machine Learning* no contexto do análise de grupo de endereços *Ethereum*, identificando grupos de endereços que compartilham características semelhantes. Esta abordagem permite compreender melhor como diferentes entidades se comportam na rede *Ethereum* e a identificar os tipos de carteiras associados a esses endereços.

Este estudo encontra-se organizado em seis capítulos. No Capítulo 2 é elaborada a introdução dos conceitos essenciais no mundo de criptomoedas, bem como a evolução de *Bitcoin* e de *Ethereum*. No Capítulo 3, são introduzidos os conceitos teóricos das carteiras com base em diferentes critérios. No Capítulo 4, é explicada a metodologia aplicada. No Capítulo 5 são apresentados e analisados os resultados obtidos. Por último, no Capítulo 6 apresentam-se as conclusões e indicações de futura investigação.

2. A tecnologia *DLT/Blockchain* e as criptomoedas

As redes de base de dados desempenham um papel crucial no mundo digital, permitindo o armazenamento eficiente e seguro dos dados. As redes fornecem mecanismos para organizar, gerir e proteger informações valiosas em diversos contextos, como empresas e instituições. Existem diversos tipos de redes, cada uma com características e usos específicos [2].

Uma rede centralizada, *Figura 1 (a)*, é caracterizada pela presença de um nó central que detém o controlo exclusivo dos dados, enquanto numa rede descentralizada, *Figura 1 (b)*, o controle dos dados é distribuído entre vários nós, permitindo que os participantes tenham autonomia na tomada de decisões e realizam transações diretamente entre si. Por fim, numa rede distribuída, *Figura 1 (c)*, o controlo e a organização de dados são compartilhados entre vários participantes (nós). Cada participante possui uma cópia do base de dados e trabalha em conjunto para alcançar consenso com os restantes nós e validar transações, por exemplo, usando a *Distributed Ledger Technology (DLT)*.

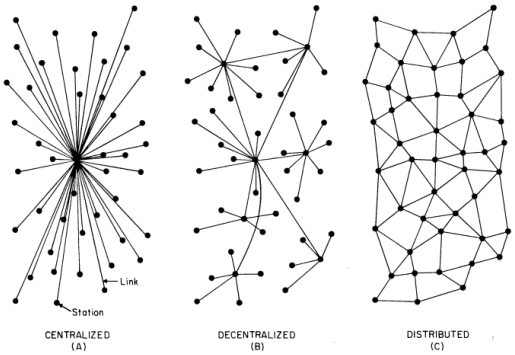


Figura 1: Tipos de Redes, sendo a figura (a) uma rede centralizada, a figura (b) uma rede descentralizada e a figura (c) uma rede distribuída [2]

A *DLT* é a tecnologia de registo distribuído de transações que pode ser compartilhada e sincronizada numa rede dispersa geograficamente [42]. A *DLT* garante que quaisquer alterações verificadas num dos nós serão refletidas nos restantes nós existentes. A *DLT* é amplamente usada para descrever tecnologias que permitem o armazenamento, a distribuição e a troca de dados entre utilizadores de redes distribuídas.

A *Blockchain*, um caso particular de *DLT*, é uma base de dados que está organizada em cadeia de blocos encadeados cronologicamente e em constante crescimento. Cada bloco contém um conjunto de registos que é adicionado à cadeia de forma segura com o recurso a criptografia [42].

2.1 Criptografia na *Blockchain*

A primeira proposta relativa ao protocolo *Blockchain* foi introduzida por *Chaum* em 1979, o qual referiu a importância da criptografia na proteção e fiabilidade do sistema da *Blockchain*

[11]. Em 1981, o autor propôs uma abordagem que visava alcançar um equilíbrio entre a privacidade e a eficiência dos sistemas de registo [12]. A proposta surgiu como resposta às preocupações relacionadas com a centralização das informações e aos riscos decorrentes da crescente informatização dos dados pessoais.

A tecnologia *Blockchain* utiliza criptografia para garantir a segurança e a integridade dos dados armazenados na rede. Diferentes formas de criptografia são utilizadas, tais como a criptografia assimétrica, que envolve o uso de chaves públicas e privadas [1,38,45].

As funções *hash* criptográficas geram sequências únicas e fixas, de forma irreversível, para proteger os dados originais[1]. Estas funções são projetadas para serem unidirecionais, facilitando a geração do *hash* a partir dos dados e, ao mesmo tempo, dificultando a reversão do processo para obter os dados originais. Estas funções são amplamente utilizadas na *Blockchain* para garantir a integridade dos blocos e a sua ligação sequencial.

A criptografia também está presente nas assinaturas digitais, que verificam a autenticidade de mensagens ou transações [38]. As assinaturas digitais são criadas com a chave privada do proprietário e podem ser verificadas com a chave pública correspondente. A utilização de assinaturas digitais nas transações e blocos permite confirmar a identidade do proprietário e garantir que os dados assinados não foram modificados desde a assinatura.

2.2 Estrutura de *Blockchain*

A estrutura básica de um bloco na *Blockchain* é composta por vários campos que armazenam informações importantes. Um bloco contém transações, um *hash* do bloco anterior, um *timestamp*, um *nonce* e uma *Merkle Root*, entre outras informações [1]. As transações são os registos da transferência de *Bitcoins*. O *hash* do bloco anterior garante a integridade da cadeia, a autenticidade das informações e protege contra falsificações. O *timestamp* regista a data e hora do evento ocorrido, assegurando a ordem cronológica das transações e proporciona transparência e fiabilidade ao registo. O *nonce* é um número que representa o histórico de transações de um endereço e permite garantir que cada transação seja processada apenas uma vez. A *Merkle Root* representa todas as transações do bloco.

As *Blockchains* são utilizadas como um Registo[2], que pode ser partilhado e corroborado por qualquer pessoa que tenha as permissões adequadas. Do ponto de vista de permissões, existem *Blockchain*s públicos, de consórcio e privados[3][45].

Numa *Blockchain* pública, qualquer indivíduo pode participar e contribuir para a manutenção da cadeia. Quanto maior for o número de indivíduos, maior será a segurança dos

[1] https://csrc.nist.gov/projects/*hash*-functions. [Acedido em 01/07/2023]
[2] Sigla em inglês "*ledger*", um *ledger* distribuído é uma base de dados de compartilhado por uma rede de participantes. Todas as partes têm cópias idênticas do registo e qualquer alteração é refletida nos restantes pontos [41].
[3] https://blog.*Ethereum*.org/2015/08/07/on-public-and-private-*Blockchain*s. [Acedido em 01/07/2023]

dados, pois os dados estão mais dispersos, considerado uma rede descentralizada. Numa *Blockchain* de consórcio, o processo de consenso é controlado por um grupo selecionado de nós. O acesso à *Blockchain* pode ser público ou restrito aos participantes, sendo considerado parcialmente descentralizada. Por fim, numa *Blockchain* privada existe uma autoridade que seleciona os participantes que contribuem para a manutenção do seu funcionamento, considerado uma rede centralizada.

2.2.1 Princípios de *Blockchain*

Os princípios de *Blockchain* permitem que a sua rede seja uma base fiável em diversos aspetos, entre os quais a descentralização dos dados, a transparência, a segurança, a imutabilidade e a privacidade [16,19,45].

A descentralização está presente na sua estrutura, pois a *Blockchain* é uma rede distribuída que permite eliminar a necessidade de uma autoridade central. A *Blockchain* é transparente, uma vez que os registos são auditáveis e acessíveis a um conjunto específico de participantes. Em *Blockchains* públicas, qualquer pessoa com acesso à *internet* pode verificar os registos, promovendo transparência e rastreabilidade. A segurança é assegurada por meio do uso de criptografia e chaves privadas, protegendo as transações de adulteração e garantindo a segurança dos dados transferidos. A imutabilidade é alcançada pelo facto dos dados registados na *Blockchain* não poderem ser alterados sem o conhecimento da rede. A utilização de *hashes* e o encadeamento de blocos tornam extremamente difícil para indivíduos ou grupos adulterarem o histórico de transações.

2.2.2 Exemplos da tecnologia *Blockchain* e de extensão de *DLT*

A tecnologia *DLT*, em particular a *Blockchain,* tem um enorme potencial de portabilidade e extensibilidade a outras áreas da atividade económica e social. Assim, tem potencial para ajudar a redefinir a relação entre as instituições e os cidadãos no que respeita à utilização de dados, nomeadamente em termos de transparência e confiança [42].

Na área de serviços financeiros e bancários, têm sido igualmente desenvolvidas tecnologias de extensão de *DLT* com o objetivo de aprimorar a eficiência, melhorando processos, reduzindo custos, aumentando a velocidade das transações e oferecendo maior segurança e transparência [14].

Os Bancos Centrais de vários países têm tentado, nos últimos anos, criar as suas próprias moedas digitais, as *Central Bank Digital Currency (CBDC)*, por forma a regular o setor [5,42,44]. A moeda digital emitida por Bancos Centrais é uma forma inovadora de dinheiro que tem como objetivo oferecer mais eficiência e segurança nos pagamentos, facilidade na inclusão financeira

das moedas digitais, bem como combater a lavagem de dinheiro e a evasão fiscal. As *CBDCs* podem ser criadas em tecnologias de Registo destruído, como *DLT* e *Blockchain*, mas esta escolha pode variar de acordo com as finalidades de cada país [5].

A tecnologia *Blockchain* também está presente noutros campos, como a prova judicial. Ao permitir o armazenamento seguro das evidências eletrónicas, como contratos, documentos e assinaturas, aumenta a segurança e transparência nos processos legais. A China tem sido pioneira nesta área, com a criação da *Blockchain LegalXchain.* Atualmente, o *LegalXchain* possui uma rede descentralizada com 39 nós distribuídos geograficamente por todo o país[4].

2.3 A Criptomoeda *Bitcoin*

Desde o final dos anos oitenta até ao início do novo milénio, várias tentativas de moedas digitais foram criadas. Entre elas, destaca-se a *eCash*, uma moeda digital criada por *David Chaum* em 1983 cujo protocolo enfrentou dificuldades devido à dependência de um intermediário centralizado [10]. Outra tentativa foi a *B-money*, uma moeda digital num sistema de ponto a ponto, criada por *Wei Dai* em 1998, cujos protocolos eram principalmente teóricos, o que tornava difícil a sua implementação prática [15]. Embora estas tentativas tenham falhado por diversas razões, as tecnologias subjacentes contribuíram para o surgimento da moeda digital *Bitcoin.*

Em 2008, um indivíduo ou grupo de pessoas, cuja verdadeira identidade permanece desconhecida até hoje, utilizou o pseudónimo *Satoshi Nakamoto* para publicar o *whitepaper* do *Bitcoin* na comunidade *Cypherpunk*. O *whitepaper*, no contexto de criptomoedas, entende-se por um documento técnico que descreve os fundamentos e detalhes de um projeto de criptomoeda, fornecendo informações sobre a tecnologia subjacente, objetivos, mecanismo de consenso, distribuição da moeda e outros aspetos relevantes [30]. Por sua vez, a comunidade *Cypherpunk*[5] é composta por indivíduos que se dedicam a defender a liberdade individual no espaço digital, com o propósito de estabelecerem um mundo em que a liberdade e a privacidade pessoal sejam preservadas por meio do uso de técnicas criptográficas.

O objetivo da *Bitcoin* foi desenvolver um sistema de pagamento eletrónico descentralizado baseado em prova criptográfica, evitando a dependência das instituições terceiras [30]. Assim, a *Blockchain Bitcoin* funciona como um livro contabilístico para registar as transações verificadas pelos participantes de rede ponto a ponto (P2P), de forma descentralizada, proporcionando transparência e integridade ao sistema [1,30].

A *Blockchain Bitcoin* resolve o problema do "gasto duplo", que se refere à tentativa de gastar a mesma unidade de valor digital mais de uma vez [30]. Nos sistemas bancários, uma autoridade central verifica e regista as transações para evitar o "gasto duplo". No caso da *Bitcoin*,

[4] https://legalxchain.com/en/technology. [Acedido em 26/02/2023]
[5] https://nakamoto.com/the-cypherpunks/ [Acedido em 26/02/2023]

sendo uma moeda digital, o problema é solucionado através da utilização de assinaturas digitais. Ao realizar uma transação, o utilizador cria uma assinatura digital única, utilizando a sua chave privada exclusiva, servindo como um comprovativo de propriedade da moeda. A chave privada exclusiva é semelhante a uma senha secreta, conhecida apenas pelo proprietário.

2.3.1 Mecanismo de Consenso: *Proof-of-Work*

O Registo da *Blockchain Bitcoin* é construído gradualmente por meio de uma estrutura encadeada de blocos [1,45]. Cada bloco contém um conjunto de transações e um registo de *hash*, isto é, uma sequência de caracteres única que garante a integridade e a autenticidade do bloco. Estes *hashes* também servem para estabelecer a conexão entre os blocos, formando assim uma cadeia contínua de informações. A estrutura encadeada assegura que qualquer modificação num bloco exija a alteração de todos os blocos subsequentes, tornando o livro contabilístico imutável e resistente a alterações fraudulentas.

O facto de ser um sistema distribuído pode levar à existência do chamado problema *Byzantine Generals*. Numa rede distribuída, como o sistema de *Bitcoin*, os diferentes nós comunicam entre si, trocando informações para chegar a um consenso e agir de acordo com o mesmo conjunto de políticas cooperativas [1]. No entanto, pode haver nós que cometem erros e enviam informações erradas, causando danos. Para evitar este problema, foi criado o mecanismo de consenso, que estabelece regras para coordenar o comportamento dos nós na rede.

A *Blockchain Bitcoin* utiliza o mecanismo de consenso chamado *PoW* para assegurar a segurança e a validação das transações [1,45]. A *PoW*, também conhecida como o processo de "mineração", incentiva os participantes, chamados mineradores, a competir pelo direito de gerar blocos, recompensando-os com uma certa quantidade de criptomoeda.

Assim, o processo de mineração inicia quando os mineradores começam a recolher e validar as transações pendentes na rede para incluir num bloco. Todas as pessoas são livres de participar no processo, no entanto, elas competem entre si pelo poder de lançar o seu bloco à cadeia da *Bitcoin*, através da resolução de um problema matemático de complexidade elevada por tentativa e erro, onde existe um gasto de energia elétrica elevado [1,45]. Apenas o vencedor pode produzir o bloco, sendo reconhecido pelos restantes utilizadores. O bloco pode ser adicionado ao último bloco da cadeia e o vencedor receberá *Bitcoin* como recompensa. Esta competição entre os mineradores garante que apenas o bloco válido e mais rápido seja adicionado ao *Blockchain*. Cada vez que um bloco é adicionado, é extremamente difícil modificar as transações nele registadas. Devido à estrutura da *Blockchain*, os blocos são encadeados sequencialmente por tempo. Assim, a quantidade de poder computacional necessário para realizar qualquer alteração na rede aumenta com o número de blocos. Isto proporciona um alto nível de segurança e imutabilidade ao sistema.

A recompensa inicial é de 50 *Bitcoin*s por bloco e é reduzida para metade a cada 210.000 blocos, de acordo com as informações estabelecidas no *whitepaper* da *Bitcoin*, especificamente através da função *Halving* do protocolo *Bitcoin*, cujo código correspondente está disponível no *GitHub*[6]. Através desta função, é possível estimar que um total de 21 milhões de *Bitcoin*s serão lançados no máximo, prevendo-se que processo de mineração continue até ao ano de 2140. Assim, os eventos periódicos da diminuição das recompensas por bloco fornecidas aos mineradores designam-se por processos de *Halving*[7]. Até ao momento, aproximadamente 92,65% do total de *Bitcoin*s já foram minerados como se pode observar na *Tabela 2*, com a recompensa por bloco atualmente estabelecida em 6,25 *Bitcoin*s, como se pode observar na *Tabela 1*.

Halving	Data / Data Estimada	Altura de bloco	Recompensa por bloco (BTC)
0	N/A	0	50
1	28/11/2012	210 000	25
2	09/07/2016	420 000	12,5
3	2020	630 000	6,25
4	2024	840 000	3,125
5	2028	1 050 000	1,5625

Tabela 1: Processo Halving. Corresponde aos eventos periódicos da diminuição das recompensas por bloco

Caraterísticas de BTC	
Total de BTC em circulação (13/08/2023)	19 455 950
Total de BTC a ser produzido	21 000 000
Percentagem de BTC já minerados	92,65%
Capitalização de mercado do BTC (USD)	572 445 801 827 $

Tabela 2: Caraterísticas da criptomoeda Bitcoin

Esta recompensa sistemática é a única forma de gerar a criptomoeda *Bitcoin.* Por outro lado, este modo de recompensa permite encorajar mais pessoas a participar, tornando a rede mais descentralizada e segura. Teoricamente, apenas um indivíduo com mais de 51% do poder de computação de toda a rede pode controlar a *Blockchain* da *Bitcoin* [1]. No entanto, a probabilidade de tal acontecer é muito reduzida. Até hoje, os comportamentos maliciosos não foram bem-sucedido. Além disso, o custo computacional envolvido seria significativo.

[6] https://gist.github.com/nelruk/d00d785f84e47288b0ab734d4aab5f49. [Acedido em 15 07 2023].
[7] https://academy.binance.com/pt/*Halving*. [Acedido em 15/07/2023].

2.4 A criptomoeda *Ethereum*

Após o surgimento da *Bitcoin*, inúmeras outras criptomoedas, conhecidas como *alternative digital assets to Bitcoin (altcoins)*, foram criadas. Estas *altcoins* compartilham propósitos semelhantes à *Bitcoin* [1,46]. Assim, durante alguns anos, a tecnologia *Blockchain* estava focada nas transações de moedas digitais, utilizando *Blockchain* principalmente como um Registo para validar transações financeiras. Isso limitou a sua utilização e expansão para outras áreas.

Esta limitação foi ultrapassada com o surgimento da *Ethereum* em 2014 [9]. A *Ethereum* é uma plataforma *Blockchain* programável que foi projetada para permitir a criação e execução de *Smart Contracts*, ou seja, programas auto executáveis de acordo com as condições pré-definidas [3,41]. Este avanço permitiu que a tecnologia *Blockchain* passasse a ser aplicada em novas áreas, como sistemas de votação, registos de propriedade ou sistemas de gestão de identidade.

2.4.1 *Smart Contract*

O termo "*Smart Contract"* foi introduzido por *Nick Szabo* em 1994, com o objetivo de minimizar exceções maliciosas ou acidentais, reduzir a necessidade de intermediários fiáveis e diminuir os custos de transação [41]. Os *Smart Contracts* são aplicados na *Blockchain* para assegurar os termos e condições entre duas partes, de forma automatizada. A *Ethereum Virtual Machine (EVM)* é a máquina virtual que fornece um ambiente de execução para os *Smart Contract*s [9]. Esta desempenha um papel fundamental na *Ethereum*, garantindo a consistência e a fiabilidade das operações realizadas. A *EVM* possui um modelo de pagamento chamado "*gas*", uma unidade de medida que representa o custo computacional [9]. Cada instrução e operação realizada na *EVM* consome uma quantidade específica de *gas*. Assim, os utilizadores devem pagar uma quantidade adequada da *Ether*, a criptomoeda criada na *blockchain Ethereum*, para cobrir o custo do *gas* necessário para executar o *Smart Contract*.

2.4.2 Aplicações descentralizadas

As aplicações descentralizadas (*DApps*), que operam na rede *Ethereum* ou noutras plataformas *Blockchains* descentralizadas [46], são construídas usando *Smart Contracts*, fornecendo serviços fiáveis sem a necessidade de intermediários. Ao contrário das aplicações tradicionais, que são executadas em servidores centralizados e controlados por uma única entidade, as *DApps* são executadas numa rede descentralizada de nós, onde cada nó possui uma cópia da *Blockchain*. Isso torna as *DApps* mais resistentes à censura, à falha de um único ponto e à

interferência de terceiros. Por outro lado, permite também aos utilizadores participarem diretamente na gestão e desenvolvimento da aplicação. As *DApps* podem ter diversos casos de uso e funcionalidades. Estas podem abranger desde jogos e entretenimento, até finanças descentralizadas (*DeFi*), votações eletrónicas, identidade digital, gestão de ativos digitais, entre outros.

A criptomoeda é uma moeda digital que utiliza os algoritmos criptográficos e a tecnologia *Blockchain* para assegurar a validade das transações. Exemplos conhecidos de criptomoedas são a *Bitcoin e a Ether.* As criptomoedas funcionam como moedas independentes, apresentando a sua própria rede de *Blockchain* e mecanismo de consenso [9,30,46].

Os *tokens* são unidades digitais criadas e executadas em cima de uma plataforma de *Blockchain* existente e são construídos utilizando *Smart Contracts* [46]. Estes podem ter várias finalidades e funcionalidades, dependendo do objetivo. Alguns *tokens* são usados como moedas internas nas *DApps*, usando para recompensar os participantes, de modo a incentivar a sua participação e contribuição na aplicação. Além disso, os *tokens* podem servir como meio de troca dentro do ecossistema da *DApps*, facilitando transações e interações entre os participantes, contribuindo para o bom funcionamento e economia das *DApps.*

Na rede *Ethereum*, existem vários tipos de *tokens* que podem ser implementados para representar diferentes ativos digitais. Um dos tipos mais comuns é o *Ethereum Request for Comment* 20 (*ERC-20*)[8]. O *ERC-20* é um conjunto de regras e interfaces que os *tokens* devem seguir para serem compatíveis com a rede *Ethereum*. Os *tokens ERC-20* são fungíveis, ou seja, cada unidade do *token* é igual e intercambiável com as outras. A padronização do *ERC-20* facilitou a interoperabilidade e a compatibilidade entre diferentes *tokens* na rede *Ethereum*, permitindo que os *tokens* fossem listados em *Exchanges*, integrados em carteiras e utilizados em diversas aplicações compatíveis com o padrão *ERC-20*.

Outro exemplo são os *tokens Ethereum Request for Comment 721 (ERC-721)* [9], conhecidos como *Non-Fungible Tokens (NFTs).* Ao contrário dos *tokens ERC-20*, os *NFTs* são únicos e não fungíveis, sendo frequentemente usados para representar objetos digitais colecionáveis, obras de arte digitais e outros ativos exclusivos.

2.4.3 *Hard fork* na rede *Ethereum*

Em 2016, foi criada uma Organização Autónoma Descentralizada [10] (*DAO*) com o objetivo de angariar fundos de capital para a inovação e o crescimento do ecossistema de *DApps* na *Ethereum* [47]. A *DAO* conseguiu obter mais de 250 milhões de dólares em *Ether* de mais de

[8] https://www.indexuniverse.eu/erc20-token-standard/. [Acedido em 02/07/2023]
[9] https://eips.*Ethereum*.org/EIPS/eip-721. [Acedido em 08/07/2023]
[10] https://www.sec.gov/files/litigation/investreport/34-81207.pdf [Acedido em 08/07/2023]

11.000 investidores[11]. Esta organização funciona automaticamente através de *Smart Contract*, sendo que todas as transações e regras do programa são mantidas pela rede *Blockchain Ethereum*[12].

Em junho de 2016, a *DAO* sofreu um ataque informático devido a uma vulnerabilidade no seu código, resultando numa perda de 3,6 milhões de *Ether* [47]. Este evento desencadeou uma proposta de *fork*, com o objetivo de corrigir a vulnerabilidade e permitir a devolução dos fundos roubados [1,45].

A *fork* é um mecanismo utilizado para atualizar e melhorar a *Blockchain*, dividindo-se em *soft fork* e *hard fork*, estando a diferença na forma de compatibilidade com as versões anteriores e o impacto na rede [1,45].

Uma *soft fork* consiste numa atualização compatível com versões anteriores da *Blockchain*. Apesar desta *fork* introduzir novas regras e funcionalidades, os nós que não adotam a atualização ainda podem validar e processar as transações. Uma *hard fork* é uma atualização na *Blockchain* que introduz mudanças incompatíveis com as versões anteriores. Com a *hard fork*, os nós que não adotam a atualização não conseguem validar ou processar as transações na nova *Blockchain*, levando à divisão da comunidade e à criação de uma nova criptomoeda, com uma rede independente da original.

Após o ataque à *DAO* na rede *Ethereum*, houve uma divisão na comunidade em relação à resposta ao evento de *fork*. Alguns participantes consideraram que a mudança proposta violava o princípio de imutabilidade da *Blockchain* de que as transações registadas não podem ser alteradas ou eliminadas. Como resultado, ocorreu uma *hard fork*, ou seja, a *Blockchain Ethereum* ramificou em duas direções distintas. Aqueles que aceitaram a *hard fork* migraram para uma nova cadeia, mantendo o nome *Ethereum*, enquanto os participantes que rejeitaram a mudança permaneceram na rede original, que ficou conhecida como *Ethereum Classic*. A *hard fork* foi concretizada em 20 de julho de 2016 [26].

2.4.4 O Trilema da descentralização de *Blockchain*

A escalabilidade é a capacidade de uma *Blockchain* lidar com um grande volume de transações de forma eficiente [19]. À medida que o número de utilizador e transações aumenta, é importante que a rede possa processar esse aumento de transações de forma rápida e eficiente, sem congestionamentos ou atrasos significativos.

O trilema da descentralização afirma que um sistema apenas apresenta dois dos seguintes três aspetos: segurança, escalabilidade ou descentralização, sendo impossível adotar os três ao mesmo tempo [20]. Por um lado, um sistema pode ser seguro e escalável, mas não descentralizado, como o *Google* (serviços baseados em nuvem). Por outro, um sistema pode ser descentralizado e

[11] https://*Ethereum*.org/en/*DAO*/. [Acedido em 02/07/2023]
[12] https://cointelegraph.com/news/takeaways-5-years-after-the-*DAO*-crisis-and-*Ethereum*-hard-fork. [Acedido em 08/07/2023]

escalável, mas não seguro, por exemplo sistemas de *Bittorrent.* Por fim, um sistema pode ser descentralizado e seguro, mas não escalável, exemplos disso são as *Blockchains públicos,* em que o tempo para processar transações individuais aumenta linearmente com o aumento do número de participantes [9]. Disso são exemplos as *Blockchain Ethereum e Bitcoin.*

Com o crescente número de indivíduos a integrar o mercado de criptomoedas, a escalabilidade tornou-se um problema, pois as *Blockchains* públicas, adotadas pela maioria das criptomoedas, começaram a ter limitações na velocidade das transações [1,46]. *Buterin,* fundador de *Ethereum*, prevê que a *Blockchain Ethereum* irá apresentar uma possível solução para este trilema [9]. O objetivo das soluções de escalabilidade procuradas é processar uma maior taxa de transferências sem sacrificar a segurança e a descentralização.

O projeto *Ethereum* é um programa para o qual estão previstas uma série de atualizações. Este processo de melhoria contém cinco fases e cada uma delas apresenta um objetivo específico de melhoria na área de escalabilidade.

A fase "*Merge*" aconteceu no dia 15 de setembro de 2022. Nesse dia o mecanismo de consenso inerente à *Ethereum* passou de *Proof-of-Work* para *Proof-of-Stake (PoS)* [27]. O *PoS* é um processo de validação de transações mais rápida e ecológica do que o *PoW*. A prova de participação exige que os participantes mantenham uma quantidade de criptomoedas na rede para se tornarem validadores, sem precisarem de equipamentos potentes para competir com outros participantes. Quanto maior for a quantidade de criptomoedas detida pelo participante na rede, maior será a probabilidade de este ser escolhido para criar o próximo bloco. Esta mudança resulta num consumo significativamente menor de energia, pois já não é necessário utilizar *hardware* de mineração e consumir eletricidade para competir na rede. Com o *PoS*, a quantidade de energia necessária para a validação das transações é reduzida em mais de 99%[13].

As outras fases de melhoramento são projeções futuras com objetivos específicos [9]. Na segunda fase, "*Surge*", será implementado o *sharding* com o objetivo de reduzir o congestionamento da rede. O *sharding* passará por uma subdivisão de blocos, reduzindo a carga de trabalho de cada avaliador, mas exigindo uma maior participação na rede. Assim, a rede tornar-se-á mais descentralizada e segura. Já na fase "*Verge*" pretende-se introduzir "*Verkle Tees*", um mecanismo de compactação de dados que tem como objetivo aumentar a escalabilidade na rede *Ethereum*. Na fase "*Purge*" pretende-se eliminar o excesso de dados históricos, reduzindo o congestionamento da rede. Por fim, a última fase, "*Splurge*", servirá para pequenas atualizações e melhorias para garantir o bom funcionamento do sistema. A data prevista para a implementação da segunda fase é antes do final do ano de 2023 e para as fases seguintes ainda não existem datas definidas.

[13] https://*Ether*eum.org/en/developers/docs/consensus-mechanisms/pos/pos-vs-pow/ [Acedido em 16/05/2023]

3. Carteiras de criptomoedas

Tendo em conta a evolução rápida das criptomoedas desde o seu início, a segurança inerente tem sido cada vez mais uma preocupação, procurando-se encontrar um ecossistema mais seguro. Por ecossistema entende-se o conjunto de elementos interdependentes que compõem o ambiente no qual as criptomoedas operam [43]. Isto inclui tecnologias e protocolos que estão envolvidos na emissão, transferência, armazenamento e utilidade das criptomoedas.

Deste modo, a importância das carteiras criptográficas é cada vez maior. Estas carteiras são aplicações ou dispositivos usados pelos utilizadores da rede para gerir com segurança os seus ativos [22,23,40]. Ou seja, para usufruir da plataforma da *Blockchain* para qualquer transação o utilizador deve criar uma cripto-carteira. Ao contrário das carteiras tradicionais de bolso, as carteiras criptográficas não armazenam criptomoedas. Na verdade, as criptomoedas não são armazenadas em nenhuma área específica nem existem fisicamente, sendo apenas representadas como dados de transações armazenadas na *Blockchain*. Portanto, não há troca real de moedas físicas, mas sim, de valores de dados de transações registadas na rede.

As carteiras criptográficas permitem que os utilizadores visualizem os seus saldos atuais e históricos, enviem e recebam transações e façam a gestão das suas chaves privadas. Na rede, ao efetuar uma transação, é necessário a utilização de uma carteira criptográfica para que a transação seja assinada digitalmente com a chave privada correspondente. Desta forma, uma carteira de criptomoedas contém três elementos relevantes[14]: a chave privada, a chave pública (derivada da chave privada) e um endereço (derivado da chave pública) [1,45]. A chave pública pode ser identificada como algo semelhante a um número de conta bancária cuja funcionalidade é consultar os movimentos. A chave privada funciona como uma senha para conseguir efetuar as transações e o endereço é chave primária da *wallet*. Assim, a carteira permite aos utilizadores assumirem o controlo total dos seus ativos, uma vez que possuem as suas chaves privadas e têm a propriedade total dos seus fundos.

Devido ao crescimento acentuado das criptomoedas, diferentes tipos de carteiras foram desenvolvidas para lidar com diferentes situações. A classificação das carteiras criptográficas não é objetiva, uma vez que depende de vários fatores, como recursos oferecidos, níveis de segurança, usabilidade e tempo de vida das criptomoedas. Existem várias maneiras de classificar as carteiras e a adequação de cada tipo pode variar conforme as preferências e necessidades dos utilizadores. Com base numa revisão da literatura, foram feitas as classificações descritas abaixo.

[14] https://blockgeeks.com/guides/cryptocurrency-wallet-guide/, 2020. [Acedido em 02/04/2023]

3.1 Classificação pelo método de manutenção de dados da *Blockchain*

As carteiras de criptomoedas podem ser classificadas com base na quantidade de dados da *Blockchain* que elas precisam manter localmente. Quanto mais espaço de armazenamento necessário, maior a independência e segurança da carteira. No entanto, a maior capacidade de armazenamento implica um dispositivo com maior capacidade de armazenamento. Por outro lado, carteiras mais leves economizam espaço, mas dependem mais de outros nós da rede para obter informações relevantes. Assim, as carteiras podem ser divididas em dois tipos: carteiras de nós completo e carteiras *Simplified Payment Verification* (*SPV*).

Carteiras de nós completos

As carteiras de nós completos exigem armazenamento local de todo o histórico de blocos, resultando assim na ocupação de uma quantidade significativa de espaço em disco.

Carteiras *Simple Payment Verification*

As carteiras *SPV* armazenam apenas informações essenciais para verificar a propriedade e o saldo das moedas associadas aos endereços do utilizador [23]. Para verificar transações, elas dependem de nós completos na *Blockchain*. Assim, a necessidade de armazenamento é bastante reduzido comparativamente com as carteiras de nós completos. Estas carteiras são convenientes para utilizadores comuns. A maioria das carteiras para dispositivos móveis e navegadores são carteiras *SPV*.

3.2 Carteiras de utilizadores finais

As carteiras de utilizadores finais permitem às pessoas usufruir dos serviços oferecidos pela tecnologia *on-chain*. Estas carteiras têm interfaces simplificadas e intuitivas, tornando o acesso e a gestão de criptomoedas mais acessíveis para todos. Estas carteiras estão disponíveis em várias formas, satisfazendo diferentes necessidades e preferências pessoais. Existem carteiras *online*, carteiras móveis, carteiras de *hardware* e carteiras de papel.

Carteiras *online*

As carteiras *online* são carteiras que se caraterizam por ser de fácil acesso a partir de qualquer local com conexão à *internet*, não exigindo a instalação de um programa especial num computador pessoal, tipicamente uma *extention plugin* no *browser*, o que as torna convenientes

para os utilizadores [22,23,40]. No entanto, por estarem sempre conectadas à *internet*, as carteiras *online* estão mais suscetíveis a ciberataques.

Carteiras móveis

As carteiras móveis caraterizam-se por serem aplicações instaladas em dispositivos móveis, o que as torna mais convenientes e fáceis de usar para transações de pagamento. Estas carteiras permitem transações de criptomoedas por meio de códigos *QR* e são especialmente adequadas para transações imediatas e de curto prazo [22,23,40].

Carteiras de *hardware*

As carteiras de *hardware* permitem gerar chaves privadas num dispositivo físico fiável de forma *offline*, geralmente num formato *USB* [22,23,25,40]. Exemplos destas carteiras incluem a *Trezor*[15] e a *Ledger Nano S*[16]. Embora estas carteiras possam realizar transações *online*, as chaves públicas e privadas são mantidas *offline*, o que proporciona uma maior segurança.

Carteiras de papel

As carteiras de papel armazenam as chaves privadas num documento impresso, geralmente em formato de sequência alfanumérica [1,22,23,40]. É possível gerar estas carteiras nos sites *online*[17]. Estas são consideradas seguras, pois estão completamente *offline* e imunes a ameaças cibernéticas. No entanto, é essencial mantê-las em locais seguros para evitar perdas ou roubos. Pelas questões de segurança, é recomendável utilizá-las como dispositivos de uso único, ou seja, cada carteira de papel deve ser usada apenas uma vez para receber ou enviar fundos.

As carteiras de papel foram o primeiro método seguro de armazenar criptomoedas *offline*[18]. Inicialmente utilizadas com a *Bitcoin*, estas foram gradualmente substituídas por opções mais avançadas de *software* e *hardware*.

3.3 Classificação pelo método de armazenamento da chave privada

Ao nível do método de armazenamento da chave privada, as carteiras de criptomoedas podem classificar-se em carteiras quentes e frias.

Relativamente às carteiras *hardware*, existem dois tipos, quentes e frias, distinguindo-se pela forma como armazenam e utilizam as chaves privadas. Quando os utilizadores não tencionam

[15] https://blog.trezor.io/paper-wallets-a-relic-of-the-past-1f711ba82b8c. [Acedido em 13/04/2023]
[16] https://www.ledger.com/ [Acedido em 13/04/2023]
[17] https://www.bitaddress.org. [Acedido em 14/04/2023]
[18] https://www.gemini.com/cryptopedia/paper-wallet-crypto-cold-storage#section-issues-with-paper-wallet-storage. [Acedido em 12/04/2023]

ligar-se à *internet*, as chaves privadas são armazenadas num dispositivo *offline*. Neste caso, estas carteiras são consideradas frias, pois são menos vulneráveis a ataques cibernéticos. No entanto, caso os utilizadores pretendam realizar transações *online*, as chaves privadas são utilizadas para assinatura de mensagens na *internet* e, neste caso, as carteiras *hardware* são consideradas quentes, sendo mais vulneráveis a ataques [23,25].

Carteiras quentes / *Hot Wallets*

As carteiras quentes correspondem a carteiras conectadas à *internet* [22,23,40]. Trata-se de carteiras vulneráveis a ataques cibernéticos. A configuração é bastante simples e os utilizadores obtêm facilmente os seus ativos, sendo assim mais conveniente para iniciantes. Estas carteiras são destinadas a efetuar transações correntes. No entanto, devido à baixa segurança associada, não deverão armazenar uma quantidade elevada de criptomoedas. As carteiras *online* e movéis são caraterizadas como carteiras quentes.

Carteiras frias / *Cold Wallets*

As carteiras frias são uma aplicação móvel *offline* e não estão conectadas à rede ou à *internet*, armazenando a chave privada num modo *offline*, num dispositivo como *USB* [22,23,40]. Desta forma, é criada uma camada extra de segurança para as transações. Estas podem ser vistas como um cofre no banco, podendo ser utilizadas para armazenar diferentes tipos de criptomoedas. Podem ser de *harware* ou de papel. As carteiras frias são as mais adequadas para os investidores de longo prazo e os *HODL (Hold On for Dear Life)*. *HODL* é um termo utilizado para designar os investidores de criptomoedas que optam por não vender as suas criptomoedas, independentemente das variações elevadas de preço[19].

3.4 Classificação pelo método de custódia

Considerando a forma de custódia, existem três tipos de carteiras de criptomoedas: carteiras com custódia, carteiras sem custódia e carteiras multiassinaturas.

Carteiras com custódia

Nas carteiras de custódia, as chaves privadas são mantidas por um terceiro de confiança, o que é atraente para utilizadores que têm dificuldade em lidar com a complexidade da criptografia de chave pública[20] [29]. Os utilizadores podem simplesmente fazer *login* usando métodos familiares de autenticação, como senhas, semelhante ao registo de um *homebaking*.

[19] https://academy.binance.com/en/glossary/hodl. [Acedido em 15/04/2023]
[20] https://www.coindesk.com/learn/custodial-wallets-vs-non-custodial-crypto-wallets/. [Acedido em 12/04/2023]

Contudo, existe um risco intrínseco associado, relacionado com a possibilidade da entidade de custódia enfrentar a falência, resultando na perda dos cripto ativos dos utilizadores. O recente colapso da *Exchange* centralizada de criptomoedas *FTX*[21], que oferecia serviços de carteira de custódia, ressaltou uma vulnerabilidade crítica no ecossistema de finanças descentralizadas. Isto destaca uma dependência excessiva destas carteiras centralizadas num ambiente que, à partida, valoriza a descentralização.

Carteiras sem custódia

Nas carteiras sem custódia, as chaves privadas são armazenadas diretamente no dispositivo do utilizador, o que aumenta o risco de perda ou roubo das criptomoedas [29]. Geralmente, carteiras como *Metamask* possuem uma opção de recuperação de chaves privadas através de uma frase *seed*, composta por 12 a 24 palavras [22]. No entanto, é importante salientar que qualquer pessoa que tenha acesso a esta frase terá controle total sobre os seus fundos.

Carteiras multiassinaturas

As carteiras multiassinaturas requerem a utilização de duas ou mais chaves privadas para concluir uma transação [22]. Por exemplo, a *BitGo*[23] é uma carteira multiassinaturas que exige que o utilizador mantenha um, enquanto a segunda chave é armazenada por uma pessoa de confiança e a terceira é mantida pela própria empresa emissora de criptomoeda.

3.5 Classificação pela atividade temporal

No que respeita a atividade temporal, existem dois tipos de carteiras de criptomoedas: carteiras ativas e as carteiras dormentes. Dado a dificuldade em encontrar definições nos artigos académicos, optou-se pela classificação e análise destes tipos de carteiras nos sites de *Exchange Binance* e site *Glassnode* que efetua análise dos comportamentos de criptomoedas, nomeadamente os cálculos de carteiras ativas e dormentes.

Carteiras ativas

Em relação às carteiras ativas, existem diversas definições utilizadas pelos provedores de criptomoedas. No entanto, em geral, são endereços da *Blockchain* que estão a ser utilizadas durante um período determinado relativamente curto. A forma de calcular pode variar, mas uma abordagem comum é contar tanto os que recebem quanto os que enviam transações ao longo de intervalos de tempo predefinidos, como dias, semanas ou meses, dependendo dos objetivos da

[21] https://www.nytimes.com/2022/11/10/technology/ftx-binance-crypto-explained.html. [Acedido em 12/04/2023]
[22] https://www.coindesk.com/learn/custodial-wallets-vs-non-custodial-crypto-wallets/. [Acedido em 12/04/2023]
[23] https://www.bitgo.com/products/custodial-wallets/ [Acedido em 14/04/2023]

análise[24]. Por outro lado, devem considerar os endereços que estiveram envolvidos em transações bem-sucedidas, tornando assim, o cálculo mais precisa[25].

Carteiras dormentes

As carteiras dormentes referem-se às carteiras que não tiveram atividades de transações por um determinado período de tempo[26] e com saldo diferente a zero. Estas carteiras podem ser classificadas por dormentes de longo prazo ou cíclicas. A métrica da média de dormência, que representa o período médio em que uma criptomoeda permanece inativa na carteira do detentor antes de ser utilizada, é usada para identificar os endereços dormentes[27]. De acordo com *Smith*, desde o final de 2012, a média de dormência para *Bitcoins* em negociação raramente caiu abaixo de 20 dias e aumentou acima de 70 dias durante períodos de alta volatilidade no preço do Bitcoin. A partir de 2017, atingiu um pico de acerca 140 dias. Estes números refletem como os detentores da *Bitcoin* tendem a reter as suas moedas por mais tempo em momentos de valorização acentuada da *Bitcoin* [39]. Portanto, a métrica da média de dormência permite compreender o comportamento dos detentores de criptomoedas e o seu período de retenção de moedas, que varia com base nas condições do mercado.

- **Carteiras dormentes por vários ciclos económicos de criptomoedas**

Estas carteiras dormentes são aquelas que não apresentam movimentações de transações ao longo de vários ciclos económicos de criptomoedas com um saldo substancialmente diferente a zero[28]. Estes ciclos referem-se aos padrões e fases de flutuações de preços que ocorrem no mercado de criptomoedas ao longo do tempo. Os ciclos económicos de criptomoedas estão diretamente associado com o processo *Halving* da *Bitcoin*, ou seja, aos eventos periódicos da diminuição das recompensas por bloco fornecidas aos mineradores. Devido à predominância da *Bitcoin* no mercado de criptomoedas, as *altcoins* seguem o padrão estabelecido por esta criptomoeda, e a *Ether* não é uma exceção, como pode ser observado na *Figura 2*, onde se constata que o histórico de preços da *Bitcoin* e da *Ether*[29] apresentam um comportamento muito similar. Conforme ilustrado na *Figura 3*, após o processo de *Halving*[30], torna-se possível identificar o início de um período de expansão na criptomoeda *Bitcoin*.

[24] https://www.binance.com/ru/feed/post/42823 [Acedido em 20/08/2023]
[25] https://studio.glassnode.com/metrics?a=BTC&m=addresses.ActiveCount [Acedido em 20/08/2023]
[26] https://crystal*Blockchain*.com/articles/2020-report-on-fund-sources-for-dormant-*Bitcoin*-addresses/ [Acedido em 16/04/2023]
[27]https://studio.glassnode.com/metrics?a=BTC&category=&chartStyle=line&ema=7&m=indicators.AverageDormancy&mAvg=0&mMedian=0&resolution=24h&s=1590889192&u=1622425192&zoom=365 [Acedido em 20/08/2023]
[28] https://www.mdpi.com/1911-8074/13/1/8, 2020/01/03 [Acedido em 06/07/2023]
[29] https://pt.investing.com/crypto/ethereum/historical-data [Acedido em 20/07/2023]
[30] https://www.gate.io/pt/explore/*Bitcoin-Halving*-countdown [Acedido em 15/07/2023]

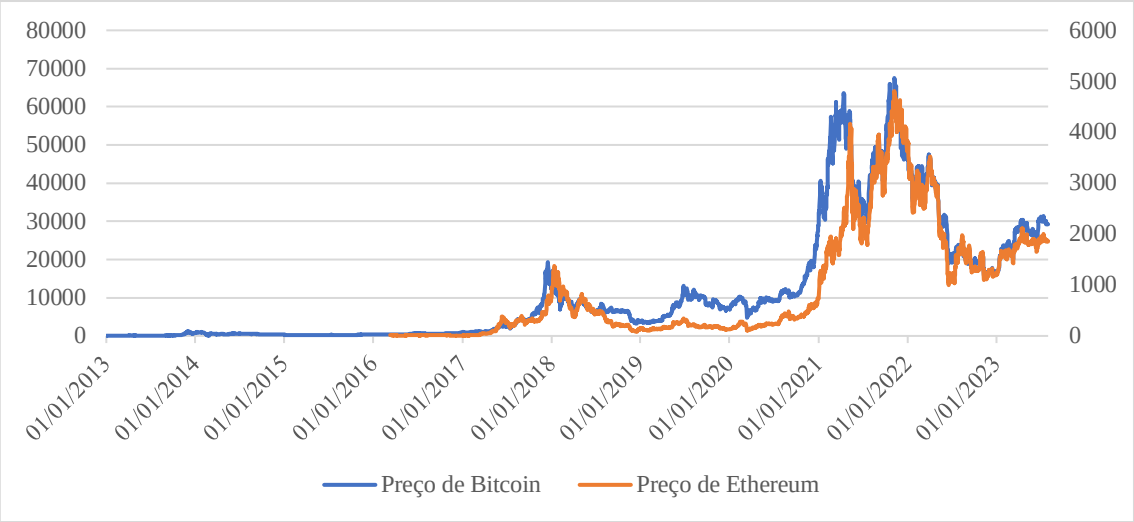


Figura 2: Preço histórico de Bitcoin e de Ether, demonstrando a forte correlação entre estas duas criptomoedas

Figura 3: Evolução de Bitcoin retirada no site gate.io, demonstrando a subida acentuada do preço após o processo de Halving

Exemplos de carteiras dormentes por vários ciclos económicos são os *HODL* de *Bitcoin*, que foram os primeiros investidores a adquirir criptomoedas e mantiveram as suas participações ao longo do tempo. No entanto, as carteiras perdidas apresentam características semelhantes. Trata-se de carteiras cujo detentores perderam a sua chave privada e, portanto, veem impossibilidade as movimentações de criptomoedas a partir dessa data. Ao analisar dados de transações, distinguir entre carteiras dormentes e perdidas é desafiador, já que apenas a ausência de atividade por longos períodos é a única informação disponível. Segundo *IntoTheBlock*, quase 30% das *Bitcoin* e estão perdidas ou dormentes em 2023[31].

- **Carteiras dormentes cíclicas, por um ciclo económico de criptomoedas**

Estas carteiras seguem um padrão de movimentação alinhado com o evento de *Halving*, que ocorre a cada quatro anos. Durante um ciclo económico completo, estas carteiras acumulam

[31] https://u.today/heres-how-many-bitcoins-are-now-lost-forever-intotheblock [Acedido em 25/07/2023]

criptomoedas durante os períodos de recessão e vendem-nas nos momentos de expansão subsequente.

3.6 Estado de arte na análise de carteiras

Existe uma variedade de trabalhos de pesquisa relacionados com a análise de grupos e tendências nos dados de transações da rede *Blockchain Ethereum*, usando as palavras *""Clustering" and "Ethereum" and "Addresses" Using "Machine" and "Learning""* e por ordem da relevância no *GoogleScholar* foram obtidos 2 290 artigos. Foram analisados os primeiros 30, entre os quais foi selecionados 3 na área de análise de grupos aplicada à caraterização de carteiras de criptomoedas e 2 na área da análise de comportamentos maliciosos. Comportamentos maliciosos em criptomoedas são ações fraudulentas ou ilegítimas que buscam lucros financeiros desonestos. A pseudonímia da *blockchain*, que permite ocultar as identidades reais dos envolvidos, torna-se uma característica atrativa para a prática dessas atividades [45].

Na *Tabela 3* alguns autores utilizam *Machine Learning* não supervisionado, ou seja, análise de grupos sem *rótulos* previamente definidos [4,32], enquanto outros utilizam *Machine Learning* supervisionado, ou seja, com *rótulos* [34]. As métricas apresentadas também são diversas. No trabalho [34], por exemplo, foram utilizadas métricas derivadas da matriz de confusão.

Análise de grupos	Métodos utilizados	Métricas de comparação	Resultados	Auto, Ano e Revista
CHARACTERIZING THE ETHEREUM ADDRESS SPACE	k-means clustering	Score Calinski Harabaz (mede a relação entre a variância intergrupos e variância intragrupos)	Com o algoritmo k-means foram criados 4 clusters com caraterísticas distintas	Payette, J., et al. (2017)
Transactional Data Analytics for Inferring Behavioural Traits in Ethereum Blockchain Network	K-means clustering e Density-based clustering	Análise de Silhouette (mede a coesão e a separação dos clusters)	O algoritmo de clustering k-means obteve um score de Silhouette mais alto em comparação com o clustering baseado em densidade	Bhargavi, M. S., et al. (2020). In 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, (pp. 485-490).
CEAT: Categorising Ethereum Addresses' Transaction Behaviour with Ensemble Machine Learning Algorithms	Random Forest, Gradient Boosting e XGBoost	Accuracy, F1-Score, precision, and Recall	Em geral, o modelo XGBoost alcançou o melhor desempenho em comparação com os classificadores Random Forest e Gradient Boosting	Pragasam, T. T. N., et al. (2023). Computation. 11(8), 156

Tabela 3: Trabalhos Relacionados com a análise de vários grupos dos endereços de Ethereum, usando Machine Learning supervisionado e não supervisionado

Na *Tabela 4* encontra-se uma seleção de trabalhos na área da análise de comportamentos maliciosos [33,35], sendo a análise baseada numa variável binária, endereço malicioso ou não malicioso. Devido à natureza das transações, o comportamento malicioso é um tema relevante. A maioria dos métodos apresentados nesta área são supervisionados.

Comportamentos maliciosos	Métodos utilizados	Métricas de comparação	Resultado	Auto, Ano e Revista
Detecting Malicious Ethereum Entities via Application of Machine Learning Classification	Logistic Regression, Support Vector Machine, Random Forest, Stacking Classifier e AdaBoost.	Accuracy, F1-Score, Precision and Recall	As avaliações demonstraram o bom desempenho nos Random Forest, Stacking Classifier e AdaBoost.	Poursafaei, F. et al. (2020). In 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services, (pp. 120-127)
Classifying Transactional Addresses using Supervised Learning Approaches over Ethereum Blockchain	Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbours, Gaussian Naïve Bayes, K-Nearest Neighbours, Random Forest, Bagging, EXtreme Gradient Boosting e Adaptive Boosting	Accuracy, F1-Score, Precision and Recall	Modelos de classificação linear e não linear tiveram um desempenho superior em relação aos modelos *ensemble (Random Forest, Bagging, Extreme Gradient, Boosting e Adaptive Boosting)*	Saxena, R., et al. (2023). Procedia Computer Science, 218, 2018-2025

Tabela 4: Trabalhos relacionados com análise de comportamentos maliciosos dos endereços de Ethereum, usando Machine Learning supervisionado

O objetivo desta Tese consiste na caraterização de endereços, efetuando análise de 4 grupos e análise de *cold wallets* e *hot wallets* no contexto da *blockchain Ethereum*. Da pesquisa realizada no *GoogleScholar*, não foram encontrados trabalhos na literatura relacionados que apliquem técnicas de *Machine Learning* aplicadas à análise de *cold e hot wallets*.

4. Metodologia – Análise de grupos e classificação

Identificar um tipo de endereço com base apenas nos dados da *Blockchain* pode ser um desafio, uma vez que estes dados não fornecem informação direta sobre o tipo de carteira que está a ser utilizada. No entanto, a vasta quantidade de transações e interações na rede *Ethereum* permite aplicar técnicas de análise de dados, como o *Machine Learning*, para descortinar propriedades que podem contribuir para uma melhor compreensão do ecossistema, incluindo a caraterização das carteiras. Neste trabalho, vão ser aplicados vários algoritmos de análise de dados com o propósito de agrupar e classificar os endereços *Ethereum*. O objetivo é identificar grupos de endereços que compartilham características semelhantes, levando a uma melhor compreensão de como diferentes entidades se comportam na rede *Ethereum* e a que tipo de carteiras estão associados os endereços.

4.1. Métodos de *Machine Learning* para análise de grupos

Nesta secção são apresentadas metodologias de análise de dados relacionadas com a análise de endereços da *Ethereum*, nomeadamente o *Machine Learning*, e a utilização de métricas para avaliação de modelos de algoritmos supervisionados. É discutida a importância da utilização de métricas adequadas na avaliação de modelos, bem como uma visão geral dos algoritmos de classificação supervisionada mais comuns utilizados em análises de dados.

Machine Learning é uma área da ciência de computação que utiliza dados existentes para capturar padrões para prever dados futuros [37]. A capacidade de capturar estes padrões é diretamente influenciada pelas caraterísticas selecionadas e a sua combinação para o conjunto de dados [21]. Assim, a seleção das características relevantes e a eliminação das irrelevantes é uma questão central na *Machine Learning* [6]. Muitos algoritmos de *Machine Learning* fornecem parâmetros para uma melhor seleção e combinação das características.

Existem três tipos de *Machine Learning*: sistema de aprendizagem supervisionado, sistema de aprendizagem não supervisionado e sistema de aprendizagem por reforço. Um sistema de aprendizagem supervisionado é o processo de treinar um modelo com dados rotulados, ou seja, a informação relativamente às categorizações estão disponíveis nos dados de treino. O objetivo é treinar o modelo a realizar classificações semelhantes nos dados de teste, que não possuem rótulos. Já no sistema de aprendizagem não supervisionado os dados são não rotulados, ou seja, não há distinção entre dados de treino e teste, o objetivo neste contexto é encontrar estruturas nos dados sem depender de informações prévias de categorização. Por fim, um sistema de aprendizagem por reforço é um paradigma na qual um sistema interage com um ambiente dinâmico para aprender a realizar ações sequenciais com base em experiências anteriores e ajusta a sua estratégia para melhorar o desempenho ao longo do tempo.

Modelos de aprendizagem supervisionados

A classificação de dados rotulados pode ser realizada usando vários métodos de aprendizagem supervisionados, incluindo a Regressão Logística, *Random Forest*, *GradientBoosting* e *AdaBoost* [7,18,31,37]. Estes serão os modelos utilizados no Capítulo 5.

Regressão Logística:

A Regressão Logística é um método estatístico que visa criar um modelo para prever os valores de uma variável categórica, frequentemente binária, a partir de um conjunto de variáveis explicativas, que podem ser contínuas ou binárias. A Regressão Logística permite estabelecer uma relação entre as características, ou variáveis independentes, presentes nos dados e a probabilidade de pertencer a uma classe específica, ou seja, variável dependente da seguinte forma (ver por exemplos [13, 17, 21, 31]):

$$\hat{p}\ (y_i = 1|X_i) = \frac{1}{1+\exp\left(-X_i w\right)}$$

onde w é um vetor de peso, X_i é o vetor do i-ésimo elemento da amostra e y_i é o valor da variável dependente para o i-ésimo elemento da amostra.

Devido à importância da seleção de variáveis e a sua combinação para prever as observações futuras, vai ser introduzido o algoritmo regularizado.

Segundo a biblioteca *scikit-learn*[32], a Regressão Logística anterior, regularizada, pode ser escrita como um problema de minimização da seguinte forma:

$$\min_w \left\{ C \sum_{i=1}^{n} [-y_i \log\left(\hat{p}(X_i)\right) - (1-y_i)\log\left(1-\hat{p}(X_i)\right)] + \left(\frac{1-p}{2} w^T w + p\left|\left|w\right|\right|_1\right) \right\}$$

onde n é o tamanho da amostra e C é uma constante positiva que é inversamente proporcional à intensidade de regularização. Quanto menor o valor de C, mais forte é a intensidade da regularização.

Definem-se os parâmetros de regularização $l1 = \left|\left|w\right|\right|_1 = \sum_{i=1}^{n} |w_i|$, ou seja, a minimização da soma de pesos absolutos das variáveis e $l2 = \frac{1}{2}||w||_2^2 = \frac{1}{2} w^T w = \frac{1}{2}\sum_{i=1}^{n} w_i^2$, ou seja, é a minimização da soma dos quadrados de pesos absolutos das variáveis. Os parâmetros de regularização "$l1$" e "$l2$" correspondem a dois métodos distintos para melhorar a capacidade de generalização do modelo, evitando o *overfitting* [31]. Quando $p = 1$ tem-se regularização do tipo "$l1$". Esta permite a seleção dos coeficientes, pois permite que alguns deles se tornem exatamente zero. Assim, apenas as variáveis independentes mais relevantes são mantidas no modelo. Em contraste, quando $p = 0$ tem-se a regularização do tipo "$l2$", que penaliza coeficientes grandes, mas não os levam necessariamente a zero.

[32] https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

O hiperparâmetro *"C"* controla o equilíbrio entre um ajuste preciso aos dados de treino (baixa regularização) e a prevenção do *overfitting* (alta regularização) pois valores maiores de *"C"* resultam numa menor regularização, tornando o modelo mais flexível para ajustar os dados de treino. Por outro lado, valores menores de *"C"* aumentam a regularização, tornando o modelo mais restrito e resistente ao *overfitting* [13].

Por fim, o valor de corte na Regressão Logística é um ponto de decisão que determina como as observações são classificadas para pertencer a uma classe ou não, com base nas probabilidades calculadas pelo modelo. A escolha deste valor é um aspeto importante no ajuste e na interpretação do modelo, sendo que, por padrão, ele é definido como 0,5.

Modelos combinados

A combinação de modelos tem vindo a ganhar cada vez mais relevância, com o objetivo de melhorar o desempenho do modelo final, reduzindo o erro e aumentando a generalização. Estes métodos envolvem a construção de um conjunto de classificadores que, posteriormente, trabalham em conjunto para classificar novos dados por meio de um processo de votação.

Random Forest (RF):

Random Forest é um algoritmo que cria um conjunto de árvores de decisão a partir de subconjuntos aleatórios dos dados de treino. Cada árvore vota numa previsão, e a previsão final é determinada pela votação por maioria (para classificação) ou pela média (para regressão) das previsões das árvores. De seguida, é apresentado o algoritmo do *Random Forest* baseando nos artigos [7,37].

Algoritmo de Random Forest

Entrada:
Conjunto de dados de treino $S = (\mathrm{X}_1, y_1), \dots, (\mathrm{X}_m, y_m)$
Número de iteração M
For $\boldsymbol{m = 1, \dots M}$:

a) **Criar uma amostra boostrap** (X_a, y_a) com n amostras de (X, y), sendo amostras aleatórias com reposição
b) **Treina e ajusta várias árvores de decisão** A para o conjunto de treino (X_a, y_a), utilizando n variáveis independentes para a escolha de cada nó.
c) Cada árvore individual é construída **na maior extensão possível**

Saída: O resultado final de classificação é determinado pela votação da maioria das classificações individuais.

O hiperparâmetro M controla o número de árvores de decisão independentes que são construídas no modelo. Aumentar o número de estimadores geralmente torna o modelo mais robusto, mas também pode aumentar o tempo de treino e o facto de aumentar exponencialmente

o número de estimadores pode não melhorar significativamente o desempenho do modelo, mas sim aumentar o custo computacional.

AdaBoost (AB):

O *AdaBoost (Boosting)* é um algoritmo meta-heurístico de combinação de múltiplas hipóteses fracas para formar uma hipótese forte de um modelo fraco. Isso é alcançado ajustando iterativamente a distribuição das observações, dando maior peso às observações que foram classificadas incorretamente para focar nos casos mais difíceis de classificar. Assim, permite melhorar o desempenho global do modelo. Um algoritmo de classificação fraco é qualquer algoritmo simples que apresenta uma taxa de erro ligeiramente melhor do que uma escolha aleatória. Por exemplo, árvore de decisão de profundidade limitada [37]. De seguida, é apresentado o algoritmo de *AdaBoost*.

Algoritmo do *AdaBoost*

Entrada:
Conjunto de dados de treino $S = (\mathrm{X}_1, y_1), \dots, (\mathrm{X}_m, y_m)$
Modelo fraco: Weak Learner (WK)
Número de iteração T
Learning rate $\eta \in (0,1]$
Inicialização: $\mathrm{D}^{(1)} = (\frac{1}{m}, \dots, \frac{1}{m})$.
For $\boldsymbol{t = 1, \dots T}$:
WL: $h_t = \mathrm{WL}(\mathrm{D}^{(t)}, S)$
Calcular o erro ponderado do hipótese fraca h_t: $\epsilon_t = \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathrm{X_i})]}$
Calcular o peso associado ao modelo fraco h_t: $w_t = \frac{1}{2}\log(\frac{1}{\epsilon_t} - 1)$
Atualizar os pesos: $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathrm{X_i}))}{\sum_{j=1}^{m} D_i^{(t)} \exp(-w_t y_i h_t(\mathrm{X_i}))}, \forall i = 1, \dots, m$
Saída: Modelo final: $h_s = sign\,(\sum_{t=1}^{T} D_i^{(t)} w_t \eta h_t(\mathrm{X}))$

Figura 4: Algoritmo do AdaBoosting retirado da pág. 135 de [37]

O processo do *AdaBoost* começa com um conjunto de dados de treino S, sendo X_m variáveis independentes e y_m as classes correspondentes (variável independente). Inicialmente todas as observações têm pesos igualmente atribuídos.

Em cada interação, o algoritmo calcula o erro ϵ_t da hipótese fraca designada por h_t. O ϵ_t é a soma de produtos dos pesos das observações $D_i^{(t)}$ pelas diferenças entre as previsões da classificação $h_t(\mathrm{X_i})$ e as classificações reais y_i da hipótese fraca h_t. A função $\mathbb{1}_{[y_i \neq h_t(\mathrm{X_i})]}$ é 1, se a condição for verdadeira e 0 caso contrário. O peso w_t associado a h_t é calculado de forma inversamente proporcional ao erro da hipótese fraca h_t, significando que as hipóteses fracas com baixo erro recebem um peso maior, o que lhes confere maior influência na hipótese final.

De seguida ocorre a atualização dos pesos $D_i^{(t+1)}$ calculados por um rácio, onde o numerador é o produto do peso inicial por um termo exponencial que depende do peso w_t, da classificação real y_i e da previsão da hipótese $h_t(\mathrm{X_i})$.O denominador do rácio é a soma dos termos exponenciais para todas as observações do treino.

O resultado final é a hipótese forte criada pela função $sign$ da combinação de todas as hipóteses fracas do modelo fraco selecionado com uma *learning rate* associado. Função essa permite realizar previsão nas observações de teste. Caso a soma dos modelos fracos for positivo a função retorna +1, caso contrário, retorna -1.

O número de estimadores T e a *learning rate* η são dois hiperparâmetros do *AdaBoost.* O número de estimadores refere-se ao número de hipóteses fracas que serão combinadas para criar a hipótese forte. Em geral, quanto mais estimadores, melhor o desempenho. No entanto, caso o aumento for excessivo pode levar ao *overfitting*, tornando o modelo muito sensível aos dados de treino. A *learning rate* é um parâmetro que controla a contribuição de cada nova hipótese fraca para o modelo final. Uma *learning rate* menor permite que cada hipótese fraca tem uma menor contribuição, tornando o processo mais lento. Por outro lado, um valor de *learning rate* maior permite ajustes mais rápidos, mas também pode levar ao *overfitting*.

GradientBoosting (GB)

O *GradientBoosting (Boosting)* também é um algoritmo meta-heurístico de combinação de múltiplas hipóteses fracas para formar uma hipótese forte de um modelo fraco. O algoritmo *GradientBoosting* é um processo iterativo, iniciando com uma previsão constante. A cada iteração, é adicionado um novo termo ao modelo corrente, com o objetivo de reduzir gradualmente o erro de previsão. O processo repete-se até que uma determinada condição de paragem seja satisfeita [18]. De seguida, é apresentado o algoritmo de *GradientBoosting.*

Algoritmo do GradientBoosting*:*

Entrada:

Conjunto de dados de treino $S = \{(\mathrm{X}_1, y_1), \ldots, (\mathrm{X}_m, y_m)\}$

Número de iteração T

Learning rate $\eta \in (0,1]$

Inicialização:

$\mathrm{F}_0(\mathrm{X}) = \mathrm{argmin}_\rho \sum_{i=1}^{N} L(y_i, \rho)$

For $\boldsymbol{m} = \mathbf{1}, \ldots \boldsymbol{M}$:

Calcular os resíduos:

$$\widetilde{y_i} = \left[\frac{\partial L(y_i, F(X_i))}{\partial F(X_i)}\right]_{F(X) = F_{m-1}(X)}, i = 1, \ldots, N$$

Ajustar um modelo fraco $h_m(X_i; a_m)$ aos resíduos:

$$a_\mathrm{m} = \mathrm{argmin}_{a,\beta} \sum_{i=1}^{N} [\widetilde{y_i} - \beta h(X_i; \mathrm{a})]^2$$

Encontrar o fator de escala ρ_m:

$\rho_\mathrm{m} = \mathrm{argmin}_{\rho>0} \sum_{i=1}^{N} L(y_i, \mathrm{F}_{m-1}(X_i) + \rho h_m(X_i; a_m))$

Atualizar: $F_m(X)$:

$$F_m(X) = F_{m-1}(\mathrm{X}) + \eta \rho_\mathrm{m} h_m(X; a_m)$$

Saída: $F_M(X) = \sum_{i=1}^{N} \eta \rho_\mathrm{m} h_m(X_i; a_m))$

Figura 5: Algoritmo do GradientBoosting retirado da pág. 1193 de [18]

A função $\mathrm{F}_0(\mathrm{X})$ é a constante que minimiza o somatório dos erros ao quadrado utilizando a função da perda $L(y, \rho)$, representando uma estimativa inicial para a variável.

Os resíduos $\widetilde{y_i}$ representam a diferença entre as classificações reais e as previsões. O índice $F(X) = F_{m-1}(X)$ refere-se à previsão criada no passo anterior (m-1).

$h_m(X_i; a_m)$ é um modelo fraco selecionado, normalmente é uma árvore de decisão.

ρ_m é um parâmetro que minimizar a função de perda considerando o modelo fraco.

O modelo final é a soma das $F_m(X)$ ajustados em cada iteração.

O número de estimadores T e a *learning rate* η são dois hiperparâmetros do *GradientBoosting*, apresentando as mesmas funcionalidades do *AdaBoost.*

Os hiperparâmetros selecionados foram os mencionados nos algoritmos discutidos anteriormente. No entanto, é importante notar que a escolha de hiperparâmetros pode variar e poderia ser exploradas outras opções ou incluídos mais hiperparâmetros para uma análise mais abrangente. Devido a restrições específicas e limitações do tempo, não foi possível realizar uma exploração mais extensa.

4.2 Métodos para a avaliação de modelos

A obtenção de uma análise de dados precisa requer a seleção de um algoritmo, dentre aqueles disponíveis, que melhor se adeque aos dados e objetivos específicos. Assim, é essencial estabelecer métodos que permitam a avaliação e comparação de diversos algoritmos e parâmetros associados.

4.2.1 Métricas de Avaliação de Modelos de Classificação

A matriz de confusão representada na *Tabela 5* é uma métrica para classificação binária que permite analisar diretamente a precisão do modelo para diferentes grupos [8]. As colunas apresentam os grupos previstos e as linhas os grupos reais.

Sim = Carteira Fria **Não = Carteira Quente**		**Previsão**	
		Sim	Não
Real	Sim	Verdadeiro Positivo	Falso Negativo (Erro Tipo II)
	Não	Falso Positivo (Erro Tipo I)	Verdadeiro Negativo

Tabela 5: Matriz de Confusão de uma classificação binária

A partir da matriz de confusão podemos inferir outras métricas [9]:

a) A Percentagem de casos Corretamente Classificados (PCC) ou *accuracy* é a medida mais usada para validar a precisão dos algoritmos, corresponde ao total de número dos casos bem classificados sobre o total das observações.

$$PCC = \frac{VP + VN}{VP + FN + FP + VN} = \frac{VP + VN}{S + N}$$

Verdadeiro Positivo (VP) representa as observações corretamente classificadas como positivas, Verdadeiro Falso (VN) representa as observações corretamente classificadas como negativas, Falso Negativo (FN) representa as observações incorretamente classificadas como negativas e Falso Positivo (FP) as observações incorretamente classificadas como positivas. S é a soma dos VP e FN, ou seja, o total das observações classificadas corretamente e N é a soma dos FP +VN, que representa o total das observações classificadas incorretamente.

No entanto, este medida pode não ser a mais adequada em situações quando esta métrica considera que os erros de predição em ambas as classes têm o mesmo impacto e importância e no caso de desequilíbrio no total de observações apresentadas. Este desequilíbrio afeta no desempenho da matriz de confusão devido à sensibilidade na proporção das grupos no conjunto de observações. Assim, deve incorporar outras métricas que ofereçam resultados mais robustos, especialmente nos casos em que a *accuracy* (PPC) não seja uma opção adequada.

b) A medida de sensibilidade mede a proporção de casos positivos que foram corretamente classificados como positivos:

$$sensibilidade(\ classe\ i) = \frac{VP(\ classe\ i)}{VP(\ classe\ i) + FN(classe\ i)}$$

No caso da análise ser multiclasse, a fórmula é ajustada de seguinte forma:

$$sensibilidade(\ classe\ i) = \frac{VP(\ classe\ i)}{VP(\ classe\ i) + FN(classe\ i)}$$

Em que VP (classe i) é o número de observações da classe i corretamente classificadas como classe i e FN (classe i) é o número de observações da classe i incorretamente classificadas como não pertencentes à classe i.

c) A precisão mede a proporção de casos corretamente classificados no total de casos classificados como positivos:

$$precisão = \frac{VP}{VP + FP}$$

d) A métrica *F-Score* relaciona a sensibilidade e a precisão. O *F-Score* é particularmente útil em situações onde não é desejável favorecer apenas destas uma métrica em detrimento da outra. Isso é relevante quando existe um desequilíbrio nas observações e o objetivo é melhorar a sensibilidade sem comprometer a precisão.

$$F\_Score = \frac{(1 + \beta^2) * sensibilidade * precisão}{\beta^2 * sensiblidade + precisão}$$

Em que o β corresponde à importância relativa entre precisão e sensibilidade. $\beta > 1$ destaca a importância de sensibilidade, $\beta < 1$ destaca a importância de precisão.

e) A curva *Receiver Operating Characteristic* (*ROC*) ilustra a relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos (1 - especificidade) num modelo de classificação binária. A *Figura 6* retiradas de [28] mostra exemplos da *ROC*. A curva *ROC* de um modelo de qualidade mais elevada terá uma maior inclinação desde o ponto zero. À medida que o número de casos verdadeiros positivos diminui, o declive diminui, tendendo para a reta horizontal y=1. Quanto maior a *ROC*, melhor é a capacidade de separar as classes positivas e negativas.

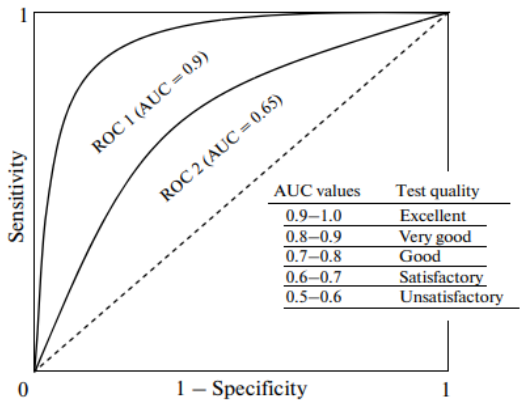


Figura 6: Curva Receiver Operating Characteristic [28]. Trata-se de uma métrica que avalia o grau de separação entre as classes positivas e negativas num modelo de classificação binária

Todas as métricas referidas anteriormente serão utilizadas para avaliação dos métodos.

4.2.2 Métodos de Avaliação

Para aplicar as métricas anteriormente referidas, é necessário que os resultados obtidos na fase de avaliação constituam uma boa representação do desempenho real do modelo quando aplicado para prever novos casos. Assim, existem dois métodos mais adotados na fase de avaliação [37]:

- Validação por divisão de amostra (*holdout validation*): divide os dados em dois conjuntos, sendo um para treinar o modelo e outro para testar o modelo.
- Validação cruzada (*cross-validation*) com *k-fold*: é uma técnica que divide os dados em k subconjuntos de tamanhos iguais, aleatoriamente, sendo que em cada iteração o modelo é treinado com k-1 subconjuntos e um dos subconjuntos é excluído. De seguida, o modelo é validado no subconjunto excluído. Este processo repete-se k vezes até que todos os subconjuntos sejam usados para validação do modelo.

Neste trabalho considera-se a validação cruzada como o método de avaliação para os modelos selecionados tendo sido adotado um 5-*fold*, ou seja, o treino é realizado com 80% das observações e 20% é para tese.

5. Análise de carteiras *Ethereum*

5.1 Recolha de dados

A transparência dos dados na *Blockchain Ethereum* é uma das características mais distintivas desta tecnologia. Enquanto nos sistemas tradicionais as transações e as informações são armazenadas de forma centralizada e muitas vezes inacessíveis ao público, a *Blockchain Ethereum* apresenta os seus dados de forma pública e transparente. Esta transparência proporciona uma fonte rica de dados para análise, permitindo uma visão detalhada e precisa das transações e informações registadas na rede.

5.1.1 Fonte dos dados

Na rede *blockchain Ethereum*, é possível consultar informações detalhadas sobre as transações, a *Tabela 6* apresenta os dados de um bloco específico retirado do site *Etherscan.io.* A estrutura de dados apresentada neste site é de fácil compreensão, tornando a interpretação das informações bastante acessível. Contudo, a sua extração pode ser desafiadora devido a algumas restrições, tais como as limitações no número de pedidos permitidos por minuto através da *Application Programming Interface* (*API*)[33]. Uma *API* atua como uma ponte de comunicação que possibiita que as pessoas acedam e partilhem informações com outros sistemas [1].

Fontes alternativas para a extração de dados incluem *Dune Analytics*, *BigQuery* e *CoinGecko*.

Campo	**Descrição**
Número do Bloco	Número sequencial do bloco na cadeia
Estado	Estado do bloco indica se este foi validado e adicionado corretamente à cadeia (por exemplo estado “confirmado” ou “pendente”)
Timestamp	Marca de data e hora da criação do bloco
Número de Transações	Número total de transações incluídas no bloco (transações internas (*Smart Contract*) e transações)
Withdrawls	Fundos ou ativos removidos do *Smart Contract* ou conta durante a execução das transações no bloco
Fee Recipient	Endereço da carteira ou contracto que recebe as taxas coletadas das transações incluídas no bloco.
Recompensa	Recompensa em criptomoeda (Ether no caso do *Blockchain Ethereum*) concedida ao minerador que adicionou o bloco à cadeia com sucesso

[33] https://etherscan.io/apis [Acedido em 10/10/2023]

Total de Dificuldade	Dificuldade de criar o bloco, é um valor que representa o grau de dificuldade que foi encontrado no *hash* do bloco.
Tamanho do Bloco	Tamanho em *bytes* do bloco
Gas utilizado	Quantidade de unidades de gas utilizada para executar as transações no bloco. O gas é uma medida de recursos computacionais necessários para executar as operações das transações
Gas limite	Limite máximo de unidades de gas permitido para todas as transações incluídas no bloco
Dados Adicionais	Dados auxiliares que podem ser incluídos no bloco, como informações de versão
Hash do Bloco	*Hash* que representada a identidade do bloco
Parent Hash	*Hash* do bloco anterior na cadeia
Root da Árvore de *Merkle*	*Hash* da raiz da árvore de *Merkle* das transações incluídas no bloco, ou seja, uma representação criptográfica de todas as transações no bloco
WithdrawalsRoot	*Hash* da raiz da árvore de *Merkle* de *Withdrawals* realizadas no bloco
Nonce	Número de sequência usado para garantir a ordem correta das transações enviadas por um carteira. Cada transação tem um *nonce* único.

Tabela 6: Informações de um Bloco na Blockchain Ethereum

A *Tabela 7* apresenta as informações que compõem uma transação na *Blockchain Ethereum* obtido no site *Etherscan.io*. Cada transação é identificada de forma exclusiva pelo seu *hash* e contém informações importantes sobre a transferência de valor e a execução de contratos inteligentes na rede. É relevante notar que na *Blockchain Ethereum* existem dois tipos de transações: transações da *Ether* e transações internas.

As transações da *Ether* referem-se à transferência direta da criptomoeda nativa da *Ethereum*, *Ether*, de um endereço para outro. Estas transações são utilizadas para enviar *Ether* entre carteiras e realizar transferências de valor entre utilizadores. Já as transações internas são transações que interagem com *Smart Contract.* Estas últimas permitem que os utilizadores executem funções e operações programadas em *Smart Contract* existentes na *Blockchain*.

No presente trabalho, o foco está nas informações das transações da *Ether*. O campo *InputData*, permite distinguir entre os dois tipos de transações. Nas transações da *Ether*, este campo é preenchido com "*0x*", enquanto nas transações internas, contém informações mais complexas, tornando a análise mais clara e específica.

Campo	Descrição
Hash da transação	*Hash* que representa a identidade da transação
Bloco	Número do bloco em que a transação está incluída
Remetente	Endereço da carteira que enviou a transação
Destinatário	Endereço da carteira ou *Smart Contract* que recebeu a transação
Valor	Valor em criptomoeda (Ether) enviado na transação
Taxa de Gás	Valor em criptomoeda pago como taxa de gas para a execução da transação
Limite de Gás	Limite máximo de unidades de gas permitido para executar a transação
Dados	Dados adicionais que podem ser incluídos na transação
Estado	Estado da transação, indicando se foi confirmada, pendente ou falhou
Timestamp	Marca de data e hora em que a transação foi enviada
InputData	Informações específicas necessárias para a execução de um contrato inteligente. Estas informações são passadas como parâmetros ou comandos para o contrato inteligente.

Tabela 7: Informações de uma transação na Blockchain Ethereum

Fases de processamento dos dados

Primeira fase: realizar uma análise abrangente aos dados disponíveis na *internet* para identificar endereços com base nas informações conhecidas de várias fontes fiáveis. Estas fontes incluem dados de diferentes plataformas e sites relevantes, como *Dune Analytics, BigQuery* e *CoinGecko.*

- *Dune Analytics* é uma plataforma de análise de dados de *Blockchain* que permite aos utilizadores explorar e visualizar dados de diversos *Blockchain*s. A linguagem de pesquisa utilizada é o *SQL*[34].
- *BigQuery* é uma plataforma de análise de dados do *Google Cloud* que permite aos utilizadores extrair quantidades elevadas de dados através da linguagem de *SQL*. Os dados são atualizados regularmente e no caso da *Blockchain Ethereum*, contem informações por exemplo sobre transações, blocos e endereços[35].
- *CoinGecko* é um site que fornece informações detalhadas sobre criptomoedas, como os preços, volume de negociação, capitalização de mercado, informações sobre *Exchanges*[36].

[34] https://dune.com/browse/dashboards [Acedido em 3/08/2023]
[35] https://console.cloud.google.com/bigquery?p=bigquery-public-data&d=crypto_*Ether*eum_classic&page=dataset&project=diesel-amulet-382015&ws=!1m5!1m4!4m3!1sbigquery-public-data!2scrypto_*Ether*eum!3stransactions [Acedido em 25/08/2023]
[36] https://www.coingecko.com/en/exchanges [Acedido em 3/08/2023]

O objetivo foi mapear e categorizar endereços com rótulos já conhecidas, tendo sido selecionadas quatro tipos de *rótulos* como *Blacklist*, os endereços mais ricos (*Top500*), lista dos mineradores e lista de endereços de *Exchange* (existem mais do que um endereço por *Exchange*), como se pode observar na *Tabela 8.* Devido às limitações na extração de dados no *BigQuery*, foi obtido um conjunto total de 1046 observações, com 213 provenientes da lista dos *Top500* e 16 oriundos dos 40 principais endereços de *Exchanges*.

Tipo	**Fonte**	**Quantidade**
Blacklist	*Dune*	359
Lista dos 500 endereços mais ricos	*Dune*	213
Lista de endereços de *Miner*	*Bigquery*	458
Lista de endereços de *Exchange*	*Coincarp*	16

Tabela 8: Fontes dos endereços com rótulo recolhidos, consistindo em 359 endereços de Blacklist, 213 da lista dos 500 endereços mais ricos, 458 da lista de endereços de Miner e 16 da Lista de endereços de Exchange

Segunda fase: os dados coletados são submetidos a um processo de tratamento e preparação, visando a criação de variáveis relevantes para análises posteriores. O objetivo deste processo é preparar os dados de forma que seja mais adequados para a análise de grupo.

Com base na pesquisa realizada sobre o tema dos endereços de *Ethereum*, vários artigos, incluindo os citados nas referências [4, 32, 33, 34, 35], abordam as variáveis utilizadas na classificação dos endereços. Assim, para a presente análise dos endereços Ethereum, foi criadas 21 variáveis, a maioria das quais foi obtida diretamente do *BigQuery*, enquanto as restantes foram derivadas dos dados recolhidos, como pode observar na *Tabela 9.* Devido à grande diferença na escala entre as variáveis criadas, optou-se por realizar a estandardização dos mesmos (média 0 e desvio padrão 1).

Input (Recebido)	Output (Enviado)
1. Número total de transações	7. Número total de transações
2. Ether total	8. Ether total
3. Média diária de número de transações	9. Média diária de número de transações
4. Média diária de Ether por transação	10. Média diária de Ether por transação
5. Ether por transação	11. Ether por transação
6. Número total de endereços	12. Número de endereços
Variável de duração	
13. Tempo (Anos) da existência dos endereços Max (data da última transação recebida; data da última transação enviada) – Min (data da primeira transação enviada, data da primeira transação recebida)	

Tabela 9: Variáveis independentes definidas. Estas relacionam-se, principalmente, com o número de transações, montante de Ehter transferida e o tempo de existência dos endereços

Terceira fase: Realizar uma análise detalhada dos dados que foram preparados na fase anterior, utilizando técnicas de *Machine Learning*. Nesta análise, são aplicados aos dados os algoritmos de *clustering* supervisionados, identificando o mais apropriado para os dados selecionados.

5.2 Análise de grupos e classificação das carteiras

Nesta seção é feita a discussão dos resultados obtidos. A implementação das técnicas através dos algoritmos supervisionados de *Machine Learning* e as comparações dos resultados realizadas em *Python*, com foco na biblioteca *scikit-learn*. Esta biblioteca apresenta uma coleção de algoritmos, métricas de avaliação e ferramentas de pré-processamento de dados.

Nesta biblioteca, o modelo fraco, por defeito, é a árvore de decisão para os algoritmos *AdaBoost* e *GradientBoosting*, assim, será este o modelo fraco aplicado no presente trabalho.

5.2.1 Classificação dos endereços de *Ethereum* com rótulos já definidas publicamente

A *Figura 7* apresenta a análise das variáveis independentes para cada tipo de grupos da base de dados recolhida. É possível observar características distintas que os diferenciam uns dos outros.

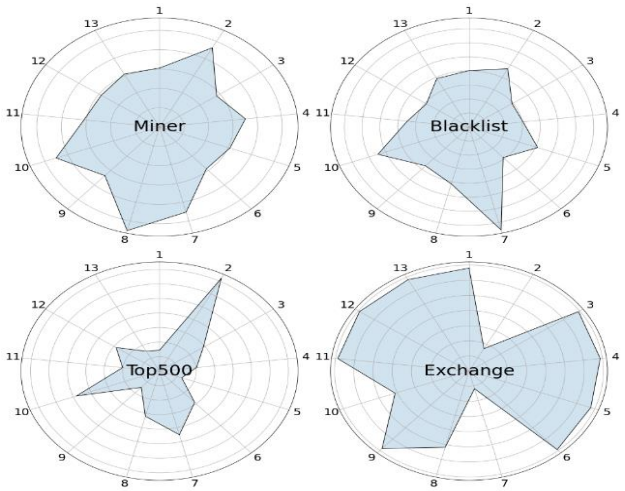
Os endereços de *Miner* apresentam um comportamento caraterizado por um baixo número de transações, mas com uma quantidade relativamente elevada de *Ether* tanto recebido

quanto enviado. Além disso, estes endereços apresentam anos de atividade relativamente elevados. Podendo deduzir que estes estão ativos na mineração de forma consistente.

Os endereços da *Blacklist* demonstram um comportamento de inatividade diária. Apresentam também um grande número de transações, porém com quantidades reduzidas de *Ether* a serem enviadas. Isto pode ser uma possível estratégia para evitar a sua deteção.

Os endereços de *Top500* apresentam um comportamento onde o valor total de *Ether* recebido é elevado, enquanto o valor total enviado é relativamente baixo. Assim, parece que estes preferem armazenar criptomoedas a longo prazo.

Os endereços de *Exchange* apresentam um comportamento de movimentos diários frequentes. Estes têm uma existência relativamente longa, indicando que estão ativos por um longo período. No entanto, nem o número total de *Ethers* recebidos nem transações enviadas nesta amostra apresenta volumes significativamente elevados.



Input (Recebido)	Output (Enviado)
1. Número total de transações	7. Número total de transações
2. Ether total	8. Ether total
3. Média diária de número de transações	9. Média diária de número de transações
4. Média diária de Ether por transação	10. Média diária de Ether por transação
5. Ether por transação	11. Ether por transação
6. Número total de endereços	12. Número de endereços
Variável de duração	
13. Tempo (Anos) da existência dos endereços	

Figura 7: Gráfico Radar das características específicas por tipo de grupos de endereços, derivada das variáveis independentes.

Validação dos modelos

Para a classificação dos 4 grupos de endereços da *Ethereum* com *rótulos* já definidos publicamente entre todos os modelos utilizados, foram selecionados os modelos *Random Forest, AdaBoost e GradientBoosting.*

Nesta fase, o objetivo é analisar o desempenho dos algoritmos de classificação. Assim, foi realizada uma otimização dos parâmetros com base num conjunto de medidas de desempenho. Todos os algoritmos foram avaliados considerando métricas como *accuracy*, *F1_Score*, precisão e sensibilidade. Foi adotada *F1-Score* como a métrica na análise dos dados, isso é, considera β igual a 1. Esta escolha implica que tanto a precisão quanto a sensibilidade têm o mesmo peso, cada um contribuindo com 50% para a métrica.

No entanto, é importante destacar que a métrica *AUC-ROC* não foi considerada devido à sua natureza binária, o que significa que é mais adequada para problemas de classificação binária, enquanto neste secção a análise envolvida é da classificação de múltiplos grupos.

Além disso, o método de avaliação utilizado é a validação cruzada. Esta proporciona uma avaliação mais robusta do desempenho do modelo, uma vez que envolve a realização de vários treinos, neste caso opta-se por 5-*fold.* Resultando numa estimativa precisa do desempenho médio do modelo.

Random Forest

De acordo com os métodos de avaliação e as respetivas métricas para determinar o melhor número de estimadores, destaca-se que o valor ótimo é alcançado com 175 estimadores para todas as métricas. É importante realçar que, em média, as métricas de avaliação mantêm-se consistentemente acima do valor de 0,8, o que indica um desempenho sólido do modelo.

Random Forest	**Validação Cruzada**			
Número de estimador	*Accuracy*	Precisão	Sensibilidade	*F1_Score*
50	0,90833	0,86145	0,81223	0,82192
75	0,90452	0,85896	0,81011	0,81970
100	0,90738	0,86075	0,81172	0,82132
125	0,90833	0,86186	0,81247	0,82227
150	0,90833	0,87050	0,82861	0,83552
175	**0,90929**	**0,87158**	**0,82914**	**0,83627**
200	0,90833	0,87074	0,82861	0,83563

Tabela 10: Hiperparâmetros do algoritmo Random Forest e as métricas de avaliação (Accuracy, Precisão, Sensibilidade e F1_Score)

AdaBoost

De acordo com os métodos de avaliação e nas métricas associadas para determinar os melhores hiperparâmetros, destacam-se duas configurações: um número de estimadores igual a 100 e uma taxa de aprendizagem de 0,1. Estes resultados indicam que o modelo adota um número médio de modelos fracos e uma taxa de aprendizagem mais baixa, o que permite ajustes mais cuidadosos durante o treino. É importante realçar que, em média, as métricas de avaliação mantêm-se consistentemente acima do valor de 0,8, o que indica um desempenho sólido do modelo.

AdaBoost		**Validação Cruzada**			
Número de estimador	*learning_rate*	*Accuracy*	Precisão	Sensibilidade	*F1_Score*
50	**0,1**	0,88350	0,80563	0,79911	0,79595
100	**0,1**	**0,88446**	**0,80748**	**0,79985**	**0,79717**
150	**0,1**	0,88062	0,79665	0,79702	0,78715
50	**1**	0,88061	0,78944	0,79574	0,78359
100	**1**	0,88253	0,79785	0,79830	0,78828
150	**1**	0,88252	0,79767	0,79855	0,78834

Tabela 11: Hiperparâmetros do algoritmo AdaBoost e as métricas de avaliação (Accuracy, Precisão, Sensibilidade e F1_Score)

GradientBoosting

De acordo com os métodos de avaliação e as métricas associadas para a determinar os melhores hiperparâmetros, destacam-se duas configurações: um número de estimadores igual a 150 e uma taxa de aprendizagem de 0,1. Estes resultados indicam que o modelo adota um número grande de modelos fracos, tornando o processo de treino mais demorado, e uma taxa de aprendizagem baixa, o que implica ajustes mais cuidadosos durante o treino. É importante realçar que, em média, as métricas de avaliação mantêm-se consistentemente acima do valor de 0,8, o que indica um desempenho sólido do modelo.

GradientBoosting		**Validação Cruzada**			
Número de estimador	***Learning rate***	*Accuracy*	Precisão	Sensibilidade	*F1_score*
50	**0,1**	0,90833	0,82115	0,81696	0,81059
100	**0,1**	0,90450	0,81764	0,81439	0,80765
150	**0,1**	**0,90928**	0,82566	**0,81726**	**0,81466**
50	**1**	0,89494	0,83093	0,80708	0,80139
100	**1**	0,90162	**0,83582**	0,81261	0,80660
150	**1**	0,90066	0,83517	0,81127	0,80557

Tabela 12: Hiperparâmetros do algoritmo GradientBoosting e as métricas de avaliação (Accuracy, Precisão, Sensibilidade e F1_Score)

Análise comparativa entre os três modelos

Entre os três algoritmos selecionados na validação cruzada, é percetível que todos apresentam os resultados semelhantes de acordo com as métricas apresentadas, como se pode observar na *Figura 8.*

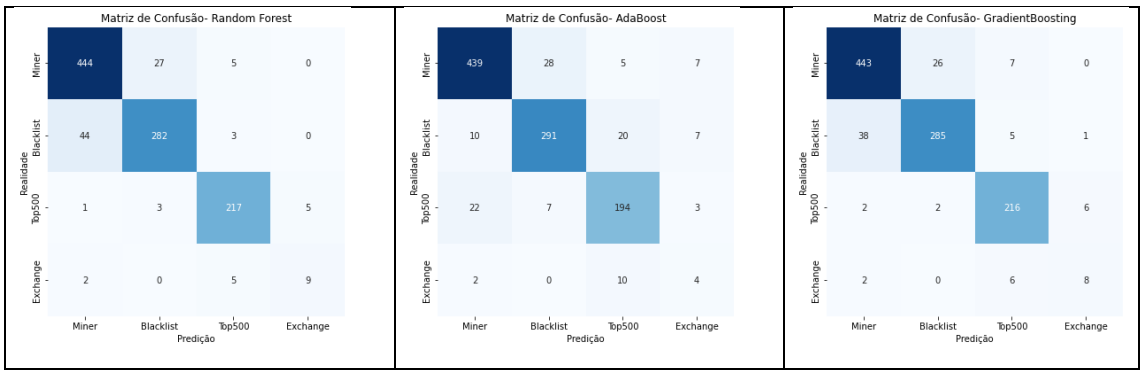


Figura 8:Comparação entre os melhores modelos selecionados

Entre os três algoritmos selecionados, é percetível que, em geral, as variáveis independentes com maior peso se concentram no número de *Ether* recebido e enviado e no anos de existência dos endereços. De salientar que a variável *Ether* por transação recebida ressalta no modelo *AdaBoost* e a variável *Ether* total enviada no modelo *GradientBoosting*, como se pode observar na *Figura 9.*

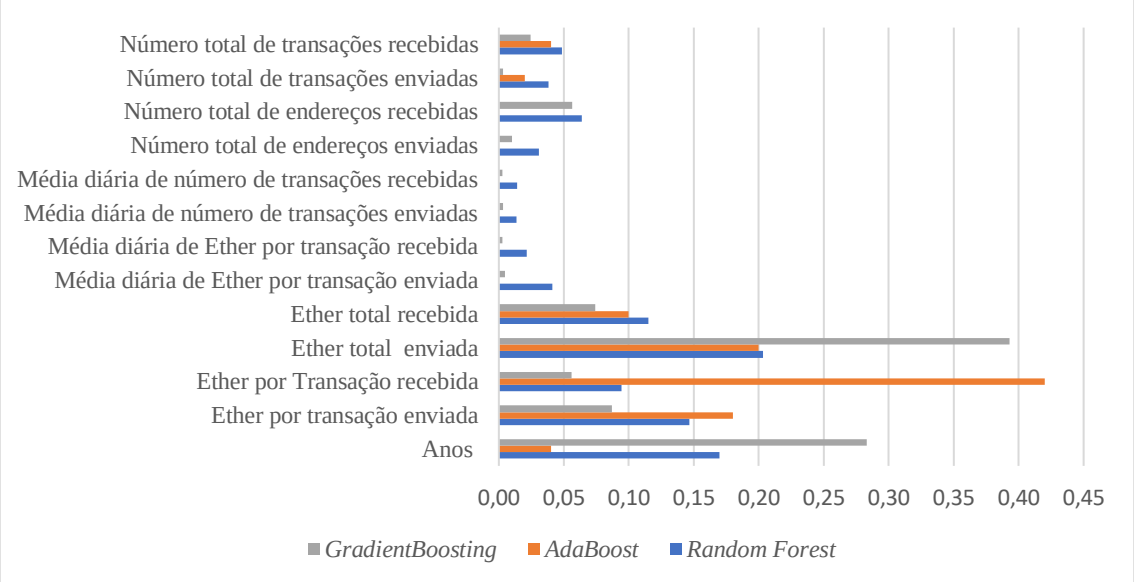


Figura 9: Comparação dos pesos de variáveis independentes nos três modelos

5.2.2 Classificação de *cold wallet* e *hot wallet* de *Ethereum*

No ecossistema das criptomoedas, a classificação de carteiras *Ethereum* em *cold wallet* e *hot wallet* desempenha um papel crucial na análise comportamental e de maturidade de mercado dos cripto ativos. No entanto, esta classificação é uma tarefa desafiadora devido à falta de uma

definição clara e à ausência de uma base de dados pública que atribua estas classificações de forma padronizada.

A distinção entre *cold* e *hot wallets* é altamente subjetiva, pois não existe uma única abordagem que se aplique a todas as situações. Os registos encontrados na *Blockchain* estão relacionados com as transações em si, pelo que não fornece informações detalhadas sobre o tipo de *wallet* utilizado como *hardware* e *online*, tornando a classificação ainda mais desafiadora. Assim, de acordo com os dados disponível na *Blockchain Ethereum*, deduzem-se nas caraterísticas mais adequadas a uma *cold wallet.* Contudo, este método de classificação enfrenta outros desafios, pois existem outros tipos de carteiras que podem ter comportamentos semelhantes, como as carteiras perdidas, foi encontrado um artigo bastante interessante que se aprofunda na análise da distinção entre uma *cold wallet* e uma carteira perdida, para obter detalhes adicionais, pode consultar à referência [24].

Nesta etapa, o objetivo reside em analisar o comportamento das transações para poder identificar *cold wallets*. Isso é fundamental, pois à medida que se analisem as suas características especificas, consegue-se entender o motivo por trás de seu uso predominante: armazenar quantidades substanciais de criptomoedas a médio e longo prazo, com um foco primordial na segurança.

A identificação de uma *cold wallet* baseia-se em diversos indicadores e práticas que diferem significativamente das *hot wallets*, as quais são mais caracterizadas por transações frequentes e acesso instantâneo aos fundos:

- Valores de transação elevados: uma *cold wallet* geralmente mantém um saldo considerável de criptomoedas na sua conta, refletindo-se em transações com valores relativamente altos.
- Número reduzido de transações: uma *cold wallets* tende a apresentar uma taxa de movimentação reduzida, alinhada com o propósito de armazenamento a médio e longo prazo. Os detentores destas carteiras não estão interessados em movimentar constantemente seus ativos. Além disso, o valor acumulado recebido na carteira é tendencialmente maior do que o valor enviado.

Com base nas observações e análises realizadas, foi desenvolvida uma abordagem para permitir a identificação e caracterização de *cold wallets* na rede *Ethereum*. É importante destacar que a classificação foi realizada com base numa metodologia própria, uma vez que não havia acesso a uma base de dados pré-classificada de *cold* e *hot wallets.*

Primeiramente, os endereços de *Blacklist* foram excluídos, pois compartilham características semelhantes com as *cold wallets*, como transações infrequentes, mas podem não estar mais ativos. Assim, a análise concentra-se em endereços dos restantes 3 grupos (*Miner*, *Top500* e *Exchange*). A amostra total para esta análise é composta por 687 observações.

Em seguida, implementou-se uma restrição associada à seleção de endereços com mais de três meses de existência. Esta restrição temporal permitiu a avaliação do comportamento destas carteiras ao longo do tempo, identificando padrões consistentes de armazenamento a médio e longo prazo.

Uma característica distintiva das *cold wallets* é a baixa quantia de número de transações. Portanto, foi introduzido um critério adicional na definição de *cold wallet*, exigindo que a média mensal de transações enviadas ou recebidas fosse inferior a 30, destacando a natureza de armazenamento dessas carteiras, que não se envolvem em atividades de grande volume, ao contrário das *hot wallet*s.

Além disso, considerou-se o valor médio por transação como um indicador relevante. A análise foi dividida em dois períodos para refletir as flutuações de valor da *Ether*, a criptomoeda nativa da rede *Ethereum*. O primeiro período, que compreende o início de 2016 até novembro de 2017 e de agosto de 2018 até outubro de 2020, foram selecionados endereços com um valor médio por transação superior a 50 *Ethers*. No segundo período, que abrange dezembro de 2017 até julho de 2018 e de novembro de 2020 até julho de 2022, quando o valor da *Ether* teve um aumento significativo, optou-se por endereços com um valor médio por transação superior a 20 *Ethers*. Em média, isso corresponde a um critério de valor médio por transação superior a 20 000 dólares. Esta abordagem levou em consideração a variação do valor da *Ether* ao longo do tempo e permitiu identificar com precisão as *cold wallets* na rede *Ethereum*.

Outro critério importante foi o número total de endereços enviados ou recebidos por uma carteira, isto é, a contagem de quantos endereços de criptomoeda de uma determinada carteira tenha utilizado para enviar ou receber cripto ativos. Cada endereço de carteira é uma sequência única de caracteres alfanuméricos que serve como identificador para aquela carteira específica. Dado o comportamento específico de uma *cold wallet* e com os dados fornecidos, foram selecionados endereços com um número total de endereços inferior a 30, ao sujeitar esta restrição pode ser mais fácil identificar padrões únicos de comportamento, como transferências grandes e infrequentes de ativos.

Após a conclusão destas etapas de filtragem, um total de 213 endereços (2 de *Exchange*, 105 de *Miner* e 106 de *Top 500*) foram identificados e categorizadas como *cold wallets*. Esta abordagem possibilita a construção de um perfil mais aproximado das *cold wallet*s na rede *Ethereum*, oferecendo *insights* sobre como os utilizadores geram e preservam os seus cripto ativos a médio e longo prazo. A *Figura 10* esquematiza o processo da seleção dos *cold wallets*.

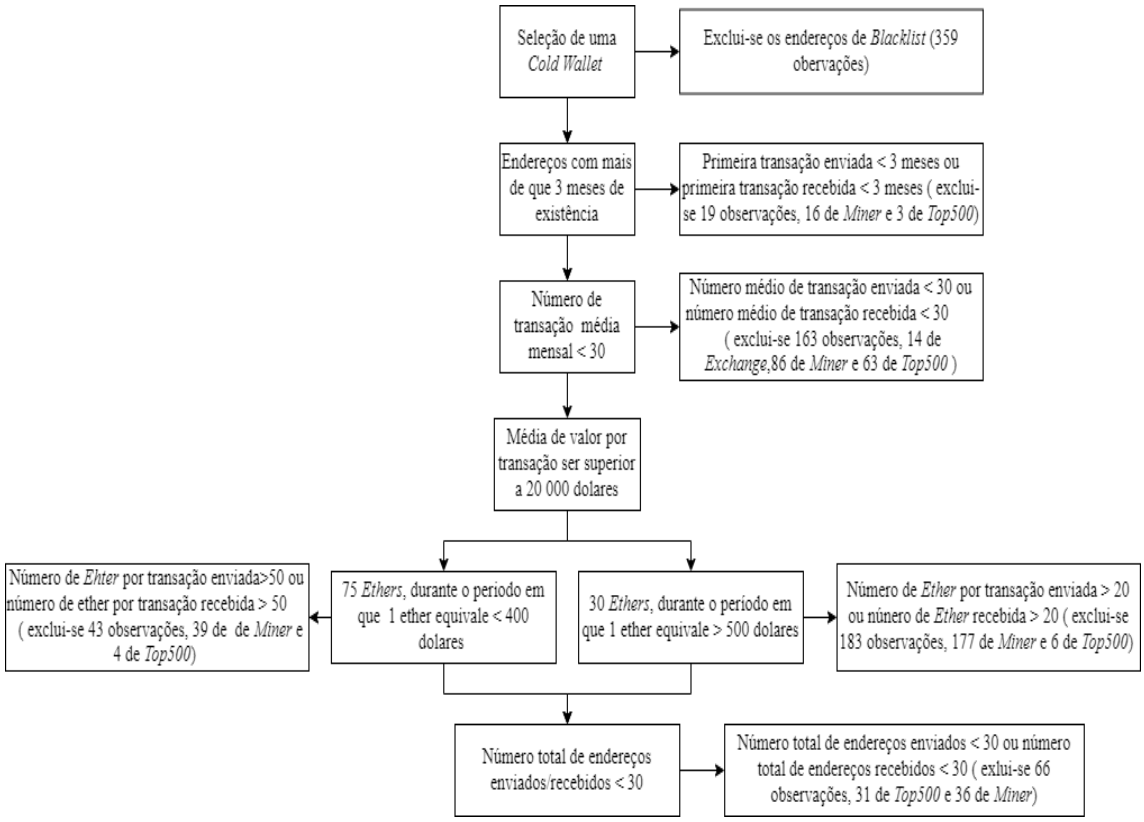


Figura 10: Processo de definição de uma Cold wallet

Das *cold wallet* selecionadas, foi realizada uma análise da evolução de montante total da *Ehter* em relação à evolução do preço de *Ether*. O objetivo era testar se estas carteiras apresentam comportamentos de investimento de longo prazo, seguindo os ciclos das criptomoedas. Estes ciclos, por sua vez, dependem do processo de *Halving* de *Bitcoin*, como ilustrado no Capítulo anterior.

A *Figura 11* ilustra o comportamento do montante total da *Ether* de 213 *cold wallets* ao longo do tempo, em relação à evolução do preço da *Ether* deste 2016 até julho de 2023.

No início do período, os movimentos da *Ether* são bastante reduzidos, tratando-se de uma fase inicial das criptomoedas. A partir de meados de 2019, observam-se mais movimentos de transferência de *Ether* de *cold wallets* para outros destinos. Esta transferência normalmente está relacionada com a venda da criptomoeda em *Exchanges* centralizadas, como a *Binance*, deduzindo que os investidores começaram a considerar a possibilidade de concretizar os seus lucros.

Um ponto a destacar é o período compreendido entre abril de 2021 e setembro do mesmo ano. Durante este intervalo, observa-se um grande número de carteiras envolvidas em transações de compra e venda da *Ether* de forma simultânea. Este comportamento pode ser um reflexo de um mercado altamente volátil, onde os investidores aproveitam as flutuações de preços para obter

ganhos substanciais. A discrepância na evolução futura do preço da *Ether*, durante este período, pode ter contribuído para a incerteza e a volatilidade do mercado.

A partir de meados de 2022, parece que o movimento de compra se tornou dominante. Isto pode indicar que os investidores apresentam uma perspetiva otimista em relação ao futuro da *Ether* e estão dispostos a manter as suas posições ou a adquirir mais moedas, especialmente considerando o processo de *Halving*, previsto para 2024.

Em suma, a *Figura 11* retrata um mercado de *Ether*, com diferentes fases de comportamentos dos detentores de criptomoedas. Salienta-se estes comportamentos podem ser influenciados por uma variedade de fatores, incluindo notícias, eventos económicos e desenvolvimentos tecnológicos, que podem afetar significativamente a dinâmica do mercado.

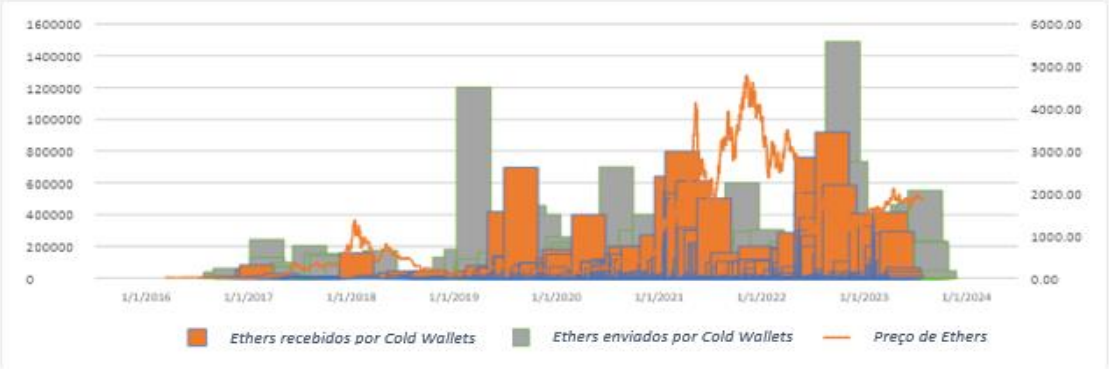


Figura 11: Preço histórico de Ether e comportamento de Ether total dos cold wallets definidos

Validação dos modelos

Para a classificação binária de endereços da *Ethereum* com rótulos definida através das características de uma *cold wallet*, foram selecionados os mesmos modelos anteriormente usados para classificação de 4 grupos de endereço. Adicionalmente foi também escolhido o modelo de Regressão Logística, devido à natureza de ser adequar numa análise binária da classificação: *cold wallet* ou *hot wallet.*

Nesta fase, o objetivo é analisar o desempenho dos algoritmos de classificação. Assim, foi realizada uma otimização dos parâmetros com base num conjunto de medidas de desempenho. Todos os algoritmos foram avaliados considerando métricas como *accuracy* e *F1_Score*, precisão, sensibilidade *e AUC*. Além disso, também foram testados os dois métodos de avaliação: validação cruzada e validação por divisão em amostras.

Random Forest

De acordo com os métodos de avaliação e as respetivas métricas para determinar o melhor número de estimadores, o valor ótimo é alcançado com 175 estimadores, de acordo com a maioria das métricas.

Random Forest	**Validação Cruzada**				
Número de estimador	*Accuracy*	Precisão	Sensibilidade	*F1_Score*	*AUC-ROC*
50	0,91123	0,90848	0,79358	0,84641	0,87885
75	0,91558	0,91101	0,80764	0,85562	0,88588
100	0,91413	0,90641	0,80786	0,85336	0,88493
125	0,91557	0,90627	0,81240	0,85601	0,88720
150	0,91703	0,90688	0,81705	0,85876	0,88952
175	**0,91849**	0,91119	**0,81717**	**0,86092**	**0,89064**
200	0,91558	**0,91403**	0,80299	0,85413	0,88461

Tabela 13: Hiperparâmetros do algoritmo Random Forest e as métricas de avaliação (Accuracy, Precisão, Sensibilidade, F1_Score e AUC-ROC)

AdaBoost

De acordo com os métodos de avaliação e as métricas associadas para determinar os melhores hiperparâmetros, destacam-se duas configurações: um número de estimadores igual a 100 e uma taxa de aprendizagem de 0,1. É importante realçar que, em média, as métricas de avaliação mantêm-se consistentemente acima do valor de 0,9, o que indica um desempenho sólido do modelo.

AdaBoost		**Validação Cruzada**				
Número de estimador	***Learning rate***	Accuracy	Precisão	Sensibilidade	F1_score	AUC-ROC
50	**0,1**	**0,93597**	**0,91943**	0,86866	0,88799	0,91692
100	**0,1**	0,93161	0,90017	**0,87807**	**0,89242**	**0,91747**
150	**0,1**	0,93307	0,90746	0,87331	0,88920	0,91664
50	**1**	0,93161	0,91090	0,86390	0,88656	0,91298
100	**1**	0,92579	0,90499	0,84983	0,87646	0,90489
150	**1**	0,92580	0,91017	0,84507	0,87618	0,90358

Tabela 14: Hiperparâmetros do algoritmo AdaBoost e as métricas de avaliação (Accuracy, Precisão, Sensibilidade, F1_Score e AUC-ROC)

GradientBoosting

De acordo com os métodos de avaliação e nas métricas associadas para determinar os melhores hiperparâmetros, destacam-se duas configurações: um número de estimadores igual a 100 e uma taxa de aprendizagem de 0,1. Também aqui, em média, as métricas de avaliação mantêm-se consistentemente acima do valor de 0,9, indicando um desempenho sólido do modelo.

GradientBoosting		Validação Cruzada				
Número de estimador	***Learning rate***	*Accuracy*	Precisão	Sensibilidade	*F1_Score*	*AUC-ROC*
50	0,1	0,92287	0,88179	0,86844	0,87417	0,90787
100	0,1	**0,92431**	**0,88523**	**0,86844**	**0,87649**	**0,90892**
150	0,1	0,92140	0,88394	0,85891	0,87098	0,90417
50	1	0,89522	0,82177	0,84994	0,83447	0,88279
100	1	0,90830	0,85859	0,84507	0,85151	0,89091
150	1	0,90684	0,85134	0,84961	0,85005	0,89107

Tabela 15: Hiperparâmetros do algoritmo GradientBoosting e as métricas de avaliação (Accuracy, Precisão, Sensibilidade, F1_Score e AUC-ROC)

Regressão Logística

De acordo com os métodos de avaliação e as métricas associadas para a determinar os melhores parâmetros, destaca-se que uma Regressão Logística regularizada do tipo "*l*1", com um valor de hiperparâmetro *C* igual a 10. A regularização "*l*1" penaliza coeficientes grandes, podendo levar a zero.

No entanto, é importante notar que, em média, as métricas de avaliação permanecem abaixo de 0,6, sugerindo que o modelo pode não ser o mais indicado a classificação. Assim, não foi realizada a comparação dos pesos das variáveis independentes. avaliação permanecem abaixo de 0,6, sugerindo que o modelo pode não ser o mais indicado.

Regressão Logística		Validação Cruzada				
Penalty	**C**	*Accuracy*	Precisão	Sensibilidade	*F1_Score*	*AUC-ROC*
L1	**0,1**	0,72929	0,60597	0,37154	0,45663	0,63095
L1	**1**	0,73218	0,60959	0,39945	0,47899	0,64069
L1	**10**	0,73655	0,61549	**0,41827**	0,49451	**0,64906**
L2	**0,1**	**0,73656**	**0,61820**	0,41795	**0,49488**	0,63836
L2	**1**	0,73218	0,60732	0,39945	0,47899	0,64069
L2	**10**	0,73073	0,60431	0,40886	0,47981	0,64646

Tabela 16: Hiperparâmetros do algoritmo Regressão Logística e as métricas de avaliação (Accuracy, Precisão, Sensibilidade, F1_Score e AUC-ROC)

Análise comparativa entre os quatros modelos

A *Tabela 17* apresenta as matrizes de confusão obtidas nos modelos *Random Forest*, *AdaBoost, GradientBoosting* e Regressão Logística.

Das 213 *cold wallets* analisadas, os modelos *AdaBoost* e *GradientBoosting* apresentaram as melhores previsões, com 185 *cold wallets* corretamente classificadas e 28 erroneamente classificadas como *hot wallets*, resultando em uma taxa de sucesso de 87%.

No que diz respeito à previsão de *hot wallets* (casos negativos), os modelos *AdaBoost* e *GradientBoosting* obtiveram resultados muito positivos, com 458 e 450 *hot wallets* corretamente classificadas, resultando em taxas de acerto de 97% e 95%, respectivamente.

O *Random Forest* apresenta uma taxa de sucesso de 82% para *cold wallets* e de 96% para *hot wallets.*

Por outro lado, o modelo de Regressão Logística apresentou o desempenho menos satisfatório, com apenas 89 *cold wallets* corretamente classificadas e 124 erroneamente classificadas como *hot wallets*, resultando em uma taxa de acerto de 42%. Em relação à previsão de *hot wallets*, o modelo de Regressão Logística obteve 417 classificações corretas, resultando em uma taxa de acerto de 88%.

Estes resultados evidenciam a eficácia dos modelos *AdaBoost, GradientBoosting e Random Forest* na classificação de *cold* e *hot wallets.*

Random Forest		**Previsão**	
		Sim (*Cold wallet*)	Não (*Hot wallet*)
Real	Sim	174	39
	Não	17	457

AdaBoost		**Previsão**	
		Sim (*Cold wallet*)	Não (*Hot wallet*)
Real	Sim	185	28
	Não	16	458

GradientBoosting		**Previsão**	
		Sim (*Cold wallet*)	Não (*Hot wallet*)
Real	Sim	185	28
	Não	24	450

Regressão Logística		**Previsão**	
		Sim (*Cold wallet*)	Não (*Hot wallet*)
Real	Sim	89	124
	Não	57	417

Tabela 17: Matriz de confusão dos quatro modelos

Relativamente à análise binária por grupo, como ilustrada na *Tabela 18*, o grupo *Miner* apresenta uma taxa de sucesso de 36%, 76%, 74% e 68% nos métodos de Regressão Logística, *AdaBoost*, *GradientBoosting* e *Random Forest*, respetivamente. O grupo *Top500* alcança uma taxa de sucesso de 74%, 96%, 98% e 93% nos mesmos métodos. Por fim, O grupo *Exchange* apresenta uma taxa de sucesso de 50% na Regressão Logística e 100% nos restantes métodos.

Com base nestes resultados, é evidente que, entre os três grupos, o grupo *Top500* apresenta um desempenho superior em todos os métodos aplicados. Por outro lado, importante de notar que o grupo *Exchange* possui apenas 2 cold wallets num conjunto de 16. Devido à amostra reduzida para a análise, pode resultar numa avaliação menos precisa.

Top500		Regressão Logística		*AdaBoost*		*GradientBoosting*		*Random Forest*	
Sim = Cold wallet		Previsão		Previsão		Previsão		Previsão	
Não = Hot wallet		Sim	Não	Sim	Não	Sim	Não	Sim	Não
Real	Sim	78	28	102	4	104	2	99	7
	Não	58	49	2	105	7	100	7	100

Exchange		Regressão Logística		*AdaBoost*		*GradientBoosting*		*Random Forest*	
Sim = Cold wallet		Previsão		Previsão		Previsão		Previsão	
Não = Hot wallet		Sim	Não	Sim	Não	Sim	Não	Sim	Não
Real	Sim	1	1	2	0	2	0	2	0
	Não	2	12	0	14	1	13	2	12

Miner		Regressão Logística		*AdaBoost*		*GradientBoosting*		*Random Forest*	
Sim = Cold wallet		Previsão		Previsão		Previsão		Previsão	
Não = Hot wallet		Sim	Não	Sim	Não	Sim	Não	Sim	Não
Real	Sim	38	67	80	25	78	27	72	33
	Não	35	318	13	340	22	331	13	340

Tabela 18: Matriz de confusão por grupo dos quatro modelos

A *Tabela 19* apresenta os melhores valores de métricas de avaliação para cada um dos modelos na análise binária global. A *AUC-ROC* é uma métrica que oferece uma visão abrangente do desempenho do modelo. Neste sentido, o modelo que obteve o melhor valor de *AUC-ROC* foi o *AdaBoost*, com uma pontuação de 92%. Os modelos *GradientBoosting* e *Random Forest* também apresentaram valores relativamente elevados para esta métrica. No entanto, quando comparados com o modelo *AdaBoost*, ambos os modelos tiveram um desempenho ligeiramente inferior em todas as métricas. Portanto, com base no valor de *AUC-ROC* e nas métricas gerais, o modelo escolhido como o mais adequado para a classificação de *hot* e *cold wallets* foi o *AdaBoost*.

Já o modelo Regressão Logística obteve o desempenho menos positivo, com um valor de 66% na métrica *AUC-ROC*.

	Hiperparâmetros		**Validação Cruzada**				
Random Forest	**Número de estimador**		*Accuracy*	Precisão	Sensibilidade	*F1_Score*	*AUC-ROC*
	175		0,91849	0,91119	0,81717	0,86092	0,89064
AdaBoost	**Número de estimador**	**Learning rate**					
	50	**0,1**	0,93597	0,91943	0,86866	0,89242	0,91747
Gradient Boosting	**Número de estimador**	**Learning rate**					
	100	**0,1**	0,92431	0,88523	0,86844	0,87649	0,90892
Regressão Logística	**Penalty**	**C**					
	L1	**10**	0,73655	0,61549	0,41827	0,49451	0,64906

Tabela 19: Avaliação dos modelos através das métricas

Na amostra o número de *hot wallets* é aproximadamente o dobro do número de *cold wallets*, o que tem um impacto significativo nas métricas de avaliação dos modelos. Os modelos de combinação, como *AdaBoost*, *GradientBoosting* e *Random Forest*, não são tão influenciados por este desequilíbrio entre os grupos, pois foram capazes de ajustar a sua previsão com base nas características das classes.

No entanto, é evidente que a Regressão Logística é significativamente afetada por este desequilíbrio, como indicado pelas métricas deste modelo. Portanto, torna-se crucial realizar uma análise mais detalhada das métricas da Regressão Logística para compreender como é que este modelo lida com o desequilíbrio das classes.

A métrica de *accuracy*, que mede a proporção de previsões corretas em relação ao total de previsões, pode conduzir a análises enganadoras no conjunto de dados desequilibrados. Neste caso, a *accuracy* parece relativamente razoável, porque o modelo tende a prever razoavelmente bem a classe maioritária (neste caso, *hot wallet*). No entanto, este não fornece uma representação completa do desempenho do modelo, porque a taxa de sucesso na classificação de *cold wallet* é apenas aproximadamente 42%.

A sensibilidade mede a proporção de verdadeiros positivos em relação a todos os casos positivos reais. Neste caso, a sensibilidade é influenciada negativamente devido ao desequilíbrio, o que significa que o modelo tem dificuldade em identificar adequadamente *cold wallets*, que é a classe minoritária. Isto pode ser problemático, pois o objetivo principal é a deteção de *cold wallets*. A métrica *F1-Score* leva em consideração tanto os verdadeiros positivos quanto os falsos positivos, oferecendo uma visão mais completa do desempenho, sendo também apenas de 50%.

De acordo com a *Figura 12*, parece que a alteração do valor de corte não terá um impacto significativo na eficácia das previsões. Isto ocorre porque, independentemente do valor escolhido para o corte, a sobreposição entre as distribuições das classes ainda é considerável, o que dificulta a definição de um valor de corte que permita uma separação clara entre as duas classes.

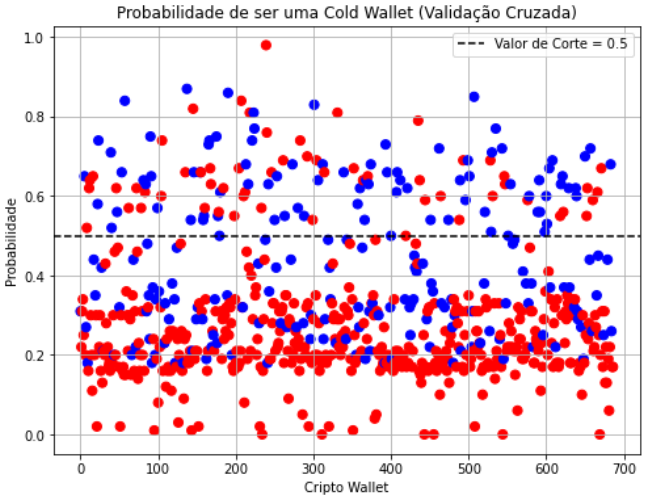


Figura 12: O gráfico de dispersão da Regressão Logística com o valor de corte de 0.5

Entre os algoritmos selecionados (excluindo o modelo de Regressão Logística devido aos seus baixos valores de métricas), é percetível que as variáveis independentes com maior peso se concentram no número total da *Ether* enviado, na média diária do número de transações enviadas, na *Ether* por transação recebida e enviada, bem como nos anos de existência dos endereços. Estas variáveis demonstram ser as mais influentes na classificação das *cold wallets*, de acordo com os algoritmos considerados, como se pode observar na *Figura 13.*

Ao analisar os pesos por grupo, verifica-se que no grupo *Miner*, a variável mais relevante

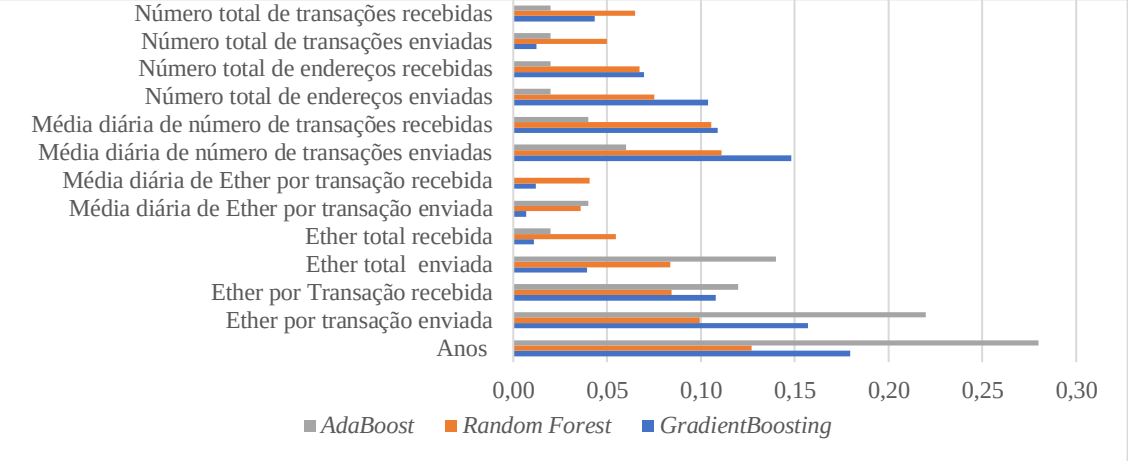
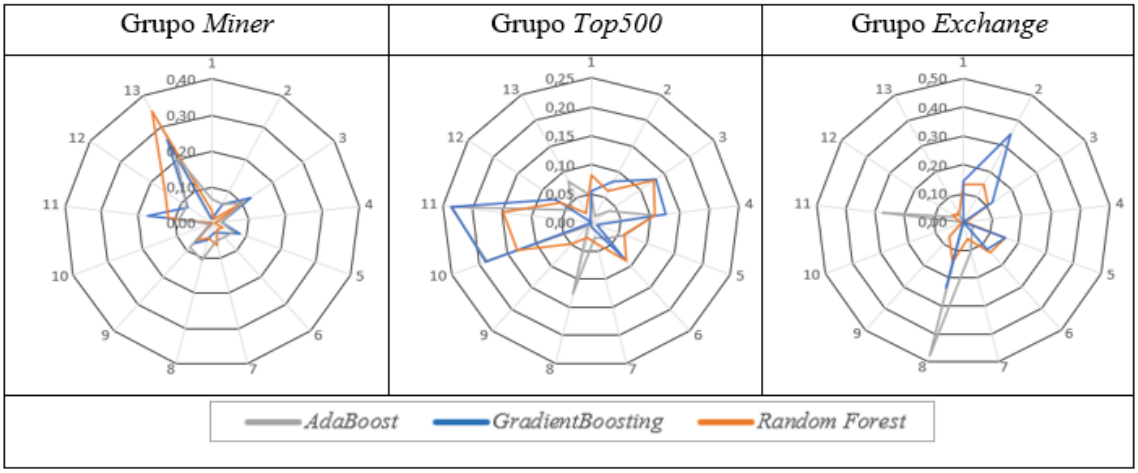


Figura 13: Comparação dos pesos de variáveis independentes nos três modelos

é os anos, deduzindo que este seja o grupo associado a investimentos de longo de prazo. No caso do grupo *Top500*, as variáveis mais relevantes são a *Ether* por transação enviada, seguida das variáveis relacionadas com médias diárias. No entanto, estas médias diárias não são de grande relevância em termos da análise binária global, de acordo com a *Figura 13.* Assim, ignorando estas médias, a variável *Ether* total enviada apresenta um peso considerável. Assim, pode sugerir-se que este grupo realiza transações com grande quantidade de *Ether e* as carteiras apresentam uma média longa duração. Quanto ao grupo *Exchange*, as variáveis *Ether* total e por transação enviada destacam-se no modelo *AdaBoost e Ether* total recebido no modelo *GradientBoosting.* Em ambos os casos, é percetível que o valor envolvido nas transações de *Ether* é de grande importância.

Na *Figura 13* observam-se os pesos na classificação binária em geral, onde se destacam as variáveis mais relevantes. A *Figura 14* permite-nos perceber que os pesos variam de grupo para grupo, entre as variáveis de destaque, realçando as caraterísticas específicas para cada um. De acordo com as matrizes de confusão por grupo anteriormente mencionadas, é notável que o grupo *Top500* apresenta melhores resultados. Assim, as variáveis que se destaquem neste grupo também revelam maior importância, ou seja, as transações envolvem uma grande quantidade de *Ether a enviar* e as carteiras são de média longa duração.



1. Número total de transações recebidas	7. Número total de transações enviadas
2. Ether total recebida	8. Ether total enviada
3. Média diária de número de transações recebidas	9. Média diária de número de transações enviadas
4. Média diária de Ether por transação recebida	10. Média diária de Ether por transação enviada
5. Ether por transação recebida	11. Ether por transação enviada
6. Número total de endereços recebidos	12. Número de endereços enviados
13. Anos da existência dos endereços	

Figura 14: Comparação dos pesos de variáveis independentes por grupo nos três modelos

5.2.3 Comparação dos pesos das variáveis para classificação de grupos e binária

A *Figura 15* apresenta uma análise de pesos das variáveis para a classificação em quatro grupos e para classificação binária, identificando se uma carteira é ou não uma *cold wallet*. Na análise de grupos, a variável mais destacada em todos os modelos é o número de anos de existência dos endereços. No entanto, na análise binária, os destaques variam, com a quantidade da *Ether* por transação recebida sendo a variável mais relevante no modelo *Random Forest* e a quantidade total da *Ether* enviada sendo a variável mais importante no modelo *AdaBoost.*

Observa-se que, em comparação com a análise binária, as variáveis apresentam pesos menores, mas há um maior número de variáveis a participar na análise de grupos. Portanto, a inclusão de um maior número de variáveis selecionadas fornece características que permitem discriminar os quatro grupos.

Na análise binária, existe uma grande variação nos pesos das variáveis, e o peso de algumas variáveis aproximam-se de valor zero, mostrando pouca influência, como as médias diárias. Isto pode ser explicado pelas características específicas de uma *cold wallet*, que geralmente tem objetivos de médio e longo prazo, tornando as variáveis relacionadas a movimentações diárias menos significativas

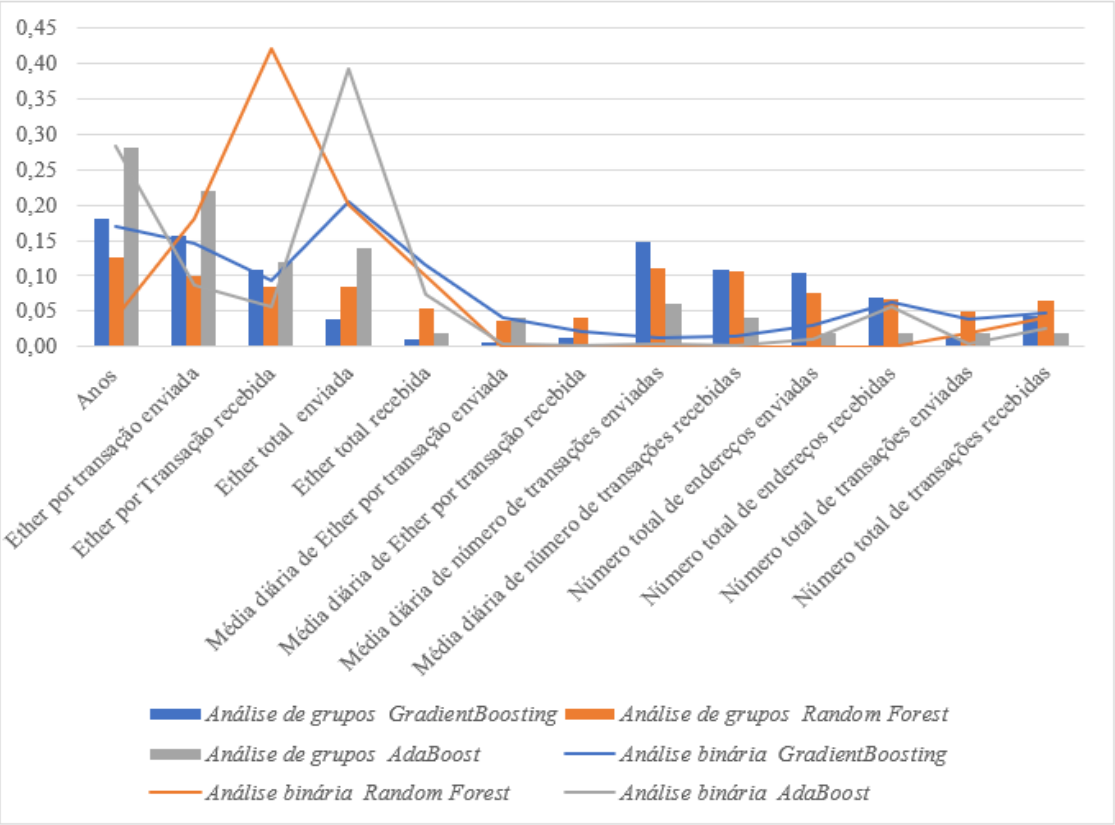


Figura 15: Comparação dos pesos de variáveis independentes na análise binária e de grupos

Conclusões

A presente Tese concentra-se na classificação de endereços da *Blockchain Ethereum* utilizando métodos de *Machine Learning* supervisionados. Primeiro foi feita uma análise a endereços com rótulos já estabelecidas publicamente, com os dados organizados em quatro grupos: *Miner, Exchange, Top500* e *Blacklist.* Neste contexto, foram utilizados três métodos de agrupamento: *Random Forest, AdaBoost* e *GradientBoosting*, obtendo todos resultados bons resultados.

Seguidamente foi feita uma análise em três grupos, *Miner, Exchange* e *Top500*, excluindo os endereços da *Blacklist*, devido às características semelhantes com *cold wallets*. Nesta fase, foi realizada uma análise para classificar endereços em *cold* e *hot wallets*, com rótulos definidas com base nas características de uma *cold wallet.* Foram utilizados os mesmos métodos da fase anterior e a Regressão Logística. Após a comparação das métricas dos modelos, embora *os modelos GradientBoosting e Random Forest* também tenham apresentado resultados bastantes favoráveis, destaca-se o modelo *AdaBoost* com uma taxa de sucesso de 87%. Ao analisar separadamente por grupo, é evidente que o grupo *Top500* obteve melhores resultados com uma taxa de sucesso de 96%. Embora o *GradientBoosting* tenha obtido uma taxa de sucesso de 98% neste grupo particular, o modelo *AdaBoost* é a escolha preferencial na análise global.

O modelo de Regressão logística alcançou um desempenho menos satisfatório comparativamente com os outros métodos na classificação binária, pois a amostra é desequilibrada, tendo significativamente mais *hot wallets* do que *cold*. Esta questão não foi superada com a redefinição do valor de corte, pois a sobreposição entre as distribuições das classes é considerável grande.

A comparação de pesos das variáveis independentes de cada modelo fornece uma visão clara de quais as variáveis que influenciam mais na classificação. Esta comparação permite-nos distinguir aspetos fundamentais relativamente às duas análises, de grupos e binária.

Por outro lado, relativamente à análise binária, ao analisar os grupos (*Miner*, *Top500* e *Exchange*) separadamente, pode observar-se que o grupo *Top500* apresenta as melhores classificações. Assim, as variáveis com maior peso neste grupo demonstram uma mais relevância, indicando com grande probabilidade que as *cold wallets* são as carteiras de média e longa duração, e estão envolvidas em transações que apresentam uma grande quantidade de *Ether*.

Por fim, é importante mencionar que este estudo se concentra especificamente na criptomoeda *Ether*, devido à sua importância. Trata-se da segunda maior criptomoeda, tendo o seu comportamento grande correlação com a *Bitcoin*. Esta correlação torna mais fácil a análise dos comportamentos ao longo do tempo, especialmente em relação a eventos significativos como o processo de *Halving*. Compreender este facto torna mais acessível a análise binária de classificação de uma carteira em *cold* ou *hot*, pois as *cold wallets* são tipicamente carteiras de

investimento de longo prazo e seguem os ciclos das criptomoedas, que são fortemente influenciados pelo processo de *Halving.*

Trabalhos futuros podem considerar a realização de uma análise mais generalizada, com a possibilidade de analisar outras criptomoedas de outras *Blockchains,* bem com outros *tokens* na rede *Ethereum.*

Referência bibliográfica

[1] *Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. Published by O'Reilly Media, Inc., Editor: Mike Loukides and Allyson MacDonald. (pp.60-65; pp.55-88; pp. 171-192; pp.220-299)*

[2] *Baran P. (1964). On Distributed Communications: 1. Introduction to Distributed Communications Networks. (pp. 3-20)*

[3] *Bartoletti, M., & Pompianu, L. (2017). An empirical analysis of smart contracts: platforms, applications, and design patterns. In Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta*

[4] *Bhargavi, M. S., Katti, S. M., Shilpa, M., Kulkarni, V. P., & Prasad, S. (2020). Transactional data analytics for inferring behavioural traits in ethereum blockchain network. In 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing. (pp. 485-490)*

[5] *Bilotta, N., & Botti, F. (2021). The (near) future of central bank digital currencies: risks and opportunities for the global economy and society. Peter Lang International Academic Publishers. (pp 31-37)*

[6] *Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial intelligence, 97(1-2). (pp.245-271)*

[7] *Breiman, L. (2001). Random forests. Machine learning, 45. (pp. 5-32)*

[8] *Burkov, A. (2019). The hundred-page machine learning book (Vol. 1, p. 32). Quebec City, QC, Canada: Andriy Burkov. (pp.13-18)*

[9] *Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. white paper*

[10] *Chaum, D. (1983). Blind signatures for untraceable payments. In Advances in Cryptology: Proceedings of Crypto 82 (pp. 199-203). Boston, MA: Springer US*

[11] *Chaum, D. L. (1979). Computer Systems established, maintained and trusted by mutually suspicious groups. Electronics Research Laboratory, University of California*

[12] *Chaum, D.L. (1985). Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM. (pp.1030-1044)*

[13] *Chen, Y., & Zhao, Q. (2021). Mineral exploration targeting by combination of recursive indicator elimination with the ℓ2-regularization logistic regression based on geochemical data. Ore Geology Reviews, 135, 104213*

[14] *Collomb, A., & Sok, K. (2016). Blockchain/distributed ledger technology (DLT): What impact on the financial sector?. Digiworld Economic Journal. (103)*

[15] *Dai, W. (1998). b-money, 1998. URL http://www. weidai. com/bmoney. Txt*

[16] *Dai, W., Deng, J., Wang, Q., Cui, C., Zou, D., & Jin, H. (2018). SBLWT: A secure Blockchain lightweight wallet based on trustzone*

[17] *Fitzpatrick, T., & Mues, C. (2016). An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market. European Journal of Operational Research, 249(2) (pp.427-439)*

[18] *Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics. (pp.1189-1232)*

[19] *Hafid, A., Hafid, A. S., & Samih, M. (2020). Scaling Blockchains: A comprehensive survey*

[20] *Halpin, H. (2020). Deconstructing the decentralization trilemma. arXiv preprint arXiv:2008.08014*

[21] *Jobson, J. D. (2012). Applied multivariate data analysis: volume II: Categorical and Multivariate Methods. Editor: Fienberg Stephen et al. Springer Science & Business Media. (pp.130-190)*

[22] *Jokić, S., Cvetković, A. S., Adamović, S., Ristić, N., & Spalević, P. (2019). Comparative analysis of cryptocurrency wallets vs traditional wallets. ekonomika, 65(3). (pp.65-75)*

[23] *Karantias, K. (2020). Sok: A taxonomy of cryptocurrency wallets. Cryptology ePrint Archive*

[24] *Khadzhi, A. S.et al. (2020). A Method for Analyzing the Activity of Cold Wallets and Identifying Abandoned Cryptocurrency Wallets. In 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). (pp. 1974-1977)*

[25] *Khan, A. G., Zahid, A. H., Hussain, M., & Riaz, U. (2019, November). Security of cryptocurrency using hardware wallet and qr code. In 2019 International Conference on Innovative Computing. (pp. 1-10)*

[26] *Kiffer, L., Levin, D., & Mislove, A. (2017). Stick a fork in it: Analyzing the Ethereum network partition. In Proceedings of the 16th ACM Workshop on Hot Topics in Networks. (pp. 94-100)*

[27] *Košťál, K., Krupa, T., Gembec, M., Vereš, I., Ries, M., & Kotuliak, I. (2018). On transition between PoW and PoS. In 2018 International Symposium ELMAR. (pp. 207-210)*

[28] *Metz, Charles E. "Basic principles of ROC analysis." Seminars in nuclear medicine. Vol. 8. No. 4. WB Saunders, 1978. (pp.283-298)*

[29] *Nair, V., & Song, D. (2023). Decentralizing Custodial Wallets with MFKDF*

[30] *Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized business review*

[31] *Ng, A. Y. (2004). Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In Proceedings of the twenty-first international conference on Machine learning. (p.78)*

[32] *Payette, J., Schwager, S., & Murphy, J. (2017). Characterizing the ethereum address space*

[33] *Poursafaei, F., Hamad, G. B., & Zilic, Z. (2020, September). Detecting malicious Ethereum entities via application of machine learning classification. In 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services. (pp. 120-127) IEEE.*

[34] *Pragasam, T. T. N., Thomas, J. V. J., Vensuslaus, M. A., & Radhakrishnan, S. (2023). CEAT: Categorising Ethereum Addresses' Transaction Behaviour with Ensemble Machine Learning Algorithms. Computation, 11(8). (p.156)*

[35] *Saxena, R., Arora, D., & Nagar, V. (2023). Classifying Transactional Addresses using Supervised Learning Approaches over Ethereum Blockchain. Procedia Computer Science, 218. (pag.2018-2025)*

[36] *Schapire R. E. and Freund Y. (2012) Boosting:Foundations and Algorithms.Editor: Dietterich T. (pp. 53-62)*

[37] *Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press. (pp.19-27; pp.130-140; pp.250-256)*

[38] *Simmons, G. J. (1979). Symmetric and asymmetric encryption. ACM Computing Surveys (CSUR), 11(4). (pp. 305-330)*

[39] *Smith, Reginald D. (2017). "Bitcoin average dormancy: A measure of turnover and trading activity." arXiv preprint arXiv:1712.10287*

[40] *Suratkar, S., Shirole, M., & Bhirud, S. (2020). Cryptocurrency wallet: A review. In 2020 4th international conference on computer, communication and signal processing. (pp. 1-7)*

[41] *Szabo, N. (1994). Smart contracts.*

[42] *UK Government Chief Scientific Adviser (2016). Distributed Ledger Technology: beyond block chain. Editor: Dr Mark Peplow*

[43] *ur Rehman, M. H., Salah, K., Damiani, E., & Svetinovic, D. (2019). Trust in Blockchain cryptocurrency ecosystem. IEEE Transactions on Engineering Management, 67(4). (pag.1196-1212)*

[44] *Ward, O., & Rochemont, S. (2019). Understanding central bank digital currencies (CBDC). Institute and Faculty of Actuaries. (pp.1-52)*

[45] *Yaga, D., Mell, P. , Roby, N. and Scarfone, K. (2018), Blockchain Technology Overview, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD*

[46] *Yano, M., Dai, C., Masuda, K., & Kishimoto, Y. (2020). Blockchain and Crypto Currency: Building a High Quality Marketplace for Crypt Data. Springer Nature. (pp.77-136)*

[47] *Zhao, X., Chen, Z., Chen, X., Wang, Y., & Tang, C. (2017). The DAO attack paradoxes in propositional logic. In 2017 4th international conference on systems and informatics (ICSAI) (pp. 1743-1746). IEEE.Xin Liu, B. F. (2020). Distributed Ledger Technology. Em Intelligent Internet of Things. (pp. 393-431)*