

Identificação de perfis de comportamento de usuários no Ethereum utilizando técnicas de aprendizado de máquina*

Júlia A. Valadares¹, Vinicius C. Oliveira¹, José Eduardo de Azevedo Sousa¹, Heder S. Bernardino¹, Saulo Moraes Villela¹, Alex B. Vieira¹, Glauber Dias Gonçalves²

¹ Universidade Federal de Juiz de Fora (UFJF) – DCC

² Universidade Federal do Piauí (UFPI) – CSHNB

{juliavaladares, vinicius.oliveira, eduardo2, heder}@ice.ufjf.br,
{saulo.moraes, alex.borges}@ufjf.edu.br, ggoncalves@ufpi.edu.br

Abstract. *Ethereum is one of the largest active cryptographic platforms today, and it is becoming a digital business environment among users. It is designed to allow decentralized transactions between anonymous users. However, the development of methods to identify user behavior profiles, keeping their identities anonymous, has the potential to leverage business on this platform. In this work, we investigate the use of machine learning to classify a user profile as professional or common based on the attributes of their transactions. This classification is challenging due to the small fraction of users publicly labeled on Ethereum and still the considerably smaller fraction of professional users. To conduct this investigation, we train models considering carefully balanced sets of transactions with labeled users. Our results show high performance models for the classification of profiles, achieving a performance greater than 90% for accuracy, precision, and other related measures. In addition, we have identified the most relevant characteristics in transactions for this classification.*

Resumo. *Ethereum é uma das maiores plataformas de cripto ativos atualmente, e vem se tornando um ambiente de negócios digitais entre usuários. O Ethereum foi concebido para permitir transações descentralizadas entre usuários anônimos. Contudo, o desenvolvimento de métodos para identificar perfis de comportamentos de usuários, mantendo suas identidades anônimas, têm o potencial para alavancar negócios nessa plataforma. Nesse trabalho, investigamos o uso de aprendizado de máquina para classificar um perfil de usuário como profissional ou comum a partir de atributos de suas transações. Essa classificação é desafiadora devido à pequena fração de usuários publicamente rotulados no Ethereum e ainda a fração consideravelmente menor de usuários profissionais. Para conduzir essa investigação, treinamos modelos considerando conjuntos cuidadosamente balanceados de transações com usuários rotulados. Nossos resultados mostram modelos de alto desempenho para a classificação de perfis, alcançando desempenho superior a 90% para acurácia, precisão, revocação e demais medidas relacionadas. Adicionalmente, identificamos as características mais relevantes em transações para essa classificação.*

*Essa pesquisa é financiada por CNPq, CNPq/Amazon AWS (Processo: 440069/2020-3), CAPES, FAPEMIG e UFJF

1. Introdução

Ethereum é uma plataforma popular para transferência e negociação de ativos digitais, ou *criptomoedas*, baseada na tecnologia blockchain [Wood 2014]. Desde a sua criação em 2014 até o presente momento, o Ethereum já registrou mais de 1 milhão de transações e tem mais de 273 bilhões de dólares em valor de mercado¹. Em adição às funcionalidades de criptomoedas populares, o Ethereum é capaz de processar programas, também conhecidos como *contratos inteligentes*, o que impulsiona essa plataforma e a torna um ambiente para desenvolvimento de aplicações descentralizadas, dado a arquitetura distribuída da tecnologia blockchain. Os contratos inteligentes vem possibilitando a formação de um ecossistema com diferentes domínios de aplicações que se estendem da área financeira para as áreas de serviços em governo e indústrias [Hu et al. 2021].

Criptomoedas, em sua essência, são desenvolvidas para permitir transações financeiras entre usuários anônimos. Isso é possível com o uso de criptografia assimétrica, componente que faz parte da tecnologia blockchain, onde os usuários se identificam pelas suas chaves públicas e usam as chaves privadas secretas para autenticar as transações que eles emitem em favor de outros usuários [Wood 2014]. Por sua vez, a identificação de perfis de usuários mantendo suas identidades anônimas é uma funcionalidade com potencial para alavancar negócios em plataformas de aplicações descentralizadas como o Ethereum. Por exemplo, aplicações financeiras descentralizadas (*DeFis*) [Schär 2020] que oferecem serviços² para pagamentos, empréstimos, doações ou *royalties* por bens digitais (e.g., *tokens não fungíveis*), podem utilizar a análise de perfis para buscar clientes adequados aos seus serviços, analisar riscos de crédito e até mesmo classificar usuários com envolvimento em esquemas fraudulentos [Bartoletti et al. 2020].

Alguns serviços Web já vem se movimentando para desenvolver ferramentas de perfis de usuários em redes blockchains. Destacam-se dentre esses as iniciativas do Etherscan, DApp.com e DApp Review, que buscam identificar as atividades das principais contas e tokens em redes blockchains³. Contudo, essa identificação ainda consiste de processos manuais em sua maior parte⁴, dependente em alguns casos da requisição do usuário para sua identificação e ainda da confirmação de suas atividades/perfil a partir de seus contatos na rede. Portanto, métodos para a identificação automática de perfis são de grande utilidade para tais serviços e, por conseguinte, o ecossistema de aplicações DeFi.

Nesse sentido, análises de comportamento de usuários baseado em transações registradas publicamente em blockchains é uma área que vem atraindo a atenção de pesquisadores. De fato, há um esforço de pesquisa em identificar esquemas fraudulentos e ataques nas redes blockchain, focando em comportamentos suspeitos de usuários [Rebello et al. 2020, Bartoletti et al. 2020, Sousa et al. 2021]. Há também estudos caracterizando o Ethereum e a maneira como seus usuários agem [Mascarenhas et al. 2018, Chen et al. 2020], considerando propriedades estruturais da rede. Modelos de aprendizado de máquina vêm sendo amplamente utilizados para inferir comunidades de usuários em criptomoedas [Norvill et al. 2017, Wang et al. 2020, Payette et al. 2017,

¹<https://etherscan.io>

²Exemplos de DeFi Ecosystem - <https://defiprime.com/ethereum>

³<https://etherscan.io/labelcloud>, <https://www.dapp.review/about>, https://www.dapp.com/token/Dapp-WhitePaper_en.pdf

⁴<https://info.etherscan.com/public-name-tags-labels/>

Wu et al. 2021]. Recentemente, buscou-se inferir em [Aspembitova et al. 2021] o perfil de usuários no Ethereum com o foco em risco de investimento (e.g., pessimista ou otimista). Todos esses trabalhos trazem contribuições relevantes na área, mas ainda não tratam de perfis de usuários quanto às atividades profissionais ou casuais na rede.

Para explorar essa lacuna, propomos neste artigo técnicas de aprendizado de máquina para a classificação automática do perfil de atividade de usuários na plataforma Ethereum. Especificamente, focamos em dois tipos de perfis: *usuários profissionais* com atividades associadas à prestação de serviços, como casas de câmbio e DeFis, e *usuários comuns*, que utilizam tais serviços ou apenas fazem transferências financeiras casuais. Nota-se que essa classificação não é trivial e requer o conhecimento de especialistas em análises de transações de criptomoedas [Chen et al. 2020]. Nossa proposta foi então desenvolver modelos supervisionados de aprendizado de máquina baseado em características extraídas de transações do Ethereum e alguns perfis rotulados extraídos de serviços Web especializados. Outro ponto a se observar é que os poucos dados rotulados são desbalanceados, ou seja, há um número muito maior de amostras de uma classe em detrimento de outra, o que dificulta o problema de aprendizado de máquina. Nesse sentido, os modelos desenvolvidos exploram técnicas para amostragem de dados com classes desbalanceadas.

Nossa abordagem de classificação foi acoplada a uma estrutura para treinamento automático e teste de modelos, usando dados reais do Ethereum. Nossos dados contêm mais de 17 mil contas únicas da rede Ethereum, coletadas de transações publicamente disponíveis na blockchain dessa criptomoeda e os rótulos das classes foram obtidos a partir dos serviços Etherscan e Etherchain⁵. Em suma, nossos resultados experimentais mostram alto desempenho para a classificação de perfis dos usuários/contas do Ethereum. De fato, os modelos de classificação tiveram altos valores de acurácia (uma média de 96,13%) juntamente com Revocação, F_1 -score, F_β -score, coeficiente de correlação de Matthews e AUC-ROC maiores que 0.29, 0.10, 0.15 e 0.66, respectivamente.

Em suma, esse artigo traz as seguintes contribuições: (I) um modelo baseado em aprendizagem de máquina para identificar perfis de usuários, especificamente a classificação como prestadores de serviços (usuários profissionais) e contas de usuários comuns; (II) a discussão sobre as características extraídas de transações disponíveis publicamente em blockchains que possibilitam o desenvolvimento desse modelo.

O restante do artigo está organizado como segue: discutimos trabalhos relacionados na Seção 2. Na Seção 3, descrevemos os dados utilizados, ao passo que a metodologia que foi usada ao longo deste trabalho é discutida na Seção 4. Por sua vez, os resultados obtidos são discutidos na Seção 5. Finalmente, as considerações finais e discussão sobre trabalhos futuros estão na Seção 6.

2. Trabalhos relacionados

Modelos de aprendizagem de máquina vêm sendo amplamente utilizados para uma variedade de análises sob a plataforma Ethereum, que envolvem desde perfis de usuários, segurança em contratos inteligentes e desempenho na execução de transações.

⁵<https://etherscan.io>, etherchain.org

Mais diretamente relacionado ao nosso trabalho estão os modelos para identificar perfis de usuários baseado em suas transações. Em [Aspembitova et al. 2021], foram desenvolvidos modelos de aprendizado de máquina para a identificação de usuários baseado nos tradicionais perfis de investimento (e.g., pessimista ou otimista). Nossa proposta segue também a linha de investigação de perfis, contudo o nosso foco é a prestação de serviços para identificar usuários comuns ou que exercem atividades profissionais.

Em [Norvill et al. 2017], os autores buscam inferir a finalidade de uma conta de usuário a partir de características extraídas do código do contrato inteligente associado a essa conta. Por sua vez, [Wang et al. 2020] extraem características de contas a partir da associação dessas com contas populares, cuja atividade já está rotulada na API Etherscan. Ambos os trabalhos usam algoritmos de agrupamento não supervisionados para distinguir as contas da plataforma em vários grupos, segundo suas finalidades ou serviços prestados. [Payette et al. 2017, Wu et al. 2021] também propõem modelos de categorização de usuários do Ethereum, utilizando algoritmos de agrupamentos. Os autores extraem características de transações como em nosso trabalho. O foco desses trabalhos, contudo, é identificar comunidades ao qual usuários pertencem.

Sobre perfis gerais de contas de usuários do Ethereum, é importante mencionar os trabalhos que analisam o relacionamento entre contas sob a perspectiva de teoria de grafos. Uma categorização de contas de usuários do Ethereum sob grafos multi fluxos foi proposta por [Chen et al. 2020], levando em consideração três tipos de relacionamento: usuário para usuário, um usuário para contrato inteligente e contrato inteligente para contrato inteligente. [Mascarenhas et al. 2018] analisam a centralidade (importância) de contas na rede Ethereum por meio de grafos multi-aspectos e modelos de redes complexas. Por fim, [Motamed and Bahrak 2019] reconstruíram o grafo temporal de transações entre contas do Ethereum para analisar a variação da densidade do grafo em função do preço da criptomoeda em dólar. Os autores identificaram os padrões mais comuns de transações baseado em características das contas. Diferentemente desses, nosso trabalho visa desenvolver modelos para classificar perfis considerando atividades profissionais ou casuais.

Um esforço de pesquisa relevante vem sendo dedicado aos aspectos de segurança do Ethereum. [Xu et al. 2020] desenvolveram um modelo de classificação com floresta aleatória para distinguir o tráfego normal de tráfego malicioso em um tipo de ataque identificado como Eclipse. Pirâmides financeiras em contratos inteligentes foram investigados aplicando métodos estatísticos usados para investigar fraudes nos mercados financeiros por [Bartoletti et al. 2020], ao passo que [Rebello et al. 2020] aplicaram modelos de aprendizagem de máquina supervisionados para identificar esse tipo de fraude. Um modelo de rede neural LSTM foi desenvolvido por [Hu et al. 2021] para identificar seis tipos de contratos inteligentes mais comuns no Ethereum, incluindo contratos de alto-risco, tendo como entrada um conjunto de transações associadas aos contratos. Diferente desses trabalhos, aplicamos técnicas de aprendizagem de máquina para identificar contas associadas a prestação de serviços e diferenciá-las de contas comuns no Ethereum, o que intuitivamente têm padrões diferentes de comportamento suspeito e fraudulento.

Outros trabalhos aplicam com sucesso modelos de aprendizado de máquina para análises de desempenho da plataforma Ethereum. Por exemplo, [Singh and Hafid 2019] propuseram modelos de aprendizado supervisionado para prever o tempo de confirmação de uma transação, explorando o impacto de classes de dados desbalanceadas no

treinamento e testes dos modelos classificadores escolhidos. Em linha com esses esforços de pesquisa, aprimoramos esses modelos para outras questões importantes sobre confirmação de transações no Ethereum em nossos trabalhos anteriores. Anteriormente, investigamos a relação entre a tarifa que os usuários pagam por transações e seu tempo de confirmação por meio de técnicas de correlação e de aprendizado não supervisionado [Sousa et al. 2020], e também desenvolvemos métodos de previsão de falhas de processamento de contratos por mineradores do Ethereum via comitês de classificadores com aprendizado supervisionado [Oliveira et al. 2021]. Essas pesquisas são ortogonais a esse trabalho, pois elas exploram características de transações em modelos de aprendizado de máquina, mas não tratam diretamente de perfis de usuários.

3. Descrição dos Dados

Neste trabalho, utilizamos a base de dados de transações do Ethereum coletada em nosso trabalho anterior [Oliveira et al. 2021]. Essa base foi obtida via as APIs Etherchain e Etherscan,⁶ que permitem obter dados detalhados de transações confirmadas ou pendentes e informações sobre as contas associadas a essas transações.

A base de dados utilizada possui 20.857.783 transações coletadas de 15 de Abril até 26 de Julho de 2019. Seleccionamos 2.519.711 transações dessa base de forma aleatória e uniforme ao longo desse período de coleta para as nossas análises. Dessa forma, buscamos reduzir efeitos diários e semanais de sazonalidade da plataforma causados por variações no valor da criptomoeda (Ether) em dólar e que podem levar à volatilidade no número de transações ocorridas diariamente na plataforma. As transações seleccionadas correspondem aproximadamente a 3.2% da quantidade de transações que o Ethereum foi capaz de processar dentro do período de coleta da base de dados.

Complementamos essa base de dados coletando informações extras de todas as contas envolvidas nas transações seleccionadas. Logo, verificamos as contas remetentes e receptoras dessas transações e criamos uma lista de todas as contas únicas observadas. Para fins de análises, i.e., treinamento e teste de modelos, consideramos cada conta como um usuário da plataforma Ethereum, embora múltiplas contas possam pertencer a um mesmo usuário (pessoa ou organização). Ao todo, foram 17.020 contas (usuários) utilizados para as análises.

Para cada uma dessas contas, extraímos informações sobre as movimentações com base nas transações seleccionadas. Especificamente, calculamos o número de transações enviadas (a conta é o remetente), o número de transações recebidas (a conta é o destinatário), número de transações enviadas para contratos inteligentes, e o número de transações recebidas cujo remetente são contratos inteligentes. Adicionalmente, coletamos informações mais recentes de cada conta: o saldo em Ether e o total de transações realizados pela conta, que considera também as transações não observadas em nossa base.⁷

Por último, coletamos a rotulação/classe para as contas utilizando a API Etherchain. Em geral, APIs de informações sobre Ethereum como Etherchain e Etherscan

⁶<https://etherscan.io> e <https://etherchain.org/>

⁷Não foi possível extrair tais informações de forma consistente da base de dados de transações. Logo, as coletamos para todas as contas tendo como referência o estado global da blockchain em um mesmo bloco.

rotulam contas que tem atividade profissional na plataforma e precisam de ampla publicidade. Esse processo tipicamente requer que o usuário proprietário da conta informe uma categoria em que suas atividades se enquadram e um nome para identificar a conta publicamente.⁸ Em alguns casos, a rotulação ainda requer que os contatos mais recentes que o usuário realizou transações confirme as informações sobre a sua atividade. Cada API estabelece categorias e padrões próprios para a identificação de contas. No caso do Etherchain, onde coletamos os rótulos, contas com atividades profissionais tem um identificador (nome) público. Logo, definimos as contas sem esse identificador em nossa base de dados como contas de usuários comuns, ou seja, usuários que não necessitam de ampla divulgação na rede Ethereum. Portanto, assumimos que esses usuários não exercem uma atividade profissional. Por outro lado, definimos as contas com o identificador como usuários profissionais. Dessa forma, obtivemos 117 (0,68%) usuários profissionais e 16.903 (99,32%) usuários comuns. A Tabela 1 sumariza todas as características extraídas das contas a serem utilizadas a seguir para nossas análises.

Tabela 1. Características extraídas das contas do Ethereum.

Atributo	Descrição
Conta do usuário	Identificador da conta do usuário (chave pública) definida na plataforma Ethereum.
Transações enviadas	Número de transações enviadas pela conta (a conta é o remetente).
Transações recebidas	Número de transações recebidas pela conta (a conta é o destinatário).
Transações enviadas para contrato	Número de transações enviadas pela conta cujo destinatário é um contrato.
Transações recebidas de contrato	Número de transações recebidas pela conta cujo remetente é um contratos.
Saldo em Ether	Saldo em Ether da conta recentemente
Total de transações enviadas	Número de transações enviadas pela conta (a conta é o remetente) incluindo transações mais recentes.
Rótulo	Dado binário, onde usuário comum tem valor 0 e usuário profissional tem valor 1.

4. Arcabouço Proposto

Nesta seção, apresentamos o nosso arcabouço para gerar os modelos de aprendizado de máquina que classificam usuários no Ethereum como perfil comum ou profissional. Antes da aplicação desses modelos, foi feito o pré-processamento dos dados a serem utilizados. Nessa parte, os dados foram normalizados dentro do intervalo [0,1] para que os modelos preditivos tenham maior acerto na determinação das classes. A seguir, descrevemos os métodos de aprendizado de máquina adotados e as métricas utilizadas para avaliação dos modelos.⁹

⁸<https://info.etherscan.com/public-name-tags-labels/>

⁹<https://github.com/juliaavaladares/accountsclassification>

4.1. Modelos de Aprendizado de Máquina

Nosso arcabouço utiliza um conjunto de classificadores para identificar de forma binária os perfis de usuários do Ethereum (i.e., usuário comum ou profissional). O conjunto de modelos escolhidos para esse trabalho foram: K-Vizinhos Mais Próximos (KNN) [Peterson 2009], Árvore de Decisão (DCT) [Safavian and Landgrebe 1991], Floresta Aleatória (RF) [Breiman 1996], Regressão Logística (LG) [Wright 1995], Máquinas de Vetor de Suporte (SVM) [Cortes and Vapnik 1995] e Comitês de Classificadores [Dietterich et al. 2002].

O KNN é um método de aprendizado supervisionado que pode ser usado para classificação ou regressão. Em ambos os casos, dado um conjunto de dados de treinamento e uma nova amostra, a predição (valor numérico ou classe) pra essa amostra é determinada usando as k instâncias mais próximas dela no conjunto de treinamento. O KNN foi usado aqui com propósito de classificação. Nota-se que é preciso escolher um valor apropriado para o número de vizinhos k .

A DCT classifica uma amostra por meio de uma sequência de decisões representadas numa estrutura de árvore. A classificação de uma amostra prossegue do nó raiz para um nó folha, onde cada nó folha representa uma classe. Os nós da árvore são formados por condições vinculadas a um dos atributos da base de dados e as possíveis decisões aparecem nas ramificações desses nós. As DCT são criadas um processo que inicia na raiz e utiliza uma métrica de qualidade para identificar a melhor separação dos dados.

As RF são combinações de árvores de decisão, de modo que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição. O conceito fundamental por trás da floresta aleatória é simples, mas muito útil para classificações já que um grande número de modelos não correlacionados (árvores) operando como um comitê terá bom desempenho quando comparado a modelos individuais.

A LR é um modelo estatístico que usa uma função logística para modelar uma variável dependente binária. Dado um problema de classificação que possui uma variável dependente com dois valores possíveis (0 ou 1), um modelo LR retorna a probabilidade de uma dada observação ser da classe 1.

O SVM classifica as instâncias via uma superfície de separação. Essa superfície é encontrada ao resolver um problema de otimização que busca pelo hiperplano que maximize a margem entre ele e dados de classes distintas. Neste trabalho, consideramos três funções *kernel* para o SVM: (i) linear, (ii) gaussiana e (iii) sigmoide.

Por último, os Comitês de Classificação combinam modelos de aprendizado de máquina diferentes e usam o voto da maioria (*hard*) ou um voto ponderado (*soft*) para prever os rótulos das classes. Os Comitês de Classificadores exploram um conjunto de modelos a fim de equilibrar suas fraquezas individuais.

Foram usadas técnicas de validação cruzada para treinar e validar os modelos. Além disso, os dados apresentam classes desbalanceadas, pois apenas 0,6% das instâncias representam usuários profissionais. Assim, estratégias para balancear os dados de treinamento podem ser aplicadas em cada etapa da validação cruzada. Implementamos dois tipos diferentes de balanceamento para esses dados, além dos dados não balanceados:

- Subamostragem aleatória (*Random Undersampling*) busca remover aleatoriamente as amostras da classe majoritária para igualar a classe minoritária;
- Sobreamostragem aleatória (*Random Oversampling*) envolve suplementar os dados de treinamento com várias cópias de algumas das classes minoritárias.

O melhor modelo na análise de desempenho pode ser usado para classificar os perfis de usuários. Além disso, pode-se analisar tipos de modelos (como RF e DT) para extrair informações sobre qual o caminho o modelo tomou para decidir a classificação.

4.2. Métricas de desempenho

Com o intuito de analisar o desempenho dos métodos e estratégias de aprendizado de máquina, usamos métricas como Accuracy, Precision, Recall, F_1 -score, F_β -score com $\beta = 2$, coeficiente de correlação de Matthews (MCC), número de verdadeiros positivos (TP) e negativos (TN) e área sob a curva ROC (AUC-ROC). Essas métricas foram calculadas usando os conjuntos de teste.

A Acurácia (*accuracy*) é o número de previsões corretas dividido pelo número total de previsões. Apesar de ser uma métrica importante, a acurácia pode ser mal interpretada em bases desbalanceadas. Nesse caso, um modelo pode prever o valor da classe majoritária para todas as previsões e atingir uma alta acurácia. A precisão (*Precision*) é uma métrica que quantifica o número de previsões positivas corretas feitas, podendo ser considerada como a acurácia para as classes minoritárias. A *Revocação* é a acurácia aplicada apenas às instâncias positivas, o que representa a proporção dos itens relevantes (positivos) que são classificados como positivos pelos modelos. Observe que os valores de *revocação* podem ser vistos como uma forma de verificar como uma técnica de balanceamento melhorou ou piorou a detecção da classe minoritária.

Os F -scores são calculados a partir de uma combinação de precisão e *revocação*. O F_1 -score é calculado a partir da média harmônica da *revocação* e da precisão, em que seu melhor valor é quando atinge 1 e seu pior valor é 0. Já o F_β -score é a generalização do F_1 -score com a adição do parâmetro β que determina o peso da métrica *revocação* no resultado final. Um $\beta < 1$ oferece peso maior para a precisão, enquanto o $\beta > 1$ favorece a *revocação*. Ao lidar com conjuntos de dados desequilibrados, o objetivo geralmente é aumentar a *revocação*, evitando uma grande diferença na precisão. Portanto, além do tradicional F_1 -score, optamos por analisar o F_2 -score, pois ele valoriza o valor de *revocação* enquanto ainda considera o valor de precisão.

O coeficiente de correlação de Matthews (MCC) [Boughorbel et al. 2017] é uma ferramenta para avaliação de modelos. Ele mede as diferenças entre os valores reais e os valores previstos. Ele é um coeficiente de correlação entre as classificações binárias observadas e previstas e retorna um valor entre -1 e +1. Um coeficiente de +1 representa uma predição perfeita, 0 não melhor do que a predição aleatória e -1 indica total assimetria entre o dado predito e o observado.

Por último, a AUC - Curva ROC é uma medida de desempenho para os problemas de classificação em várias configurações de limite. ROC é uma curva de probabilidade e AUC representa o grau ou medida de separabilidade, que diz o quanto o modelo é capaz de diferenciar as classes. Quanto maior a AUC, melhor será o modelo em prever 0s como 0s e 1s como 1s.

Tabela 2. Fórmulas de algumas das métricas para a análise das classificações, tendo como TP , TN , FP , e FN a quantidade de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente.

Acurácia	Precisão (P)	Revocação (R)	F_{β} -score	MCC
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{(1+\beta^2) \cdot R \cdot P}{\beta^2 \cdot R + P}$	$\frac{TP \cdot FN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}$

5. Resultados

Os resultados obtidos com os modelos e metodologia de avaliação propostos são discutidos nessa seção. Primeiramente apresentamos o desempenho dos classificadores que compõem nosso arcabouço de aprendizado de máquina. Na sequência, discutimos sobre as características mais relevantes para as classificações de perfis de usuários.

5.1. Desempenho de Classificadores

Na Tabela 3, apresentamos os resultados para a avaliação de desempenho dos classificadores treinados com as três formas de pré-processamento de dados descritas na Seção 4.1, respectivamente: nenhum pré-processamento, subamostragem e sobreamostragem. Para cada uma dessas formas, apresentamos os resultados via oito métricas de desempenho. Nota-se que a acurácia é a métrica com maior desempenho (acima de 90%) para todos os classificadores. Contudo, é importante observar que os dados analisados têm classes altamente desbalanceadas, ou seja, a classe de usuários profissionais corresponde a menos de 1% do total de usuários. Nesse caso, as métricas que capturam a especificidade dos classificadores em identificar corretamente a classe minoritária (usuários profissionais) ganham importância, e essas métricas são o foco da avaliação de desempenho nessa seção.

Primeiramente, discutimos os resultados obtidos sem nenhuma técnica para tratar o desbalanceamento dos dados. Observamos que dessa forma há uma tendência de alta acurácia mesmo que o modelo apenas retorne a classe majoritária. No caso abordado aqui, os classificadores identificam menos usuários profissionais, que é a nossa classe positiva. Além disso, pode haver um viés na precisão, pois apenas instâncias claramente positivas serão classificadas como tal. Nesse caso, como pode ser observado na Tabela 3, a precisão atinge valores superiores a 52% para a maioria dos classificadores ao passo que a revocação fica abaixo de 41%, e foram identificados menos de 50 usuários profissionais (TP menor que 42%). Consideramos que o arcabouço dessa forma tem especificidade baixa porque tende a classificar usuários como comuns, que é a vasta maioria.

Ao treinarmos os modelos com subamostragem, observamos que os classificadores em geral aumentam o desempenho para as métricas revocação e AUC-ROC e, por consequência, aumentam o acerto para usuários profissionais. Definimos o AUC-ROC como critério do *Comitê Soft* para o voto dos classificadores por apresentar uma alta relação com a probabilidade de acerto dos usuários profissionais (TP) sem impactar no acerto dos usuários comuns (TN). Com a melhora dessa métrica, todos os classificadores tem TP acima de 37%, alcançando até 94% em alguns casos, mas mantendo o TN acima de 90%. Dessa forma, um maior AUC-ROC representa maiores chances de identificar usuários profissionais e comuns. Quanto ao desempenho dos classificadores individuais,

Tabela 3. Desempenho dos classificadores treinados com três formas de pré-processamento de dados: nenhum, subamostragem e sobreamostragem.

TB	Classificador	Acurácia	Precisão	Revocação	F_1 -score	F_2 -score	MCC	TP	TN	AUC-ROC
Nenhuma	KNN	0,994066	0,629032	0,333333	0,435754	0,367925	0,455296	39 (33,3%)	16880 (99,8%)	0,665986
	Árvore de Decisão	0,993948	0,594595	0,376068	0,460733	0,405904	0,470043	44 (37,6%)	16873(99,8%)	0,687147
	Floresta Aleatória	0,994771	0,705882	0,410256	0,518919	0,447761	0,535810	48 (41,0%)	16883 (99,9%)	0,704537
	Regressão Logística	0,987838	0,255435	0,401709	0,312292	0,360429	0,314490	47 (40,2%)	16766 (99,2%)	0,696802
	SVM Linear	0,994418	0,689655	0,341880	0,457143	0,380228	0,483216	40 (34,2%)	16885 (99,9%)	0,670408
	SVM Gaussiano	0,994477	0,744681	0,299145	0,426829	0,339806	0,469891	35 (29,9%)	16891 (99,9%)	0,649218
	SVM Sigmoides	0,993420	0,528736	0,393162	0,450980	0,414414	0,452723	46 (39,3%)	16862 (99,8%)	0,695368
	<i>Comitê Hard</i>	0,994536	0,676471	0,393162	0,497297	0,429104	0,513265	46 (39,3%)	16881 (99,9%)	0,695930
<i>Comitê Soft</i>	0,994536	0,662162	0,418803	0,513089	0,452029	0,524081	49 (41,8%)	16878 (99,8%)	0,711382	
Subamostragem	KNN	0,929553	0,079316	0,871795	0,145403	0,290764	0,250653	102 (92,3%)	15719 (92,9%)	0,900874
	Árvore de Decisão	0,936369	0,092749	0,940171	0,168841	0,332527	0,284443	110(94,0%)	15827 (93,6%)	0,938257
	Floresta Aleatória	0,943772	0,104520	0,948718	0,188295	0,362745	0,304867	111 (94,8%)	15952 (94,4%)	0,946228
	Regressão Logística	0,903349	0,061424	0,914530	0,115116	0,242081	0,222928	107 (91,1%)	15268(90,3%)	0,908901
	SVM Linear	0,968214	0,161342	0,863248	0,271871	0,461609	0,365314	101 (86,3%)	16378 (96,9%)	0,916094
	SVM Gaussiano	0,992421	0,449153	0,452991	0,451064	0,452218	0,447252	53 (45,3%)	16838 (99,6%)	0,724573
	SVM Sigmoides	0,993302	0,517647	0,376068	0,435644	0,397830	0,437955	44 (37,6%)	16862 (99,8%)	0,686821
	<i>Comitê Hard</i>	0,962397	0,145183	0,914530	0,250585	0,443983	0,356124	107 (91,5%)	16273 (96,3%)	0,938629
<i>Comitê Soft</i>	0,945777	0,107212	0,940172	0,192476	0,368139	0,307571	110 (94,0%)	15987 (94,6%)	0,965530	
Sobreamostragem	KNN	0,927791	0,081955	0,931624	0,150657	0,303115	0,264562	109 (93,2%)	15682 (92,7%)	0,929694
	Árvore de Decisão	0,937720	0,095973	0,957265	0,174455	0,342508	0,292573	112(95,7%)	15848 (93,7%)	0,947425
	Floresta Aleatória	0,942832	0,104436	0,965812	0,188490	0,364516	0,307654	113 (96,5%)	15934 (94,3%)	0,954242
	Regressão Logística	0,742832	0,024341	0,931624	0,047443	0,110190	0,126317	109 (93,1%)	12534 (74,1%)	0,836575
	SVM Linear	0,967509	0,158307	0,863248	0,267550	0,456600	0,361686	101 (86,3%)	16366 (96,8%)	0,915739
	SVM Gaussiano	0,992362	0,444444	0,444444	0,444444	0,444444	0,440599	52 (44,4%)	16838 (99,6%)	0,720299
	SVM Sigmoides	0,993243	0,511628	0,376068	0,433498	0,397112	0,435345	44 (37,6%)	16861 (99,8%)	0,686792
	<i>Comitê Hard</i>	0,962456	0,145380	0,914530	0,250879	0,444352	0,356379	107 (91,5%)	16274 (96,3%)	0,938659
<i>Comitê Soft</i>	0,947062	0,110338	0,948718	0,197684	0,376526	0,313850	111(94,6%)	16008 (94,7%)	0,952651	

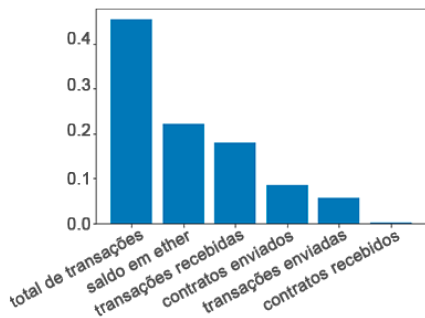
a Floresta Aleatória tem o melhor desempenho com AUC-ROC e TP , acima de 94%. Como consequência, este classificador é o que mais impacta na decisão do Comitê Soft.

O treinamento dos modelos com sobreamostragem também permite o aumento da revocação, o que favorece a identificação de usuários da classe profissional. Os resultados para TP se mantiveram acima de 37%, alcançando até 95% em alguns classificadores, com TN acima de 92% para a maioria dos classificadores (a exceção é a Regressão Logística com TN de 74%). A Floresta Aleatória se destacou novamente como classificador com o melhor desempenho individual. Contudo, o treinamento com sobreamostragem permitiu que os outros classificadores também tenham votos na decisão do *Comitê Soft*. Neste caso, os SVM Gaussiano e Sigmoides, que obtiveram os piores desempenhos para TN e AUC-ROC, foram os únicos excluídos deste comitê.

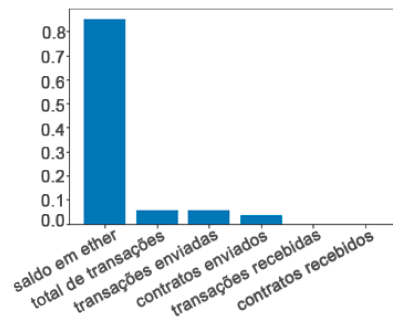
Em suma, observamos que ambos os treinamentos com subamostragem e sobreamostragem atingem uma alta especificidade e aumentam o número de verdadeiros positivos (TP) na classificação. A subamostragem se mostrou a melhor opção de treinamento aqui, pois obteve o maior desempenho para o AUC-ROC, que é a métrica principal de decisão do Comitê Soft.

5.2. Características mais relevantes para a Classificação

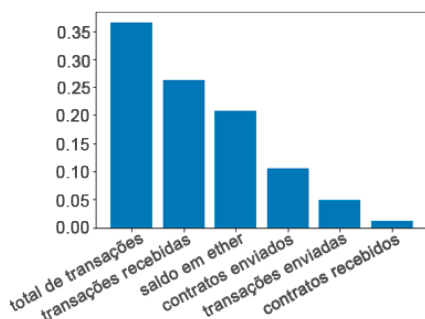
Na Figura 1, mostramos as características mais relevantes para a tomada de decisão do *Comitê Soft*. Especificamente, mostramos nas figuras (a) e (c) a ordem de relevância das características nos modelos de Floresta Aleatória que teve voto decisivo (único) para o arcabouço treinado com subamostragem e ainda participa juntamente a outros modelos no comitê treinado com sobreamostragem. Por sua vez, nas figura (b) e (d), mostramos relevância das características para o classificador Árvore de Decisão, que teve voto



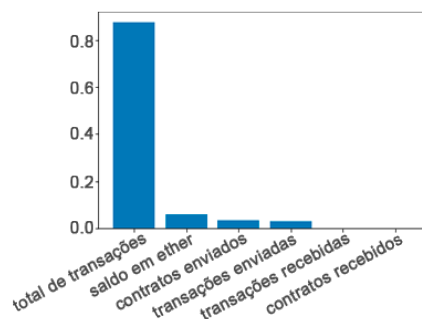
(a) Floresta Aleatória (subamostragem)



(b) Árvore de Decisão (subamostragem)



(c) Floresta Aleatória (sobreamostragem)



(d) Árvore de Decisão (sobreamostragem)

Figura 1. Relevância de características para a classificação de usuários considerando dois classificadores com bons resultados de AUC-ROC: Floresta Aleatória e Árvore de Decisão em treinamentos com subamostragem nas figuras (a) e (b) e com sobreamostragem nas figuras (c) e (d) respectivamente.

juntamente a outros classificadores para classificação no arcabouço treinado com sobreamostragem.

Como pode-se observar, a Floresta Aleatória classifica os perfis de usuários considerando principalmente o total de transações. Essa característica tem aproximadamente 45% e 36% de relevância para a decisão desse tipo de modelo treinado, respectivamente, com subamostragem e sobreamostragem. Contudo, outras características ainda são exploradas com algum grau de relevância por esse classificador, como mostram as Figuras 1 (a) e (c). Apenas o número de transações recebidas de contratos (contratos recebidos) é irrelevante para a classificação com Floresta Aleatória. Entendemos que o uso de informações extraídas de um maior número de características torna a Floresta Aleatória o tipo de modelo com o melhor desempenho geral mostrado na Tabela 3 com voto no *Comitê Soft* treinado com subamostragem e sobreamostragem.

A Árvore de Decisão, que é um dos classificadores com voto no *Comitê Soft* treinado com sobreamostragem, realiza classificações com relevância quase totalmente atribuída ao (i) saldo em ether (característica com aproximadamente 80% de relevância)

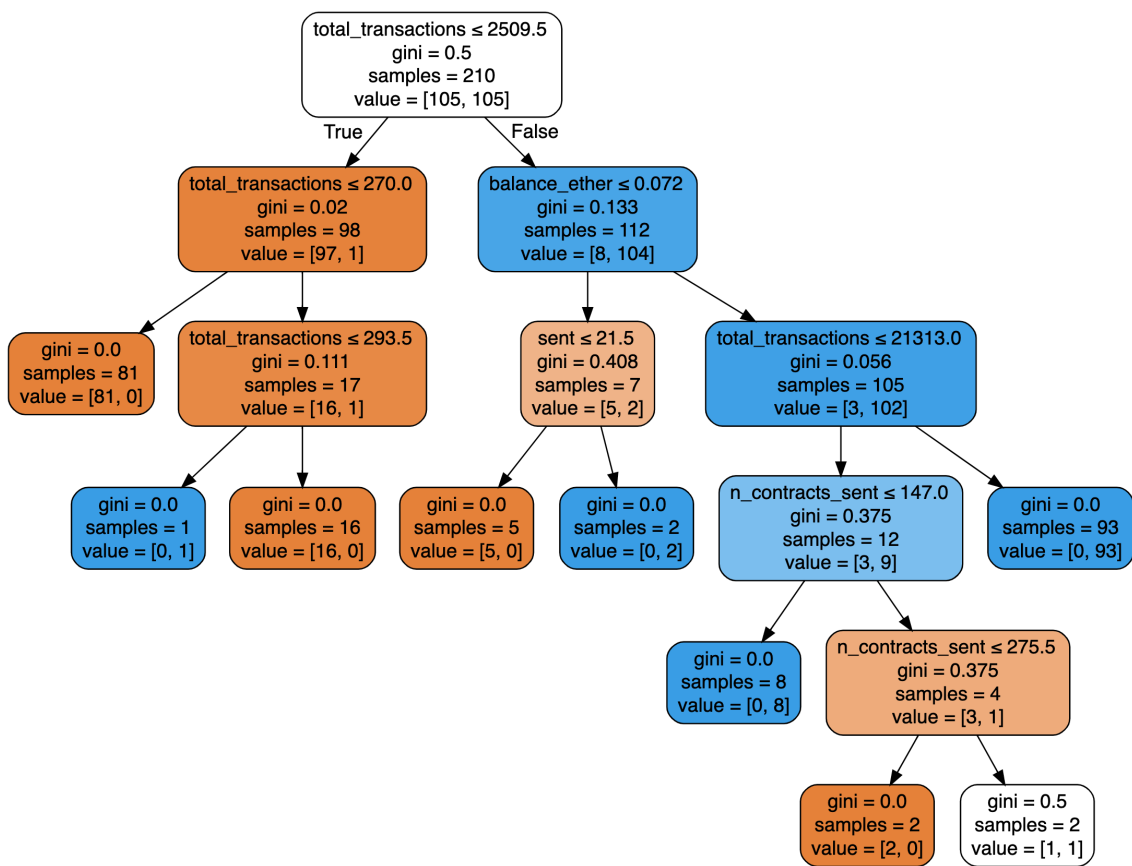


Figura 2. Árvore de decisão obtida com os dados de treinamento dos modelos usando sobreamostragem.

quando há subamostragem, e (ii) total de transações (também com aproximadamente 80% de relevância) usando sobreamostragem, como mostra as Figuras 1 (b) e (d). As demais características são praticamente irrelevantes para a Árvore de Decisão e para entendermos esse efeito investigamos adicionalmente como as características são usadas no processo de decisão. A Figura. 2 apresenta uma árvore de decisão obtida quando os dados do treinamento dos modelos são balanceados por sobreamostragem. Cada nó da árvore contém: (a) sua condição associada a uma característica dentre as seis utilizadas; (b) o índice Gini, que é o critério de divisão utilizado; (c) o número de amostras de contas de usuários antes da divisão em 2 conjuntos, um para cada aresta; e (d) o número de amostras de cada classe, que iniciam iguais e são separadas ao longo das divisões. As arestas da árvore são os resultados das condições (verdadeiro ou falso) e as cores do nó representam a porção de contas de usuários comuns (azul) e profissionais (laranja).

A árvore da Figura. 2 é capaz de identificar ambas as classes e o número de transações recentes é sua característica mais importante, utilizada na maioria dos nós para decisões. O índice Gini captura a variação da característica em um nó: quanto mais próximo a zero mais homogêneo é a distribuição dos dados no nó e mais confiável é a decisão. Logo, a árvore da Figura. 2 confirma nossas observações da relevância do total de transações. Depois de dividir os dados de acordo com o gás usado, o valor do recurso também pode ser considerado relevante.

Em suma, observamos que o total de transações é a mais relevante para os modelos com melhor desempenho dentre as seis características avaliadas. Além disso, o treinamento dos modelos com subamostragem permite que a Floresta de Árvore seja o modelo com voto único para o *Comitê Soft*. Esse modelo reduz adequadamente o peso da característica total de transações recentes na tomada de decisão para que outras características sejam também exploradas, obtendo o melhor desempenho geral.

6. Conclusões

Neste trabalho, apresentamos a aplicação de modelos de aprendizado de máquina para analisar perfis de usuários no Ethereum os classificando como usuários (contas) profissionais ou comuns. Esses modelos podem ser adequadamente treinados usando características de transações coletadas de transações publicamente disponíveis na blockchain do Ethereum, e nossas avaliações mostram tais modelos podem alcançar alta acuracidade e um bom compromisso entre precisão e revocação para a classificação de usuários profissionais e comuns. Contudo, é importante o uso de estratégias de balanceamento de dados para treinar modelos de aprendizado de máquina adequadamente dado que fração de usuários profissionais na rede é consideravelmente menor que a de usuários comuns.

Para trabalhos futuros, pretendemos melhorar o desempenho do arcabouço proposto. Além disso, planejamos a adoção de mais classes dentro dos perfis profissionais, assim como nos perfis de usuários comuns.

Referências

- Aspembitova, A. T., Feng, L., and Chew, L. Y. (2021). Behavioral structure of users in cryptocurrency market. *Plos one*, 16(1):e0242600.
- Bartoletti, M., Carta, S., Cimoli, T., and Saia, R. (2020). Dissecting ponzi schemes on ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277.
- Boughorbel, S., Jarray, F., and El-Anbari, M. (2017). Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6).
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- Chen, T., Li, Z., Zhu, Y., Chen, J., Luo, X., Lui, J. C.-S., Lin, X., and Zhang, X. (2020). Understanding ethereum via graph analysis. *ACM Transactions on Internet Technology (TOIT)*, 20(2):1–32.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Dietterich, T. G. et al. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125.
- Hu, T., Liu, X., Chen, T., Zhang, X., Huang, X., Niu, W., Lu, J., Zhou, K., and Liu, Y. (2021). Transaction-based classification and detection approach for ethereum smart contract. *Information Processing & Management*, 58(2):102462.
- Mascarenhas, J. Z., Vieira, A. B., and Ziviani, A. (2018). Análise da rede de transações do ethereum. In *Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain-SBRC 2018)*, volume 1. SBC.

- Motamed, A. P. and Bahrak, B. (2019). Quantitative analysis of cryptocurrencies transaction graph. *Applied Network Science*, 4(1):131.
- Norvill, R., Pontiveros, B. B. F., State, R., Awan, I., and Cullen, A. (2017). Automated labeling of unknown contracts in ethereum. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE.
- Oliveira, V. C., Valadares, J. A., Sousa, J. E. A., Bernardino, H. S., Vieira, A. B., Villela, S. M., and Gonçalves, G. D. (2021). Analyzing Transaction Confirmation in Ethereum Using Machine Learning Techniques. *SIGMETRICS Perform. Eval. Rev.*, 48(4):12–15.
- Payette, J., Schwager, S., and Murphy, J. (2017). Characterizing the ethereum address space. <https://pdfs.semanticscholar.org/db53/a0281ea25f0041ca4aa812be5c9013f33f26.pdf>.
- Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.
- Rebello, G., Hu, Y., Thilakarathna, K., Batista, G., Seneviratne, A., and Duarte, O. C. M. B. (2020). Melhorando a acurácia da detecção de lavagem de dinheiro na rede bitcoin. In *Anais do XXXVIII SBRC*, pages 728–741. SBC.
- Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674.
- Schär, F. (2020). Decentralized finance: On blockchain-and smart contract-based financial markets. *Available at SSRN 3571335*.
- Singh, H. J. and Hafid, A. S. (2019). Prediction of transaction confirmation time in ethereum blockchain using machine learning. In *International Congress on Blockchain and Applications*, pages 126–133. Springer.
- Sousa, J. E. A., Oliveira, V., Valadares, J., Gonçalves, G. D., Villela, S. M., Bernardino Soares, H., and Vieira, A. B. (2020). An analysis of the fees and pending time correlation in ethereum. *International Journal of Network Management*, page e2113.
- Sousa, J. E. A., Oliveira, V. C., Valadares, J. A., Vieira, A. B., Bernardino, H. S., Villela, S. M., and Gonçalves, G. D. (2021). Fighting Under-price DoS Attack in Ethereum with Machine Learning Techniques. *SIGMETRICS Perform. Eval. Rev.*, 48(4):24–27.
- Wang, M., Ichijo, H., and Xiao, B. (2020). Cryptocurrency address clustering and labeling. *arXiv preprint arXiv:2003.13399*.
- Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32.
- Wright, R. E. (1995). *Reading and understanding multivariate statistics*, chapter Logistic regression, page 217–244. American Psychological Association.
- Wu, S. X., Wu, Z., Chen, S., Li, G., and Zhang, S. (2021). Community detection in blockchain social networks. *Journal of Communications and Information Networks*, 6(1):59–71.
- Xu, G., Guo, B., Su, C., Zheng, X., Liang, K., Wong, D. S., and Wang, H. (2020). Am i eclipsed? a smart detector of eclipse attacks for ethereum. *Computers & Security*, 88:101604.