Modelagem Bayesiana para Recomendação Adaptativa de Proxies em Sistemas Automatizados

Paulo Henrique Cardoso de Souza
Escola de Engenharia Elétrica, Mecânica e de Computação
Universidade Federal de Goiás
Goiânia, Goiás, Brasil
paulo.souza@discente.ufg.br

Leonardo da Cunha Brito
Escola de Engenharia Elétrica, Mecânica e de Computação
Universidade Federal de Goiás
Goiânia, Goiás, Brasil
leonardo_brito@ufg.br

Thyago Carvalho Marques
Escola de Engenharia Elétrica, Mecânica e de Computação
Universidade Federal de Goiás
Goiânia, Goiás, Brasil
thyago@ufg.br

***Resumo*—Este artigo apresenta um sistema de recomendação de rotação de proxies baseado em modelagem Bayesiana para otimizar a coleta automatizada de dados em larga escala. As abordagens tradicionais de rotação de proxies não se adaptam dinamicamente ao desempenho dos proxies nem consideram o contexto temporal dos eventos, resultando em baixa eficiência e necessidade de intervenção manual. O método proposto utiliza a distribuição Beta conjugada com Thompson Sampling para estimar probabilidades de sucesso dos proxies, incorporando um componente de decaimento exponencial que prioriza eventos recentes e penaliza falhas com peso três vezes maior que acertos. O sistema foi avaliado com 72 robôs automatizados operando com quatro provedores de proxies durante oito dias (7 a 15 de setembro de 2024). Os resultados demonstram uma melhoria significativa na taxa de acertos, aumentando de 47% para 77% na média e de 49% para 77% na mediana, comparado ao sistema de rotação simples. Casos específicos evidenciaram ganhos ainda mais expressivos, como o robô 15 que passou de 15% para 85% de acertos. O sistema demonstrou adaptabilidade dinâmica, reduzindo a necessidade de intervenção manual e garantindo continuidade operacional em aplicações de web scraping em larga escala.**

***Index Terms*—Web Scraping, Rotação de Proxies, Modelagem Bayesiana, Coleta de Dados, Thompson Sampling**

I. Introdução

O aumento exponencial no volume de dados digitais tem criado uma demanda crescente por sistemas eficientes de coleta automatizada de informações [1]. O web scraping, definido como o processo automatizado de extração de dados de websites [3], tornou-se uma técnica fundamental para organizações que dependem de dados atualizados para tomada de decisões estratégicas [5].

Com o crescimento da digitalização global, estima-se que o volume de dados criados, capturados e replicados mundialmente atingirá 175 zettabytes até 2025 [2]. Nesse contexto, a capacidade de extrair e processar informações de forma automatizada representa uma vantagem competitiva significativa para empresas e instituições de pesquisa.

Entretanto, a captura de dados em larga escala enfrenta desafios significativos [4]. Com a crescente vigilância digital e a proteção de dados, muitos serviços implementam restrições rigorosas para impedir acessos automatizados. O bloqueio de IPs é uma das principais estratégias empregadas para conter essa atividade, tornando o processo de extração de dados mais complexo. Proxies são amplamente utilizados como uma solução para contornar esses bloqueios, permitindo que sistemas de coleta de dados mantenham sua operação e acessem informações de maneira mais eficaz e segura.

A utilização de proxies, no entanto, por si só não garante um funcionamento eficiente dos sistemas de captura de dados. Para evitar a utilização de proxies bloqueados ou inativos, estratégias de gerenciamento são necessárias, assegurando que os recursos disponíveis sejam utilizados de maneira inteligente e adaptativa. Dessa forma, a seleção apropriada dos proxies disponíveis se torna essencial para a continuidade e desempenho otimizado da captura de dados.

A. Funcionamento dos Proxies

Os proxies atuam como intermediários entre o cliente e a internet, mascarando o endereço IP original do usuário. Quando um cliente faz uma requisição para acessar um site, essa requisição é primeiro enviada ao servidor proxy, que então encaminha a solicitação ao destino final [6]. O site alvo responde ao proxy, que por sua vez retransmite a resposta ao cliente. Esse processo permite ocultar a identidade real do solicitante e distribuir acessos por diferentes endereços IP [7], reduzindo o risco de bloqueios. A Fig. 1 ilustra esse funcionamento.

B. Principais Abordagens Utilizadas Atualmente

Atualmente, diversas abordagens são empregadas para mitigar os desafios enfrentados na captura de dados em larga escala [4], [5]. Contudo, na literatura acadêmica, não existem ferramentas propostas especificamente para solucionar esse problema. As soluções atualmente disponíveis são desenvolvidas por fornecedores de proxies ou discutidas em sites e fóruns técnicos. As principais abordagens incluem:

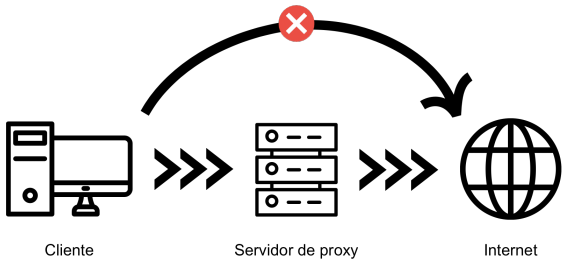

Figura 1. Esquema de funcionamento de um proxy, demonstrando o tráfego da requisição do cliente para a internet através do servidor proxy.

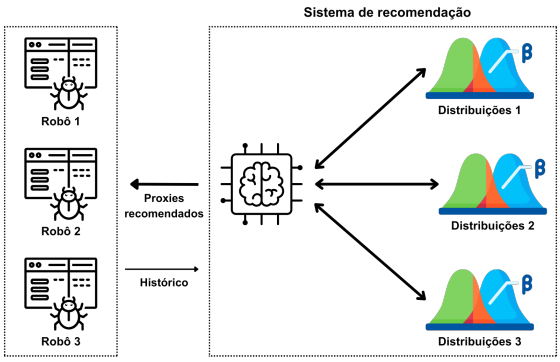

Figura 2. Esquemático do objetivo do projeto, sistema de recomendação retornando proxies adequados para os robôs.

- **Listas de Proxies**: Utilização de listas predefinidas de proxies, onde os endereços são utilizados em sequência [8].
- **Proxies Dinâmicos**: Serviços que fornecem proxies rotativos automaticamente, alterando endereços IP periodicamente [9].
- **Técnicas de Randomização**: Algumas abordagens utilizam variações aleatórias de uma lista de proxies, aplicando mudanças nos padrões de requisição para reduzir a chance de bloqueios, simulando, portanto, um comportamento mais próximo ao humano [10].

Embora essas estratégias sejam amplamente utilizadas, elas possuem limitações intrínsecas que comprometem a eficiência em larga escala. A principal delas é a falta de adaptação ao contexto: os mesmos proxies são utilizados independentemente do domínio-alvo, ignorando que cada site possui mecanismos de defesa distintos. Isso torna o desempenho de métodos como a rotação simples imprevisível e altamente variável, podendo ser aceitável para um alvo e completamente ineficaz para outro.

Além disso, tais abordagens não aprendem com o histórico de eventos, insistindo no uso de proxies com baixo desempenho e exigindo intervenções manuais para removê-los. Essa incapacidade de considerar a performance temporal e a saúde de cada proxy individualmente representa uma lacuna significativa, que este trabalho se propõe a resolver com um modelo adaptativo.

É crucial destacar que, embora a gestão de proxies seja uma necessidade prática e amplamente discutida em ambientes comerciais e fóruns técnicos, ela representa um campo notavelmente subexplorado na literatura acadêmica. As soluções existentes são, em sua maioria, de natureza industrial, focadas em oferecer um serviço funcional, mas sem a profundidade de uma modelagem adaptativa e contextual.

C. Objetivo do Projeto

Diante dos desafios mencionados anteriormente, este trabalho tem como objetivo desenvolver um sistema de recomendação de proxies que seja escalável, eficiente e adaptativo. A proposta busca melhorar a operacionalidade de sistemas de captura de dados em massa, reduzindo a ocorrência de bloqueios e indisponibilidade de proxies. A abordagem proposta se diferencia das soluções existentes ao utilizar um modelo probabilístico para avaliar e recomendar proxies com base em seu desempenho passado e nas condições atuais da rede.

D. Público-Alvo e Contexto de Aplicação

O sistema de recomendação proposto é voltado a profissionais e organizações que dependem de web scraping e captura de dados automatizada em diversos cenários. Por exemplo, pesquisadores e desenvolvedores podem empregar a solução em estudos e investigações científicas; empresas de tecnologia podem utilizá-la para análise de mercado e otimização de processos; e provedores de proxies podem melhorar sua infraestrutura de distribuição de endereços. Além disso, o sistema se mostra relevante em áreas como comércio eletrônico, monitoramento de redes sociais, coleta de notícias e análise de informações governamentais, evidenciando sua adaptabilidade a diferentes necessidades de captura de dados em larga escala.

II. Metodologia

Nesta seção, são abordados os principais conceitos teóricos que fundamentam o sistema de recomendação proposto e como eles foram implementados, incluindo a modelagem Bayesiana e as técnicas utilizadas para avaliar e selecionar proxies de forma eficiente.

A. Modelagem Bayesiana

A modelagem Bayesiana é um método probabilístico que permite a tomada de decisões sequenciais com base na incerteza e na adaptação contínua a novas observações [14]. O sistema de recomendação proposto emprega uma abordagem inspirada no Classificador Bayesiano, utilizando a amostragem de Thompson (Thompson Sampling) para estimar a qualidade de proxies em diferentes contextos [15]. Esse método possibilita equilibrar exploração e explotação, garantindo que tanto novos proxies quanto aqueles com melhor desempenho sejam utilizados de maneira eficiente.

O modelo se baseia na inferência Bayesiana para ajustar dinamicamente as probabilidades associadas ao desempenho dos proxies, atualizando-as continuamente conforme novos dados são coletados [11]. Essa abordagem permite lidar de forma robusta com cenários onde os dados são escassos ou esparsos, tornando-se mais eficiente em comparação a métodos tradicionais de seleção de proxies [12].

Dentre as principais vantagens dessa abordagem em relação a outros métodos, como a Filtragem Colaborativa, Máquina de Boltzmann, Aprendizagem por Reforço, Fatoração Matricial e Autoencoders [13], destacam-se:

- **Oferece exploração e explotação equilibradas:** O método balanceia a necessidade de se explorarem diferentes opções de proxies e a necessidade de aproveitar os melhores resultados obtidos até o presente;
- **Provê flexibilidade de adaptação a mudanças:** o método bayesiano possui uma natureza adaptativa, podendo se ajustar a mudanças nos padrões de bloqueios dos proxies ao longo do tempo, por meio de atualização contínua de suas estimativas de probabilidade [14];
- **Tem menos sensibilidade a hiper parâmetros:** Alguns métodos, como a Máquina de Boltzmann, Aprendizagem por Reforço, Fatoração Matricial e Autoencoders, exigem conjuntos de hiper parâmetros com valores adequados para proverem desempenho adequado, o que pode exigir conhecimento especializado para os ajustarem. Por sua vez, o método bayesiano requer menos ajustes, o que facilita sua implementação e uso prático;
- **Apresenta grande eficiência computacional:** A abordagem Bayesiana é computacionalmente eficiente [12]. Embora seja um algoritmo estocástico, ela não requer grandes recursos computacionais, o que a torna escalável para lidar com grandes volumes de dados e cenários em tempo real.

B. Distribuição Beta

A distribuição beta (conforme a equação 1) desempenha um papel fundamental no contexto do ranqueamento bayesiano, especialmente quando se lida com variáveis aleatórias contínuas definidas no intervalo [0, 1], como proporções ou probabilidades. Sua relevância se destaca por várias razões, o que justifica sua escolha no objeto em análise [14].

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \tag{1}$$

onde $B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$.

Flexibilidade na Modelagem de Distribuições: A distribuição beta é parametrizada por dois parâmetros positivos, α e β, que controlam a forma da distribuição. Essa parametrização permite modelar uma ampla variedade de comportamentos, desde distribuições uniformes até distribuições com forte assimetria ou concentração em valores específicos.

Propriedades Conjugadas em Inferência Bayesiana: Uma característica notável da função beta é sua propriedade de conjugação com a distribuição binomial. Em inferência Bayesiana, ao utilizar uma distribuição beta como priori para uma probabilidade desconhecida e atualizar essa crença com dados binomiais, a distribuição posterior resultante também é uma distribuição beta.

Interpretação Intuitiva dos Parâmetros: Os parâmetros α e β da distribuição beta possuem interpretações intuitivas em termos de contagens de sucessos e fracassos, respectivamente. Essa característica facilita a incorporação de conhecimento prévio no processo de modelagem Bayesiana [14].

C. Erros e Acertos

Para avaliar o desempenho dos proxies no sistema de recomendação, é necessário definir critérios claros para determinar quando um proxy foi bem-sucedido ou falhou em sua operação.

Definição de Acerto: Um acerto ocorre quando uma sessão de captura de dados utilizando um determinado proxy é bem-sucedida, ou seja, quando a sessão de captura de dados retorna o conteúdo esperado sem erros de conexão, timeouts ou códigos de status HTTP que indiquem bloqueio (ex: 403 **Forbidden**, 429 **Too Many Requests**).

Definição de Erro: Um erro ocorre quando uma sessão de captura de dados utilizando um proxy falha. Essa falha pode ser decorrente de bloqueios, indisponibilidade do proxy ou erros de requisição.

Portanto, a principal métrica de avaliação utilizada é a taxa de acerto, calculada pela razão entre o número de requisições bem-sucedidas e o total de requisições em uma janela de tempo.

D. Componente Temporal

Os impactos dos erros e acertos na recomendação de proxies variam de acordo com o tempo em que esses eventos ocorreram. Para considerar essa influência temporal, foi incorporado um componente de decaimento exponencial ao cálculo das pontuações de erro e acerto.

A pontuação de cada proxy é ajustada utilizando uma função exponencial decrescente:

$$f(t) = e^{-0,03t} \times M \tag{2}$$

onde t representa o tempo decorrido desde o evento, $0,03$ é a taxa de decaimento exponencial e M é um multiplicador (3 para erros e 1 para acertos).

Dessa forma, as equações para a penalização dos erros e acertos são:

$$P_{erro}(t) = 3 \times e^{-0,03t}, \quad P_{acerto}(t) = 1 \times e^{-0,03t} \tag{3}$$

A Fig. 3 mostra a evolução das pontuações ao longo do tempo.

E. Cálculo de Pontuações

A pontuação de cada proxy é determinada a partir do somatório ponderado dos erros e acertos registrados ao longo do tempo:

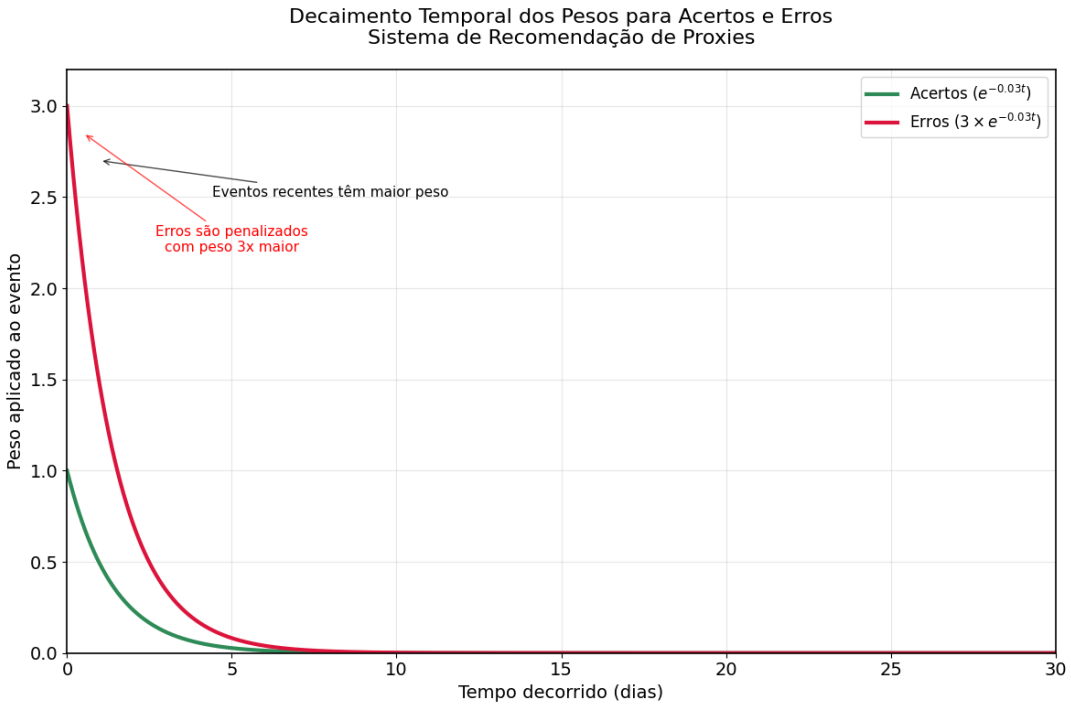

Figura 3. Decaimento temporal dos pesos aplicados para eventos de acertos e erros ao longo do tempo.

$$\alpha = \sum_{i=1}^{n} e^{-0,03t_i}, \quad \beta = \sum_{j=1}^{m} 3 \times e^{-0,03t_j} \tag{4}$$

onde t_i representa o tempo decorrido desde o evento de acerto i e t_j representa o tempo decorrido desde o evento de erro j.

Para garantir que proxies recém-inseridos ou que não tenham sido utilizados recentemente ainda possam ser avaliados, são atribuídos valores iniciais específicos. Se um proxy for recém-inserido ou se ambas as pontuações α e β forem inferiores a 1, os valores são ajustados para $\alpha = 100$ e $\beta = 1$.

A Fig. 4 ilustra a curva da distribuição da probabilidade considerando os valores iniciais $\alpha = 100$ e $\beta = 1$.

O sistema implementa um mecanismo de decaimento temporal que prioriza eventos mais recentes ao calcular os parâmetros da distribuição Beta. A Fig. 3 apresenta o comportamento das funções de peso aplicadas aos eventos de acertos e erros ao longo do tempo. Conforme demonstrado, eventos mais antigos recebem pesos progressivamente menores, enquanto erros são penalizados com peso 3 vezes maior que os acertos, garantindo que proxies com histórico de falhas recentes sejam menos propensos a serem selecionados.

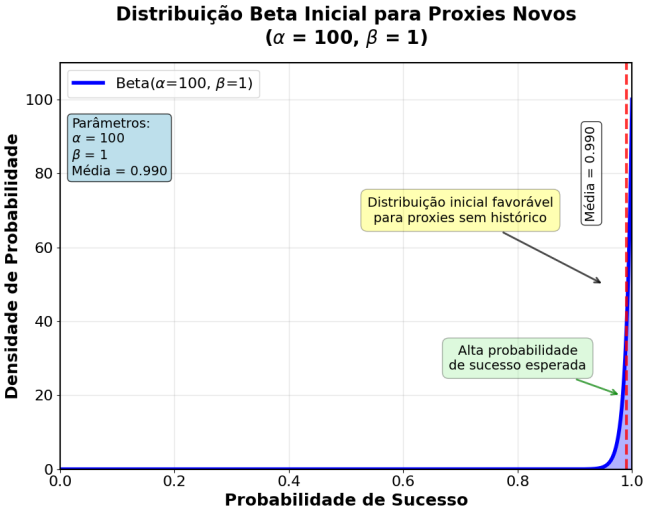

Figura 4. Curva da distribuição de probabilidade com $\alpha = 100$ e $\beta = 1$.

F. Funcionamento Geral do Sistema

O sistema de recomendação segue uma abordagem probabilística para selecionar o proxy mais adequado. O processo consiste em:

1) Iteração sobre todos os proxies disponíveis;
2) Para cada proxy, percorrer todos os registros de erros e acertos associados ao robô;
3) Calcular as pontuações α e β utilizando a função exponencial de decaimento temporal apresentada na Fig. 3;
4) Gerar um valor aleatório para o proxy com base na distribuição $Beta(\alpha, \beta)$;
5) Selecionar o proxy com a maior pontuação.

Input: Lista de proxies, robô
Output: Proxy recomendado
$melhores\_proxies \leftarrow \emptyset$;
for *cada* $proxy$ *em* $lista\_proxies$ **do**
  $\alpha \leftarrow 0$, $\beta \leftarrow 0$;
  for *cada* $registro$ *em* $proxy.registros(robo)$ **do**
    if $registro == acerto$ **then**
      $\alpha \leftarrow \alpha + e^{-0{,}03 * tempo\_decorrido(registro)}$;
    else
      $\beta \leftarrow \beta + 3 * e^{-0{,}03 * tempo\_decorrido(registro)}$;
    end
  end
  if $\alpha < 1$ *e* $\beta < 1$ **then**
    $\alpha \leftarrow 100$, $\beta \leftarrow 1$;
  end
  $pontuacao \leftarrow sorteio\_beta(\alpha, \beta)$;
  Adicionar $(proxy, pontuacao)$ em $melhores\_proxies$;
end
return proxy com maior pontuação;

Algoritmo 1: Selecionar Proxy Ótimo

O Algoritmo 1 ilustra a lógica de seleção.

G. Desenvolvimento de Ambiente de Testes

O sistema foi desenvolvido e testado utilizando as seguintes especificações técnicas e configurações para garantir a reprodutibilidade dos experimentos:

1) Infraestrutura e Software:

- **Linguagem de programação:** Python 3.9.7
- **Bibliotecas principais:** NumPy 1.21.0, SciPy 1.7.0, Flask 2.0.1
- **Banco de dados:** PostgreSQL 13.4
- **Ambiente de execução:** Docker containers
- **Sistema operacional:** CentOS Linux 7 (Core)
- **Processador:** Intel Xeon Platinum 8275CL CPU @ 3.00GHz
- **Memória RAM:** 32GB
- **Armazenamento:** SSD NVMe 500GB

2) Configuração dos Robôs de Teste:

- **Quantidade:** 72 robôs automatizados
- **Tecnologia:** Desenvolvidos em C# .NET Framework
- **Domínios-alvo:** Cada robô foi configurado para acessar domínios públicos distintos na internet
- **Distribuição de requisições:** Tempos aleatórios entre requisições, variando de 5 a 30 segundos
- **Timeout de requisição:** 15 segundos por tentativa
- **User-agents:** Rotação aleatória entre diferentes user-agents de navegadores populares

3) Configuração dos Proxies:

- **Provedores:** 4 diferentes fornecedores de proxies comerciais
- **Tipos:** Proxies HTTP/HTTPS fixos residenciais
- **Localização geográfica:** Distribuídos globalmente (América do Norte, Europa, Ásia)
- **Pool total:** Aproximadamente 40 endereços IP únicos

4) Parâmetros do Sistema de Recomendação:

- **Taxa de decaimento temporal:** 0,03 (conforme equações apresentadas)
- **Multiplicador de penalização:** 3x para erros, 1x para acertos
- **Valores iniciais da distribuição Beta:** $\alpha = 100$, $\beta = 1$
- **Frequência de atualização:** Tempo real após cada requisição
- **Janela de histórico:** 7 dias de eventos anteriores

III. Resultados

Os testes foram conduzidos entre **7 de setembro de 2024 e 15 de setembro de 2024**, utilizando **72 robôs** operando com **4 diferentes provedores de proxies**. Inicialmente, foi empregado um modelo de rotação simples, sendo substituído pelo sistema de recomendação às **23 horas do dia 10 de setembro**.

Para cada robô existem diferentes curvas de distribuição de probabilidades. A Fig. 5, referente ao robô 19, exemplifica isso. É importante notar que essas curvas variam ao longo do tempo, refletindo a adaptação dos pesos conforme novos dados são processados. As Fig. 5 e Fig. 6 mostram a diferença das curvas após um intervalo de 24 horas, indicando essa adaptabilidade.

A Fig. 7 apresenta a proporção de erros e acertos ao longo do tempo, evidenciando uma clara divisão entre os períodos antes e depois da aplicação do sistema de recomendação.

As métricas gerais indicam que o sistema de recomendação melhorou significativamente a taxa de acertos:

- **Rotação Simples:** Média de 47%, mediana de 49%, desvio padrão de 6%;
- **Sistema de Recomendação:** Média de 77%, mediana de 77%, desvio padrão de 7%.

Casos específicos, como o do **robô 15** (Fig. 8), evidenciam ganhos ainda mais expressivos. A taxa de acertos aumentou de **15% para 85%**, e a mediana passou de **16% para 85%**.

O desempenho excepcional do **robô 15**, que saltou de 15% para 85% de acertos, pode ser atribuído a uma combinação de fatores:

- **Especificidade do Domínio-Alvo**: O site acessado pelo robô 15 possivelmente implementava um sistema de bloqueio rigoroso que identificava e bloqueava rapidamente padrões de requisição vindos de um mesmo grupo de IPs.
- **Adaptação Dinâmica**: O sistema de recomendação, ao analisar o histórico de erros e acertos em tempo real, foi capaz de identificar quais provedores de proxy eram ineficazes para aquele domínio específico e priorizar os proxies que conseguiam contornar as restrições. A rotação simples, por sua vez, continuava a insistir em proxies já bloqueados, resultando em uma taxa de sucesso muito baixa.

IV. Conclusão

Os resultados obtidos demonstram a eficácia do sistema de recomendação de proxies em relação ao modelo tradicional de

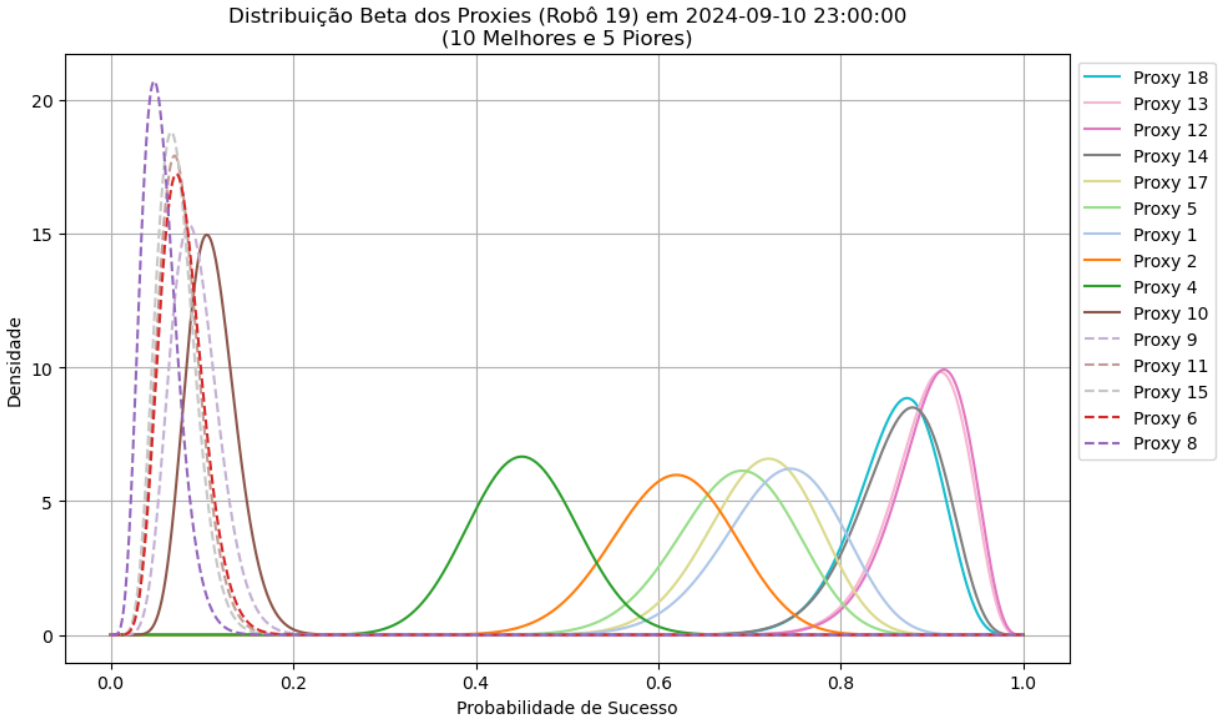

Figura 5. Distribuição Beta do robô 19 (Anterior à utilização do sistema de recomendação).

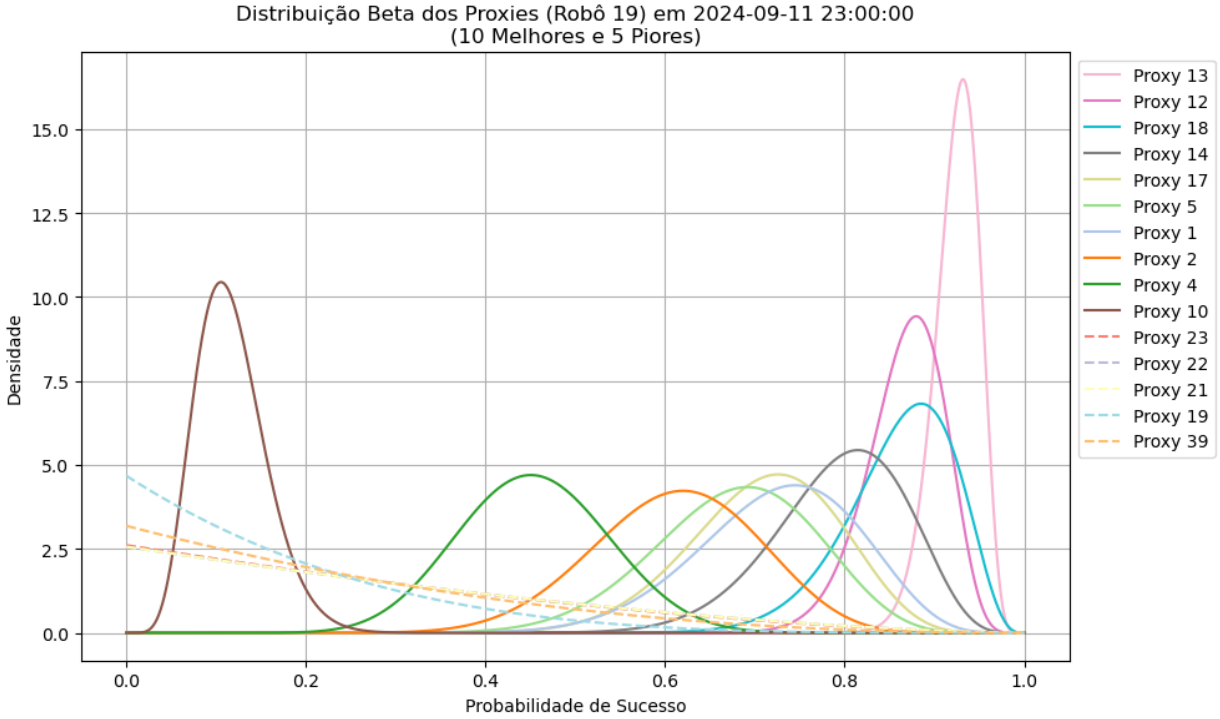

Figura 6. Distribuição Beta do robô 19 (Posterior à utilização do sistema de recomendação).

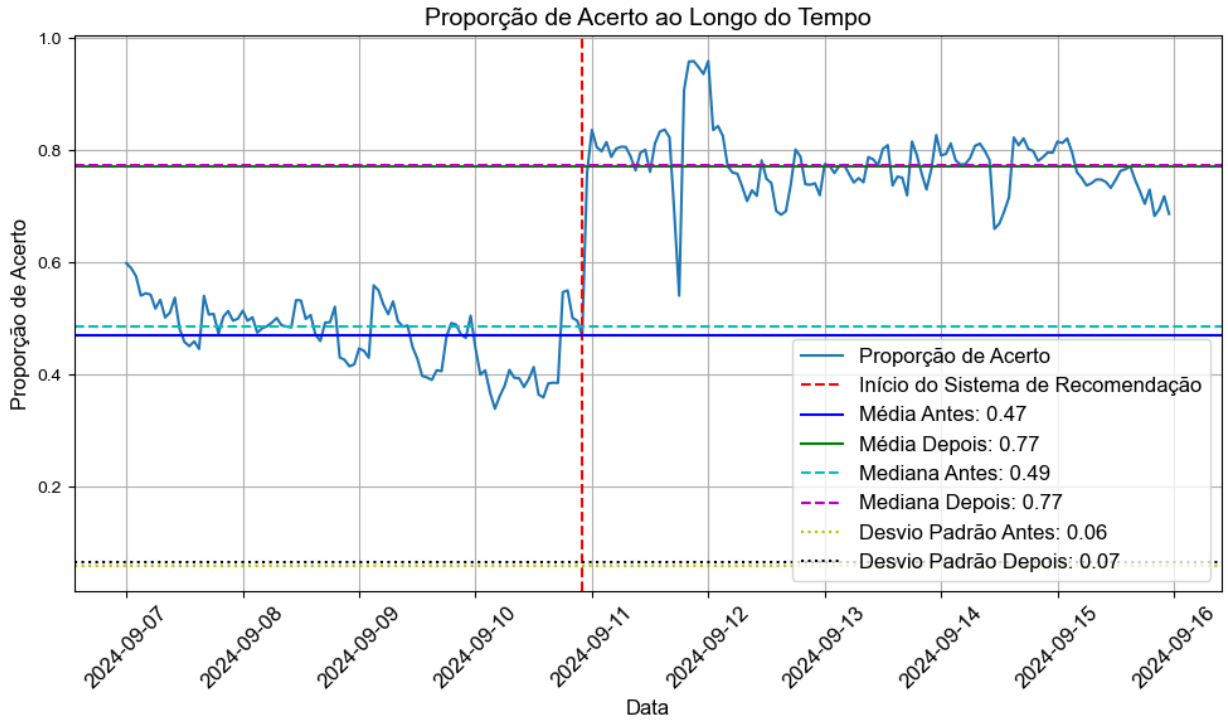

Figura 7. Proporção de erros e acertos ao longo do tempo.

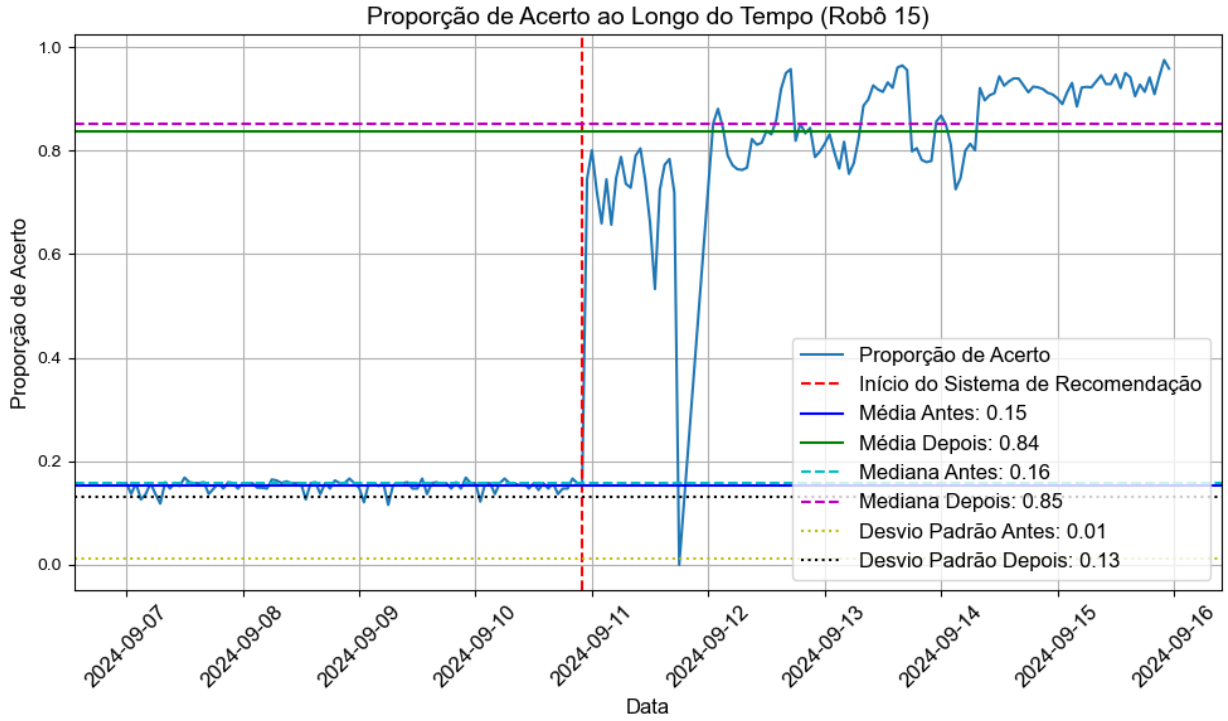

Figura 8. Evolução da taxa de acertos do robô 15.

rotação simples. A análise das métricas indicou um aumento significativo na taxa de acertos, passando de uma média de **47% para 77%**, enquanto a mediana subiu de **49% para 77%**. A estabilidade do sistema foi mantida, com desvio padrão de **6%** na abordagem tradicional e **7%** no modelo de recomendação.

Casos específicos evidenciaram ainda maiores ganhos, mostrando que a personalização do uso dos proxies melhora a eficiência da coleta de dados, reduzindo a necessidade de intervenção manual e garantindo maior disponibilidade de proxies ativos.

A adaptação dinâmica proporcionada pela modelagem Bayesiana demonstrou ser essencial para evitar a utilização de proxies bloqueados ou indisponíveis, melhorando a continuidade das capturas de dados e otimizando o desempenho dos robôs. As distribuições Beta geradas ao longo do tempo confirmaram que o sistema é capaz de se ajustar às variações de desempenho dos proxies, garantindo decisões mais precisas.

Os resultados reafirmam a importância de sistemas inteligentes para a gestão de proxies, evidenciando que o uso de técnicas probabilísticas pode trazer benefícios significativos para aplicações que dependem de web scraping em larga escala [3].

No entanto, apesar dos resultados positivos, o sistema de recomendação possui limitações:

- **Cold Start para Novos Robôs**: O sistema depende de um histórico de eventos (erros e acertos) para modelar as distribuições de probabilidade. Um robô recém-configurado para um novo domínio-alvo passa por uma fase inicial de "aprendizagem", onde a taxa de acertos pode ser baixa até que dados suficientes sejam coletados para uma recomendação mais precisa. A atribuição de valores iniciais α=100 e β=1 mitiga parcialmente esse problema, conferindo um viés otimista inicial, mas não o elimina completamente.
- **Dependência da Qualidade dos Proxies**: A eficácia do sistema está intrinsecamente ligada à qualidade do pool de proxies disponíveis. Se todos os proxies de todos os provedores forem de baixa qualidade ou estiverem bloqueados para um determinado site, o sistema poderá otimizar a seleção entre as piores opções, mas a taxa de sucesso geral permanecerá baixa.

Uma linha promissora para trabalhos futuros consiste em uma análise comparativa de diferentes distribuições de probabilidade para a modelagem do desempenho dos proxies. Embora a distribuição Beta, utilizada neste estudo, tenha se mostrado eficaz e sido escolhida por sua flexibilidade e propriedades de conjugação, não há garantia de que ela represente a dinâmica de sucesso e falha de um proxy de forma ótima em todos os cenários. Portanto, a exploração de alternativas poderia refinar ainda mais o modelo.

Referências

[1] A. Chatterjee et al., "Data-driven decision making in business: A comprehensive survey," *Journal of Business Analytics*, vol. 15, no. 2, pp. 45-67, 2023.

[2] D. Reinsel, J. Gantz, and J. Rydning, "The digitization of the world from edge to core," *IDC White Paper*, Nov. 2017.

[3] L. A. Schintler and C. L. McNeely, Eds., "Web scraping," in *Encyclopedia of Big Data*. Springer International Publishing, May 2017, pp. 1-3. [Online]. Available: https://doi.org/10.1007/978-3-319-32001-4_483-1

[4] R. Baumgartner, G. Gottlob, and M. Herzog, "Scalable web data extraction for online market intelligence," *Proc. VLDB Endowment*, vol. 2, pp. 1512-1523, 2009.

[5] C. Lotfi, S. Srinivasan, M. Ertz, and I. Latrous, "Web scraping techniques and applications: A literature review," in *Advances in Data Science and Management*. Springer, 2021, pp. 381-394.

[6] M. Oliveira, "Proxy servers and network security," *Computer Networks*, vol. 54, no. 12, pp. 2045-2058, 2010.

[7] R. Gunawardana and S. Chen, "IP masking techniques for web scraping," *Proc. IEEE Conf. Network Security*, pp. 123-130, 2023.

[8] "Free proxy lists and rotation techniques," *Web Scraping Tools*, 2023. [Online]. Available: https://example.com

[9] J. Belurk, "Dynamic proxy rotation systems," *Technical Report*, 2023.

[10] F. Geigl et al., "Randomization techniques in web scraping," *Proc. Int. Conf. Web Intelligence*, pp. 45-52, 2015.

[11] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1-22, May 2012.

[12] Y. Zhang and J. Koren, "Efficient Bayesian hierarchical user modeling for recommendation system," *Proc. ACM SIGIR*, pp. 47-54, 2007.

[13] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261-273, 2015.

[14] A. Gelman et al., *Bayesian Data Analysis*, 3rd ed. Chapman and Hall/CRC, 2013.

[15] D. J. Russo et al., "A tutorial on Thompson sampling," *Foundations and Trends in Machine Learning*, vol. 11, no. 1, pp. 1-96, 2018.