Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Mapeamento do Comportamento de Usuários para Identificar Contas Profissionais no Ethereum Utilizando Aprendizado Semissupervisionado

Júlia Almeida Valadares

JUIZ DE FORA
JANEIRO, 2023

Mapeamento do Comportamento de Usuários para Identificar Contas Profissionais no Ethereum Utilizando Aprendizado Semissupervisionado

Júlia Almeida Valadares

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciências da Computação

Bacharelado em Ciência da Computação

Orientador: Alex Borges Vieira

Coorientador: Heder Soares Bernardino

JUIZ DE FORA

JANEIRO, 2023

Mapeamento do Comportamento de Usuários para Identificar Contas Profissionais no Ethereum Utilizando Aprendizado Semissupervisionado

Júlia Almeida Valadares

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Alex Borges Vieira
Doutor

Heder Soares Bernardino
Doutor

Edelberto Franco Silva
Doutor

Luaciano Jerez Chaves
Doutor

JUIZ DE FORA

11 DE JANEIRO, 2023

Aos meus amigos e irmãos.

Aos pais, pelo apoio e sustento.

Resumo

Ethereum é uma das maiores plataformas blockchain atualmente e que se tornou um ambiente de negócios digital. Esta plataforma permite transações descentralizadas entre usuários anônimos. Assim, o desenvolvimento de métodos para identificar os comportamentos dos usuários e mantê-los anônimos pode potencializar os negócios nesta plataforma. Neste trabalho, propomos uma combinação de abordagens de aprendizado de máquina de diferentes categorias, supervisionado, não supervisionado e semissupervisionado, para mapear os comportamentos das contas dos usuários e identificar usuários com atividades profissionais no Ethereum. Essas são tarefas desafiadoras devido à pequena fração de dados rotulados publicamente referentes às contas de usuários que prestam serviços nesta plataforma, como troca, pagamento e entretenimento, entre os usuários de comportamento mais casual. Inicialmente, usamos técnicas de aprendizado não supervisionado para agrupar as contas dos usuários não rotulados e identificar um conjunto deles com comportamento casual. Como resultado, obtém-se um conjunto de dados contendo instâncias rotuladas (casuais ou profissionais) e não rotuladas. Métodos de aprendizado semi-supervisionado são então aplicados (i) para gerar modelos que classificam os comportamentos das contas em casuais ou profissionais e (ii) para descobrir contas com comportamentos profissionais entre as não rotuladas. Experimentos computacionais foram conduzidos, e os resultados obtidos pelo procedimento proposto são comparados aos obtidos por técnicas de aprendizado supervisionado da literatura. A proposta superou os da literatura e atingiu valores superiores a 95% para acurácia, precisão, reconvocação, F_β-scores, MCC e AUC-ROC.

Palavras-chave: Criptomoeda, transações, Ethereum, aprendizado de Máquina, Blockchain.

Abstract

Ethereum is one of the largest blockchain platforms currently that has become a digital business environment. This platform allows for decentralized transactions between anonymous users. Thus, the development of methods to identify users' behaviors and keep them anonymous can potentially leverage business on this platform. In this work, we propose a combination of machine learning approaches of different categories, namely, unsupervised and semi-supervised, to map the behaviors of users' owned accounts and identify users with professional activities in Ethereum. These are challenging tasks due to the small fraction of publicly labeled data referring to users' accounts that provide services on this platform, such as exchange, payment, and entertainment, among most casual behavior users. Initially, we use unsupervised learning techniques to cluster the unlabeled users' accounts and to identify a set of them with casual behavior. As an outcome, a dataset containing labeled (casual or professional) and unlabeled instances is obtained. Semi-supervised learning methods are then applied (i) to generate models that classify accounts' behaviors into casual or professional ones and (ii) to discover accounts with professional behaviors among the unlabeled ones. Computational experiments were conducted, and the results obtained by the proposed procedure are compared to those achieved by supervised learning techniques from the literature. The proposal outperformed those from the literature and reached values higher than 95% for the Acurácia, Precisão, revocação, F_β-scores, MCC, and AUC-ROC.

Keywords: Cryptocurrency, Blockchain, Ethereum, Transactions, Machine Learning.

Agradecimentos

A todos os meus parentes, pelo encorajamento e apoio.

Aos professores Alex, Heder, Saulo e Glauber pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

"Lembra que o sono é sagrado e alimenta de horizontes o tempo acordado de viver".

Beto Guedes (Amor de Índio)

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

DCC Departamento de Ciência da Computução

UFJF Universidade Federal de Juiz de Fora

1 Introdução

Este trabalho tem como objetivo entender o comportamento dos perfis de usuários dentro da rede Blockchain do Ethereum. Para isso se propõe o uso de técnicas de aprendizado de máquina para classificação e balanceamento dos dados.

1.1 Apresentação do tema

Ethereum é uma plataforma popular baseada na tecnologia blockchain para negociação de ativos digitais (CHEN et al., 2020). Desde o seu início em 2014, o Ethereum registrou mais de 1 milhão de transações e tem mais de $273 bilhões em valor de mercado, tornando-se a segunda maior plataforma atualmente.[1] Ele estende o Bitcoin, a primeira plataforma blockchain bem conhecida, permitindo o controle de ativos digitais por meio de códigos de programa denominados contratos inteligentes. Ao fornecer ao usuário várias possibilidades para determinar como usar seus ativos e transferi-los para qualquer lugar, o Ethereum contribuiu muito para aplicativos blockchain disruptivos para produtores e consumidores de serviços (XU; WEBER; STAPLES, 2019).

Plataformas baseadas em Blockchain, como Ethereum, geralmente são projetadas para permitir transações entre usuários anônimos (ZHANG; XUE; LIU, 2019). Isso é possível graças aos esquemas de assinatura digital, um componente da tecnologia blockchain. A partir deste componente, são criadas duas chaves para funcionar como endereços de usuários na plataforma: par de chaves pública e privada. Os usuários se identificam por sua chave pública e usam a chave privada secreta para autenticar as transações que emitem em favor de outros usuários. Os pares de chaves podem ser gerados quantos usuários quiserem e são realmente usados como pseudônimos dos usuários, fornecendo identidades anônimas nas plataformas blockchain.

Como a maioria das plataformas Blockchain promove o anonimato, identificar perfis de usuários mantendo suas identidades anônimas é um desafio.Especificamente no

[1] ⟨https://www.binance.com/en/markets⟩

Ethereum, está se tornando popular para os usuários realizarem atividades associadas à prestação de serviços, como troca de ativos digitais ou investimentos, trabalhando assim como usuários profissionais. Por outro lado, a maioria dos usuários tem comportamento normal e executa a transferência casual de ativos digitais, ou seja, usuários comuns. Tanto os usuários profissionais quanto os casuais são contas Ethereum operadas por humanos, diferente dos contratos inteligentes, que são contas programadas para executar transações.

1.2 Problema

O ecossistema blockchain pode se beneficiar da identificação desses perfis de usuários profissionais e casuais. De fato, esse recurso tem um alto potencial para alavancar negócios em aplicativos descentralizados no Ethereum. Por exemplo, aplicativos financeiros descentralizados (*DeFis*) (SCHÄR, 2020) que oferecem serviços para pagamentos, empréstimos, doações ou *royalties* para tokens não fungíveis (CASALE-BRUNET et al., 2021) podem usar análise de perfil buscar clientes adequados para seus serviços e analisar riscos de crédito (HARVEY; RAMACHANDRAN; SANTORO, 2021).

Alguns serviços da web já estão se movendo para o desenvolvimento de ferramentas de perfil de usuário para redes blockchain. Entre elas estão as iniciativas de Etherscan (ETHERSCAN, 2022) e DApp.com (DappReview, 2019), que pretendem identificar as atividades das principais contas e tokens em redes blockchain. No entanto, a identificação que essas ferramentas fornecem é baseada principalmente manualmente,[2] dependendo, em alguns casos, das solicitações dos usuários para seu perfil identificação e confirmação de suas atividades a partir de seus contatos de rede. Portanto, métodos para identificação automática de perfis são muito úteis para tais serviços, bem como para o ecossistema de aplicações DeFi.

1.3 Justificativa

Analisar o comportamento do usuário com base em transações registradas publicamente em blockchains é uma área que atualmente está atraindo a atenção de pesquisadores. O

[2] ⟨https://info.etherscan.com/public-name-tags-labels⟩

aprendizado de máquina tem sido usado para identificar perfis de usuários em plataformas blockchain com foco no risco de investimento, por exemplo, traders otimistas e pessimistas (ASPEMBITOVA; FENG; CHEW, 2021), e para classificar contratos inteligentes de acordo com suas semelhanças de código (NORVILL et al., 2017). Um conjunto de pesquisas caracteriza o comportamento do usuário por meio da teoria dos grafos, que modela as propriedades estruturais das redes blockchain (CHEN et al., 2020; WU et al., 2021b; MOTAMED; BAHRAK, 2019; THARANI et al., 2021). Também vale a pena notar que um volume considerável de trabalho usou modelos de aprendizado de máquina para segurança, identificação de fraudes e ataques a redes blockchain, com foco em comportamentos de usuários suspeitos. (XU et al., 2020; BARTOLETTI et al., 2020; HU et al., 2021).

Todos esses trabalhos trazem contribuições relevantes para o campo. No entanto, eles não abordam perfis de usuários para comportamento profissional ou casual dos usuários em plataformas blockchain.

Atualmente, classificar perfis de usuários em profissionais e casuais não é trivial e requer o conhecimento de especialistas em análise de transações de criptomoedas (CHEN et al., 2020). Investigamos ainda mais esses perfis profissionais no Etherscan e descobrimos uma pequena parcela de usuários que autoidentificam suas contas como serviços (por exemplo, Defi, Gaming, Charity) associados a vários contratos inteligentes. Essas informações sobre usuários no Ethereum nos trouxeram duas percepções. Primeiro, um sistema automático para identificar o perfil do usuário é um desafio devido ao pequeno número de dados rotulados publicamente e usuários profissionais no Ethereum para desenvolver e treinar um modelo de previsão. Outro ponto a ser observado é que um conjunto de dados baseado em instâncias rotuladas é desbalanceado, ou seja, um número muito pequeno de amostras de classes profissionais, dificultando o desenvolvimento de modelos de aprendizado de máquina para suportar sistemas automáticos.

1.4 Arcabouço

Para lidar com a identificação de perfis de usuários no Ethereum, especificamente usuários profissionais e casuais, propomos neste artigo a utilização de uma combinação de diferentes técnicas de aprendizado de máquina para mapear os comportamentos dos usuários e

identificar seus perfis. Mais profundamente, combinamos métodos de aprendizado não supervisionados e semisupervisionados para mapear os comportamentos dos usuários e identificar perfis profissionais no Ethereum. Para isso, desenvolvemos um procedimento que inicia agrupando os usuários não rotulados, separando-os dos usuários identificados. Esta etapa cria um conjunto de dados com instâncias rotuladas (casuais ou profissionais) e não rotuladas. Em seguida, abordagens de aprendizado semissupervisionado são aplicadas (i) para gerar modelos para classificar usuários em casuais e profissionais e (ii) descobrir usuários com comportamentos profissionais entre as contas não rotuladas.

Vale ressaltar que aplicamos a classificação não supervisionada para indicar instâncias com características semelhantes aos usuários profissionais rotulados e que poderiam ser rotulados também nesta categoria. A abordagem semi-supervisionada é então proposta para melhorar a classificação de perfis profissionais e casuais por meio de modelos de aprendizado de máquina supervisionados.

Para avaliar o método proposto, coletamos várias transações no blockchain Ethereum usando a API do Etherscan, um explorador de blockchain muito popular.

Em seguida, criamos um grande conjunto de dados compilando características sobre usuários extraídos de transações disponíveis publicamente no blockchain Ethereum com alguns usuários profissionais rotulados que reunimos manualmente com a API Etherscan.

Nossos resultados experimentais mostram alto desempenho para classificar ambos os perfis de usuários no Ethereum. Na verdade, um de nossos modelos mais propostos tem alta precisão (acima de 99%) junto com Precisão, revocação, F_1-score, F_2-score, coeficiente de correlação de Matthews e AUC-ROC maior que 97% , 95%, 96%, 96%, 96% e 97%, respectivamente.

Em resumo, nossas contribuições são as seguintes:

i) um conjunto de dados de contas de usuários no Ethereum identificados por sua chave pública anônima com sete recursos representativos além do rótulo profissional para 142 usuários que realizam serviços no Ethereum e autoidentificaram sua atividade no popular blockchain explorador Etherscan;

ii) um procedimento que combina abordagens de aprendizado de máquina de diferentes

categorias, ou seja, não supervisionado, supervisionado e semisupervisionado, para identificar automaticamente o comportamento de usuários profissionais, levando em consideração um grande número de instâncias não rotuladas em sistemas blockchain devido ao anonimato de usuários e

iii) avaliação do procedimento proposto usando várias abordagens de aprendizado de máquina supervisionado de última geração.

2 Trabalhos relacionados

Modelos de aprendizado de máquina têm sido amplamente utilizados para uma variedade de análises aplicadas sobre transações publicamente disponíveis em plataformas blockchain como Ethereum e Bitcoin. Nesta seção, fornecemos uma visão geral da literatura, que cobre essa área de pesquisa, também conhecida como *chain analytics*. Primeiro, discutimos os esforços de pesquisa em relação à classificação do comportamento do usuário usando modelos de aprendizado de máquina. Em seguida, estendemos nossa discussão ao desempenho do sistema e aos aplicativos de segurança, aproveitando essas abordagens.

2.1 Classificação do comportamento do usuário

Aplicativos descentralizados (dApps) no blockchain Ethereum têm suportado diferentes tipos de negócios em áreas como finanças, jogos, jogos de azar e saúde, como exemplos analisados por (WU et al., 2021a). Esses dApps aproveitam os efeitos de rede da combinação de vários serviços e trazem mais participação de mercado dos aplicativos tradicionais centralizados. Por exemplo, Finanças Descentralizadas (DeFi) (SCHÄR, 2020)) procura construir e combinar vários blocos de construção de dApps em produtos financeiros sofisticados com custos de transação minimizados e valor maximizado para os usuários. Por exemplo, qualquer cliente pode simplesmente pagar a taxa fixa para usar um contrato inteligente e se beneficiar das inovações do ecossistema DeFi dApps como stablecoins, tokens fungíveis e não fungíveis, custódia de fundos, ajustes de fornecimento de tokens, recompensas apostadas e outros benefícios financeiros implementados por meio de contratos inteligentes, descritos por (HARVEY; RAMACHANDRAN; SANTORO, 2021). Assim, os dApps têm atraído diversos usuários interessados em explorar os benefícios de um novo ecossistema de serviços descentralizados para atividades profissionais ou casuais.

O crescimento de dApps e DeFis exigirá métodos para extrair características dos usuários com base em seu comportamento, i.e., a partir de suas transações. Dessa forma, algumas pesquisas abordaram os desafios de desenvolver tais métodos. Recentemente,

(ASPEMBITOVA; FENG; CHEW, 2021) desenvolveram modelos para identificar perfis de usuários em Bitcoin e Ethereum. Os autores identificaram usuários nos perfis de investimento tradicionais, especificamente quatro grupos: traders pessimistas, otimistas, negativos e positivos. (NORVILL et al., 2017) propuseram técnicas para classificar contratos inteligentes no Ethereum em grupos de similaridade usando recursos extraídos do código do contrato inteligente. Através dos resultados de agrupamentos, perceberam que é possível identificar o propósito de um contrato inteligente.

Em (ASPEMBITOVA; FENG; CHEW, 2021; NORVILL et al., 2017) os autores aplicaram modelos através de uma abordagem não supervisionada, em que algoritmos de agrupamento indicavam grupos de usuários e suas características eram analisadas. Diferente deles, contamos com classes de usuários definidas por especialistas e desenvolvemos modelos para aprender tal tarefa de classificação. Nossa abordagem pode apoiar pesquisas de perfis de usuários de interesse em blockchains.

Em relação à análise do comportamento dos usuários em plataformas blockchain, é importante considerar o relacionamento entre as contas dos usuários com base em teoria dos grafos. Uma categorização de contas de usuários Ethereum em gráficos multifluxo foi proposta por (CHEN et al., 2020). Eles consideram três tipos de relacionamentos: usuário para usuário, um usuário para um contrato inteligente e um contrato inteligente para um contrato inteligente. (WU et al., 2021b) analisou as diferenças estruturais entre as redes Bitcoin e Ethereum para identificar as comunidades de usuários. Para o Bitcoin, eles propuseram análise espectral de agrupamento, enquanto um gráfico bipartido que modela a relação entre contratos inteligentes e contas foi proposto para o Ethereum.

(MOTAMED; BAHRAK, 2019) reconstruíram o gráfico temporal das transações entre contas Ethereum para analisar a variação na densidade do gráfico em função do preço da criptomoeda em dólares. (THARANI et al., 2021) propuseram técnicas de visualização de gráficos para facilitar a análise de fluxos de dinheiro suspeitos em criptomoedas como Ramsonware e transações marcadas como ilícitas. Nesses trabalhos, foram encontrados diversos recursos de rede baseados em transações entre contas e seus padrões que emergem de redes blockchain modeladas como grafos. Nosso trabalho, por sua vez, visa uma aplicação desses recursos para a classificação de perfis de usuários.

2.2 Desempenho e segurança de sistema

Um esforço de pesquisa relevante tem sido dedicado aos aspectos de segurança das plataformas. (XU et al., 2020) desenvolveram um modelo de classificação de floresta aleatória para distinguir o tráfego normal do tráfego malicioso que ocorre em um tipo de ataque identificado como Eclipse. Pirâmides financeiras em contratos inteligentes foram investigadas por meio de métodos estatísticos para identificar fraudes em mercados financeiros em (BARTOLETTI et al., 2020).

Um modelo de rede neural LSTM (Long Short Term Memory) foi desenvolvido por (HU et al., 2021) para identificar seis tipos de contratos inteligentes mais casuais no Ethereum, incluindo contratos de alto risco, tendo como entrada um conjunto de transações associadas aos contratos. Ao contrário desses trabalhos, aplicamos técnicas de aprendizado de máquina para identificar provedores de serviços e diferenciá-los de contas de usuários casuais, o que intuitivamente é uma tarefa diferente da detecção de comportamentos suspeitos e fraudulentos.

Outros trabalhos aplicaram com sucesso modelos de aprendizado de máquina para analisar o desempenho da plataformo Ethereum. Por exemplo, (SINGH; HAFID, 2019) propôs modelos de aprendizado supervisionado para prever o tempo de confirmação de uma transação, explorando o impacto de classes de dados desbalanceadas no treinamento e teste dos modelos. Em linha com esses esforços de pesquisa, aprimoramos esses modelos para outras questões importantes sobre a confirmação de transações no Ethereum em nossos trabalhos anteriores. Em (SOUSA et al., 2020), investigamos a relação entre a taxa que os usuários pagam pelas transações e seu tempo de confirmação por meio de métricas de correlação e aprendizado não supervisionado, enquanto métodos para prever falhas de processamento de contratos por mineradores Ethereum via comitês classificadores com aprendizado supervisionado foram desenvolvidos em (OLIVEIRA et al., 2021).

Esses trabalhos são ortogonais a este trabalho, pois exploram recursos de transação em modelos de aprendizado de máquina, mas não abordam diretamente a identificação ou classificação de perfis de usuários.

3 Metodologia

Nesta seção, apresentamos a metodologia proposta neste trabalho para gerar os modelos de aprendizado de máquina que classificam usuários no Ethereum como perfil comum ou profissional. Para isso, descrevemos quais os paradigmas de aprendizado de máquina e os modelos utilizados, como foi realizada a coleta e o tratamento dos dados e, por fim, como se deu a classificação.

3.1 Paradigmas de aprendizado de máquina

As abordagens de aprendizado de máquina podem ser classificadas em aprendizado supervisionado, não supervisionado, por reforço (SUTTON; BARTO et al., 1998) e semissupervisionado (CHAPELLE; SCHOLKOPF; ZIEN, 2009) . Os processos de aprendizado de máquina com objetivo de classificação (supervisionado e semissupervisionado) são divididos em dois passos: treino e teste. Na fase de treino, as amostras nos dados de treinamento são tomadas como entrada, nas quais os recursos são aprendidos pelo algoritmo de aprendizado e constroem o modelo de aprendizado. Já na fase de teste, o modelo de aprendizado usa o mecanismo de execução para fazer a previsão para os dados. Os dados marcados são a saída do modelo de aprendizado que fornece a previsão final ou dados classificados.

O aprendizado supervisionado é usado para descrever tarefas de previsão em que o objetivo é prever ou classificar um resultado específico de interesse. Os vários algoritmos geram uma função que mapeia as entradas para as saídas desejadas, ou seja, cada exemplo consiste em um par de entrada (objeto conhecido) e saída (objeto esperado). A função gerada pode ser usada para classificar ou predizer entradas não conhecidas. O objetivo principal desse tipo de aprendizado é construir um estimador capaz de prever o rótulo de um objeto dado pelo conjunto de características (NASTESKI, 2017).

Os métodos de aprendizado de máquina não supervisionados são particularmente úteis em tarefas de descrição pois visam encontrar relacionamentos em uma estrutura de

dados sem ter um resultado mensurável em diversas métricas. Essa categoria de aprendizado de máquina é chamada de não supervisionada porque não possui uma variável de resposta que possa supervisionar a análise (GARETH et al., 2013). O objetivo do aprendizado não supervisionado é encontrar decisões de agrupamento diretamente nas propriedades de um determinado conjunto de amostras sem o uso de amostras de treinamento, ao contrário do aprendizado supervisionado. O algoritmo mais comum e mais usado do aprendizado não supervisionado é o k-means.

O aprendizado semissupervisionado combina exemplos rotulados e não rotulados para gerar uma função ou classificador apropriado (BENGIO et al., 2008). Esse tipo de aprendizado é utilizado quando existem poucas amostras de dados rotuladas, uma vez que o aprendizado supervisionado precisa de uma grande quantidade de dados com rótulos para treino. Assim, os algoritmos do semissupervisionado são usados para classificar uma parte dos dados não rotulados. Existem diversos algoritmos usados na abordagem semi-supervisionada como COP-k-means, assemble e máquinas de vetores de suporte transdutiva.

Os algoritmos usados neste trabalho são descritos na seção a seguir.

3.2 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; LANDGREBE, 1991), Floresta Aleatória (RF) (BREIMAN, 1996), Regressão Logística (LG) (WRIGHT, 1995), Máquinas de Vetor de Suporte (SVM) (CORTES; VAPNIK, 1995) e Comitês de Classificadores (DIETTERICH et al., 2002) representando o aprendizado supervisionado. Para o não supervisionado foi usado o K-means (MACQUEEN, 1967) e para o semissupervisionado a Máquina de Vetor de Suporte Transdutiva (TSVM).

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. 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.

Os Comitês de Classificação combinam modelos de aprendizado de máquina 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.

O K-means utiliza, principalmente, técnica de otimização que possui como objetivo particionar o conjunto de dados em k grupos. Cada dado pertence ao grupo com a média mais próxima de um dos centros (centróide).

Ao contrário das SVMs tradicionais, as SVMs transdutivas (TSVM) levam em consideração os pontos de dados não rotulados. Além disso, o objetivo do SVM Trans-

dutivo é diferente do SVM regular, agora ao resolver o problema de otimização, não precisamos apenas encontrar o hiperplano de separação, mas também os rótulos para os pontos de dados não rotulados.

3.3 Metodologia proposta

Os dados contêm rótulos/valores a serem previstos por métodos de aprendizado supervisionado, enquanto nenhuma informação supervisionada está presente em problemas não supervisionados. Além disso, um problema semi-supervisionado aparece quando se tem dados rotulados e não rotulados. Nas seções a seguir, é discutida a proposta de combinar abordagens de aprendizado de máquina de diferentes categorias. Por fim, são definidas as métricas utilizadas para avaliar os modelos.

Uma estratégia de aprendizado supervisionado para classificar usuários profissionais e casuais foi proposta em nosso trabalho anterior (VALADARES et al., 2021). Nesse caso, todos os dados não rotulados foram considerados usuários casuais. O primeiro fluxograma na Figura 3.1 ilustra essa abordagem. Embora essa abordagem tenha fornecido uma estratégia inicial para lidar com os dados não rotulados, pode-se observar que os usuários profissionais que não fornecem sua categoria explicitamente são considerados usuários casuais.

Para evitar esse ruído nos rótulos, propomos aqui uma análise dos dados não rotulados via aprendizado não supervisionado para identificar aquelas instâncias que, de fato, representam usuários casuais. Para isso, uma abordagem não supervisionada é aplicada aos dados não rotulados e os grupos de usuários são obtidos. Esses conjuntos são caracterizados, e um comportamento regular pode ser observado nos usuários de um deles. Os usuários desse grupo são então identificados como casuais. Pode-se observar que outros grupos além do que contém os usuários casuais podem ser gerados. Essas instâncias permanecem sem rótulo, pois não temos informações neste momento para identificá-los como usuários profissionais ou casuais. O segundo e terceiro fluxogramas na Figura 3.1 contêm este procedimento.

Aqui, usamos aprendizado não supervisionado para fornecer um conjunto de dados supervisionado composto por usuários profissionais e casuais. Com esse resultado,

uma abordagem de aprendizado supervisionado pode ser aplicada e modelos para identificar usuários profissionais podem ser criados. Essa abordagem é ilustrada no segundo fluxograma da Figura 3.1.

É importante destacar que o procedimento apresentado anteriormente para gerar modelos para classificar os usuários em profissionais ou casuais contém uma desvantagem: alguns dados permanecem sem rótulo e são descartados. No entanto, esses dados podem conter informações para melhorar os modelos de classificação. Assim, propomos aqui o uso do aprendizado semissupervisionado para gerar modelos considerando um conjunto de dados contendo dados de usuários identificados como profissionais e casuais, bem como, aqueles não identificados (não rotulados). Este procedimento proposto é ilustrado no terceiro fluxograma da Figura 3.1.

Por fim, o conjunto de dados pode apresentar classes desbalanceadas. Por exemplo, espera-se que o número de usuários casuais seja muito maior do que o de profissionais. Assim, estratégias para balancear os dados de treinamento devem ser aplicadas. Esta etapa é apresentada em todos os fluxogramas mostrados na Figura 3.1 usando um quadrado tracejada.

3.4 Descrição dos dados

A base de dados utilizada possui 20.857.783 transações coletadas de 15 de Abril até 26 de Julho de 2019. Selecionamos 2.519.711 transações dessa base de forma aleatória e uniforme ao longo desse período de coleta para as nossas análises. As transações selecionadas correspondem a 3.2% da quantidade de transações que o Ethereum foi capaz de processar dentro do período de coletada da base de dados.

Complementamos essa base de dados coletando informações extras de todas as contas envolvidas nas transações selecionadas. 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 plataformo Ethereum, embora múltiplas contas possam pertencer a um mesmo usuário (pessoa ou organização). Ao todo, foram 17.020 contas (usuários) utilizados para as análises.

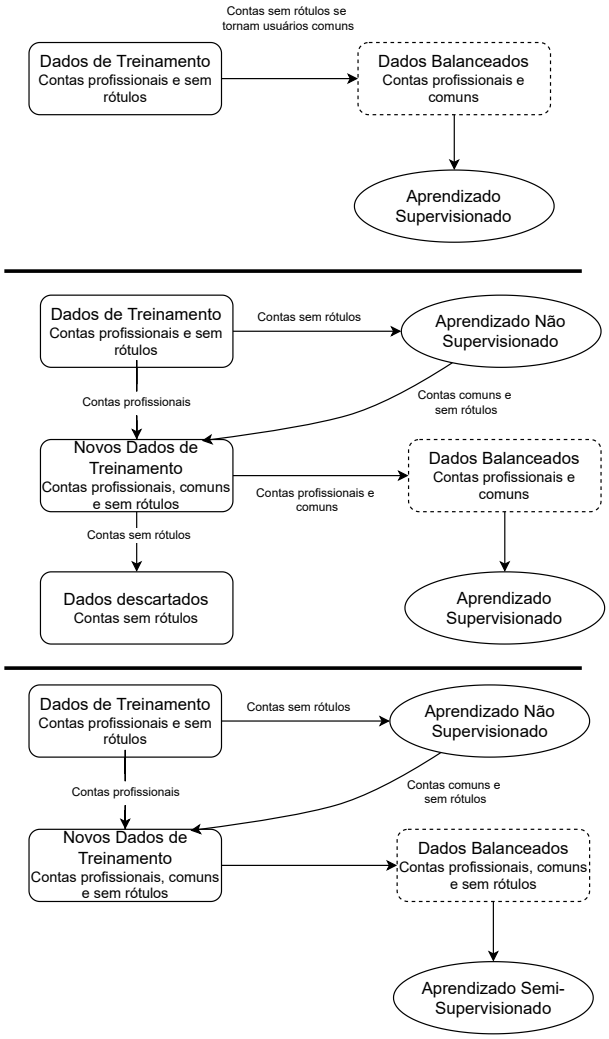


Figura 3.1: Fluxograma da abordagem a partir da literatura (primeira) e das propostas que combinam diferentes categorias de aprendizagem para identificar relatos profissionais.

Para cada uma dessas contas, extraímos informações sobre as movimentações com base nas transações selecionadas. 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.

Por último, coletamos a rotulação/classe para as contas utilizando a API Etherchain [3]. Em geral, APIs de informações sobre Ethereum como Etherchain e Etherscan rotulam contas que tem atividade profissional na plataforma e precisam de ampla publicidade. Esse processo tipicamente requer que o usuário proprietário da conta informe

[3] ⟨https://etherscan.io⟩

Tabela 3.1: Características extraídas das contas do Ethereum.

Atributo	Descrição
Conta do usuário	Identificador da conta do usuário (chave pública) definida na plataformo 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 rementente 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.

uma categoria em que suas atividades se enquadram e um nome para identificar a conta publicamente. 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.

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 Subamostragem*) que busca remover aleatoriamente as amostras da classe majoritária para igualar a classe minoritária;

- Sobreamostragem aleatória (*Random sobreamostragem*) que envolve suplementar os dados de treinamento com várias cópias de algumas das classes minoritárias;

- SMOTE que gera novas instâncias de amostras minoritárias existentes que são fornecidas como entrada.

O melhor modelo na análise de desempenho foi usado para classificar os perfis de usuários. Além disso, analisamos tipos de modelos (como RF e DCT) para extrair informações sobre qual o caminho o modelo tomou para decidir a classificação.

3.5 Métricas de desempenho

Com o intuito de analisar o desempenho dos métodos e estratégias de aprendizado de máquina, usamos métricas como Acurácia, Precisão, revocação, 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 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 é uma métrica que quantifica, de forma proporcional, do número de previsões positivas corretas feitas sobre o número total de previsões 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 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; JARRAY; EL-ANBARI, 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 preditas 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.

Além disso, temos as métricas de verdadeiros positivos (TP) e verdadeiros negativos (TN), que mostram a quantidade da amostra de dados que obteve uma predição correta.

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 3.2: 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
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{(1+\beta^2)\cdot R\cdot P}{\beta^2\cdot R+P}$	$\frac{TP\cdot FN-FP\cdot FN}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}}$

4 Resultados alcançados

Os resultados obtidos com os modelos propostos e são discutidos nesta seção. Eles são apresentados nas subseções a seguir, nas quais são mostrados os resultados da abordagem da literatura usando apenas aprendizado supervisionado, e aqueles em que uma técnica não supervisionada é usada para identificar contas de usuários casuais a partir de dados não rotulados.

Os experimentos computacionais foram realizados em Python com a biblioteca *Scikit-learn*[4]. Os valores de parâmetro padrão do Scikit-learn foram usados para as técnicas de aprendizado de máquina. Todos os dados e código fonte estão disponíveis publicamente.[5]

4.1 Aprendizado supervisionado

Começamos discutindo os resultados de nossa abordagem proposta em trabalhos anteriores, ou seja, aprendizado supervisionado aplicado a dados não rotulados tratados como usuários casuais (VALADARES et al., 2021). Consideramos esses resultados como a linha de base para os resultados descritos nas próximas subseções. Na Tabela 4.1, mostramos o desempenho de classificadores treinados com os três métodos de balanceamento de dados descritos anteriormente, a saber, subamostragem, sobreamostragem e SMOTE além da opção com dados sem pré-processamento. Para cada um desses tipos, apresentamos resultados por meio de oito métricas de desempenho. Como se pode observar, a acurácia é a métrica com maior desempenho para quase todos os classificadores. No entanto, é importante destacar que os dados analisados possuem classes altamente desbalanceadas, ou seja, a maior parte das amostras representa a classe de usuários comuns (mias de 99%) e os usuários profissionais representam menos de 1% do total de usuários. Nesse caso, métricas que capturam a especificidade dos classificadores em identificar corretamente a classe minoritária (usuários profissionais) ganham importância, e essas métricas são o foco

[4] ⟨https://scikit-learn.org/⟩
[5] ⟨http://netlab.ice.ufjf.br/ethereum⟩

de nossa avaliação de desempenho nesta seção.

Tabela 4.1: Desempenho de classificadores sem pré-processamento e com subamostragem, sobreamostragem e SMOTE.

Pre	Classificador	Acurácia	Precisão	Revocação	F_1-score	F_2-score	MCC	TP	TN	AUC-ROC
Sem pré-processamento	KNN	0.992244	0.580645	0.253521	0.352941	0.285714	0.380442	36 (25.35%)	16852 (99.85%)	0.625990
	Árvore de Decisão	0.992891	0.606061	0.422535	0.497925	0.449775	0.502638	60 (42.25%)	16839 (99.77%)	0.710112
	Floresta aleatória	**0.993596**	0.708861	0.394366	**0.506787**	0.432767	**0.525918**	56 (39.44%)	16855 (99.86%)	0.696502
	Regressão logísticas	0.991716	0.666667	0.014085	0.027586	0.017513	0.096098	2 (1.41%)	16877 (99.99%)	0.507013
	SVM Linear	0.991892	**1.000000**	0.028169	0.054795	0.034965	0.167154	4 (2.82%)	**16878 (100.0%)**	0.514085
	SVM Gaussiano	0.991539	0.479167	0.161972	0.242105	0.186688	0.275275	23 (16.20%)	16853 (99.85%)	0.580245
	SVM Sigmóide	0.987955	0.200000	0.147887	0.170040	0.156018	0.166011	21 (14.79%)	16794 (99.50%)	0.571455
	Comitê rígido	0.992303	0.689655	0.140845	0.233918	0.167504	0.309450	20 (14.08%)	16869 (99.95%)	0.570156
	Comitê ponderado	0.992891	0.800000	0.197183	0.316384	0.232172	0.395087	28 (19.72%)	16871 (99.96%)	0.598384
Subamostragem	KNN	0.887250	0.059494	0.845070	0.111163	0.232108	0.206193	120 (84.51%)	14981 (88.76%)	0.866338
	Árvore de Decisão	0.913572	0.080227	0.894366	0.147246	0.295212	0.253074	127 (89.44%)	15422 (91.37%)	0.904050
	Floresta aleatória	0.927732	0.096439	**0.915493**	0.174497	0.339248	0.284050	**130 (91.55%)**	15660 (92.78%)	**0.921664**
	Regressão logísticas	0.987603	0.319372	0.429577	0.366366	0.401845	0.364286	61 (42.96%)	16748 (99.23%)	0.710938
	SVM Linear	0.991833	0.530612	0.183099	0.272251	0.210697	0.308527	26 (18.31%)	16855 (99.86%)	0.590868
	SVM Gaussiano	0.982256	0.245223	0.542254	0.337719	0.436508	0.357036	77 (54.23%)	16641 (98.60%)	0.764106
	SVM Sigmóide	0.984019	0.259259	0.492958	0.339806	0.417661	0.350234	70 (49.30%)	16678 (98.82%)	0.740554
	Comitê rígido	0.982961	0.258170	0.556338	0.352679	0.451945	0.371633	79 (55.63%)	16651 (98.66%)	0.771444
	Comitê ponderado	0.931257	0.097809	0.880282	0.176056	0.338570	0.280252	125 (88.03%)	15725 (93.17%)	0.905984
Sobreamostragem	KNN	0.984195	0.256705	0.471831	0.332506	0.404101	0.340750	67 (47.18%)	16684 (98.85%)	0.730168
	Árvore de Decisão	0.951821	0.126652	0.809859	0.219048	0.389566	0.308774	115 (80.99%)	16085 (95.30%)	0.881437
	Floresta aleatória	0.992538	0.575758	0.401408	0.473029	0.427286	0.477155	57 (40.14%)	16836 (99.75%)	0.699460
	Regressão logísticas	0.979377	0.222812	0.591549	0.323699	0.444444	0.354873	84 (59.15%)	16585 (98.26%)	0.787095
	SVM Linear	0.981199	0.241279	0.584507	0.341564	0.455044	0.367811	83 (58.45%)	16617 (98.45%)	0.784522
	SVM Gaussiano	0.954054	0.112591	0.654930	0.192149	0.333572	0.258842	93 (65.49%)	16145 (95.66%)	0.805750
	SVM Sigmóide	0.888719	0.059799	0.838028	0.111632	0.232604	0.205843	119 (83.80%)	15007 (88.91%)	0.863587
	Comitê rígido	0.978966	0.230000	0.647887	0.339483	0.475207	0.378051	92 (64.79%)	16570 (98.18%)	0.814819
	Comitê ponderado	0.972679	0.196998	0.739437	0.311111	0.476839	0.372920	105 (73.94%)	16450 (97.46%)	0.857039
SMOTE	KNN	0.956580	0.121673	0.676056	0.206230	0.353721	0.274704	96 (67.61%)	16185 (95.89%)	0.817498
	Árvore de Decisão	0.945006	0.110020	0.788732	0.193103	0.353090	0.281944	112 (78.87%)	15972 (94.63%)	0.867527
	Floresta aleatória	0.981375	0.253521	0.633803	0.362173	**0.487541**	0.393412	90 (63.38%)	16613 (98.43%)	0.809051
	Regressão logísticas	0.978555	0.216285	0.598592	0.317757	0.442248	0.351468	85 (59.86%)	16570 (98.18%)	0.790171
	SVM Linear	0.980082	0.228650	0.584507	0.328713	0.445757	0.357551	83 (58.45%)	16598 (98.34%)	0.783959
	SVM Gaussiano	0.946475	0.107252	0.739437	0.187333	0.339367	0.268638	105 (73.94%)	16004 (94.82%)	0.843827
	SVM Sigmóide	0.909871	0.071956	0.823944	0.132353	0.266636	0.227291	117 (82.39%)	15369 (91.06%)	0.867269
	Comitê rígido	0.964982	0.164201	0.781690	0.271394	0.446141	0.348480	111 (70.42%)	16313 (97.44%)	0.874107
	Comitê ponderado	0.972150	0.187970	0.704225	0.296736	0.454545	0.354730	100 (78.17%)	16446 (96.65%)	0.839315

Primeiramente, discutimos os resultados obtidos sem manipular os dados desbalanceados. Observamos uma tendência de alta precisão mesmo que o modelo retorne apenas a classe majoritária. Nesse caso, os classificadores identificam menos usuários profissionais, que é nossa classe positiva. Além disso, pode haver um viés de precisão, pois apenas instâncias claramente positivas são classificadas corretamente. Neste caso, como podemos ver na Tabela 4.1, a precisão atinge valores maiores que 52% para a maioria dos classificadores enquanto o revocação fica abaixo de 41%, e menos de 50 usuários são identificados na classe de usuários profissionais (TP inferior a 42%). Consideramos que nosso arcabouço, dessa forma, possui baixa especificidade, pois tende a classificar os usuários como casuais, que é a classe majoritária.

Agora nos concentramos no pré-processamento de dados de subamostragem e observamos que os classificadores geralmente aumentam o desempenho das métricas de revocação e AUC-ROC e, como resultado, aumentam os acertos para a classe de usuários

profissionais. Assim, definimos AUC-ROC como o critério do comitê ponderado (soft), pois esta métrica apresenta uma alta relação com a probabilidade de acertos para a classe de usuários profissionais (TP) sem impactar os acertos para a classe de usuários casuais (TN). Com o aprimoramento desta métrica, todos os classificadores possuem TP acima de 37%, chegando até 94% em alguns casos, mantendo o TN acima de 90%. Dessa forma, observamos que uma AUC-ROC maior representa maiores chances de identificar ambas as classes. Quanto ao desempenho dos classificadores individualmente, Floresta aleatória tem o melhor desempenho com AUC-ROC e TP acima de 94%. Consequentemente, este classificador tem o maior impacto na decisão do Comitê ponderado.

O pré-processamento dos dados com sobreamostragem também melhora a revocação, o que favorece a identificação da classe de usuários profissionais. Os resultados para TP permanecem acima de 37%, chegando até 95% em alguns classificadores, com TN acima de 92% para a maioria dos classificadores (exceção é a Regressão Logística com TN de 74%). Floresta aleatória supera novamente outros classificadores com o melhor desempenho individual. No entanto, o sobreamostragem também permite que diferentes classificadores melhorem seus resultados e participem do Comitê ponderado. Neste caso, os SVM Gaussiano e Sigmóide, que obtiveram os piores desempenhos para TN e AUC-ROC, são os únicos excluídos deste comitê.

Em resumo, observamos que tanto as técnicas de pré-processamento de subamostragem quanto de sobreamostragem atingem alta especificidade e melhoram o número de verdadeiros positivos (TP) na classificação. A subamostragem, no entanto, tem se mostrado a melhor técnica de pré-processamento, pois alcançou o maior desempenho para o AUC-ROC, que é a principal métrica de decisão do Comitê ponderado.

4.2 Aprendizado não supervisionado

Aqui discutimos os resultados alcançados usando o aprendizado não supervisionado (a técnica de clusterização k-means é adotada neste trabalho) nos dados não rotulados para identificar grupos de usuários. Esta nova informação de agrupamento é utilizada para melhorar o desempenho dos classificadores uma vez que foi identificado um grande grupo que consideramos ser usuários casuais, e melhorar o desempenho dos classificadores utilizando

este novo grupo e dados rotulados como profissionais.

Dessa forma, primeiro exploramos quantos grupos podem ser distinguidos entre dados não rotulados. Para isso, aumentamos gradualmente o número de grupos (parâmetro k em k-médias) e observamos suas características, principalmente o número de instâncias de dados (ou seja, contas) em cada grupo. Na Tabela 4.2, apresentamos todos os grupos gerados, onde cada linha apresenta o número de instâncias nos grupos para diferentes valores de k, com $k = 2, \ldots, 10$. Como se pode notar, um grande grupo mantém aproximadamente seu número de contas à medida que separamos o conjunto de dados em mais grupos. Paramos em dez grupos à medida que o número de contas no maior grupo se torna estável.

Além disso, observamos que as contas que saíram do grupo maior apresentaram características bem distintas deste grupo. Assim, consideramos a interseção entre contas do maior grupo de cada valor de k, ou seja, 16576 contas, como usuários casuais no Ethereum a ser explorado na análise a seguir. As 302 contas restantes são mantidas sem rótulo e descartadas na abordagem atual.

Tabela 4.2: Grupos k-means.

k	Grupos									
	0	1	2	3	4	5	6	7	8	9
2	16836	42	–	–	–	–	–	–	–	–
3	16797	79	2	–	–	–	–	–	–	–
4	16796	76	4	2	–	–	–	–	–	–
5	16640	199	33	4	2	–	–	–	–	–
6	16640	200	28	4	4	2	–	–	–	–
7	16640	199	31	4	2	1	1	–	–	–
8	16638	188	28	17	3	2	1	1	–	–
9	16583	196	50	25	17	3	2	1	1	–
10	16580	212	37	24	18	3	1	1	1	1
$\cap$	16576									

Com um conjunto de dados composto por contas de usuários profissionais e casuais, avaliamos a classificação do perfil dos usuários. Na tabela 4.3, mostramos o desempenho de classificadores treinados com três tipos de pré-processamento de dados e oito métricas de desempenho, como na seção anterior, para comparar com esse aprendizado supervisionado como parâmetro.

A precisão é novamente a métrica com o melhor desempenho para todos os classificadores. No entanto, as métricas que capturam a especificidade dos classificadores em identificar corretamente a classe minoritária (usuários profissionais) são as mais importantes.

Nos resultados obtidos sem manipulação dos dados desbalanceados, a precisão chega a 100% visto que alguns classificadores atingem todos os usuários casuais. Ao mesmo tempo, a média da revocação é de cerca de 43%, e a maioria dos TP está acima de 50%. Este resultado mostra melhorias na especificidade em comparação com a primeira abordagem de aprendizagem.

A vantagem do aprendizado não supervisionado proposto é melhor observada nos resultados com pré-processamento de dados. A subamostragem aumenta o desempenho das métricas de revocação e AUC-ROC e, como resultado, aumenta os acertos para a classe de usuários profissionais. Todos os classificadores atingem TP entre 46% e 100% mantendo TN acima de 91% (exceto para SVM Linear). Floresta aleatória tem o melhor desempenho com AUC-ROC e TP atingindo 83% e 100%, respectivamente. O uso de sobreamostragem também melhora a precisão e a revocação simultaneamente, e os resultados para TP permanecem acima de 68%, chegando a 95% na Árvore de Decisão, com TN acima de 97% para a maioria dos classificadores. Por sua vez, SMOTE segue a mesma tendência, e os resultados para TP permanecem acima de 68%, chegando até 93%, mantendo TN acima de 95% para todos os classificadores. Floresta aleatória supera os demais classificadores com 97% em AUC-ROC.

Em suma, os resultados apresentados na Tabela 4.3 mostram que as contas removidas impactam no desempenho dos classificadores. A identificação do maior grupo de contas com comportamento semelhante via k-means (abordagem não supervisionada) melhorou o desempenho dos classificadores em geral. Tomando o modelo de floresta aleatória com os dados balanceados através do SMOTE como exemplo, TP e AUC-ROC aumentaram 58% e 28% em relação à abordagem anterior.

Tabela 4.3: Desempenho de classificadores sem pré-processamento e com subamostragem, sobreamostragem e SMOTE em dados sem contas não rotuladas.

Pre	Classificador	Acurácia	Precisão	Revocação	F_1-score	F_2-score	MCC	TP	TN	AUC-ROC
Sem pré-processamento	KNN	0.995813	0.973684	0.521127	0.678899	0.574534	0.710751	74 (52.11%)	16574 (99.99%)	0.760503
	Árvore de Decisão	0.997009	0.960000	0.676056	0.793388	0.718563	0.804308	96 (67.61%)	16572 (99.98%)	0.837908
	Floresta Aleatória	**0.997248**	**1.000000**	0.676056	**0.806723**	**0.722892**	**0.821088**	96 (67.61%)	**16576 (100.0%)**	0.838028
	Regressão logística	0.991686	**1.000000**	0.021127	0.041379	0.026270	0.144745	3 (2.11%)	**16576 (100.0%)**	0.510563
	SVM Linear	0.992523	**1.000000**	0.119718	0.213836	0.145299	0.344706	17 (11.97%)	**16576 (100.0%)**	0.559859
	SVM Gaussiano	0.995992	0.941176	0.563380	0.704846	0.612557	0.726538	80 (56.34%)	16571 (99.97%)	0.781539
	SVM Sigmóide	0.988276	0.245283	0.183099	0.209677	0.192878	0.206112	26 (18.31%)	16496 (99.52%)	0.589136
	Comitê rígido	0.995992	**1.000000**	0.528169	0.691244	0.583204	0.725288	75 (52.82%)	**16576 (100.0%)**	0.764085
	Comitê ponderado	0.996112	**1.000000**	0.542254	0.703196	0.596899	0.734939	77 (54.23%)	**16576 (100.0%)**	0.771127
Subamostragem	KNN	0.904355	0.070713	0.845070	0.130506	0.264901	0.227885	120 (84.51%)	14999 (90.49%)	0.874966
	Árvore de Decisão	0.909858	0.079482	0.908451	0.146176	0.294386	0.253648	129 (90.85%)	15082 (90.99%)	0.909160
	Floresta Aleatória	0.941680	0.118939	**0.915493**	0.210526	0.391331	0.318307	**130 (91.55%)**	15613 (94.19%)	**0.928698**
	Regressão logística	0.995155	**1.000000**	0.429577	0.600985	0.484897	0.653826	61 (42.96%)	**16576 (100.0%)**	0.714789
	SVM Linear	0.993121	**1.000000**	0.190141	0.319527	0.226891	0.434547	27 (19.01%)	**16576 (100.0%)**	0.595070
	SVM Gaussiano	0.993779	0.658333	0.556338	0.603053	0.574128	0.602101	79 (55.63%)	16535 (99.75%)	0.776932
	SVM Sigmóide	0.995514	0.946667	0.500000	0.654378	0.552100	0.686275	71 (50.00%)	16572 (99.98%)	0.749879
	Comitê rígido	0.994856	0.780000	0.549296	0.644628	0.583832	0.652154	78 (54.93%)	16554 (99.87%)	0.773984
	Comitê ponderado	0.947960	0.127049	0.873239	0.221825	0.401554	0.321674	124 (87.32%)	15724 (94.86%)	0.910920
Subreamostragem	KNN	0.990848	0.472362	0.661972	0.551320	0.612777	0.554791	94 (66.20%)	16471 (99.37%)	0.827819
	Árvore de Decisão	0.966922	0.181395	0.823944	0.297332	0.482275	0.377425	117 (82.39%)	16048 (96.81%)	0.896045
	Floresta Aleatória	0.996710	0.914286	0.676056	0.777328	0.713224	0.784692	96 (67.61%)	16567 (99.95%)	0.837757
	Regressão logística	0.990549	0.458333	0.619718	0.526946	0.578947	0.528354	88 (61.97%)	16472 (99.37%)	0.806722
	SVM Linear	0.990908	0.472826	0.612676	0.533742	0.578457	0.533764	87 (61.27%)	16479 (99.41%)	0.803412
	SVM Gaussiano	0.945149	0.108190	0.753521	0.189213	0.343610	0.272412	107 (75.35%)	15694 (94.68%)	0.850156
	SVM Sigmóide	0.904056	0.074505	0.901408	0.137634	0.279965	0.243436	128 (90.14%)	14986 (90.41%)	0.902743
	Comitê rígido	0.990011	0.444444	0.704225	0.544959	0.630517	0.554854	100 (70.42%)	16451 (99.25%)	0.848342
	Comitê ponderado	0.983910	0.315942	0.767606	0.447639	0.596933	0.486315	109 (76.76%)	16340 (98.58%)	0.876684
SMOTE	KNN	0.967520	0.177134	0.774648	0.288336	0.462574	0.360939	110 (77.46%)	16065 (96.92%)	0.871910
	Árvore de Decisão	0.962316	0.165753	0.852113	0.277523	0.466102	0.366169	121 (85.21%)	15967 (96.33%)	0.907686
	Floresta Aleatória	0.988037	0.395683	0.774648	0.523810	0.650118	0.548650	110 (77.46%)	16408 (98.99%)	0.882256
	Regressão logística	0.989951	0.435644	0.619718	0.511628	0.571429	0.514759	88 (61.97%)	16462 (99.31%)	0.806420
	SVM Linear	0.990370	0.450777	0.612676	0.519403	0.571616	0.520845	87 (61.27%)	16470 (99.36%)	0.803141
	SVM Gaussiano	0.944431	0.112315	0.802817	0.197061	0.360076	0.287627	114 (80.28%)	15675 (94.56%)	0.874231
	SVM Sigmóide	0.890059	0.064234	0.880282	0.119732	0.248608	0.220456	125 (88.03%)	14755 (89.01%)	0.885212
	Comitê rígido	0.976791	0.243750	0.823944	0.376206	0.558206	0.440753	117 (78.87%)	16213 (98.39%)	0.901022
	Comitê ponderado	0.982235	0.295515	0.788732	0.429942	0.591341	0.476344	112 (82.39%)	16309 (97.81%)	0.886312

4.3 Aprendizado semi-supervisionado

Por fim, discutimos os resultados obtidos usando o aprendizado semissupervisionado com um conjunto de dados composto por contas de usuários profissionais e casuais, além de instâncias não rotuladas. Aqui, usamos os dados não rotulados que foram removidos do conjunto de dados de treinamento na seção anterior. Dessa forma, é possível ver como essas contas podem melhorar o desempenho dos classificadores para o problema aqui resolvido. Todas as 302 contas foram adicionadas à parte de treinamento de nossa estrutura. Os resultados apresentados na Tabela 4.4 são maiores em quase todas as métricas quando comparados com os obtidos pelas outras duas abordagens (seus resultados estão nas tabelas 4.1 e 4.2) . A exceção é a precisão que, como comentado anteriormente, não é o foco da discussão para o problema atual devido ao desequilíbrio dos dados. O uso de dados não balanceados para classificação ainda gera resultados com os valores mais baixos de TP, AUC-ROC e Precisão. No entanto, quando a técnica semi-supervisionada é combinada

com técnicas de balanceamento, as métricas TP e AUC-ROC melhoram.

Ao subamostrar dados balanceados, os valores de TP estão acima de 46% e chegaram a 100% com Floresta Aleatória. A métrica AUC-ROC apresenta o maior valor igual a 96,8%, enquanto os maiores valores usando as abordagens anteriores são 91% e 92%. A Floresta Aleatória conseguiu identificar todas as contas profissionais (TP igual a 100%) e encontrou resultados com uma AUC-ROC igual a 96,8%. Além disso, acurácia e evocação apresentam valores satisfatórios, respectivamente, iguais a 93,6% e 100%. Ela também tem o melhor desempenho com Subreamostragem e SMOTE, atingindo 95,8% e 97,8%, respectivamente, em TP e AUC-ROC.

Tabela 4.4: Desempenho de classificadores sem pré-processamento e com subamostragem, subreamostragem e SMOTE aplicando a técnica semi-supervisionada.

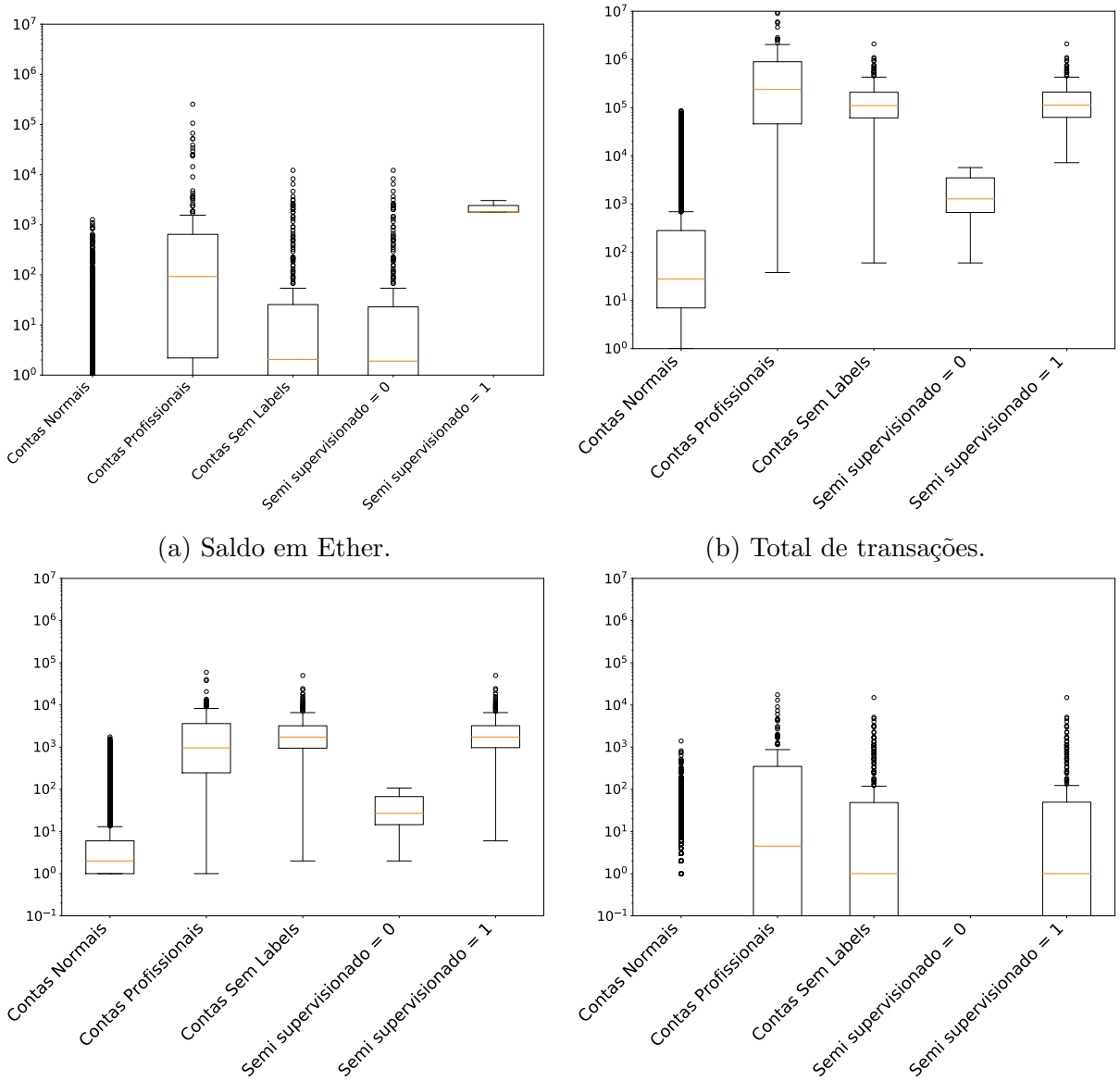
Pre	Classificador	Acurácia	Precisão	Revocação	F_1-score	F_2-score	MCC	TP	TN	AUC-ROC
Sem pré-processamento	KNN	0.995813	0.973684	0.521127	0.678899	0.574534	0.710751	74 (52.11%)	16574 (99.99%)	0.760503
	Árvore de Decisão	0.997129	0.960784	0.690141	0.803279	0.731343	0.813030	98 (69.01%)	16572 (99.98%)	0.844950
	Floresta Aleatória	0.997248	**1.000000**	0.676056	0.806723	0.722892	0.821088	96 (67.61%)	**16576 (100.0%)**	0.838028
	Regressão Logística	0.991686	**1.000000**	0.021127	0.041379	0.026270	0.144745	3 (2.11%)	**16576 (100.0%)**	0.510563
	SVM Linear	0.992523	**1.000000**	0.119718	0.213836	0.145299	0.344706	17 (11.97%)	**16576 (100.0%)**	0.559859
	SVM Gaussiano	0.995992	0.941176	0.563380	0.704846	0.612557	0.726538	80 (56.34%)	16571 (99.97%)	0.781539
	SVM Sigmóide	0.988276	0.245283	0.183099	0.209677	0.192878	0.206112	26 (18.31%)	16496 (99.52%)	0.589136
	Comitê rígido	0.995992	**1.000000**	0.528169	0.691244	0.583204	0.725288	75 (52.82%)	**16576 (100.0%)**	0.764085
	Comitê ponderado	0.996172	**1.000000**	0.549296	0.709091	0.603715	0.739718	78 (54.93%)	**16576 (100.0%)**	0.774648
Subamostragem	KNN	0.915421	0.083770	0.901408	0.153293	0.305344	0.260159	128 (90.14%)	15176 (91.55%)	0.908474
	Árvore de Decisão	0.928759	0.102952	0.957746	0.185919	0.359979	0.301491	136 (95.77%)	15391 (92.85%)	0.943129
	Floresta Aleatória	0.936476	0.117940	**1.000000**	0.210996	0.400677	0.332241	**142 (100.0%)**	15514 (93.59%)	0.967966
	Regressão Logística	0.995454	**1.000000**	0.464789	0.634615	0.520505	0.680197	66 (46.48%)	**16576 (100.0%)**	0.732394
	SVM Linear	0.397595	0.010453	0.746479	0.020617	0.049491	0.026504	106 (74.65%)	16541 (39.46%)	0.570543
	SVM Gaussiano	0.986601	0.345865	0.647887	0.450980	0.551559	0.467454	92 (64.79%)	16402 (98.95%)	0.818695
	SVM Sigmóide	0.990429	0.452632	0.605634	0.518072	0.567282	0.518901	86 (60.56%)	16472 (99.37%)	0.79968
	Comitê rígido	0.994676	0.757282	0.549296	0.636735	0.581222	0.642432	78 (54.93%)	16551 (99.85%)	0.773894
	Comitê ponderado	0.948080	0.128834	0.887324	0.225000	0.407503	0.326873	126 (88.73%)	15724 (94.86%)	0.917962
Subreamostragem	KNN	0.993600	0.579909	0.894366	0.703601	0.806861	0.717373	127 (89.44%)	16484 (99.44%)	0.944408
	Árvore de Decisão	0.972066	0.225042	0.936620	0.362892	0.573770	0.451723	133 (93.66%)	16118 (97.24%)	0.954495
	Floresta Aleatória	**0.999462**	0.978417	0.957746	**0.967972**	**0.961810**	**0.967756**	136 (95.77%)	16573 (99.98%)	**0.978783**
	Regressão Logística	0.969972	0.176259	0.690141	0.280802	0.435943	0.339072	98 (69.01%)	16118 (97.24%)	0.831255
	SVM Linear	0.976193	0.215556	0.683099	0.327703	0.476424	0.375266	97 (68.31%)	16223 (97.87%)	0.830901
	SVM Gaussiano	0.950114	0.124729	0.809859	0.216165	0.385906	0.306005	115 (80.99%)	15769 (95.13%)	0.880587
	SVM Sigmóide	0.991087	0.482587	0.683099	0.565598	0.630689	0.569897	97 (68.31%)	16472 (99.37%)	0.838412
	Comitê rígido	0.990310	0.455357	0.718310	0.557377	0.643939	0.567461	102 (71.83%)	16454 (99.26%)	0.855475
	Comitê ponderado	0.981876	0.306024	0.894366	0.456014	0.645982	0.517275	127 (89.44%)	16288 (98.26%)	0.938496
SMOTE	KNN	0.974937	0.243993	0.929577	0.386530	0.595131	0.469287	132 (92.96%)	16167 (97.53%)	0.952452
	Árvore de Decisão	0.955258	0.152523	0.936620	0.262327	0.461806	0.368170	133 (93.66%)	15837 (95.54%)	0.946019
	Floresta Aleatória	0.997368	0.781609	0.957746	0.860759	0.916442	0.863971	136 (95.77%)	16538 (99.77%)	0.977727
	Regressão Logística	0.965127	0.153846	0.690141	0.251605	0.406639	0.315236	98 (69.01%)	16037 (96.75%)	0.828812
	SVM Linear	0.975476	0.209957	0.683099	0.321192	0.470874	0.370092	97 (68.31%)	16211 (97.80%)	0.830539
	SVM Gaussiano	0.961778	0.158184	0.809859	0.264672	0.444015	0.347795	115 (80.99%)	15964 (96.31%)	0.886469
	SVM Sigmóide	0.992583	0.548913	0.711268	0.619632	0.671543	0.621228	101 (71.13%)	16493 (99.50%)	0.853130
	Comitê ponderado	0.977988	0.267490	0.915493	0.414013	0.616698	0.488343	130 (78.17%)	16220 (98.47%)	0.947008
	Comitê rígido	0.982953	0.304110	0.781690	0.437870	0.594855	0.481255	111 (91.55%)	16322 (97.85%)	0.883183

Por fim, também analisamos os rótulos atribuídos pelo TSVM às instâncias não rotuladas.A tabela 4.5 mostra o número de instâncias rotuladas como contas de usuários profissionais e casuais. É possível observar que apenas 3 dessas contas foram classificadas como casuais. Dessa forma, podemos concluir que a abordagem proposta utilizando tanto o aprendizado não supervisionado quanto o semissupervisionado auxiliou na identificação

dos relatos profissionais.

Tabela 4.5: Resultado da aplicação de SVM transdutivo a dados não rotulados.

Tipo	Contas
Profissionais	299
Casuais	3



(a) Saldo em Ether.

(b) Total de transações.

(c) Transações recebidas.

(d) Transações enviadas.

Figura 4.1: Boxplots mostrando as características das contas não rotuladas quando comparadas às características dos grupos profissionais e casuais.

De acordo com os boxplots apresentados na Figura 4.1, as contas classificadas pela TSVM como profissionais (semi-supervisionadas = 1) são semelhantes àquelas rotuladas como Contas Profissionais quanto ao saldo Ether, total de transações e transações

recebidas e enviadas. Isso mostra a eficiência do classificador semissupervisionado na identificação das contas profissionais.

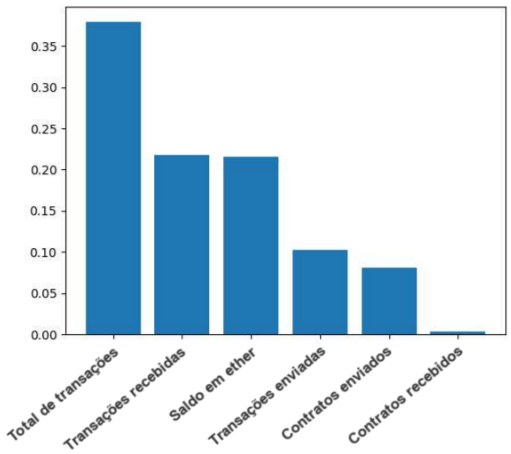


Figura 4.2: Importância dos atributos da Floresta Aleatória com sobreamostragem.

Com relação à importância dos atributos mais relevantes do classificador com melhor AUC-ROC (Figura 4.2), pode-se observar que total de transações, total de transações recebidas e saldo em Ether são relevantes para a identificação das contas profissionais. Isso reflete em mais contas profissionais classificadas corretamente, como floresta aleatória, dando mais importância para essas características, foi a única que conseguiu acertar todas as contas profissionais.

5 Conclusões e trabalhos futuros

Nesse trabalho, propomos abordagens de aprendizado de máquina de diferentes categorias, a saber, aprendizado não supervisionado e semissupervisionado, para mapear comportamentos de usuários e identificar seu perfil em plataformas blockchain. A combinação dessas abordagens pode alcançar alto desempenho para classificar usuários em perfis de interesse e descobrir tais perfis com base no comportamento do usuário entre dados não rotulados. Vale ressaltar que as técnicas de balanceamento de dados também são um recurso importante para treinar classificadores de aprendizado de máquina adequadamente. De fato, a pequena fração de dados rotulados publicamente sobre contas de usuários em plataformas blockchain representa uma tarefa desafiadora para a classificação automática dos perfis desses usuários.

Nossa contribuição neste trabalho oferece uma abordagem importante para a questão da classificação de perfis de usuários em plataformas blockchain. Primeiro, desenvolvemos vários procedimentos que constituem um arcabouço para combinar diferentes abordagens de aprendizado de máquina para essa classificação. Diante dessa estrutura, construímos modelos para identificar contas profissionais, distinguindo-as de contas casuais de comportamento geral, usando recursos extraídos de transações publicamente disponíveis no Ethereum.

Para trabalhos futuros, pretendemos melhorar o desempenho das abordagens de aprendizado de máquina propostas, que abrangem a exploração de mais técnicas de aprendizado de máquina e a extração de novos recursos de transações blockchain. Por fim, pretendemos estender nosso arcabouço para as diferentes plataformas blockchain e diversas categorias de perfis de usuários.

Bibliografia

ASPEMBITOVA, A. T.; FENG, L.; CHEW, L. Y. Behavioral structure of users in cryptocurrency market. *Plos one*, Public Library of Science San Francisco, CA USA, v. 16, n. 1, p. e0242600, 2021.

BARTOLETTI, M. et al. Dissecting ponzi schemes on ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, Elsevier, v. 102, p. 259–277, 2020.

BENGIO, Y. et al. Advances in neural information processing systems 19. *Chinese Medical Ethics*, v. 23, p. 80–83, 2008.

BOUGHORBEL, S.; JARRAY, F.; EL-ANBARI, M. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, Public Library of Science, v. 12, n. 6, 2017.

BREIMAN, L. Bagging predictors. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996.

CASALE-BRUNET, S. et al. Networks of ethereum non-fungible tokens: a graph-based analysis of the erc-721 ecosystem. In: IEEE. *2021 IEEE International Conference on Blockchain (Blockchain)*. [S.l.], 2021. p. 188–195.

CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542–542, 2009.

CHEN, T. et al. Understanding ethereum via graph analysis. *ACM Trans. on Internet Technology (TOIT)*, ACM New York, NY, USA, v. 20, n. 2, p. 1–32, 2020.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

DappReview. *About Dapp Review*. 2019. ⟨https://www.dapp.com/token/Dapp_WhitePaper_en.pdf⟩.

DIETTERICH, T. G. et al. Ensemble learning. *The handbook of brain theory and neural networks*, MIT press Cambridge, Massachusetts, v. 2, p. 110–125, 2002.

ETHERSCAN. *Label Word Cloud*. 2022. ⟨https://etherscan.io/labelcloud⟩.

GARETH, J. et al. *An introduction to statistical learning: with applications in R*. [S.l.]: Spinger, 2013.

HARVEY, C. R.; RAMACHANDRAN, A.; SANTORO, J. *DeFi and the Future of Finance*. [S.l.]: John Wiley & Sons, 2021.

HU, T. et al. Transaction-based classification and detection approach for ethereum smart contract. *Information Processing & Management*, Elsevier, v. 58, n. 2, p. 102462, 2021.

MACQUEEN, J. Classification and analysis of multivariate observations. In: *5th Berkeley Symp. Math. Statist. Probability*. [S.l.: s.n.], 1967. p. 281–297.

MOTAMED, A. P.; BAHRAK, B. Quantitative analysis of cryptocurrencies transaction graph. *Appl. Netw. Sci.*, v. 4, n. 1, p. 131, Dec 2019.

NASTESKI, V. An overview of the supervised machine learning methods. *Horizons. b*, v. 4, p. 51–62, 2017.

NORVILL, R. et al. Automated labeling of unknown contracts in ethereum. In: *Proc. of the IEEE International Conference on Computer Communication and Networks*. [S.l.: s.n.], 2017. p. 1–6.

OLIVEIRA, V. C. et al. Analyzing Transaction Confirmation in Ethereum Using Machine Learning Techniques. *SIGMETRICS Perform. Eval. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 48, n. 4, p. 12–15, 2021.

PETERSON, L. E. K-nearest neighbor. *Scholarpedia*, v. 4, n. 2, p. 1883, 2009.

SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 21, n. 3, p. 660–674, 1991.

SCHÄR, F. Decentralized finance: On blockchain-and smart contract-based financial markets. *Available at SSRN 3571335*, 2020.

SINGH, H. J.; HAFID, A. S. Prediction of transaction confirmation time in ethereum blockchain using machine learning. In: *Proc. of the Int. Congress on Blockchain and Applications*. [S.l.: s.n.], 2019. p. 126–133.

SOUSA, J. E. A. et al. An analysis of the fees and pending time correlation in ethereum. *Int. J. Netw. Manag.*, Wiley Online Library, p. e2113, 2020.

SUTTON, R. S.; BARTO, A. G. et al. *Introduction to reinforcement learning*. [S.l.]: MIT press Cambridge, 1998. v. 135.

THARANI, J. S. et al. Graph based visualisation techniques for analysis of blockchain transactions. In: *Proc. of the IEEE Conference on Local Computer Networks*. [S.l.: s.n.], 2021. p. 427–430.

VALADARES, J. A. et al. Identifying user behavior profiles in ethereum using machine learning techniques. In: IEEE. *2021 IEEE International Conference on Blockchain*. [S.l.], 2021. p. 327–332.

WRIGHT, R. E. Reading and understanding multivariate statistics. In: _____. [S.l.]: APA, 1995. cap. Logistic regression, p. 217–244.

WU, K. et al. A first look at blockchain-based decentralized applications. *Software: Practice and Experience*, Wiley Online Library, v. 51, n. 10, p. 2033–2050, 2021.

WU, S. X. et al. Community detection in blockchain social networks. *Journal of Communications and Information Networks*, v. 6, n. 1, p. 59–71, 2021.

XU, G. et al. Am i eclipsed? a smart detector of eclipse attacks for ethereum. *Computers & Security*, Elsevier, v. 88, p. 101604, 2020.

XU, X.; WEBER, I.; STAPLES, M. *Architecture for blockchain applications*. [S.l.]: Springer, 2019.

ZHANG, R.; XUE, R.; LIU, L. Security and privacy on blockchain. *ACM Computing Surveys*, ACM New York, NY, USA, v. 52, n. 3, p. 1–34, 2019.