UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Pedro Leale

Detecção de Mixers na Ethereum

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Pedro Leale

Detecção de Mixers na Ethereum

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Ivan da Silva Sendin

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

Pedro Leale

Detecção de Mixers na Ethereum

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 18 de setembro de 2025:

Ivan da Silva Sendin
Orientador

Rodrigo Sanches Miani

Luis Fernando Faina

Uberlândia, Brasil
2025

Resumo

Este trabalho propõe uma abordagem baseada em aprendizado de máquina para a detecção automatizada de contratos do tipo *mixer* na blockchain Ethereum, com foco no protocolo *Tornado Cash*. Foram utilizados dois algoritmos supervisionados — *Random Forest* e *Support Vector Machine (SVM)* — treinados com dados *on-chain* extraídos da plataforma *Blockchair*. As métricas selecionadas incluíram entropia de valores transferidos, entropia de taxas de gás, chamadas internas e variações de padrão em transações. O conjunto de treinamento foi composto exclusivamente por interações com a versão de 10 ETH do *Tornado Cash*, a fim de avaliar a capacidade de generalização dos modelos para outras denominações do protocolo. A avaliação foi realizada com base em um cenário real de 29 de outubro de 2020, no qual foram registradas mais de 1,1 milhão de transações na rede Ethereum, das quais apenas 239 estavam associadas ao Tornado Cash. O modelo *Random Forest* obteve melhor desempenho, identificando corretamente os contratos ativos do Tornado Cash, enquanto o SVM apresentou maior número de falsos positivos. Os resultados indicam que é possível detectar comportamentos suspeitos na rede Ethereum com uso de aprendizado supervisionado, embora melhorias como a filtragem de contratos legítimos e a incorporação de dados *off-chain* sejam necessárias para reduzir imprecisões.

Palavras-chave: Blockchain, Ethereum, anonimato, mixer.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de máquina
EVM	Ethereum Virtual Machine
DAO	Descentralized Autonomous Organization
DeFi	Descentralized Finance
ERC	Ethereum Request for Comments
EIP	Ethereum Improvement Proposal
PoW	Proof-of-Work
PoS	Proof-of-Stake
SVM	Support Vector Machine
TC	Tornado Cash
ZKP	Zero Knowledge Proof

Sumário

1 Introdução

Os Contratos Inteligentes (SZABO, 1997) trazem uma nova ”camada” de aplicações para os já revolucionários avanços advindos das criptomoedas (NAKAMOTO, 2009). Entre as plataformas existentes, destaca-se o **Ethereum**, que se consolidou como a principal infraestrutura para a criação e execução de contratos inteligentes, ao disponibilizar a *Ethereum Virtual Machine* (EVM) como ambiente de execução descentralizado (WOOD et al., 2014). Essa tecnologia possibilita — entre outras coisas — que um novo conjunto de aplicações financeiras seja disponibilizado (WERNER et al., 2023).

O suporte tecnológico utilizado nesse contexto é a *blockchain*: um sistema descentralizado e distribuído de livro-razão que armazena transações de forma pública e transparente (NAKAMOTO, 2009). Entretanto, essa transparência também expõe informações sensíveis de seus usuários, como valores, horários e endereços envolvidos em transferências. Como resposta, surgiram mecanismos para aumentar a privacidade transacional, entre eles os contratos *mixers*, que procuram ofuscar a origem e o destino dos fundos utilizando protocolos criptográficos avançados, como as *Zero Knowledge Proofs* (ZKP).

Os *mixers* desempenham um papel controverso no ecossistema das criptomoedas. De um lado, oferecem uma camada adicional de privacidade a usuários que desejam proteger informações sensíveis de seus fluxos financeiros; de outro, podem ser explorados em atividades ilícitas, como a lavagem de dinheiro e financiamento de crimes cibernéticos. O funcionamento desses serviços geralmente se baseia em agregar múltiplos depósitos de diferentes usuários em contratos inteligentes e redistribuí-los de forma embaralhada, dificultando a rastreabilidade direta entre origem e destino. Essa característica, embora legítima em cenários de defesa da privacidade, levanta preocupações significativas para reguladores e autoridades de segurança.

A popularização dos *mixers*, especialmente após o caso do *Tornado Cash* — que se tornou referência no uso de ZKPs para anonimização em larga escala — evidenciou os desafios dessa tecnologia. Lançado em 2019, o Tornado Cash rapidamente se consolidou na Ethereum, sendo amplamente utilizado tanto por usuários legítimos quanto por agentes maliciosos. Em 2022, o protocolo foi alvo de sanções impostas pelo Departamento do Tesouro dos Estados Unidos, sob a acusação de facilitar a lavagem de milhões de dólares oriundos de ataques hackers e outras atividades ilícitas. Esse episódio reforçou a importância de desenvolver métodos que permitam a identificação desses contratos e a análise de seus padrões transacionais. Assim, ferramentas que possibilitam a detecção automatizada de possíveis *mixers* contribuem tanto para a transparência regulatória quanto para

o avanço de pesquisas forenses na blockchain (WANG et al., 2023; CASH, 2019; YOUN; CHIN; OMOTE, 2023).

1.1 Objetivos

Este trabalho tem como objetivo desenvolver um método de detecção automatizada de contratos inteligentes do tipo *mixers* na rede Ethereum (WOOD et al., 2014), com foco em variações do *Tornado Cash*.

A metodologia consiste na coleta de transações *on-chain* associadas a contratos *mixers* e *não-mixers*, seguida da extração de métricas sobre valores transacionados e estatísticas de chamadas internas. Em seguida, aplica-se um modelo de classificação baseado em aprendizado de máquina. A análise utiliza dados históricos com o intuito de validar a eficácia do modelo e identificar características relevantes para a distinção entre *mixers* e *não-mixers*.

Os dados utilizados neste trabalho são obtidos a partir da plataforma *Blockchair* (Blockchair, 2023) e do *Etherscan*[1], acessados por meio de suas APIs públicas. Os algoritmos são implementados em Python, permitindo tanto a manipulação dos dados quanto a construção e avaliação dos modelos de aprendizado supervisionado.

[1] https://etherscan.io/

2 Revisão Bibliográfica

Este capítulo reúne os principais conceitos que embasam a pesquisa, começando pelos fundamentos da tecnologia *blockchain* e seus desdobramentos. São explorados desde os conceitos básicos de *Blockchain* até as inovações trazidas pelo *Ethereum*, incluindo contratos inteligentes e os chamados *mixers*. Também são apresentados conceitos de *Zero Knowledge Proofs*, fundamentais para a construção dos *mixers*. Por fim, são abordados os conceitos de aprendizado de máquina que sustentam o desenvolvimento do trabalho.

2.1 Blockchain

Blockchain é um sistema *peer-to-peer* para transações financeiras, concebido de forma que não seja necessário confiar em entidades mediadoras, mas sim em provas criptográficas (PIERRO, 2017), possuindo potencial para evitar diversos tipos de fraudes.

Todas as transações são validadas e posteriormente registradas em blocos, assemelhando-se a um livro-razão, o qual é de acesso público. Esses blocos são organizados sequencialmente e podem ser verificados por meio das mesmas funções criptográficas com as quais foram criados (NAKAMOTO, 2009).

Avanços recentes como contratos inteligentes (SZABO, 1997) trazem uma nova ”camada” de aplicações para este meio. Essa nova tecnologia permite - entre outras coisas - que um novo conjunto de aplicações financeiras seja disponibilizado (WERNER et al., 2023). Dentre essas aplicações, destacam-se as finanças descentralizadas (DeFi, do inglês *Decentralized Finance*), os centros de troca de *tokens* (ativos digitais) e até mesmo as organizações descentralizadas autônomas (DAO, do inglês *Decentralized Autonomous Organization*).

As **DeFi** consistem na provisão descentralizada de serviços financeiros através de um conjunto de infraestrutura, mercados e tecnologias como *smart contracts* (CHEN; BELLAVITIS, 2020). Nesse ecossistema, usuários podem realizar operações típicas do setor bancário — como empréstimos, poupança, negociação de ativos e derivativos — sem a necessidade de intermediários tradicionais.

Já as **DAOs** representam organizações que funcionam sem autoridade central ou hierarquia formal, sendo geridas de forma autônoma por meio de contratos inteligentes (WANG et al., 2019). Nessas estruturas, as regras de governança são codificadas diretamente no protocolo, e as decisões são tomadas coletivamente pelos detentores de *tokens* de governança. Esses ativos digitais funcionam como instrumentos de participação e conferem aos seus titulares o direito de voto proporcional à quantidade possuída, permitindo que

influenciem em propostas de alteração do protocolo, financiamento de iniciativas ou alocação de recursos. Um exemplo notável é a *MakerDAO*, responsável pela emissão da moeda digital DAI, cujo funcionamento e parâmetros de risco são definidos pela comunidade por meio desse mecanismo. Outro caso relevante é a *VitaDAO*, voltada ao financiamento de pesquisas na área da longevidade, demonstrando como DAOs podem ir além das finanças e atuar em campos como ciência, cultura e impacto social. (VITADAO, 2021; MAKERDAO, 2014)

Como exemplos de Blockchains, além da Bitcoin que é especializada em fazer transações, existem outras em que focam em aspectos diferentes. Como *Monero* (MöSER et al., 2018) que é conhecida por sua privacidade e fungibilidade, ou *Ethereum* (WOOD et al., 2014) que foi uma das primeiras a dar suporte nativo a contratos inteligentes.

2.2 Contratos Inteligentes

Contratos inteligentes (*smart contracts*) são programas armazenados e executados em uma *blockchain*, cujo comportamento é regido por regras imutáveis definidas em código. O conceito foi proposto por Szabo em 1997 (SZABO, 1997), com o objetivo de criar acordos autoexecutáveis que dispensassem intermediários. A implementação prática ganhou relevância a partir de 2015, com o lançamento da plataforma Ethereum, que incorporou uma máquina virtual própria — a *Ethereum Virtual Machine* (EVM) — capaz de executar contratos de forma descentralizada e determinística (WOOD et al., 2014).

Do ponto de vista técnico, contratos inteligentes interagem com dados e ativos digitais por meio de funções públicas e eventos, podendo manter estado interno e acessar variáveis globais da *blockchain*. Sua execução é desencadeada por transações, e cada instrução consome uma unidade de *gas*, que deve ser paga pelo remetente. Essa arquitetura garante segurança contra abusos de recursos, transparência na execução e auditabilidade do código, mas também implica na imutabilidade do contrato após o *deploy*, exigindo rigor no desenvolvimento e auditoria prévia (CHEN; BELLAVITIS, 2020).

2.2.1 Ethereum

Projeto inicialmente idealizado como uma plataforma Blockchain de tecnologia generalizada, dando suporte à criação de aplicações para o desenvolvedor final (WOOD et al., 2014). Este suporte é representado na capacidade de hospedar contratos inteligentes, que são aplicativos que combinam protocolos de segurança com interfaces para usuários para garantir e assegurar transações entre usuários através da Blockchain (SZABO, 1997).

A *Ethereum Virtual Machine* (EVM) é o ambiente de execução global responsável por processar contratos inteligentes na rede Ethereum. Ela garante segurança e isolamento das execuções por meio de mecanismos como o sistema de *Gas* — uma taxa paga em ETH

por cada operação computacional, que previne abusos de recursos — e pela imutabilidade, na qual o código implantado na *blockchain* torna-se permanente e auditável publicamente (WOOD et al., 2014).

Os contratos inteligentes da Ethereum podem ser desenvolvidos em diferentes linguagens de programação, sendo a **Solidity** a mais amplamente utilizada, com sintaxe semelhante a JavaScript e orientação a objetos. Outra opção é a **Vyper**, que busca simplicidade e segurança, adotando sintaxe inspirada em Python. Mais recentemente, a **Rust** tem ganhado espaço como alternativa focada em segurança e desempenho, possuindo sintaxe próxima à de C/C++ (WOOD et al., 2014) (SZABO, 1997).

O ecossistema da Ethereum abriga uma ampla gama de aplicações que vão além de simples transferências de valor. No campo de *DeFi*, plataformas como Uniswap e Aave utilizam contratos inteligentes para viabilizar empréstimos, trocas de ativos (*swaps*) e negociação de derivativos (CHEN; BELLAVITIS, 2020). Outro segmento de destaque é o de *tokens não fungíveis* (NFTs), implementados por meio do padrão ERC-721, que permite a tokenização de ativos digitais como obras de arte, músicas e até imóveis. Além disso, as Organizações Autônomas Descentralizadas (DAOs) representam um modelo inovador de governança decentralizada, no qual contratos inteligentes coordenam votações e alocação de recursos.(WANG et al., 2019).

A evolução da rede Ethereum tem sido marcada por importantes marcos tecnológicos. Um deles foi o *The Merge* (2022), que promoveu a transição do mecanismo de consenso *proof-of-work* (PoW) para *proof-of-stake* (PoS), reduzindo o consumo energético em mais de 99% (JAIN; JAIN; KRYSTYNIAK, 2023). Entre as atualizações recentes, destaca-se o *Dencun* (2024), que introduziu o *proto-danksharding* (EIP-4844), possibilitando a redução de até 90% nos custos de transação em soluções de segunda camada (L2). Já a atualização *Pectra* (2025) ampliou o saldo máximo permitido para validadores e implementou mecanismos para atingir finalidade de transações em menos de cinco segundos.

Tabela 1 – Evolução Técnica da Ethereum

Fase	**Principal Inovação**	**Impacto**
Frontier (2015)	Lançamento inicial da *blockchain*	Suporte a contratos inteligentes básicos
Beacon Chain (2020)	Camada de consenso PoS	Preparação para *The Merge*
Dencun (2024)	EIP-4844 (dados *blob*)	Redução drástica de custos em L2
Pectra (2025)	EIP-7251 (validador único)	Maior escalabilidade e segurança

2.3 Provas de Conhecimento Zero

Prova de Conhecimento Zero (*Zero-Knowledge Proof* — ZKP) é uma técnica criptográfica que permite a um provador (*prover*) demonstrar a veracidade de uma afirmação

a um verificador (*verifier*) sem revelar nenhuma informação adicional além do fato de que a afirmação é verdadeira (YANG; LI, 2020).

Exemplo de uma prova de conhecimento zero, imagine que um *prover* puxa uma carta de um baralho comum, a carta puxada é Ás de ouro cor vermelha. Este indivíduo deseja provar que a carta é vermelha, ele pode provar mostrando a carta ou pode provar mostrando todas as cartas pretas que sobraram no baralho, não demonstrando nenhuma informação sobre seu segredo, mas provando que sua carta é vermelha (NITULESCU, 2020).

Os protocolos de ZKP podem ser classificados em **interativos** e **não-interativos**. Nos *interativos*, o provador e o verificador realizam múltiplas trocas de mensagens até atingir um nível de confiança estatístico; já nos *não-interativos*, como nos *zk-SNARKs* (*Zero-Knowledge Succinct Non-interactive Arguments of Knowledge*), a prova é gerada em uma única etapa e validada pelo verificador sem necessidade de novas interações, sendo mais eficiente para ambientes descentralizados (NITULESCU, 2020).

Os algoritmos de ZKP podem ser implementados no contexto das Blockchains, especialmente em contratos inteligentes em plataformas como Ethereum, com o objetivo de garantir privacidade em um ambiente transparente, possibilitando a prevenção de falhas de segurança decorrentes da divulgação de determinados dados (ČAPKO; VUKMIROVIć; NEDIć, 2022).

2.4 *Mixers*

Os serviços de *mixing* (ou *misturadores*) são protocolos criptoeconômicos que permitem aos usuários depositar *tokens* e misturá-los com os de outros participantes em um *pool* comum, para posterior resgate em endereços não vinculáveis. Esse processo tem como objetivo principal **quebrar a heurística de análise de cadeia** (*chain analysis*), garantindo anonimato forte por meio de técnicas criptográficas avançadas (WANG et al., 2023).

Além da operação básica de um *mixer*, alguns conceitos são fundamentais para a efetividade do anonimato oferecido. Um deles é o **conjunto de anonimidade** (*anonymity set*), que representa o número de participantes entre os quais uma transação pode estar indistintamente inserida: quanto maior esse conjunto, maior a dificuldade de associação entre depósito e saque (YANG; LI, 2020). Outro fator relevante é o **atraso de resgate**, no qual a introdução de aleatoriedade temporal no momento do saque dificulta correlações diretas (WANG et al., 2023). Por fim, a prática de **depósitos de mesmo valor**, definida já na criação e implantação do contrato inteligente, impede que a quantia transferida seja utilizada como pista para vincular endereços. Esses princípios podem ser sintetizados conforme segue:

- **Conjunto de anonimidade/Anonimity Set**: Quanto maior o número de participantes, maior a indistinguibilidade (YANG; LI, 2020).
- **Atraso de Resgate**: Aleatoriedade temporal impede correlações entre depósitos e saques (WANG et al., 2023).
- **Depósitos de mesmo valor**: O valor aceito pelo *mixer* no momento do depósito é definido na criação e no deploy do contrato, a fim de evitar a associação entre endereços com base no valor transferido.

Embora comumente associados a atividades ilícitas, os *mixers* possuem também **aplicações legítimas** em diferentes contextos. Empresas e indivíduos podem recorrer a essas ferramentas para proteger patrimônio contra ataques direcionados, ocultando saldos ou movimentações significativas de terceiros mal-intencionados (YOUN; CHIN; OMOTE, 2023). Além disso, em alguns cenários de *compliance* financeiro, a dissociação de endereços durante auditorias públicas pode ser necessária para preservar a confidencialidade de estratégias corporativas ou informações sensíveis (YANG; LI, 2020). Tais aplicações podem ser resumidas como:

- **Proteção de Patrimônio**: Empresas usam *mixers* para ocultar saldos de ataques direcionados (YOUN; CHIN; OMOTE, 2023).
- **Compliance Financeiro**: Dissociação de endereços em auditorias públicas (YANG; LI, 2020).

Apesar dos usos legítimos, o funcionamento de *mixers* também impõe riscos significativos e desafios regulatórios. Casos como o do *Tornado Cash*, sancionado por autoridades norte-americanas por suposta participação na lavagem de aproximadamente 455 milhões de dólares (YOUN; CHIN; OMOTE, 2023), ilustram a magnitude do problema. Além disso, desenvolvedores e colaboradores da plataforma enfrentaram retaliações legais, incluindo sanções e prisões, evidenciando o dilema entre privacidade financeira e combate a crimes digitais. Esses riscos podem ser sintetizados como:

- **Lavagem de Dinheiro**: *Tornado Cash* foi sancionado por lavagem de $455M (YOUN; CHIN; OMOTE, 2023).
- **Retaliação**: Desenvolvedores do *Tornado Cash* enfrentaram sanções e prisões.

2.4.1 Tornado Cash

No protocolo *Tornado Cash*, o processo de anonimização se inicia com o depósito de uma quantia fixa de Ether (ETH) em um contrato inteligente, acompanhado de um

compromisso criptográfico (*commitment*) gerado por meio de um *hash* de Pedersen sobre um segredo aleatório. Esse *commitment* é então inserido como uma folha em uma estrutura de dados do tipo árvore de Merkle binária, com profundidade 20, cujos nós internos são computados utilizando a função *hash* MiMC, otimizada para operações em provas de conhecimento zero (*zk-SNARKs*).

No momento do saque, o usuário — ou um intermediário chamado *relayer* — deve apresentar uma prova *zk-SNARK* que ateste, de forma não reveladora, que o compromisso correspondente está incluído na árvore, que o caminho de verificação é válido e que o *nullifier* gerado (um identificador único derivado do segredo) ainda não foi utilizado. Essa verificação impede que um mesmo depósito seja gasto mais de uma vez.

Se a prova for válida, o contrato executa a transferência dos fundos para um novo endereço, rompendo o vínculo direto com o endereço original. Além disso, durante o processo de depósito observa-se uma quantidade maior de transações internas em comparação ao saque, devido às chamadas internas do *hash* MiMC utilizadas na árvore de Merkle binária. Na base de dados do *Blockchair* (Blockchair, 2023), são registradas 41 transações internas, nomeadas como *call_count*, para depósitos e 18 para saques.

Relayer é um intermediário opcional responsável por submeter a transação de retirada em nome do usuário, recebendo uma taxa paga diretamente pelo contrato. Essa função permite que o saque seja realizado sem que o endereço IP ou a conta original do usuário interaja diretamente com a blockchain, preservando assim o anonimato total.(CASH, 2019; YOUN; CHIN; OMOTE, 2023)

Para assegurar a característica de depósitos com valores fixos, o protocolo mantém três contratos distintos implantados na rede *Ethereum Mainnet*, cada um associado a um valor específico: 0,1 ETH (`0x12D66f87A04A9E220743712cE6d9bB1B5616B8Fc`), 1 ETH (`0x47CE0C6eD5B0Ce3d3A51fdb1C52DC66a7c3c2936`) e 10 ETH (`0x910Cbd523D972eb0a6f4cAe4618aD62622b39DbF`).

A parte *on-chain* do protocolo restringe-se ao contrato inteligente, enquanto as aplicações responsáveis pela geração de *commitment* e pela atuação como *relayers* são implementadas de forma *off-chain*. Tais aplicações estão disponíveis publicamente nos repositórios oficiais do GitHub, denominados *tornado-core* e *tornado-relayer* (CASH, 2019).

Todo esse protocolo 1 só é eficaz se o conjunto de anonimidade for suficientemente grande e os usuários utilizarem, de fato, endereços distintos e não relacionados para depósito e saque. Como discutido em (WANG et al., 2023), o uso inadequado do protocolo pode, inclusive, comprometer ainda mais a privacidade.

O *Tornado Cash* foi implantado na rede Ethereum em agosto de 2019 e em 8 de agosto de 2022, o protocolo e seus desenvolvedores passaram a ser alvo de sanções impostas pelo Escritório de Controle de Ativos Estrangeiros dos Estados Unidos (*Office of Foreign*

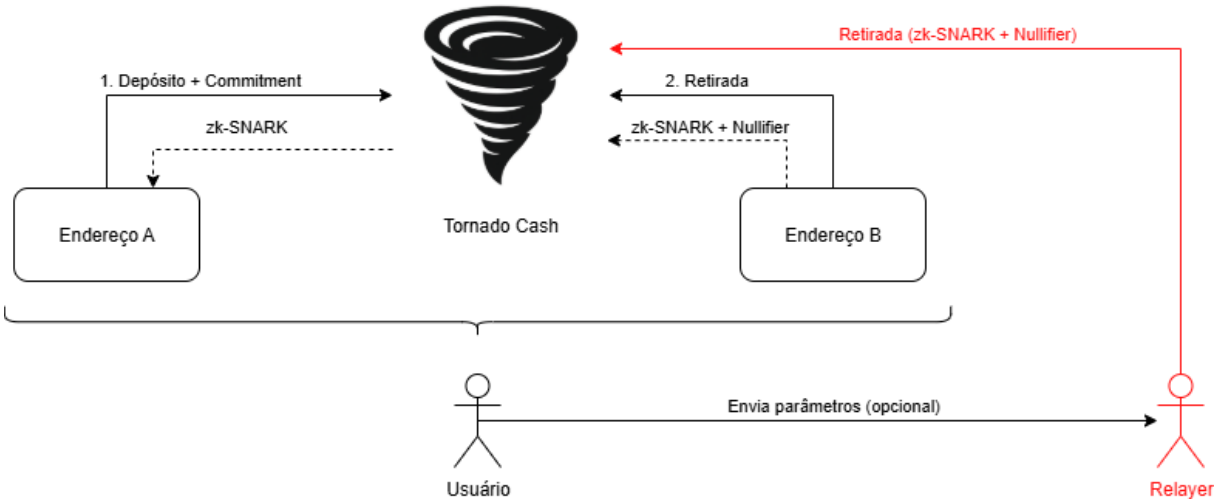

Figura 1 – Diagrama Tornado Cash

Assets Control — OFAC), sob a acusação de facilitar a lavagem de aproximadamente 455 milhões de dólares em fundos ilícitos. Essas sanções resultaram no bloqueio de endereços vinculados ao protocolo e na prisão de colaboradores, incluindo a detenção de Alexey Pertsev nos Países Baixos dias após o anúncio oficial.

2.5 Aprendizado de máquina

Aprendizado de Máquina (AM) é um subcampo da inteligência artificial que capacita sistemas a extrair padrões e inferir conhecimentos a partir de dados, sem necessidade de programação explícita em cada regra (DIETTERICH, 1990). Clássicamente, AM se divide em três paradigmas principais:

- **Aprendizado Supervisionado**: utiliza conjuntos de dados rotulados para treinar modelos capazes de predizer saídas conhecidas, com aplicações em classificação de fraudes e detecção de anomalias.
- **Aprendizado Não Supervisionado**: emprega dados não rotulados para descobrir estruturas subjacentes, como agrupamentos e redução de dimensionalidade, sendo útil em segmentação de clientes e análise exploratória.
- **Aprendizado por Reforço**: agentes aprendem políticas ótimas por meio de interação com o ambiente, recebendo recompensas ou penalidades (*rewards*) (NASTESKI, 2017).

2.5.1 Fundamentos de Aprendizado Supervisionado

Nesse paradigma, modelos são treinados a partir de pares *(entrada, saída)* previamente conhecidos. As principais tarefas incluem:

- **Classificação**: atribuição de rótulos discretos (e.g., spam/não-spam, mixer/não-mixer).
- **Regressão**: predição de valores contínuos (e.g., estimativa de preços).

O fluxo típico de trabalho abrange:

1. Coleta e pré-processamento dos dados (limpeza, normalização, tratamento de valores faltantes).
2. Separação em conjuntos de treino e teste.
3. Seleção e treinamento de algoritmos, com ajuste de hiperparâmetros.
4. Avaliação por métricas adequadas: acurácia, precisão, recall, F1-score (classificação); MSE, MAE (regressão).

Entre os desafios destacam-se o *overfitting* e o *underfitting*, mitigados por técnicas como validação cruzada e regularização.

2.5.2 Algoritmos Supervisionados Populares

2.5.2.1 Random Forest

Proposto por Breiman (BREIMAN, 2001), o *Random Forest* é um método de ensemble que combina múltiplas árvores de decisão treinadas com amostras Bootstrap e subconjuntos de atributos (*feature bagging*). A predição final resulta de votação majoritária (classificação) ou média (regressão). Suas principais vantagens são:

- Resistência ao *overfitting*, devido à diversidade gerada pelas árvores.
- Cálculo de importância de variáveis, auxiliando na interpretação do modelo.
- Robustez a valores ausentes e outliers.

2.5.2.2 Support Vector Machines (SVM)

As *Support Vector Machines* são algoritmos de classificação (e regressão) que buscam hiperplanos ótimos para separar classes em espaços de alta dimensão. Utilizam *kernels* (RBF, polinomial) para mapear dados não lineares em espaços onde são linearmente separáveis. Entre as vantagens:

- Eficácia em espaços de alta dimensionalidade.
- Controle de margem que confere robustez a ruídos, utilizando apenas vetores de suporte.

2.6 Trabalhos relacionados

Faqir-Rhazoui, Arroyo e Hassan (2021) quantifica e analisa métricas de adoção, tais como crescimento, atividade e utilização do sistema de votação, bem como os fundos de DAOs criadas por três plataformas distintas (Aragon, DAOstack, DAOhaus). Esta análise também abrange a comparação de preço e tempo de transações, como as de votos em DAOs na rede principal da Ethereum network (mainnet) e na rede paralela (xDAI). Tal estudo se relaciona com a presente monografia devido às comparações de diversos aspectos das transações, muitas vezes em contratos idênticos, porém em circunstâncias distintas, assemelhando-se à análise que será conduzida nesta monografia para a comparação de transações com e sem a implementação de *zero-knowledge proofs*.

Oliva, Hassan e Jiang (2020) analisa conjuntos de dados de contratos inteligentes na rede Ethereum para compreender o nível de atividade, categorias e a complexidade do código-fonte. Como resultado, constata-se que uma parcela muito pequena (aproximadamente 0.05%) dos contratos são responsáveis por 80% das transações em contratos. Além disso, em relação às categorias, a maioria dos contratos é utilizada para desenvolver aplicativos simples centrados em *tokens*. Quanto à complexidade, os contratos com maior atividade tendem a ser simples, no entanto, esses contratos costumam ter um número significativo de comentários da comunidade, até mesmo mais do que projetos mais populares no GitHub. Essa análise assemelha-se a esta monografia na investigação da atividade e sua relação com outras características do contrato, uma vez que o objetivo deste trabalho é analisar de maneira semelhante a adoção de contratos que implementam provas de conhecimento zero.

Lee et al. (2020) reune vários *datasets* de interações na Ethereum Network, tanto de usuário para usuário, quanto entre usuários e contratos inteligentes, extraem diversas informações e as compara com trocas de informação na Web tradicional. As transações são organizadas em grafos dependendo do tipo de análise e do tipo de transação, facilitando o percorrimento e as correlações. Além de mostrar técnicas eficientes para analisar grandes volumes de dados na Blockchain, se assemelha com esta monografia no quesito análise de transação.

3 Desenvolvimento

Nesta seção são apresentadas as etapas de desenvolvimento do trabalho, abrangendo desde a aquisição e organização dos dados até a aplicação dos modelos de aprendizado de máquina. O objetivo principal foi estruturar um processo capaz de identificar padrões característicos de contratos do tipo *mixer* na rede Ethereum, utilizando técnicas supervisionadas de classificação.

A primeira parte desta seção dedica-se ao processo de obtenção dos dados utilizados nos experimentos, incluindo a seleção das fonte e os procedimentos de pré-processamento utilizados.

Na sequência, apresentam-se os modelos de aprendizado de máquina empregados no estudo, destacando-se as motivações para a escolha de cada método e os parâmetros de configuração definidos.

3.1 Aquisição e Pré-processamento

A metodologia adotada para construção do modelo de detecção de *mixers* iniciou-se com a aquisição de registros de transações na blockchain Ethereum, referentes ao mês de março de 2025, obtidos através da plataforma *Blockchair* (Blockchair, 2023). O conjunto de dados brutos contemplava diversos atributos on-chain, incluindo identificadores de bloco e transação (`block_id`, `index`), `hash`, `timestamp`, status de execução (`failed`), tipo de chamada (`type`), endereços envolvidos (`sender`, `recipient`), número de chamadas internas (`call_count`), valores transferidos e internos (`value`, `internal_value`), suas respectivas versões em dólares, taxas de transação, métricas de gás, payload e assinaturas digitais. Esses campos podem ser melhor visualizados na Tabela 2.

Com os dados obtidos, realizou-se a etapa de engenharia de atributos, com foco na captura de padrões associados à anonimização financeira. Foram extraídas quatro métricas principais: a entropia dos valores transferidos (`value_entropy`), das taxas de gás (`fee_entropy`), dos valores internos (`internal_value_entropy`) e a média do número de chamadas internas por transação (`call_count_mean`). Esses atributos foram calculados a partir do *dataset* da plataforma Blockchair (Blockchair, 2023), sendo computados de forma agregada por endereço de remetente (*sender*) e destinatário (*receiver*).

A entropia, neste contexto, quantifica o grau de dispersão ou aleatoriedade dos valores, e é calculada conforme a Equação 3.1 em que $p(x_i)$ representa a frequência relativa de ocorrência do valor x_i na distribuição observada.

Tabela 2 – Principais atributos extraídos da Blockchair

Campo	**Descrição**
`block_id`, `index`	Identificadores de bloco e posição da transação
`hash`, `time`	Hash da transação e timestamp
`failed`, `type`	Status de execução e tipo de chamada
`sender`, `recipient`	Endereço de origem e destino
`call_count`	Número de chamadas internas
`value`, `value_usd`	Valor transferido em ETH e USD
`internal_value`, `internal_value_usd`	Valores internos
`fee`, `fee_usd`	Taxa da transação em ETH e USD
`gas_used`, `gas_limit`, `gas_price`	Informações sobre uso de gás
`input_hex`, `nonce`	Dados brutos e número sequencial
`v`, `r`, `s`	Componentes da assinatura digital

$$H(X) = -\sum_{i=1}^{n} p(x_i) \cdot \log_2 p(x_i) \tag{3.1}$$

Na sequência, as transações originadas de um endereço conhecido do *Tornado Cash* — operando com depósitos fixos de 10 ETH — foram rotuladas como pertencentes à classe *"possível_mixer"* (PERTSEV ROMAN SEMENOV, 2019). Para representar a classe negativa (*"não_mixer"*), foram selecionados aleatoriamente 100 endereços distintos com no mínimo 10 transações cada, inspecionados via Etherscan para assegurar que não apresentavam vínculos com *mixers* (Etherscan Blockchain Explorer,).

Com o conjunto de dados rotulado, preparou-se o modelo para treinamento. Primeiramente, foram extraídas quatro métricas principais: entropia dos valores transferidos, entropia das taxas de gás, entropia dos valores internos e média das chamadas internas. Em seguida, aplicou-se a técnica de **Random Over Sampling (ROS)** antes da divisão em treino e teste, replicando aleatoriamente amostras da classe minoritária para corrigir o desequilíbrio e garantir maior representatividade da classe *"possível_mixer"*.

Tabela 3 – Atributos extraídas na etapa de engenharia de atributos

Atributo	**Identificador**
Entropia dos valores transferidos	`value_entropy`
Entropia das taxas de gás	`fee_entropy`
Entropia dos valores internos	`internal_value_entropy`
Média de chamadas internas	`call_count_mean`

3.2 Random Forest

A escolha pelo uso do *Random Forest* deve-se à sua ampla adoção em tarefas de classificação supervisionada e à facilidade de implementação. Esse algoritmo é amplamente reconhecido por sua robustez em diferentes cenários e pela capacidade de lidar com variáveis heterogêneas, reduzindo riscos de sobreajuste por meio da combinação de múltiplas árvores de decisão.

Neste trabalho, o modelo foi implementado com a biblioteca `scikit-learn`, configurado com 100 árvores (`n_estimators=100`), profundidade máxima de cinco níveis (`max_depth=5`) e balanceamento interno de pesos por classe (`class_weight='balanced'`).

Além disso, foi implementada uma função de inferência que recebe um vetor com as métricas extraídas de uma transação e estima a probabilidade de pertencer à classe *"possível_mixer"*. Se a probabilidade for superior a 0,5, a transação é rotulada como tal; caso contrário, como *"não_mixer"*. Parâmetros como número de árvores e profundidade são escolhas recorrentes em aplicações práticas de aprendizado de máquina, recomendadas em guias de referência como (GÉRON, 2022).

3.3 Support Vector Machine (SVM)

Com o objetivo de avaliar uma abordagem alternativa ao *Random Forest*, foi implementado também um classificador utilizando *Support Vector Machine* (SVM). Trata-se de um método bem estabelecido na literatura, amplamente utilizado devido à sua robustez teórica e à capacidade de fornecer fronteiras de decisão claras mesmo em conjuntos de dados relativamente reduzidos. Além disso, a implementação prática do SVM, disponibilizada em bibliotecas consolidadas como a `scikit-learn`, facilita a reprodução experimental e possibilita uma comparação direta com outros modelos supervisionados, como o *Random Forest*.

Para manter a consistência na comparação entre os modelos, utilizaram-se as mesmas métricas derivadas previamente extraídas — entropia dos valores transferidos, entropia das taxas de gás, entropia dos valores internos e média das chamadas internas — bem como o mesmo conjunto rotulado de transações.

O modelo foi implementado com kernel radial (`rbf`), ativando a estimativa probabilística das classes (`probability=True`) e utilizando a opção de balanceamento automático de classes (`class_weight='balanced'`). A validação cruzada foi aplicada com cinco partições (`cv=5`), seguindo recomendações práticas amplamente utilizadas na literatura (HSU et al., 2003).

4 Resultados

Neste capítulo são apresentados os resultados obtidos no uso dos classificadores *Random Forest* e *Support Vector Machine* na detecção de transações do *mixer*.

4.1 Classificador Random Forest

Para validar a efetividade do modelo proposto na detecção de contratos do tipo *mixer*, foi selecionado o dia **29 de outubro de 2020** como principal cenário de teste. Essa data foi escolhida pois em uma análise prévia detectamos um volume expressivo de transações envolvendo os contratos do *Tornado Cash*, distribuídas entre as três principais denominações: 0,1 ETH, 1 ETH e 10 ETH. Segundo dados da plataforma *Blockchair*, foram registradas **1.114.702 transações** na blockchain Ethereum ao longo do dia, englobando **230.658 endereços únicos**.

Especificamente, foram identificadas **239 transações** direcionadas ao *Tornado Cash* nesse período, com a seguinte distribuição por contrato: **73 transações** para o contrato de 0,1 ETH, **55 transações** para o de 1 ETH e **111 transações** para o contrato de 10 ETH. Esses contratos operam sob a premissa de valores fixos por depósito, o que facilita análises específicas de movimentação e comportamento.

O modelo de classificação baseado em *Random Forest*, treinado previamente com transações do *Tornado Cash* da denominação 10 ETH e balanceado via *Random Over Sampling*, foi aplicado sobre todas as transações registradas nesse dia. Como resultado, **63 endereços** distintos foram rotulados como prováveis *mixers*.

Dentre esses, **3 endereços** correspondiam exatamente aos endereços utilizados pelo *Tornado Cash* nas transações legítimas identificadas no conjunto. Ou seja, o modelo conseguiu reconhecer, mesmo em um cenário adverso e heterogêneo, as instâncias reais do protocolo de anonimização. Os **60 endereços restantes**, classificados como *falsos positivos*, foram majoritariamente associados a contratos inteligentes de categorias como *NFT marketplaces*, *liquidity pools*, exchanges descentralizadas (DEXs) e aplicações de jogos em blockchain (*GameFi*).

A confusão do modelo nesses casos pode ser atribuída ao fato que tais contratos frequentemente compartilham características estruturais com *mixers*, tais como elevado número de transações internas, baixa variabilidade nos volumes transacionados (entropia) e presença recorrente de interações automatizadas.

Adicionalmente, observou-se um comportamento interessante no que tange à entropia dos valores transferidos pelo *Tornado Cash*. Apesar do uso de contratos com va-

lores fixos, a entropia calculada não foi nula. Isso se explica pelo fato de que, durante os saques, o campo externo `value` da transação permanece com valor `0 ETH`, e o valor real transferido ao usuário é registrado exclusivamente nos campos de chamadas internas (`internal_value`). Esse fator introduz variações nos dados que, embora pequenas, são suficientes para que a métrica de entropia registre dispersão.

Esses resultados reforçam que, mesmo com um conjunto de treinamento limitado a uma única denominação de *mixer*, o modelo é capaz de capturar padrões estruturais associados ao uso de serviços de anonimização na Ethereum. No entanto, também evidenciam a necessidade de incluir métricas adicionais — como padrões temporais, recorrência de uso e interações cruzadas entre contratos — para melhorar a precisão e reduzir a taxa de falsos positivos em contextos reais de uso.

4.2 Classificador Support Vector Machine

Além do *Random Forest*, um segundo experimento foi conduzido utilizando o modelo de *Support Vector Machine (SVM)*, com os mesmos dados de treinamento, conjunto de atributos e cenário de teste descritos anteriormente. O objetivo foi avaliar se uma abordagem distinta de classificação apresentaria desempenho semelhante ou complementar.

No cenário de teste do dia 29 de outubro de 2020, o classificador SVM identificou **77 endereços** como prováveis *mixers*. Dentre esses, os **mesmos 3 endereços** reais utilizados pelo *Tornado Cash* — previamente identificados pelo modelo Random Forest — foram corretamente classificados.

Comparado ao Random Forest, o SVM apresentou um número ligeiramente maior de falsos positivos (77 contra 63), mantendo, no entanto, a mesma taxa de detecção de casos reais. Tal comportamento pode ser atribuído à sensibilidade do SVM a variações marginais nos atributos, especialmente após o processo de normalização e balanceamento das classes.

4.3 Comparação entre os modelos

Nesta seção, apresentamos uma análise comparativa entre os dois modelos utilizados. Na Figura 2, observa-se o resultado das classificações realizadas, confirmando os pontos já discutidos nas subseções anteriores.

Na Figura 3, nota-se que os falsos positivos identificados pelo *Random Forest* apresentaram, em média, um número elevado de chamadas internas. Por outro lado, os falsos positivos do modelo SVM exibiram uma média reduzida de chamadas internas.

Ainda nessa figura, observa-se que a entropia dos valores internos (*internal value*

entropy) para os contratos verdadeiros foi próxima de zero, coerente com a premissa de depósitos de valores fixos do *Tornado Cash*. Já para os falsos positivos de ambos os classificadores, a entropia apresentou valores significativamente mais altos (entre 0.8 e 5), refletindo o comportamento mais heterogêneo de contratos legítimos de alto volume.

Por fim, a Figura 4 apresenta uma visão consolidada das métricas de avaliação. Nota-se que tanto o *Random Forest* quanto o SVM obtiveram resultados semelhantes em termos de precisão e F1-Score e Revocação (*Recall*), com diferenças marginais entre si.

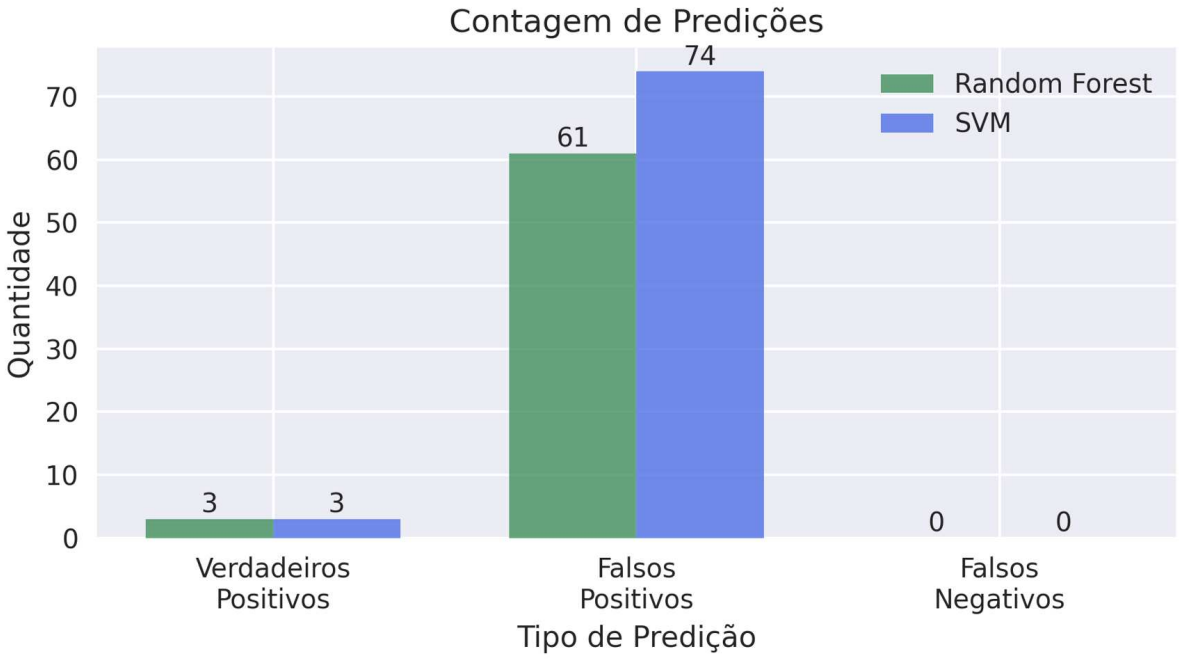

Figura 2 – Contagem de predições

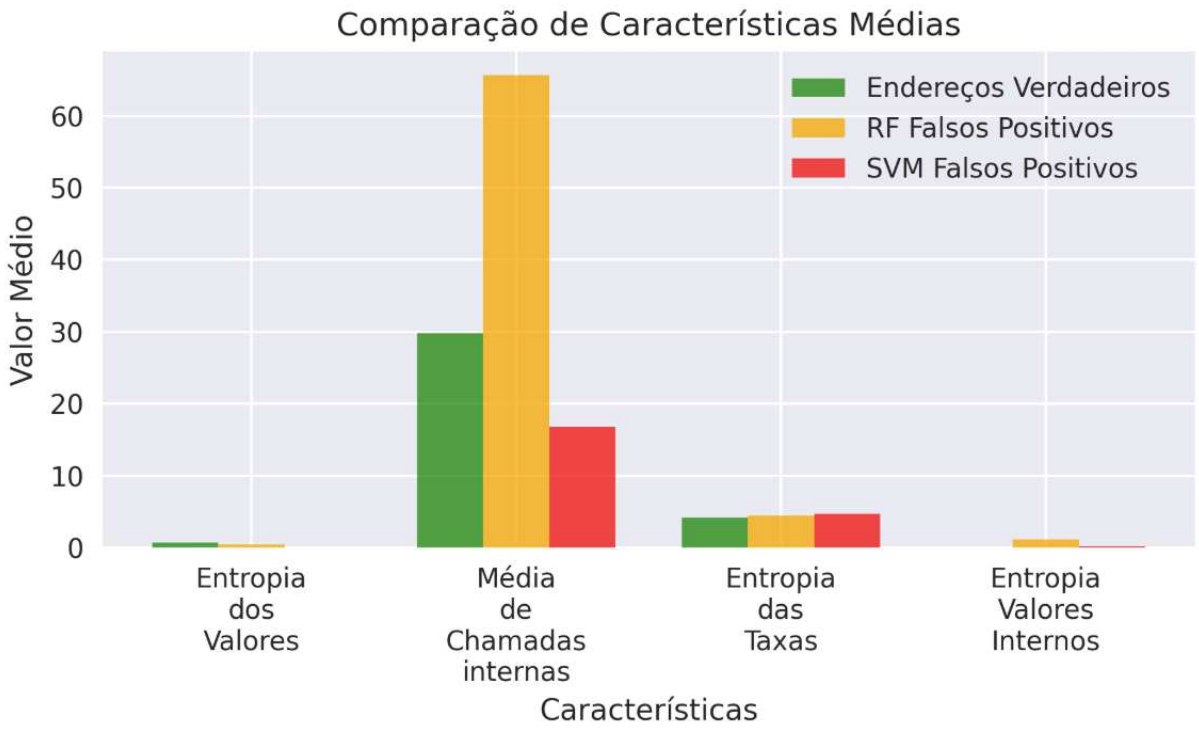

Figura 3 – Comparação de características médias

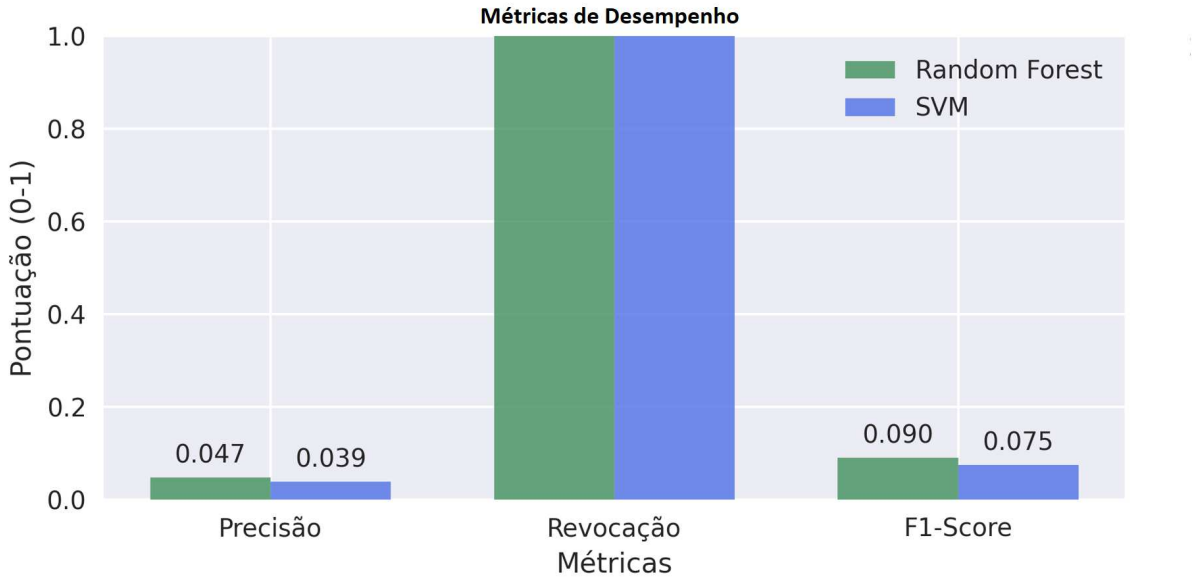

Figura 4 – Métricas de desempenho

5 Conclusão

As criptomoedas e os contratos inteligentes tem um impacto sigificativo na nossa sociedade, ao mesmo tempo que facilitam transações financeiras e trazem uma gama do novos serviços eles também introduzem novidades em atividades ilegais. Neste aspecto destacamos a lavagem de dinheiro, com o *mixer Tornado Cash*. Atualmente este *mixer* não está em atividade, mas o seu legado, tanto de código quanto de tecnologias, estão disponíveis para serem utilizados.

Desta forma, este trabalho propôs e avaliou uma abordagem baseada em aprendizado de máquina para a detecção automatizada de contratos do tipo *mixer* na plataforma Ethereum, utilizando como estudo de caso o protocolo *Tornado Cash*. Foram exploradas duas técnicas supervisionadas — *Random Forest* e *Support Vector Machine (SVM)* — treinadas com métricas *on-chain* como: entropia de valores transferidos; chamadas internas nas execução dos contratos e valores das taxas de transação pagas. O conjunto de treinamento foi construído exclusivamente com transações da versão de 10 ETH do *Tornado Cash*, estratégia adotada deliberadamente para testar a capacidade dos modelos em generalizar e identificar outras denominações do mesmo serviço, como as versões de 0,1 ETH e 1 ETH. Nos experimentos realizados, o modelo *Random Forest* obteve precisão de 0,047 e F1-Score de 0,090, enquanto o modelo SVM alcançou precisão de 0,039 e F1-Score de 0,075.

Os resultados demonstraram que os modelos foram capazes de capturar padrões estruturais típicos do processo de anonimização. O classificador baseado em *Random Forest* obteve melhor desempenho na tarefa, identificando corretamente os contratos ativos do *Tornado Cash* entre mais de um milhão de transações em um cenário real. O modelo SVM, embora com desempenho similar, apresentou leve aumento no número de falsos positivos.

Esses achados confirmam que abordagens supervisionadas podem ser eficazes para detectar comportamentos suspeitos em sistemas descentralizados. No entanto, também revelam limitações importantes: a similaridade entre *mixers* e contratos legítimos de alto volume — como exchanges descentralizadas, marketplaces de NFTs e aplicações de jogos — eleva a taxa de falsos positivos. Uma possível melhoria consiste na filtragem prévia de contratos legítimos, por meio da integração com bases externas de contratos verificados (como *Etherscan* ou *Infura* (Etherscan Blockchain Explorer, ; WUEHLER, 2016)), além da inclusão de atributos temporais como frequência de uso e recorrência de interação.

Do ponto de vista prático, soluções como a apresentada neste trabalho podem ser integradas a sistemas de monitoramento em corretoras e serviços de *cash-out*, fortalecendo

os mecanismos de *compliance* e mitigando riscos relacionados à lavagem de dinheiro. Como proposta para trabalhos futuros, recomenda-se a incorporação de dados *off-chain* — como listas públicas de endereços sancionados, fontes OSINT (Open Source Intelligence) e metadados provenientes de plataformas como *Etherscan* e *Infura* — de modo a enriquecer o contexto de análise e reduzir falsos positivos. Além disso, pode-se explorar a adaptação da metodologia para ambientes *multi-chain*, que envolvem redes como *Polygon* e *Binance Smart Chain*, bem como a utilização de sistemas mais robustos de decisão baseados na combinação de múltiplos classificadores em um esquema de votação majoritária (*majority voting*), técnica já consolidada em ensembles de aprendizado de máquina (KUNCHEVA, 2014), a fim de aumentar a confiabilidade do processo de detecção.

Referências

Blockchair. **Blockchair: Universal Blockchain Explorer and API**. 2023. Acessado em: 2 maio 2025. Disponível em: <https://blockchair.com>. Citado 3 vezes nas páginas 9, 15 e 19.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1010933404324>. Citado na página 17.

CASH, T. **Tornado cash: Privacy Solution for Ethereum Github Repository**. 2019. <https://github.com/tornadocash>. Citado 2 vezes nas páginas 9 e 15.

CHEN, Y.; BELLAVITIS, C. Blockchain disruption and decentralized finance: The rise of decentralized business models. **Journal of Business Venturing Insights**, v. 13, p. e00151, 2020. ISSN 2352-6734. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352673419300824>. Citado 3 vezes nas páginas 10, 11 e 12.

DIETTERICH, T. G. Machine learning. **Annual review of computer science**, Citeseer, v. 4, n. 1, p. 255–306, 1990. Citado na página 16.

Etherscan Blockchain Explorer. **Etherscan Blockchain Explorer**. Acesso em: maio de 2025. Disponível em: <https://etherscan.io>. Citado 2 vezes nas páginas 20 e 26.

FAQIR-RHAZOUI, Y.; ARROYO, J.; HASSAN, S. A comparative analysis of the platforms for decentralized autonomous organizations in the ethereum blockchain. **Journal of Internet Services and Applications**, v. 12, 12 2021. Citado na página 18.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. [S.l.]: "O'Reilly Media, Inc.", 2022. Citado na página 21.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. et al. A practical guide to support vector classification. Taipei, 2003. Citado na página 21.

JAIN, A.; JAIN, C.; KRYSTYNIAK, K. Blockchain transaction fee and ethereum merge. **Finance Research Letters**, v. 58, p. 104507, 2023. ISSN 1544-6123. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1544612323008796>. Citado na página 12.

KUNCHEVA, L. I. **Combining pattern classifiers: methods and algorithms**. [S.l.]: John Wiley & Sons, 2014. Citado na página 27.

LEE, X. T.; KHAN, A.; GUPTA, S. S.; ONG, Y. H.; LIU, X. Measurements, analyses, and insights on the entire ethereum blockchain network. In: **Proceedings of The Web Conference 2020**. New York, NY, USA: Association for Computing Machinery, 2020. (WWW '20), p. 155–166. ISBN 9781450370233. Disponível em: <https://doi.org/10.1145/3366423.3380103>. Citado na página 18.

MAKERDAO. **MakerDAO – An Unbiased Global Financial System**. 2014. Acessado em agosto de 2025. Disponível em: <https://makerdao.com/>. Citado na página 11.

MöSER, M.; SOSKA, K.; HEILMAN, E.; LEE, K.; HEFFAN, H.; SRIVASTAVA, S.; HOGAN, K.; HENNESSEY, J.; MILLER, A.; NARAYANAN, A.; CHRISTIN, N. **An Empirical Analysis of Traceability in the Monero Blockchain**. 2018. Disponível em: <https://arxiv.org/abs/1704.04299>. Citado na página 11.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. maio 2009. Disponível em: <http://www.bitcoin.org/bitcoin.pdf>. Citado 2 vezes nas páginas 8 e 10.

NASTESKI, V. An overview of the supervised machine learning methods. **Horizons. b**, v. 4, n. 51-62, p. 56, 2017. Citado na página 16.

NITULESCU, A. **zk-SNARKs: a gentle introduction**. [S.l.]: Technical report, 2020. Citado na página 13.

OLIVA, G. A.; HASSAN, A. E.; JIANG, Z. M. An exploratory study of smart contracts in the ethereum blockchain platform. **Empirical Software Engineering**, v. 25, n. 3, p. 1864–1904, Mar 2020. Citado na página 18.

PERTSEV ROMAN SEMENOV, R. S. A. **Tornado.Cash: 10 ETH Mixer Contract**. 2019. <https://etherscan.io/address/0x910cbd523d972eb0a6f4cae4618ad62622b39dbf>. Acesso em: 15-Mar-2025. Citado na página 20.

PIERRO, M. D. What is the blockchain? **Computing in Science Engineering**, v. 19, n. 5, p. 92–95, 2017. Citado na página 10.

ROSA, G. J. **The elements of statistical learning: Data mining, inference, and prediction by Hastie, T., Tibshirani, R., and Friedman, J.** [S.l.]: Oxford University Press, 2010. Nenhuma citação no texto.

SZABO, N. Formalizing and securing relationships on public networks. **First Monday**, University of Illinois Libraries, v. 2, n. 9, set. 1997. ISSN 1396-0466. Disponível em: <http://dx.doi.org/10.5210/fm.v2i9.548>. Citado 4 vezes nas páginas 8, 10, 11 e 12.

VITADAO. **VitaDAO – The Longevity DAO**. 2021. Acessado em agosto de 2025. Disponível em: <https://www.vitadao.com/>. Citado na página 11.

WANG, S.; DING, W.; LI, J.; YUAN, Y.; OUYANG, L.; WANG, F.-Y. Decentralized autonomous organizations: Concept, model, and applications. **IEEE Transactions on Computational Social Systems**, v. 6, n. 5, p. 870–878, 2019. Citado 2 vezes nas páginas 10 e 12.

WANG, Z.; CHALIASOS, S.; QIN, K.; ZHOU, L.; GAO, L.; BERRANG, P.; LIVSHITS, B.; GERVAIS, A. On how zero-knowledge proof blockchain mixers improve, and worsen user privacy. In: **Proceedings of the ACM Web Conference 2023**. New York, NY, USA: Association for Computing Machinery, 2023. (WWW '23), p. 2022–2032. ISBN 9781450394161. Disponível em: <https://doi.org/10.1145/3543507.3583217>. Citado 4 vezes nas páginas 9, 13, 14 e 15.

WERNER, S.; PEREZ, D.; GUDGEON, L.; KLAGES-MUNDT, A.; HARZ, D.; KNOTTENBELT, W. Sok: Decentralized finance (defi). In: **Proceedings of the 4th ACM Conference on Advances in Financial Technologies**. New York, NY, USA: Association for Computing Machinery, 2023. (AFT '22), p. 30–46. ISBN 9781450398619. Disponível em: <https://doi.org/10.1145/3558535.3559780>. Citado 2 vezes nas páginas 8 e 10.

WOOD, G. et al. Ethereum: A secure decentralised generalised transaction ledger. **Ethereum project yellow paper**, v. 151, n. 2014, p. 1–32, 2014. Citado 4 vezes nas páginas 8, 9, 11 e 12.

WUEHLER, E. G. M. **Infura - Build, Scale, Disrupt The world's most powerful suite of high availability blockchain APIs and developer tools**. 2016. <https://infura.io/>. Acessado em julho de 2025. Citado na página 26.

YANG, X.; LI, W. A zero-knowledge-proof-based digital identity management scheme in blockchain. **Computers Security**, v. 99, p. 102050, 2020. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404820303230>. Citado 2 vezes nas páginas 13 e 14.

YOUN, M.; CHIN, K.; OMOTE, K. Empirical analysis of cryptocurrency mixer: Tornado cash. In: **2023 Congress in Computer Science, Computer Engineering, Applied Computing (CSCE)**. [S.l.: s.n.], 2023. p. 2324–2331. Citado 3 vezes nas páginas 9, 14 e 15.

ČAPKO, D.; VUKMIROVIć, S.; NEDIć, N. State of the art of zero-knowledge proofs in blockchain. In: **2022 30th Telecommunications Forum (TELFOR)**. [S.l.: s.n.], 2022. p. 1–4. Citado na página 13.

Apêndices

APÊNDICE A – Github

Os artefatos de código usados estão presentes no repositório do Github.

<https://github.com/PedroLeale/TCC-BCC-UFU>