Identificador de bots sociais em redes de comentários do YouTube

Járdesson Ribeiro[1], Samuel Silva[2], Matheus Do Vale[3]

[1,2,3]Centro de Ciências da Natureza – Universidade Federal do Piauí (UFPI)
Tópicos em Inteligência Artificial
{jardessonrs1117,matheusvale0202, samuelslv7}@gmail.com

Abstract. *This study aimed to identify and build a bot network in YouTube video comments. To achieve this, data mining techniques were employed, including the consumption of YouTube API v3 on Google Cloud to collect information about videos, comments, replies, and channel data. Based on this data, relevant characteristics about the comments and channels that were published were calculated. The study includes two categories of bots: Conversational Bots, which exhibit more organized behavior and well-crafted texts in comments, and Spam Bots, which focus on generating a large volume of comments. It was observed that the accuracy of the SGDClassifier model, used in the study, does not always reflect the ability to identify both types of bots.*

Resumo. *O presente trabalho teve como objetivo identificar e construir uma rede de bots em comentários de vídeos do YouTube. Para isso, foram empregadas técnicas de mineração de dados, incluindo o consumo da API v3 do YouTube no Google Cloud para a coleta de informações de vídeos, comentários, respostas e dados de canais. Com base nesses dados, foram calculadas features relevantes sobre os comentários e os canais que os publicaram. O estudo identificou duas categorias de bots: Bots Conversacionais, que exibem um comportamento mais organizado e textos bem elaborados nos comentários, e Bots Spam, com foco em gerar um grande volume de comentários. Foi observado que a acurácia do modelo SGDClassifier, utilizado no trabalho, nem sempre reflete a capacidade de identificar ambos os tipos de bots.*

Palavras-chave: *Detecção de Bots, YouTube, Bots Sociais, Aprendizagem de Máquina, Análise de Redes Sociais.*

1. Introdução

A crescente popularidade de plataformas digitais como o YouTube transformou a interação social, mas também abriu espaço para a proliferação de perfis automatizados, conhecidos como bots sociais. Programados para simular comportamento humano, esses bots são frequentemente usados para manipular o engajamento de forma artificial ou para disseminar golpes financeiros em larga escala. Eles operam em seções de comentários, publicando falsos depoimentos, convidando usuários para plataformas externas e simulando diálogos para criar uma falsa sensação de legitimidade.

A constante adaptação das estratégias dos bots torna a detecção manual uma tarefa complexa e pouco escalável. Diante deste cenário, este trabalho propõe o desenvolvimento de um sistema de detecção que combina técnicas de Aprendizagem de Máquina com Análise de Redes Sociais.

O objetivo principal é construir e validar um modelo de classificação, utilizando o algoritmo SGDClassifier, capaz de identificar bots com base em metadados de seus comentários e canais. Adicionalmente, o projeto vai além da classificação individual, modelando as interações entre as contas detectadas como grafos direcionados. Essa abordagem busca não apenas identificar os perfis automatizados, mas também revelar seus padrões de comportamento coordenado, expondo a estrutura de suas redes de operação e reforçando a eficácia da detecção.

2. Metodologia

2.1. Aquisição dos dados

Para iniciar o processo de identificação de bots, buscamos vídeos que apresentassem comportamentos suspeitos na seção de comentários. A escolha foi baseada em observações de vídeos dos segmentos **fitness** e **criptomoedas**, onde identificamos um padrão recorrente: comentários oferecendo dinheiro ou promessas de lucro rápido, seguidos de diversas respostas que reforçam essas promessas. Esses padrões sugerem uma possível atuação coordenada de contas automatizadas (bots), principalmente em conteúdos com apelo financeiro.

A coleta de dados foi realizada por meio da **YouTube Data API v3**[GOOGLE], onde consultamos as informações de comentários, respostas (replies) e dados dos canais envolvidos. A API permite o acesso estruturado a informações.

“https://developers.google.com/youtube/v3/docs/videos/list”

Exemplo 1. Para buscar informações de vídeos

Requisição utilizada para obter dados sobre o vídeo, como publicação, estatísticas, usado como parâmetro a propriedade ‘part’ com valor ‘snippet‘**,** ‘id’ com o valor do ‘id do vídeo’ e a ‘key’ com o valor da key do usuário que deve ser gerada pelo Google Cloud**[**GOOGLE].

“https://developers.google.com/youtube/v3/docs/commentThreads/list”

Exemplo 2. Para buscar informações de comentários do vídeo

A requisição foi utilizada para obter os comentários principais de um vídeo. Os parâmetros definidos incluíram 'part': 'snippet' para acessar o conteúdo e metadados do comentário, 'videoId' para indicar o vídeo-alvo, 'maxResults': 100 para limitar o número de resultados por página, 'pageToken' para controle de paginação, e 'key' para autenticação via chave da API.

“https://developers.google.com/youtube/v3/docs/comments/list”

Exemplo 3. Para buscar informações de comentários respostas(replies)

A requisição foi utilizada para coletar comentários-resposta (replies) a partir de um vídeo. Os parâmetros incluíram 'part': 'snippet' para obter os dados do comentário, 'videoId' para identificar o vídeo de origem, 'maxResults': 100 para controlar o volume por requisição, 'textFormat': 'plainText' para retornar texto limpo, 'pageToken' para paginação e 'key' para autenticação.

“https://developers.google.com/youtube/v3/docs/channels/list”

Exemplo 4. Para buscar informações do canal

A requisição foi utilizada para obter informações dos canais autores dos comentários. Os parâmetros utilizados foram 'part': 'snippet,statistics' para acessar dados como nome, data de criação, inscritos, vídeos e visualizações, 'id' para identificar o canal, e 'key' para autenticação via chave da API.

2.2. Pré-processamento dos dados

Após a coleta, os dados dos comentários foram inicialmente organizados em arquivos JSON, um para cada vídeo analisado. Na primeira etapa do pré-processamento, consolidamos todos esses arquivos em um único dataset unificado. Em seguida, realizamos a rotulagem manual dos comentários, marcando quais eram de bots com base em indícios como conteúdo repetitivo, menções a dinheiro ou golpes, e características suspeitas nos canais — essa validação foi feita com apoio da interface do próprio YouTube.

Com os dados rotulados, extraímos e organizamos as variáveis relevantes em um arquivo CSV. Foram aplicadas transformações como: conversão de datas para timestamps, cálculo de diferenças temporais (por exemplo, tempo após a publicação do vídeo), tratamento de valores ausentes e padronização dos tipos de dados. Esse conjunto final foi utilizado como base para a construção do modelo de classificação.

2.3. Extração de features

Com o conjunto de dados estruturado e rotulado, foi realizada a etapa de extração e engenharia de atributos. Algumas features foram diretamente obtidas da API do YouTube, enquanto outras precisaram ser calculadas ou transformadas, como o tempo desde a criação do canal ou a identificação de comentários puramente numéricos. Também foram aplicadas normalizações para garantir que os dados pudessem ser processados corretamente pelo modelo, como pode ser visualizado na **Tabela 1**.

Coluna	Descrição
video_id	ID do vídeo onde o comentário foi publicado.
comment_id	ID único do comentário
is_reply	Indicador binário (0 ou 1) se o comentário é uma resposta.
reply_to	ID do comentário pai (caso seja uma resposta)
comment_lenght	Quantidade de caracteres do comentário
author	Nome do autor do comentário
channel_id	ID do canal que fez o comentário
root_channel_id	ID do canal do comentário pai (relacionamento

	entre usuários)
comment_publishedAt	Timestamp da publicação do comentário (convertido de data ISO)
seconds_after_comment	Tempo entre o comentário e sua resposta (0 para comentários principais)
seconds_after_video	Segundos desde a publicação do vídeo até o comentário
subscriber_count	Número de inscritos do canal autor do comentário
video_count	Total de vídeos publicados pelo canal
view_count	Total de visualizações do canal
created_channel	Dias desde a criação do canal até o momento do comentário
comment_is_number	Indicador binário: 1 se o comentário for apenas numérico, 0 caso contrário
bot	Rótulo binário: 1 para bot, 0 para não-bot (definido manualmente)

Tabela 1. Tabela de metadados extraídos

2.4. Modelo de Classificação

Para a tarefa de classificação de bots, utilizamos o algoritmo Stochastic Gradient Descent (SGDClassifier) da biblioteca scikti-learn da linguagem Python, conhecido por sua eficiência em grandes volumes de dados e simplicidade de implementação. A divisão entre treino e teste foi ajustada empiricamente, testando diferentes proporções e observando o impacto na acurácia. A escolha do SGD foi baseada em um estudo do **CiDAMO** (Grupo de Ciência de Dados, Aprendizagem de Máquina e Otimização) do departamento de matemática da **UFPR**, que apontou esse modelo como o mais eficaz na detecção de contas automatizadas. Sua natureza linear, combinada com boa generalização, o torna adequado para conjuntos com muitas features numéricas e dados ruidosos.

3. Resultados

Nesta seção, apresentamos os principais resultados obtidos durante o desenvolvimento do classificador de bots em redes sociais de comentários no YouTube. Os experimentos foram conduzidos sobre um conjunto de dados composto por comentários extraídos de vídeos com indícios de comportamento de bots. O conjunto final utilizado para

treinamento continha tanto comentários rotulados manualmente como bots, quanto comentários legítimos.

O modelo foi treinado com features extraídas dos comentários e dos canais, como tempo de publicação, contagem de inscritos, número de vídeos e padrões no conteúdo textual, entre outras features.

A avaliação do modelo foi feita com diferentes divisões entre treino e teste e, posteriormente, aplicamos o classificador sobre novos conjuntos de comentários, sem rótulos, para identificar possíveis bots. Os resultados obtidos foram analisados tanto individualmente quanto em termos de relacionamento em rede, visando detectar padrões coordenados de atividade automatizada.

3.1. Avaliação do modelo de classificação - com menos features

Com um conjunto de 7941 comentários anotados manualmente, sendo 3437 identificados como bots e o restante como não bots, treinamos o modelo utilizando diferentes configurações de divisão dos dados e balanceamento entre as classes. Utilizamos nesses testes as seguintes features [**comment_publishedAt, seconds_after_comment, subscriber_count, video_count, view_count, created_channel, comment_lenght**]. E os resultados podem ser vistos nas Figura 1 e 2 abaixo.

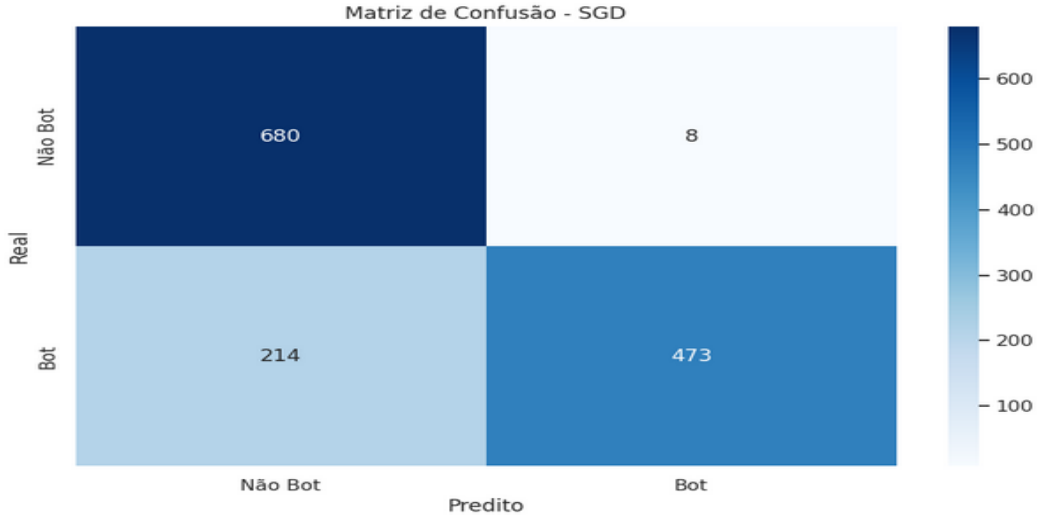

Figura 1. Matriz de confusão com dados desbalanceados para o treinamento

Neste cenário, a acurácia caiu para 73%, mostrando desempenho inferior ao balanceado. Para “Não Bot”, a precisão foi 0.80, mas o recall caiu para 0.70, indicando que o modelo deixou de identificar alguns “Não Bots”. Para “Bot”, a precisão foi 0.66 e recall 0.77, revelando que o modelo erra mais ao prever bots, gerando mais falsos positivos. O f1-score ficou em torno de 0.72-0.75, indicando redução na performance geral. Aqui, o desbalanceamento prejudicou o modelo, resultando em perda de equilíbrio entre as classes.

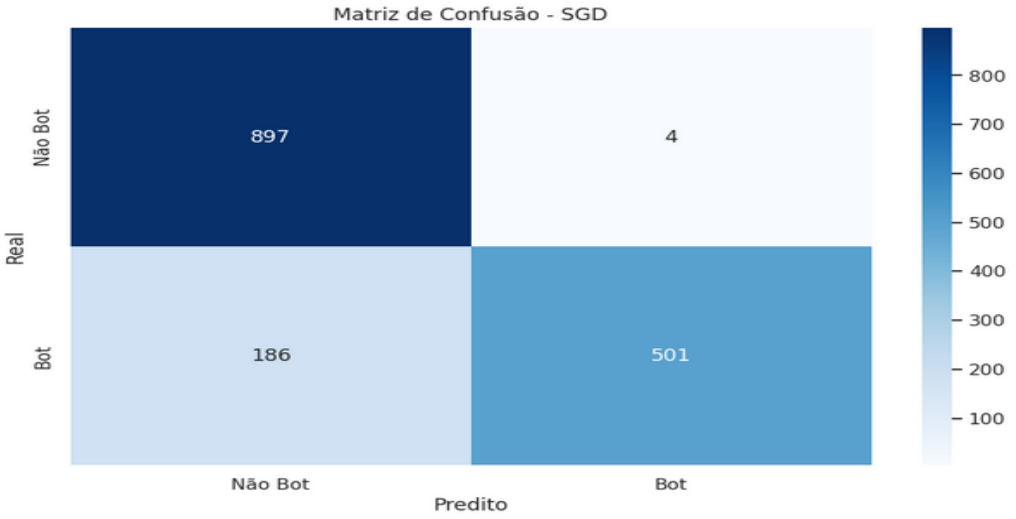

Figura 2. Matriz de confusão com dados balanceados para o treinamento

O modelo obteve **acurácia de 77%**, indicando desempenho moderado, com equilíbrio entre classes. Para "Não Bot", a precisão foi 0.79 e recall 0.75, mostrando que acerta bem, mas há falsos negativos. Para "Bot", a precisão ficou em 0.76 e o recall em 0.80, mostrando ligeira vantagem em identificar bots corretamente. O f1-score de 0.77-0.78 indica equilíbrio entre precisão e recall em ambas as classes. O balanceamento ajudou o modelo a aprender os padrões de ambas as classes de forma próxima, reduzindo viés e mantendo desempenho consistente..

Para tanto, podemos identificar quais features contribuíram mais para a identificação dos bots na rede de comentários, como pode ser visto na Figura 3.

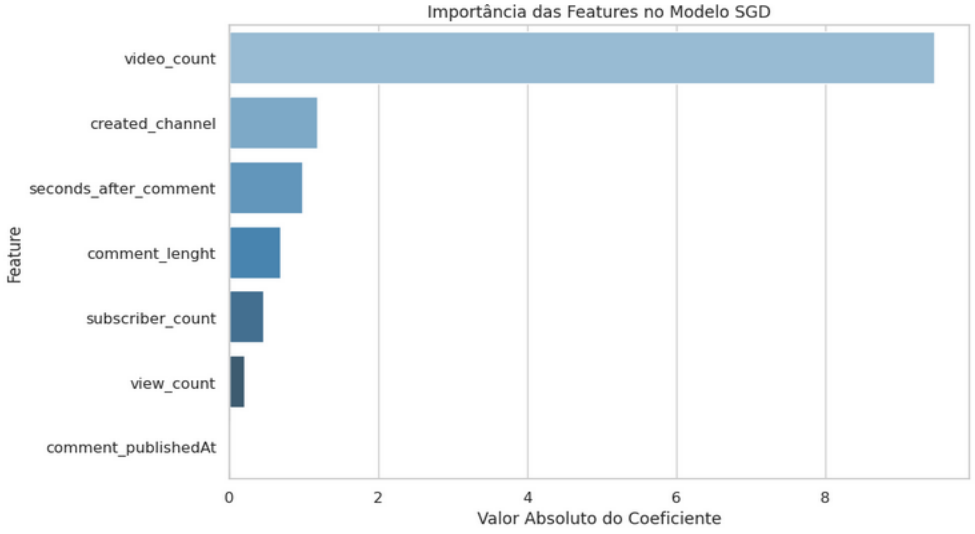

Figura 3. Features mais importantes com menos features no modelo

Percebemos dessa forma que **video_count** e **created_channel** foram as que tiveram maior impacto na detecção. Embora a acurácia não tenha sido tão boa, a considerar como boa pelo menos 90% de acurácia, conseguimos identificar um maior número de bots com comportamentos humanos.

Com a remoção de features podemos identificar esses bots com um comportamento mais humano, chamados de **Bots conversacionais avançados**, como um texto de comentário mais estruturado, com um tempo de comentário maior em relação ao vídeo, como mostrado na Figura 4, para um conjunto predominante desses bots, com 297 nós identificados, com um total de 265 arestas, com que nos evidencia um valor bom de detecção com desses tipos de bots.

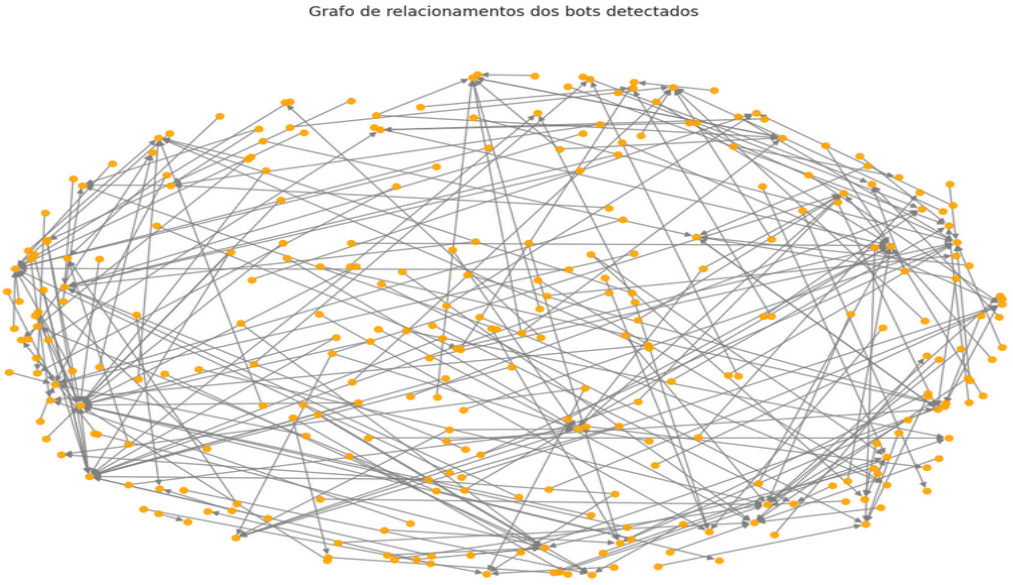

Figura 4. Bots detectados em vídeos não rotulados de Cripto

3.2. Avaliação do modelo de classificação - com todas as features

Seguindo os mesmos conjuntos de dados do cenário proposto no item **3.1** agora aplicamos os mesmos experimentos, no entanto, colocamos todas as features de treinamento para observarmos o comportamento da classificação dos bots em rede.

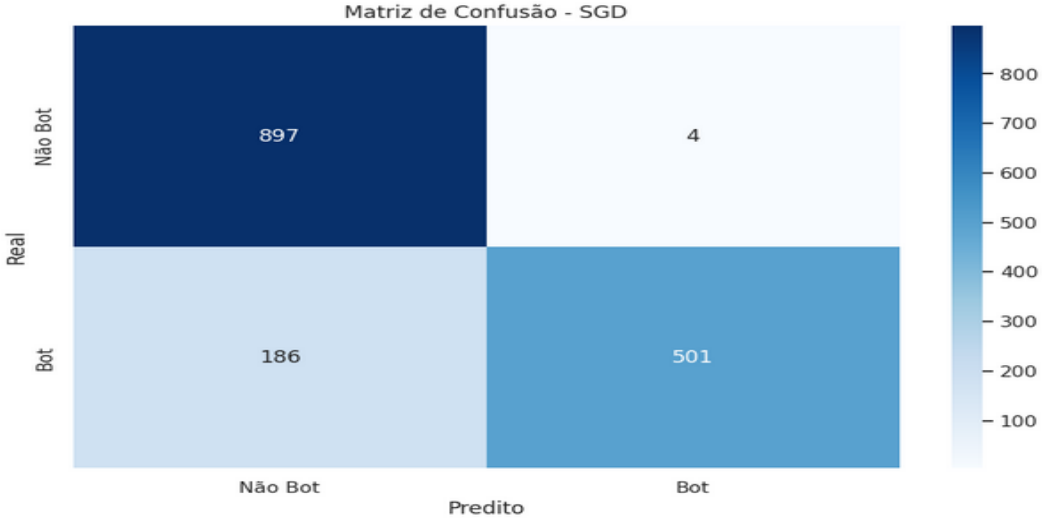

Figura 5. Com dados desbalanceados e mais features

Neste cenário, a acurácia subiu para 88%, mostrando melhoria com os dados desbalanceados. Para “Não Bot”, houve precision de 0.83 e recall de 1.00, identificando

quase todos os “Não Bots” sem deixar escapar nenhum, mas com alguns bots classificados erroneamente como “Não Bot”. Para “Bot”, a precision foi alta (0.99) e o recall subiu para 0.73, melhorando em relação ao balanceado. O f1-score também subiu, reforçando o ganho de desempenho. Isso indica que o modelo teve facilidade em identificar “Não Bots” quando em maior quantidade, mantendo bom desempenho na detecção de bots.

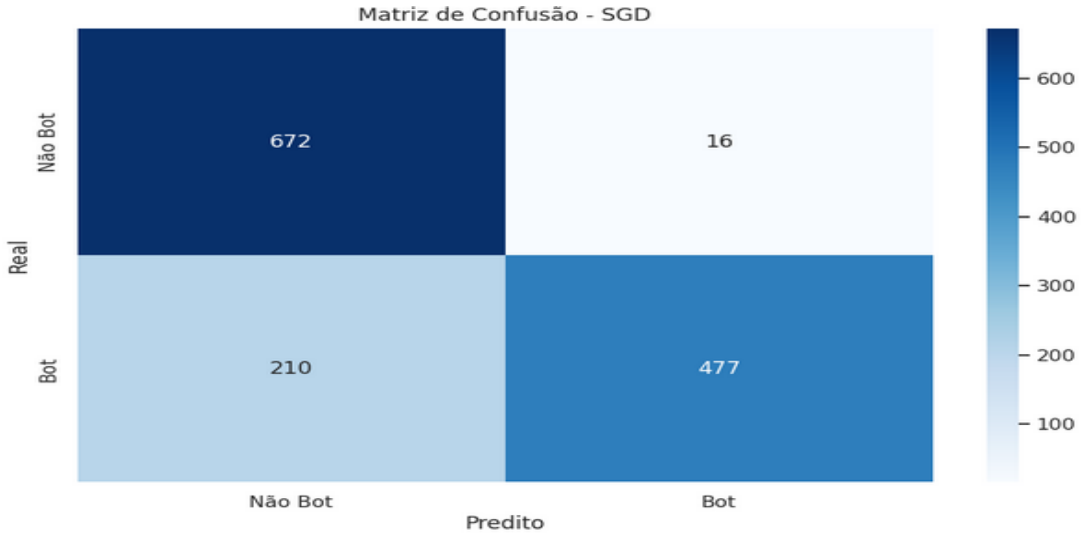

Figura 6. Com dados balanceados e mais features

Comparando ambos, o modelo performa melhor em dados desbalanceados, aumentando acurácia e recall de “Bot” mesmo em cenário onde há mais “Não Bots”, com uma acurácia de 84%. Isso indica que o modelo está aprendendo bem os padrões de bots mesmo quando eles são minoria. Entretanto, o balanceamento ainda pode ser útil caso o foco seja reduzir vieses de classe e ter métricas equilibradas, mas, neste caso específico, os dados desbalanceados resultaram em melhor desempenho geral e melhor identificação de “Bots”, mostrando que a qualidade dos dados e a separabilidade das classes podem ser mais importantes que o balanceamento puro em certos contextos.

No entanto, quando observamos as features mais importantes podemos notar uma mudança no comportamento do modelo, que passou a ter como features mais importantes justamente features que não estavam sendo usadas antes, como mostra a Figura 7.

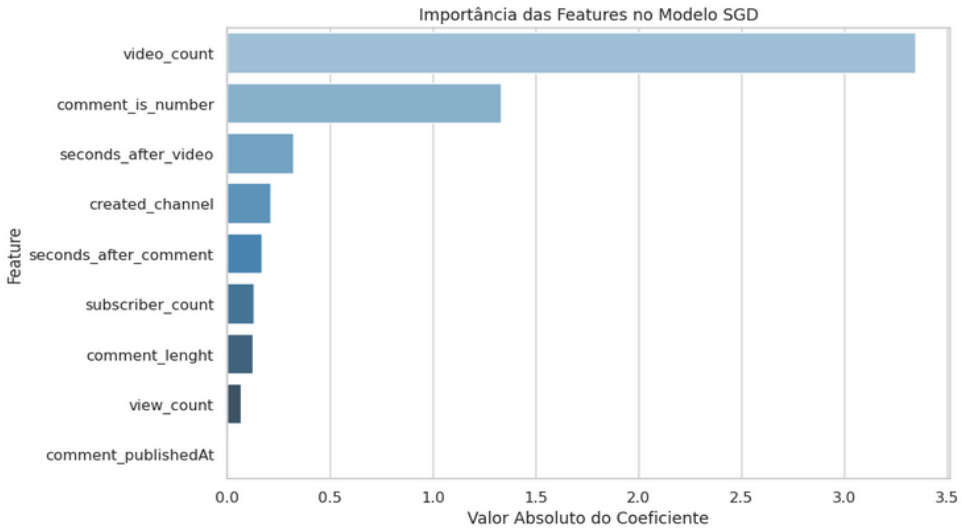

Figura 7. Features mais importantes com mais features no modelo

Essa mudança evidencia que a detecção de bots necessita considerar aspectos de comportamento desses bots, no qual mudanças de features pode levar considerar bots quando não sendo, e vice-versa, como é o caso mostrado nesse exemplo, que pode se visualizado na Figura 8, em comparação com a Figura 4.

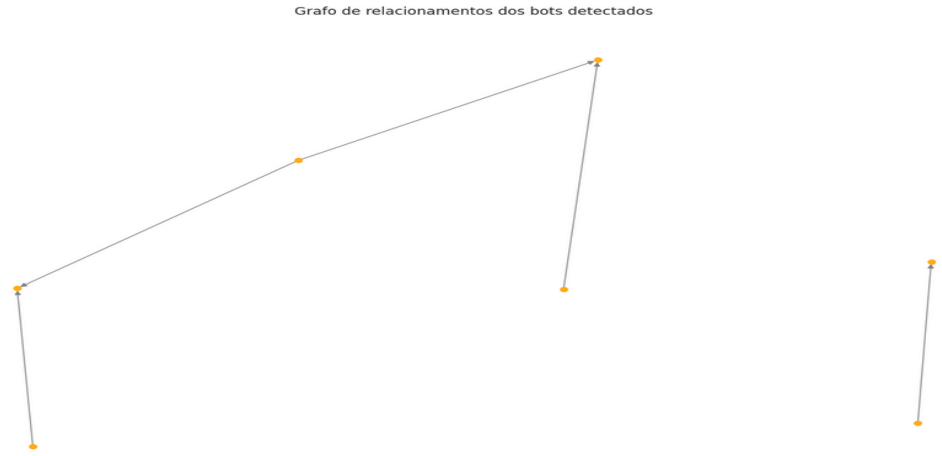

Figura 8. Bots detectados em vídeos não rotulado de Cripto

Notamos, de forma óbvia, que ao colocarmos mais features, e comparando com com o outro experimento com menos features no item **3.1,** Figura 4, percebemos que determinados bots com comportamentos diferentes deixam de ser detectados, gerando uma rede menor, com menos bots. Esses bots que deixaram de ser detectados, são da categoria de **Bots conversacionais avançados**, que possuem, como característica ter comentários com maior número de caracteres e adicionar comentários com intervalo maior de tempo, com exemplos de 2 dias para começar a comentar em um vídeo.

Em continuidade, observamos que, de acordo com a Figura 7, podemos notar que as features mais importantes são justamente as features que possuem informações sobre o tamanho do comentário e o tempo desde a publicação do vídeo.

3.3. Detecção de redes de bots

Com outro viés, partimos para identificação de comportamentos em rede, buscando responder os questionamentos sobre os bots, e inferir hipóteses com base na topologia da rede, com dados de grau e componentes, e assim comprovar, através dessas observações, o comportamento de bots nos comentários do YoutTube.

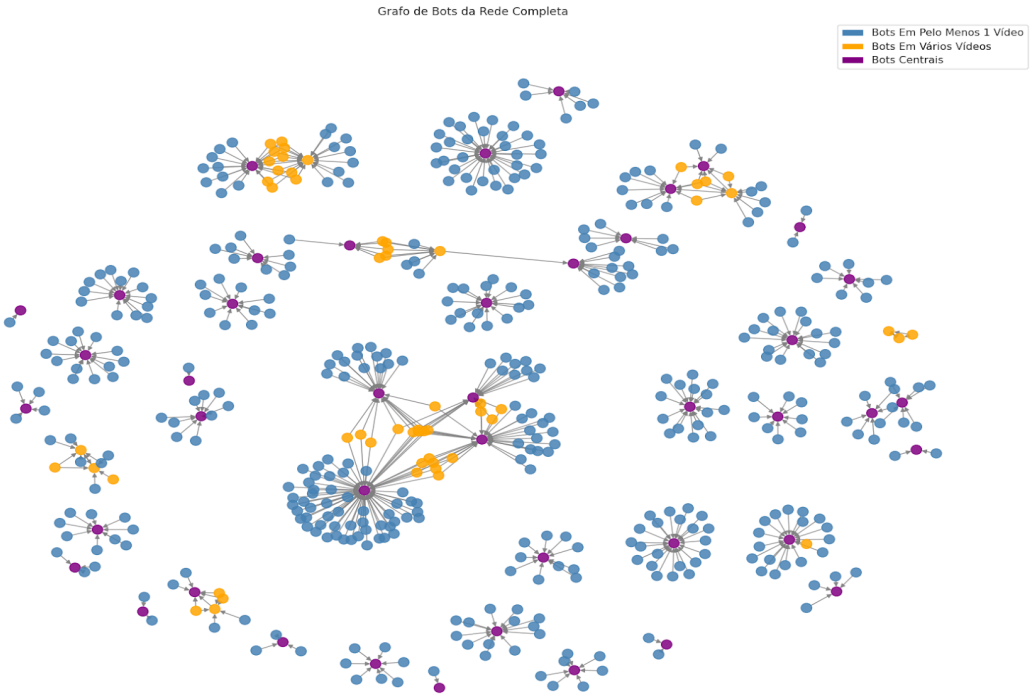

Figura 9. Rede completa dos bots identificados

Com a rede completa de bots identificados em comentários do YouTube, podemos começar a compreender melhor o comportamento da rede de bots, assim como bots centrais, bots que iniciam os comentários no vídeo, também podem agir como bots secundários, que apenas respondem (replay), em outros vídeos, e assim compreender um pouco mais sobre ações coordenadas em grupo de vídeos de um segmento de vídeo, por exemplo.

Os bots centrais são os bots com maior grau de entrada, relação a aresta de reply, nesse caso, alguns bots centrais iniciam os comentários. Um outro ponto percebido, que existem grupos de bots relacionados em diferentes vídeos, com uma comportamento de evitar comentar sempre como bot central, como exemplo um bot A inicia os comentários daquele grupo em um vídeo A, mas no vídeo B, ele não será mais o bot central, mas sim um secundário pois existe outro bot B como principal. Mas quando observamos o comportamento da rede observamos essa ação.

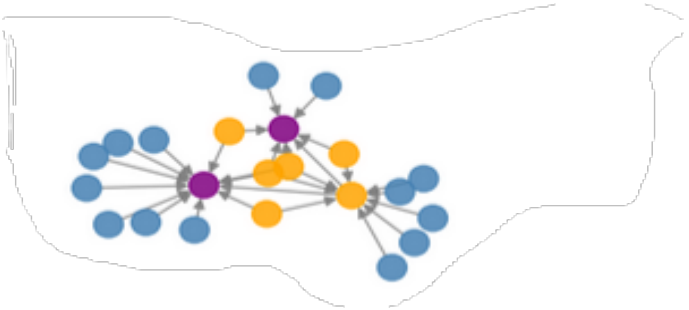

Figura 10. Bots centrais agindo como secundários

A Figura 10 evidencia bem esse comportamento dos bots, no qual em um dos grupos de comentários o bot apresenta-se como central, iniciando os comentários, mas em outros 2 grupos, ele é apenas um bot secundário gerando uma pequena rede.

Em continuação na análise, mas aplicando outras métricas, como Componentes Fortemente Conectados (SCC), obtivemos 523 fortemente conectados, o que nos indica rede altamente fragmentada, o que também é indicativo por conter na base de dados vários segmentos de vídeos. Além disso, Bots em vários vídeos (laranjas) aparecem espalhados entre os clusters, indicando bots mais ativos, mas ainda isolados por cluster, o que indica estratégia de bots de engajamento ou spam, atuando em ilhas para inflar métricas de canais específicos.

3.4. Detecção de redes de bots em segmento de vídeos

A análise da subrede de bots revelou um cenário interessante para o comportamento de contas automatizadas em segmentos específicos de vídeos. Apesar de o grafo apresentar uma estrutura visual densa, com múltiplas conexões entre nós, a análise dos Componentes Fortemente Conectados (SCC) mostrou que foram encontrados 167 SCCs, todos de tamanho 1. Isso indica que cada nó está isolado em termos de conexões recíprocas, formando um componente unitário, mesmo dentro de uma estrutura aparentemente conectada.

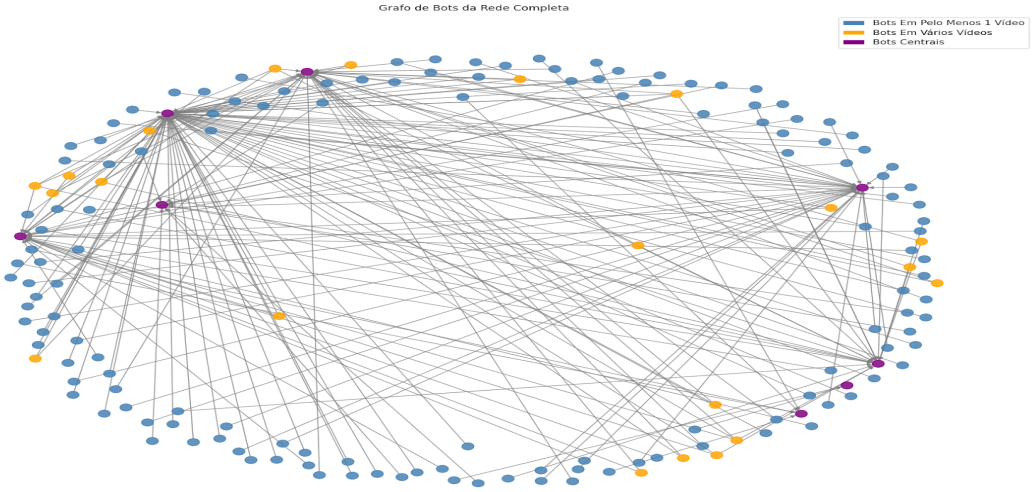

Figura 11. Sub rede de bots por um segmento de vídeo

Os bots desta sub rede se conectam de forma unidirecional a nós centrais ou canais, sem criar caminhos de ida e volta entre si. Essa configuração revela que, apesar

da rede parecer densa, não há laços de reciprocidade entre os bots, o que impede a formação de comunidades detectáveis por algoritmos de análise de grafos. Esse comportamento é uma estratégia para que os bots operem de forma dispersa, mas ainda focados em gerar engajamento em canais ou vídeos específicos. Assim, conseguem inflar visualizações e interações de forma coordenada, enquanto evitam rastros que facilitem sua identificação em análises de rede.

Com isso, a rede se mantém ativa e eficiente em manipular métricas de relevância, mas permanece silenciosa em termos de conexões internas entre os bots. Este padrão é característico de bots de engajamento e promoção, que priorizam a atuação em segmentos específicos sem criar estruturas visíveis de cooperação entre si.

3.5. Interação com componentes fracamente conectados

Outra métrica utilizada em comparação entre a rede e sub rede, foi Componentes Fracamente Conectados (WCC), no qual utilizamos a rede completa e a sub rede, e assim comparar o comportamento geral e local da rede de bots.

Na rede completa, foram encontrados 40 componentes fracamente conectados (WCC), sendo o maior com 115 nós. Isso indica que a rede está fragmentada em diversos grupos isolados, mas ainda existe um grande bloco conectado, mesmo que apenas fracamente, em que os nós estão interligados por caminhos ignorando a direção das arestas.

Na subrede, foram encontrados apenas 5 WCCs, sendo também o maior com 115 nós, indicando que a subrede concentra um grande bloco conectado que se mantém coeso, mesmo em um recorte da rede. O fato do maior WCC ter o mesmo tamanho sugere que esta sub rede é justamente o maior componente encontrado na rede completa, confirmando sua relevância estrutural. Essas métricas indicam que, enquanto os bots operam de forma isolada em muitos casos (gerando fragmentação em SCCs), há grupos que compartilham conexões indiretas, formando blocos maiores de interação quando a direção é ignorada. Isso pode indicar bots que interagem em torno dos mesmos vídeos ou canais, comentando em espaços similares, criando um tecido de conexões que os mantêm relacionados de forma indireta.

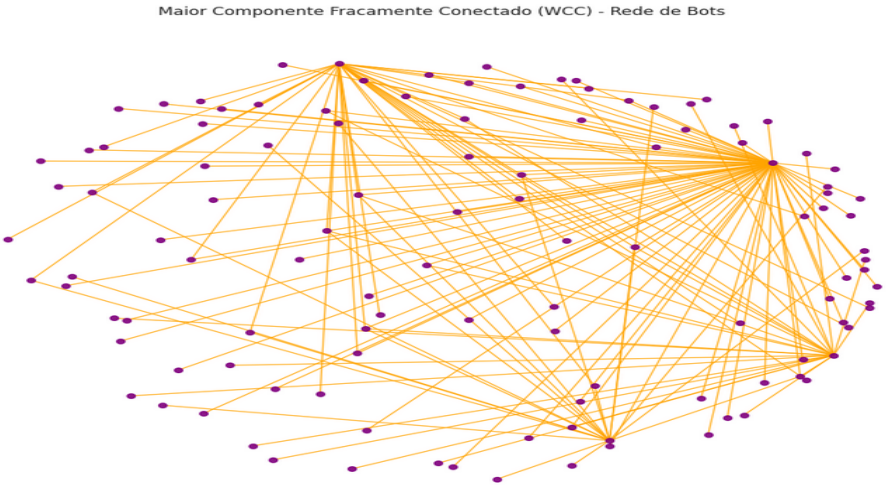

Figura 12. Maior componente fracamente conectados

A Figura 12 demonstra o maior componente fracamente conectado, que é o mesmo tanto na rede completa como na sub rede.

4. Conclusões

Este trabalho apresentou um sistema híbrido para a detecção de bots sociais nos comentários do YouTube, demonstrando a eficácia da combinação entre o classificador de Aprendizagem de Máquina. SGDClassifier e a Análise de Redes Sociais. O objetivo de identificar perfis automatizados e expor seus padrões de interação coordenada foi alcançado com sucesso.

A principal contribuição deste projeto reside na validação de que uma abordagem multifacetada é significativamente mais robusta para o combate a atividades fraudulentas e engajamento artificial. A metodologia não apenas permitiu a classificação de contas individuais com base em seus metadados , mas também, de forma crucial, mapeou a infraestrutura de suas operações por meio da modelagem de grafos.

A análise das redes revelou comportamentos coordenados, como a alternância de papéis entre bots "centrais" e "secundários" em diferentes vídeos, uma tática para evitar a detecção. Os resultados demonstraram que a seleção de *features* impacta diretamente a capacidade de detecção. Enquanto o modelo com mais *features* alcançou uma acurácia superior, chegando a 88% em dados desbalanceados , a versão com menos *features* se mostrou mais apta a identificar "bots conversacionais avançados", que exibem comportamentos mais próximos aos de humanos. A análise da importância das *features* destacou que atributos como a contagem de vídeos do canal (video_count) e o tempo desde sua criação (created_channel) foram determinantes para o modelo.

As análises de rede, utilizando métricas como Componentes Fortemente e Fracamente Conectados, confirmaram que os bots atuam em grupos fragmentados, mas exibem conexões indiretas que formam blocos de interação em torno de vídeos ou canais específicos.

Como trabalhos futuros, sugere-se a expansão do conjunto de *features* para incluir uma análise semântica mais aprofundada do conteúdo textual dos comentários, utilizando modelos de Processamento de Linguagem Natural (PLN) para identificar padrões de discurso e tópicos recorrentes. Adicionalmente, o sistema pode ser evoluído para uma ferramenta de monitoramento em tempo real, oferecendo um recurso valioso para criadores de conteúdo e administradores de plataformas na moderação proativa de interações suspeitas e na proteção de suas comunidades.

5. Referências

PYTHON SOFTWARE FOUNDATION. *Python documentation*. Disponível em: https://docs.python.org/3/.

DAVIS, C. A.; VAROL, O.; FERRARA, E.; FLAMMINI, A.; MENCZER, F. BotOrNot: A system to evaluate social bots. In: Proceedings of the 25th International Conference Companion on World Wide Web. 2016.

MORAIS, D. M. G.; DIGIAMPIETRI, L. A. Methods and challenges in social bots detection: A systematic review. In: XVII Brazilian Symposium on Information Systems (SBSI 2021). 2021.

PANDAS DEVELOPMENT TEAM. *pandas: Python Data Analysis Library*. Disponível em: https://pandas.pydata.org/docs/.

SCIKIT-LEARN DEVELOPERS. *scikit-learn: Machine Learning in Python*. Disponível em: https://scikit-learn.org/stable/.

HUNTER, J. D. *Matplotlib: Visualization with Python*. Disponível em: https://matplotlib.org/stable/contents.html.

WASKOM, M. L. et al. *Seaborn: Statistical Data Visualization*. Disponível em: https://seaborn.pydata.org/.

GOOGLE. *Google Colaboratory Documentation*. Disponível em: https://research.google.com/colaboratory/.

GOOGLE. *YouTube Data API v3 – Developer Guide*. Disponível em: https://developers.google.com/youtube/v3

GOOGLE. *API Keys Documentation*. Disponível em: https://cloud.google.com/api-keys/docs.

CIDAMO. *Bots: A Brief Overview*. Disponível em: https://cidamo.github.io/2020-08-06-bots/.

SCIKIT-LEARN. s*klearn.linear_model.SGDClassifier*. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html