Abrindo a Caixa-Preta – Aplicando IA Explicável para Aprimorar a Detecção de Sequestros de Prefixo

**Adriano B. Carvalho, Brivaldo A. da Silva Jr,
Carlos Alberto da Silva e Ronaldo A. Ferreira**

[1]Universidade Federal de Mato Grosso do Sul (UFMS)
{adriano.bastos,brivaldo.junior,carlos.silva,ronaldo.ferreira}@.ufms.br

Abstract. *The BGP protocol lacks native security mechanisms, allowing malicious actors to hijack prefixes. Recent studies use machine learning to detect these hijacks, but the models are black boxes, making it difficult to determine if they use the most suitable features. This work applies eXplainable Artificial Intelligence (XAI) techniques to evaluate and improve a recently proposed model for prefix-hijack detection. Through extensive analysis of the original model with 28 features, we developed two models with 11 and 5 features that yield results without statistical difference to the complete model while reducing processing time by over 30% and storage space by over 59%.*

Resumo. *O protocolo BGP não possui mecanismos nativos de segurança, permitindo que atacantes sequestrem prefixos. Trabalhos recentes utilizam aprendizado de máquina para identificar esses sequestros, mas os modelos são caixas-pretas, tornando inviável determinar se utilizam as features mais adequadas. Este trabalho aplica técnicas de Inteligência Artificial Explicável (XAI) para avaliar e melhorar um modelo de detecção de sequestros de prefixo proposto recentemente. A partir de uma análise extensiva do modelo original com 28 features, foram criados dois modelos com 11 e 5 features, que produzem resultados sem diferenças estatísticas do modelo completo, mas reduzem o tempo de processamento em mais de 30% e o espaço de armazenamento em mais de 59%.*

1. Introdução

A Internet é formada por redes que são administradas de forma independente e que se conectam para prover conectividade para seus usuários. Uma rede, ou um conjunto de redes, administrada por uma determinada entidade é denominada de Sistema Autônomo (AS – *Autonomous System*). Um AS define suas próprias políticas de roteamento e acordos com outras redes visando implementar um serviço de entrega de pacotes fim-a-fim. Cada AS possui um conjunto de endereços IP, que é representado por um ou mais prefixos de rede, e um número que o identifica denominado ASN (*Autonomous System Number*).

Os sistemas autônomos na Internet trocam informações de roteamento utilizando o protocolo BGP (*Border Gateway Protocol*) [Rekhter and et al. 2006], que oferece vários mecanismos para suportar políticas complexas de roteamento. A construção de uma rota em BGP começa quando um *AS origem* anuncia um prefixo IP aos seus ASes vizinhos. As rotas são então propagadas por meio de mensagens de atualização BGP entre os ASes. O *AS-path* é a sequência de ASes atravessados pela rota até atingir o AS origem. O *AS-path* é empregado pelo BGP para prevenir *loops* e também no processo de seleção da

melhor rota para um prefixo de destino. BGP escolhe a melhor rota para um destino (*i.e.,* prefixo) utilizando uma sequência de critérios que inclui o atributo de preferência local (*LocalPref*), origem da rota, comprimento de *AS-path*, etc. [Rekhter and et al. 2006].

Apesar de toda sua flexibilidade, BGP não implementa nativamente técnicas de validação e autenticação de anúncios de rota recebidos. A ausência de mecanismos de segurança em BGP permite que um AS anuncie prefixos de outros ASes e altere os anúncios das rotas antes de repassá-las, incluindo modificações no *AS-path*. Alterações maliciosas no *AS-path* podem ocultar a origem ilegítima de um prefixo e *sequestrar* o tráfego destinado a ele [Cho et al. 2019]. As técnicas de sequestro de prefixo têm se aprimorado e agora também são usadas para crimes. Em 3 de fevereiro de 2022, por exemplo, um sequestro de prefixo permitiu que hackers roubassem cerca de 1,9 milhão de dólares em criptomoedas da plataforma KlaySwap [Siddiqui 2022]. O sequestro intencional de prefixos pode ter vários objetivos, como divulgar páginas maliciosas (*phishing*), enviar spams, roubar informações, falsificar certificados digitais, entre outros [Birge-Lee et al. 2018, Testart et al. 2019, Cho et al. 2019].

O número crescente de problemas de segurança com BGP [Lychev et al. 2016], que tem causado indisponibilidades frequentes de partes da Internet, levou pesquisadores e operadores de rede a propor vários mecanismos para melhorar sua segurança. Entre eles estão o RPKI (*Resource Public Key Infrastructure*) [Bush and Austein 2017], que oferece uma infraestrutura de chaves públicas para autorização e validação de rotas, BGP-Sec [Lepinski and Sriram 2017], que permite a assinatura de todos os anúncios de rota, e MANRS (*Mutually Agreed Norms for Routing Security*) [Freedman et al. 2019], que define um conjunto de ações que os ASes devem implementar, como filtragem e validação de rotas, para evitar sequestros de prefixo, vazamento de rotas e falsificação (*spoofing*) de endereços. Entretanto, essas propostas ainda enfrentam resistências e não são amplamente implementadas, deixando o sistema de roteamento da Internet vulnerável a ataques e falhas de configuração.

Como ainda não é possível eliminar sequestros de prefixo na Internet, diversos trabalhos recentes se concentram em detectá-los para alertar os operadores e assim mitigar o problema [Lad et al. 2006, Shi et al. 2012, Sermpezis et al. 2018, Qin et al. 2022, Holterbach et al. 2024]. Muitas dessas abordagens utilizam aprendizado de máquina devido à grande quantidade de dados de roteamento disponíveis em coletores públicos [Meyer 1997] e informações sobre ASes em bancos de dados de registros regionais (RIRs, *Regional Internet Registries*) [Merit Network, Inc 2024], que são usados para treinar os modelos[Testart et al. 2019, Shapira and Shavitt 2022, Holterbach et al. 2024]. No entanto, esses modelos são caixas-pretas (*black boxes*) e estão se tornando cada vez mais complexos, com um grande número de *features* que consomem muito tempo para serem calculadas e nem sempre melhoram o desempenho.

Sem uma análise detalhada do funcionamento de um modelo de aprendizado de máquina e de como suas inferências ocorrem, *features* de pouca ou nenhuma relevância podem ser utilizadas, resultando em desperdício de espaço de armazenamento e tempo de processamento para cálculo das *features*, treinamento do modelo e inferência. Alguns trabalhos recentes propõem o uso de Inteligência Artificial Explicável (XAI, do inglês *eXplainable Artificial Intelligence*) para se entender e explicar as decisões de um modelo caixa-preta [Ribeiro et al. 2016, Jacobs et al. 2022]. Trustee [Jacobs et al. 2022],

por exemplo, é uma ferramente de explicabilidade global que gera um modelo interpretável na forma de uma árvore de decisão que captura e explica o comportamento de um modelo caixa-preta. Diferente de métodos tradicionais de seleção de *features*, a saída de um método de explicabilidade global (*i.e.,* árvore de decisão) pode ser analisada por um operador de redes para verificar se o modelo está tomando decisões coerentes com o conhecimento existente do domínio do problema.

Este trabalho utiliza técnicas e ferramentas de XAI para avaliar e aperfeiçoar um modelo caixa-preta para detecção de sequestros de prefixo proposto recentemente e considerado como o estado-da-arte [Holterbach et al. 2024]. O modelo, utilizado no sistema DFOH (*Detects Forged Origin Hijacks*) e descrito na Seção 3, utiliza 28 *features* extraídas de anúncios BGP [Meyer 1997], dados de registros regionais [Merit Network, Inc 2024], bases da CAIDA [CAIDA 2015] e do PeeringDB [PeeringDB 2010]. Na análise de funcionamento do DFOH, os autores obtiveram uma taxa de verdadeiros positivos de 0,909 e de falsos positivos de 0,019, resultando numa precisão de 0,9795 [Holterbach et al. 2024].

Uma avaliação extensiva do modelo de DFOH com a ferramenta Trustee possibilitou a criação de dois modelos reduzidos, um com 11 e outro com apenas 5 das 28 *features* originais (Seção 4). Validações dos modelos reduzidos, utilizando os métodos descritos em [Holterbach et al. 2024], mostraram que os modelos reduzidos produziram resultados de precisão e *recall* dentro do mesmo intervalo de confiança do modelo completo, não apresentando, portanto, diferenças estatisticamente significativas. Por exemplo, o modelo com 5 *features*, apesar de uma redução superior a 80% nas *features*, teve uma diferença na média do F1-Score de apenas 0,0001 (Seção 6). O número reduzido de *features* nos novos modelos diminuiu os tempos de processamento para cálculo das *features*, treinamento e inferência em mais de 30% e 37% para os modelos com 11 e 5 *features*, respectivamente. O espaço necessário para armazenar as *features* também reduziu em mais de 59% e 71% para os modelos com 11 e 5 *features*, respectivamente.

Utilizando os três modelos para inferir as classes de novos enlaces surgidos ao longo de um período de 40 dias, foram observadas diferenças de 6% e 7,8% entre o modelo original e os modelos reduzidos com 11 e 5 *features*, respectivamente. Como não há uma verdade absoluta (*ground truth*) para esses novos enlaces, aqueles que ainda foram observados nos coletores em períodos posteriores (superiores a um mês) foram considerados legítimos. A justificativa para essa classificação é que sequestros de prefixo são realizados por períodos curtos de tempo [Testart et al. 2019], ou seja, enlaces suspeitos, que podem indicar sequestros de prefixo, só são observados nos coletores por períodos curtos de tempo. Nessa avaliação, o modelo com 11 *features* gerou um número menor de falsos positivos ao identificar corretamente um número maior de enlaces legítimos. Por outro lado, o modelo com 5 *features* obteve precisão e *recall* superiores ao modelo original para os enlaces suspeitos. Esses resultados surpreendentes sugerem que o modelo original contém *features* que não contribuem na separação das classes e em alguns casos até prejudicam o desempenho do modelo. A Seção 6 apresenta uma discussão qualitativa das *features* utilizados nos modelos e possíveis motivos para melhorias de desempenho dos modelos reduzidos. Uma vantagem adicional dos modelos reduzidos é que, por serem computacionalmente mais leves, eles podem ser retreinados com maior frequência e assim capturar mais rapidamente mudanças nas distribuições dos dados.

2. Trabalhos Relacionados

Sequestros de prefixo têm sido estudados e analisados em diversos trabalhos ao longo do tempo [Liu et al. 2012, Shi et al. 2012, Sermpezis et al. 2018, Cho et al. 2019, Testart et al. 2019, Holterbach et al. 2024]. Esta seção apresenta os principais trabalhos que buscam caracterizar os ASes responsáveis por sequestro bem como identificar sequestros no plano de dados, no plano de controle ou em ambos.

2.1. Análise de Sequestros de Prefixo

[Testart et al. 2019] estudaram o comportamento de ASes que sequestram prefixos frequentemente e os denominaram de *sequestradores seriais*. O trabalho mostra que esses ASes originam muitos prefixos únicos, mas por curtos períodos. Além disso, eles ficam ausentes das tabelas de roteamento com frequência, ao contrário dos ASes legítimos, que normalmente anunciam ininterruptamente os mesmos prefixos por mais de um ano.

Os sequestros de prefixos podem ser classificados pela forma, prefixo, objetivo ou efeito. Quando um AS sequestra um prefixo e o anuncia como seu, ou seja sem nenhum ASN antes dele no *AS-path*, é um sequestro Tipo-0 [Sermpezis et al. 2018] ou uma mudança de origem da rota [Cho et al. 2019]. Nesse tipo de sequestro, múltiplas origens (MOAS, *Multiple Origem Autonomous System*) para um prefixo são comuns. Ele pode ser evitado com RPKI, que bloqueia origens inválidas, mas a proteção completa depende da ampla implementação de RPKI nos equipamentos de borda dos ASes. Entretanto, sequestradores de prefixo forjam *AS-paths* para evitar detecção por sistemas que usam RPKI. Eles inserem o ASN do proprietário do prefixo ou outros ASes no início do *AS-path* para ofuscar sua identidade. Esse tipo de sequestro é chamado de Tipo-X [Sermpezis et al. 2018], em que X é o número de ASes inseridos, ou sequestro com manipulação do *AS-path* [Cho et al. 2019]. A desvantagem é que quanto maior o *AS-path* forjado menor será o alcance do sequestro, já que um dos critérios de desempate de BGP para seleção da melhor rota é o comprimento do *AS-path*.

Os sequestros podem diferir de três maneiras quanto ao prefixo sequestrado: quando o prefixo tem o mesmo tamanho do anúncio legítimo e compete pela rota, quando é mais específico e tem prioridade na tabela de roteamento, ou quando envolve um prefixo não alocado ou anunciado pelo proprietário (*IP prefix squatting* [Sermpezis et al. 2018]). Geralmente, esses sequestros são não intencionais, resultantes de erros de configuração ou digitação, e causam grande impacto no AS vítima.

2.2. Identificação de Sequestros de Prefixo

A identificação de sequestros no plano de dados envolve a análise de tráfego de pacotes sem depender das tabelas de roteamento BGP. LDC (*Load Distribution Change*) [Liu et al. 2012] monitora mudanças no volume de tráfego para um destino específico, pois um aumento no tráfego para o sequestrador e uma redução para a vítima podem indicar um sequestro. No entanto, essa abordagem só identifica o sequestrador com precisão se o monitor estiver diretamente conectado a ele. Oscilloscope [Bühler et al. 2023], por outro lado, analisa mudanças no RTT (*Round Trip Time*) para um destino, pois mudanças no RTT podem sinalizar uma alteração no caminho devido a um sequestro, mas, como o LDC, não identifica precisamente o sequestrador.

Sequestros de prefixo também podem ser identificados com informações do plano de controle devido a existência de vários coletores de informações BGP,

como os projetos Route Views [Meyer 1997] e RIPE RIS [Mcgregor et al. 2010]. [Milolidakis and et al. 2023] mostram que dificilmente atividades maliciosas com o BGP não são registradas nos coletores. Segundo [Sermpezis et al. 2018], apenas ataques que afetam menos de 1% da Internet podem passar despercebidos pelos coletores.

O sistema PHAS (*Prefix Hijack Alert System*) [Lad et al. 2006] monitora os anúncios dos coletores e envia um e-mail para um usuário cadastrado quando detecta uma nova origem para um prefixo. Já o Artemis (*Automatic and Real-Time dEtection and MItigation System*) [Sermpezis et al. 2018] busca detectar todos os tipos de sequestros, incluindo Tipo-0 e Tipo-1. Entretanto, ele identifica sequestros apenas dos prefixos do AS onde está implantado.

Os sistemas Argus [Shi et al. 2012] e Themis [Qin et al. 2022] usam o plano de controle para identificar sequestros e o plano de dados para confirmá-los, concentrando-se em sequestros que causam *blackholing*. O Argus [Shi et al. 2012] usa três módulos: o primeiro busca anomalias no plano de controle, o segundo identifica sequestros quando uma anomalia é encontrada e usa `ping` em dispositivos remotos, como servidores de rotas e *looking glasses*, para verificação, enquanto o terceiro busca endereços ativos prováveis para verificação. O Themis [Qin et al. 2022], por sua vez, aprimora o Argus com um módulo adicional que diferencia entre MOAS legítimos e ilegítimos, descartando alertas para prefixos anunciados por outro AS com autorização do proprietário.

2.3. Técnicas de Seleção de *Features*

A seleção adequada de *features* é fundamental no desenvolvimento de modelos de aprendizado de máquina. Diversas técnicas foram propostas na literatura para identificar as *features* mais relevantes de um modelo, considerando seus impactos na predição. Os trabalhos de [Hammood and Al-Musawi 2021] e [Arai et al. 2019] aplicam técnicas conhecidas de redução de *features* em modelos de detecção de anomalia em BGP. Esses modelos resultaram em reduções na quantidade de *features* em relação aos modelos originais de mais de 80%. Entretanto, as técnicas tradicionais analisam o impacto de uma *feature* de forma isolada ou sem fornecer informações suficientes para se entender o processo global de decisão do modelo. Por outro lado, a aplicação de métodos de Inteligência Artificial Explicável na seleção de *features* permite não somente a redução de *features* como também o entendimento das decisões do modelo, permitindo que um especialista no domínio do problema valide essas decisões.

3. Detecção de Sequestros de Origem Forjada

DFOH [Holterbach et al. 2024] é um sistema projetado para identificar sequestros de prefixo com origem forjada. Esses sequestros ocorrem quando um atacante manipula o *AS-path* de um anúncio para incluir o AS de origem do prefixo, fazendo o anúncio parecer legítimo. O objetivo do sistema é detectar anúncios em que a conexão entre dois ASes consecutivos no *AS-path* seja forjada. Para isso, o sistema utiliza um modelo de aprendizado de máquina do tipo caixa-preta baseado em uma floresta aleatória (*random forest*) com 28 *features* calculadas a partir de diversas fontes públicas de informação de roteamento, como coletores de rota [Meyer 1997, Mcgregor et al. 2010], bases de relacionamento da CAIDA [CAIDA 2015], informações de registros regionais [Merit Network, Inc 2024] e peeringDB [PeeringDB 2010].

Em [Holterbach et al. 2024], os autores apresentam o DFOH como o sistema mais avançado e atual para detecção de sequestros de prefixo com origem forjada e mostram que ele atinge altas taxas de precisão e *recall*, sem gerar muitos alarmes falsos.

As *features* do DFOH, mostradas na Tabela 1, são divididas em quatro categorias: *topológicas, peering, padrão do AS-path e bidirecionalidade*. Por limitação de espaço, a tabela apresenta uma breve descrição de cada *feature*, mas a descrição detalhada pode ser encontrada em [Holterbach et al. 2024]. Para calcular as *features* topológicas, DFOH constrói um grafo direcionado representando as conexões entre os ASes usando anúncios BGP coletados durante os 300 dias anteriores ao treinamento. Como DFOH não utiliza dados de todos os coletores públicos disponíveis, ele complementa o grafo topológico com informações de relacionamento entre ASes fornecidas em [CAIDA 2015]. As *features* de *peering* são calculadas a partir de dados do PeeringDB [PeeringDB 2010]. Para calcular as *features* de padrão do *AS-path*, DFOH treina uma floresta aleatória utilizando sequências de grau de AS e tamanhos de cone de clientes a partir dos ASes presentes nos *AS-paths*. O cone de clientes é o conjunto de ASes clientes diretos e indiretos de um determinado AS. O grau de um AS é calculado a partir do grafo topológico e os tamanhos de cone de clientes são obtidos do ASRank [CAIDA 2001]. A *feature* de bidirecionalidade indica se uma ligação entre dois ASes é bidirecional. Ela é construída com informações do grafo topológico e complementada com informações dos RIRs [Merit Network, Inc 2024]. Por último, o valor da *feature nb_vps* é calculado com informações dos vizinhos observados no grafo topológico e a relação dos ASes que fornecem informações aos coletores BGP públicos [Meyer 1997, Mcgregor et al. 2010].

Para treinar o modelo, DFOH gera sinteticamente mil amostras de enlaces legítimos e mil de enlaces suspeitos envolvendo grupos de ASes determinados pelo algoritmo de clusterização `K-means` [Ahmed et al. 2020] e com dados dos 60 dias anteriores ao treinamento. A ideia é criar um conjunto de amostras representativas para ambas as classes que contenham ASes de todos os tipos (*i.e.,* Tier-1, *stub*, trânsito e *multi-homed*), pois uma geração totalmente aleatória favoreceria enlaces de ASes do tipo *stub* por serem em maior quantidade na Internet. Note que o treinamento é realizado com dados dos 60 dias anteriores enquanto o grafo topológico é gerado com dados dos 300 dias anteriores. O período maior para a construção do grafo topológico visa capturar relações mais estáveis entre os ASes.

DFOH opera continuamente e retreina seu modelo diariamente para analisar todos os novos enlaces surgidos por dia. Os novos enlaces classificados como legítimos podem ser usados para retreinar o modelo nos dias subsequentes, enquanto os classificados como forjados são removidos do grafo topológico por 30 dias, permitindo que um operador de rede verifique possíveis sequestros envolvendo os ASes do enlace.

4. Desvendando o Funcionamento de DFOH com XAI

Nos últimos anos, aprendizado de máquina tem sido utilizado para resolver diversos problemas, empregando modelos cada vez mais complexos baseados em redes neurais profundas, florestas aleatórias e LLMs (*Large-Language Models*). Esses modelos são considerados caixas-pretas (*black boxes*) e geralmente são preferidos aos modelos interpretáveis, como árvores de decisão, devido ao seu desempenho superior. Entretanto, apesar da atenção que recebem, operadores de rede ainda hesitam em adotar modelos

Tabela 1. *Features* utilizadas no modelo de floresta aleatória do DFOH.

Categoria			*Feature*	Informação utilizada para computar a *feature*
Topológicas	Por AS	Centralidade	*degree_centrality_as1*	Fração dos ASes existentes conectados no AS1
			degree_centrality_as2	Fração dos ASes existentes conectados no AS2
			closeness_centrality_as1	Comprimento médio do caminho mais curto de todos os ASes até o AS1
			closeness_centrality_as2	Comprimento médio do caminho mais curto de todos os ASes até o AS2
			harmonic_centrality_as1	Média harmônica dos caminhos mais curtos do AS1 até os demais ASes
			harmonic_centrality_as2	Média harmônica dos caminhos mais curtos do AS2 até os demais ASes
		Vizinhos	*average_neighbord_degree_as1*	Valor médio da quantidade de ASes conectados aos vizinhos do AS1
			average_neighbord_degree_as2	Valor médio da quantidade de ASes conectados aos vizinhos do AS2
			eccentricity_as1	Distância máxima do AS1 até os demais ASes
			eccentricity_as2	Distância máxima do AS2 até os demais ASes
		P. Topo.	*triangles_as1*	Número de ligações em triângulo que envolvem o AS1
			triangles_as2	Número de ligações em triângulo que envolvem o AS2
			clustering_as1	Fração dos possíveis triângulos existentes que incluem o AS1
			clustering_as2	Fração dos possíveis triângulos existentes que incluem o AS2
	Par de ASes	Prox.	*jaccard*	Similaridade de Jaccard entre os vizinhos dos AS1 e AS2
			adamic_adar	Proximidade de AS1 e AS2 baseada nos vizinhos compartilhados
			preferencial_attachment	Probabilidade do AS1 e AS2 se conectar com base na quantidade de vizinhos
		Dist.	*shortest_path*	Comprimento do caminho mais curto entre AS1 e AS2 com base no histórico
Peering			*country_dist*	Países onde os ASes vizinhos são registrados
			ixp_dist	PTT (IXP) onde os ASes vizinhos possuem conexões
			facility_fac_dist	Locais onde os ASes vizinhos possuem instalações
			facility_cities_dist	Cidades onde os ASes vizinhos possuem instalações
			facility_country_dist	Países onde os ASes vizinhos possuem instalações
AS-path			*degree*	Quantidade de ASes vizinhos
			cone	Quantidade de clientes diretos e indiretos
			cone_degree	Valor de cone e de degree
Bidirec.			*bidi*	Se o enlace foi observado (1) ou não (0) em ambos os sentidos
			nb_vps	Quantidade de vizinhos conectados em fornecedores de informações para os coletores

caixas-pretas em situações críticas, como bloqueio de tráfego, por não entenderem como as decisões são tomadas [Jacobs et al. 2022, Willinger et al. 2023]. Além disso, esses modelos exigem uma quantidade maior de *features* que dependem de informações de fontes nem sempre confiáveis e demandam maior tempo de processamento para computá-las.

Diversos esforços na área de Inteligência Artificial Explicável (XAI) propõem métodos para explicar ou examinar a validade das classificações geradas por modelos caixa-preta e determinar a relevância das *features* no processo de decisão. As abordagens de XAI podem ser divididas em dois grupos principais: *explicabilidade local* e *explicabilidade global*. Os métodos de explicabilidade local procuram identificar a importância de cada *feature* em uma decisão específica [Ribeiro et al. 2016]. Por outro lado, a explicabilidade global visa explicar como o modelo caixa-preta toma decisões de forma geral e não em casos individuais. Geralmente, os métodos de explicabilidade global extraem um modelo caixa-branca (como uma árvore de decisão ou regras de decisão) que pode ser compreendido por um especialista no domínio do problema [Lakkaraju et al. 2016, Jacobs et al. 2022].

A ferramenta Trustee [Jacobs et al. 2022], destinada à explicabilidade global, gera árvores de decisão de alta fidelidade para explicar o funcionamento de um modelo caixa-preta. Cada árvore de decisão é gerada utilizando aprendizado por imitação (*imitation learning*) [Lakkaraju et al. 2016] usando uma dinâmica professor-aluno em que um conjunto de amostras é passado para o modelo caixa-preta classificar. As amostras classificadas pelo modelo caixa-preta (professor) são usadas para treinar a árvore de decisão (aluno). O objetivo é gerar árvores de decisão que imitam o funcionamento do modelo

caixa-preta. Assim, um especialista pode analisar as árvores de decisão para verificar se as decisões tomadas pelo modelo estão de acordo com o conhecimento existente no domínio do problema. Portanto, a árvore de decisão gerada por Trustee atua como um modelo substituto compreensível para o especialista. Trustee já foi utilizada para identificar problemas em modelos caixas-pretas em diferentes domínios de aplicação, especialmente na área de segurança de redes [Jacobs et al. 2022, Beltiukov et al. 2023].

4.1. Abrindo a Caixa-Preta de DFOH com Trustee

DFOH utiliza uma floresta aleatória com 28 *features* calculadas a partir de várias fontes, como coletores BGP, registros regionais e PeeringDB. O modelo é bastante complexo, pois várias *features* não possuem interpretações diretas, sendo derivadas de cálculos que capturam o posicionamento de um AS na topologia da Internet ou suas interações com outros ASes. A situação se complica ainda mais porque as *features* de *AS-path* (veja Tabela 1) são calculadas a partir de outro modelo de floresta aleatória, tornando a compreensão do modelo inviável até mesmo para especialistas no domínio do problema. Entretanto, ao utilizar a ferramenta Trustee para explicar o modelo utilizado por DFOH, ela gera uma árvore de decisão, mostrada na Figura 1, com apenas cinco *features*, mas com fidelidade de 95% ao modelo de floresta aleatória original. Isso indica que, apesar de sua complexidade, o modelo não utiliza todas as *features* em seu processo de decisão e pode ser substituído por um modelo mais simples. As seções a seguir detalham a análise realizada sobre o modelo de DFOH, explicando como ele pode ser modificado para eliminar certas *features* sem afetar o seu desempenho. Inicialmente, são descritos os conjuntos de dados utilizados na análise e em seguida os resultados obtidos.

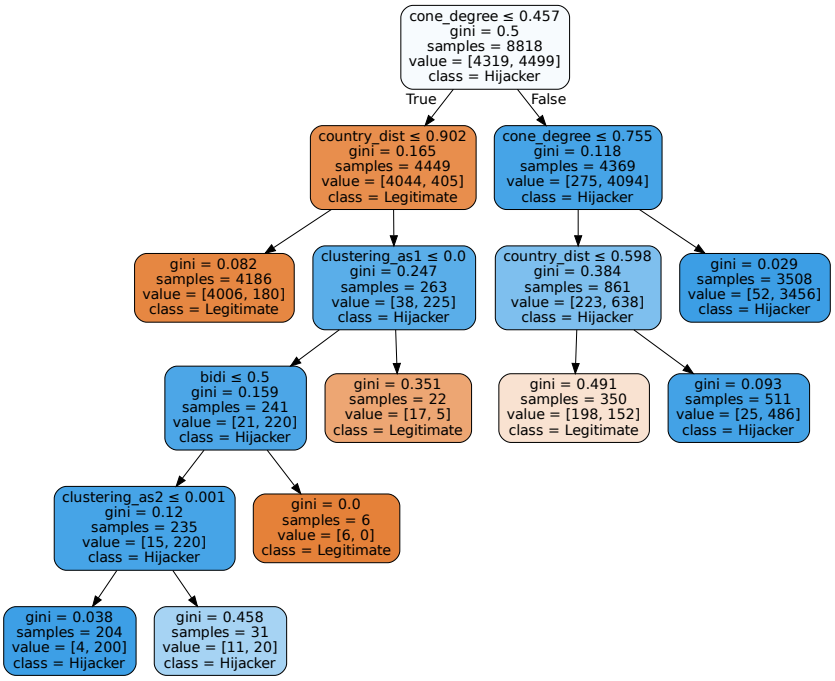


Figura 1. Árvore de decisão gerada por TRUSTEE para o modelo de DFOH.

5. Ambiente de Análise e Conjuntos de Dados

O código fonte de DFOH está disponível publicamente, o que facilita a reprodução dos resultados apresentados em [Holterbach et al. 2024] e a experimentação com mudanças no modelo. Durante a execução de DFOH, diversos arquivos são buscados de bases de dados públicas da Internet para cálculo das *features* do modelo de floresta aleatória. Para evitar que o tempo de download desses arquivos interfira nas medições de tempo que serão apresentadas na Seção 6, todos os arquivos necessários para cálculos das *features*, treinamento do modelo e inferências foram copiados e salvos localmente no servidor usado nos experimentos. O servidor é uma máquina virtual com 10 núcleos de processamento Intel Xeon E5-2670 de 2,60 GHz isolados exclusivamente para as medições de tempo de execução e 124 GB de memória RAM rodando o sistema operacional Linux Ubuntu Server 20.04. O código fonte desenvolvido para a avaliação está disponível em [Carvalho et al. 2024].

Para avaliar o modelo de DFOH e suas modificações, foram utilizados conjuntos de dados gerados para dois períodos distintos em dezembro de 2022 e dezembro de 2023 com as seguintes características:

- **Dias analisados**: para cada período, foram analisados 20 dias, de 1 a 20 de dezembro, totalizando 40 dias;
- **Informações dos coletores**: foram utilizadas as bases de dados de anúncios BGP de 200 pontos que fornecem informações aos coletores [Meyer 1997, Mcgregor et al. 2010] selecionados de acordo com o descrito em [Alfroy et al. 2022] para um período de 300 dias imediatamente anteriores ao dia de teste, resultando em 1.024.242 enlaces direcionais únicos durante todo o período analisado.
- **Conjunto de dados de treinamento**: são geradas 1.000 amostras para cada uma das duas classes (legítimos/suspeitos) por dia para um período de 60 dias consecutivos imediatamente anteriores ao dia de teste, totalizando 120 mil amostras;
- **Novos enlaces**: são aqueles enlaces identificados no dia de teste e que não foram observados no período de 300 dias anteriores ou que não foram adicionados a base para esse período por serem considerados suspeitos;
- **Amostras para avaliação dos modelos**: foram geradas 1.000 amostras, distintas da de treinamento, para cada uma das classes para cada dia de teste para se calcular as métricas de desempenho precisão, *recall* e *F1-score*.

6. Avaliação dos Modelos

Para escrutinizar o modelo de DFOH e definir novos modelos, a ferramenta Trustee foi executada com diferentes tamanhos de amostra e iterações para gerar 56 árvores de decisão, sendo 28 com poda e 28 sem poda. A relevância de cada *feature* foi determinada pela frequência de sua presença nas árvores geradas, sendo as mais frequentes consideradas mais relevantes. Com base nesse critério, foram definidos os seguintes modelos:

- **M1**: modelo formado com todas as *features* originalmente utilizadas por DFOH;
- **M2**: modelo original excluindo-se as *features degree_centrality_as1*, *degree_centrality_as2* e *adamic_adar*, que não apareceram em nenhuma das 28 árvores de decisão completas;
- **M3**: modelo M2 menos as três *features* seguintes que menos apareceram nas árvores completas, *eccentricity_as1*, *eccentricity_as2* e *shortest_path*;

- **M4**: modelo original de DFOH excluindo as 17 *features* que apareceram em no máximo uma das árvores com poda, permanecendo apenas as 11 *features* seguintes: *cone_degree*, *country_dist*, *clustering_as1*, *clustering_as2*, *bidi*, *triangles_as1*, *triangles_as2*, *preferential_attachement*, *degree*, *nb_vps* e *ixp_dist*; e
- **M5**: modelo formado com as cinco *features* que mais apareceram nas árvores com poda: *cone_degree*, *country_dist*, *clustering_as1*, *clustering_as2* e *bidi*.

Os cinco modelos foram inicialmente avaliados para as métricas de precisão, *recall* e *F1-Score* utilizando 10 conjuntos de teste, cada um com mil enlaces legítimos e mil forjados. O cálculo das *features* foi realizado para o dia primeiro de dezembro de 2022, usando os conjuntos de dados descritos na Seção 5. A Tabela 2 mostra o valor médio dos resultados para os 10 conjuntos de teste com o intervalo de confiança (IC) de 95%. Pode-se observar que não há diferença estatisticamente significativa entre os modelos. Assim, foram selecionados os modelos M4 e M5, por possuírem menos *features*, para uma comparação mais detalhada com M1.

Tabela 2. Médias de dez testes dos modelos, com os respectivos intervalos de confiança (IC) de 95%, para as métricas de precisão, *recall* e *F1-score*

	Legítimos						Suspeitos					
M	**Precisão ± IC**		**Recall ± IC**		**F1-Score ± IC**		**Precisão ± IC**		**Recall ± IC**		**F1-Score ± IC**	
M1	0,9059	0,0054	0,9831	0,0029	0,9429	0,0035	0,9815	0,0032	0,8978	0,0064	0,9378	0,0041
M2	0,9059	0,0054	0,9830	0,0029	0,9429	0,0035	0,9814	0,0031	0,8979	0,0064	0,9378	0,0041
M3	0,9062	0,0053	0,9831	0,0028	0,9431	0,0034	0,9816	0,0031	0,8982	0,0063	0,9380	0,0039
M4	0,9062	0,0051	0,9829	0,0029	0,9430	0,0034	0,9814	0,0031	0,8982	0,0061	0,9379	0,0040
M5	0,9060	0,0052	0,9831	0,0028	0,9430	0,0033	0,9815	0,0030	0,8979	0,0062	0,9378	0,0039

A Tabela 3 mostra que houve redução de *features* em todas as categorias. Porém, não houve eliminação de nenhuma categoria, o que mostra que todas as categorias contribuem para o processo de decisão.

Tabela 3. Comparativo da quantidade de *features* entre os modelos M1, M4 e M5.

Categoria	M1	M4	Redução M4	M5	Redução M5
Topológicas	18	5	72,22%	2	88,89%
Peering	5	2	60,00%	1	80,00%
Padrão do AS-path	3	2	33,33%	1	66,67%
Bidirecionalidade	2	2	0,00%	1	50,00%
Total	28	11	60,71%	5	82,14%

6.1. Impacto das *Features*

A ausência de diferença estatisticamente significativa entre os modelos mesmo com a remoção de várias *features* pode ser justificada por pelo menos dois motivos. Primeiro, pela *feature cone_degree*, que aparece na raiz da árvore de decisão da Figura 1. Ao calcular as métricas de desempenho de um modelo com apenas essa *feature* utilizando os valores presentes na árvore de decisão, verifica-se que ela sozinha atinge valores de precisão e *recall* acima de 0,9 para ambas as classes. Das 4.319 amostras legítimas, ela identifica corretamente 4.044, e das 4.499 amostras suspeitas, ela identifica corretamente 4.094.

Segundo, algumas *features* impactam no desempenho dos modelos de forma mais significativa do que outras. Para se entender o motivo de algumas *features* terem grande

relevância para o modelo e outras não, foram calculadas as distribuições dos valores dessas *features*. A Figura 2a mostra a distribuição dos valores de uma *feature* que não foi utilizada nos modelos reduzidos. Pode-se observar que os valores para a classe de enlaces legítimos estão contidos nos valores das classe de enlaces suspeitos, tornando a *feature* irrelevante para a separação das classes. Por outro lado, a Figura 2b mostra a distribuição de uma *feature* que foi incluída nos modelos reduzidos. Nota-se que os valores para as duas classes são bem distintos, contribuindo, portanto, para a separação das classes. Situação semelhante foi observada para as demais *features* que foram removidas ou mantidas nos modelos reduzidos.

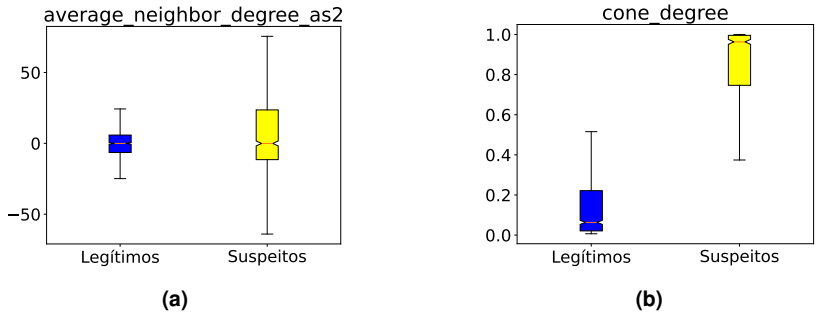


Figura 2. Boxplots com as distribuições dos valores de duas *features* entre as classes para as amostras geradas. A *feature* de (a) faz parte somente do modelo M1 enquanto a *feature* de (b) faz parte de todos os modelos.

6.1.1. Avaliação Qualitativa das *Features* Removidas de M4 e M5

As *features* para os modelos M4 e M5 foram escolhidas com base no número de vezes que elas apareceram nas 56 árvores de decisão geradas por Trustee. Entretanto, esse critério quantitativo não explica a razão das *features* removidas não contribuírem significativamente no processo de decisão dos modelos. A seguir é apresentada uma analise qualitativa que auxilia no entendimento da falta de contribuição de algumas *features*.

- *Padrão AS-path*: essa categoria é composta por três *features*, sendo que uma delas, *cone_degree*, é gerada por uma floresta aleatória que possui como entrada os mesmos valores utilizados na geração das outras duas. Assim, ela contém informações das outras duas em seu valor.
- *Topológicas*: essa categoria é dividida em subcategorias, como mostrado na Tabela 1. As *features* relacionadas a Centralidade, Distância (Dist.), Proximidade (Prox.) e Vizinhos se baseiam na quantidade de ASes conectados a um determinado AS e no comprimento dos *AS-paths*. Com o aumento dos pontos de troca de tráfego (PTTs) e de ASes conectados a eles, a hierarquia da Internet está se tornando mais plana, reduzindo portanto a relevância dessas *features*. Os ASes *stubs* se destacam em relação aos demais tipos de AS por terem uma única ligação, mas essa diferença também é capturada pelas *features* de Padrão Topológico (P. Topo.), que se mostraram mais relevantes. As *features clustering_as1* e *clustering_as2* se mostraram mais relevantes que as *triangles_as1* e *triangles_as2*, provavelmente por usar o valor das últimas duas em seus cálculos.
- *Peering*: As informações utilizadas para cálculo das *features* dessa categoria são fornecidas pelos administradores dos ASes e podem estar incompletas e desatualizadas, não sendo totalmente confiáveis para caracterização do

AS [Du et al. 2023]. A informação mais perene e confiável é o país onde o AS possui registro e está na *feature country_dist* que foi mantida em todos os modelos.
- *Bidirecionalidade*: uma ligação observada de forma bidirecional é suficiente para ser caracterizada como legítima [Sermpezis et al. 2018, Holterbach et al. 2024], tornando esta *feature* altamente relevante para as inferências. A *feature nb_vps*, que é calculada no código juntamente com a *feature bidi*, representa ligações com um grupo de ASes e pode estar perdendo a relevância com o aumento dos PTTs.

6.2. Tempo de Execução e Espaço Utilizado

DFOH é um sistema bastante modular e possui um módulo separado para calcular as *features* de cada categoria. Além disso, ele contém um módulo *broker* que calcula as *features* dos novos enlaces observados por dia, realiza o treinamento do modelo e a inferência das classes dos novos enlaces com o modelo treinado. A Tabela 4 mostra os tempos médios de execução, com os respectivos intervalos de confiança de 95%, dos módulos de DFOH para o período de 40 dias analisados. Pode-se observar que há uma redução em relação ao tempo de processamento de M1 para o cálculo das *features* superior a 31% e 37% para os modelos M4 e M5 (linha "Soma parcial" da tabela) e superior a 32% e 38% para a execução do módulo *broker* (linha "Broker"). Os tempos de inferência são ínfimos para todos os modelos avaliados e não estão na tabela.

Tabela 4. Comparação entre os tempos médios para execução dos modelos M1, M4 e M5. Os valores de tempo estão em segundos e os valores em porcentagem correspondem a redução de tempo gasto em comparação com M1.

Módulo	M1	IC	M4	IC	% Redução M4	M5	IC	% Redução M5
Padrão *AS-path*	168	1	124	1	26,32%	97	1	41,96%
Bidirecionalidade	51	8	50	7	0,34%	46	7	9,10%
Peering	229	5	56	1	75,59%	37	1	83,87%
Topológicas	1427	47	1061	41	25,67%	1000	37	29,93%
Soma Parcial	1875	45	1291	36	31,15%	1180	32	37,04%
Broker	1064	84	718	64	32,55%	653	61	38,60%

Os experimentos desta seção geraram arquivos de *features* para 160 dias (60 dias para o treinamento e 20 dias de testes para os dois períodos analisados) separados por categoria e tipos de amostra. Para as *features* de padrão de *AS-path* ainda é gerado um modelo por dia para cada uma das *features*, sendo três *features* para M1, duas para M4 e uma para M5. A Tabela 5 mostra o espaço de armazenamento necessário para processamento do modelo. Comparando o volume de dados utilizado para armazenamento dos arquivos relativos às *features* necessárias para os treinamentos dos modelos (diretório *features*), houve um redução de quase 60% no uso do espaço de armazenamento para M4 e de 71% para o M5 em relação ao espaço ocupado pelo modelo M1.

6.3. Avaliação das Métricas de Desempenho dos Modelos

Apesar dos ganhos em tempo de processamento e espaço de armazenamento apresentados na seção anterior, um modelo com um conjunto menor de *features* deve ter desempenho semelhante ao original para ser útil. Os modelos foram analisados com enlaces forjados sintéticos, usando o mesmo procedimento descrito em [Holterbach et al. 2024]. Foram geradas mil amostras de enlaces legítimos e mil de suspeitos para cada um dos 40 dias

Tabela 5. Comparação entre o espaço de armazenamento necessário para armazenar as *features* dos modelos M1, M4 e M5. Os valores estão em bytes.

Diretório do sistema	M1	M4	Redução M4	M5	Redução M5
aspath_models_clusters	2651986704	1721000442	35,11%	842243747	68,24%
features	144141619	57872896	59,85%	41575973	71,16%
features/negative/aspath	11616412	8401632	27,67%	5193468	55,29%
features/negative/bidirectionality	2708608	2708609	0,00%	2269618	16,21%
features/negative/peeringdb	17637457	8240107	53,28%	5103018	71,07%
features/negative/topological	41857488	10378604	75,20%	8646301	79,34%
features/positive/aspath_clusters	11131982	8070168	27,50%	5046440	54,67%
features/positive/bidirectionality_clusters	2714919	2714914	0,00%	2316226	14,69%
features/positive/peeringdb_clusters	16351922	7594905	53,55%	4784388	70,74%
features/positive/topological_clusters	40106447	9747573	75,70%	8196034	79,56%

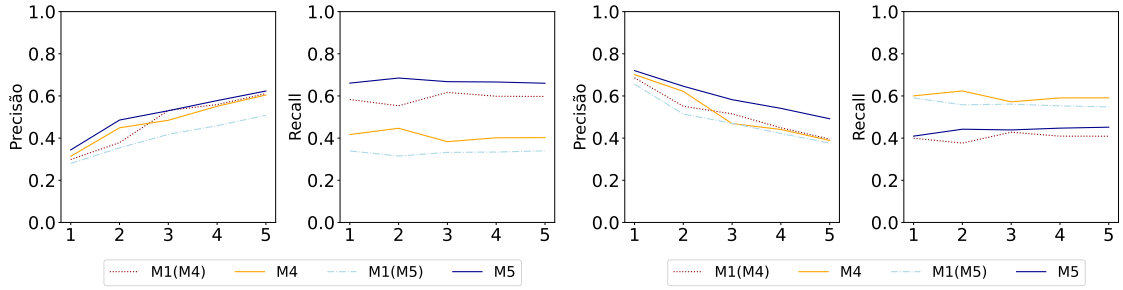
avaliados. As *features* de cada um dos modelos M1, M4 e M5 foram calculadas para cada um dos conjuntos de amostra. Os valores médios das métricas de desempenho estão na Tabela 6 e mostram que os valores obtidos para M1 e M4 ficaram praticamente iguais e estão nos intervalos de confiança um do outro. Entretanto, os valores obtidos por M5 ficaram um pouco abaixo dos valores de M1, com diferenças de 0,0092 e 0,0091 nas médias do F1-score dos enlaces legítimos e suspeitos, respectivamente, e com distância máxima entre os intervalos de confiança de 0,0063.

Tabela 6. Métricas para os modelos M1, M4 e M5, calculadas com base nas amostras geradas para os 40 dias de análise. São apresentados os valores médios, com os respectivos intervalos de confiança (IC) de 95%.

	Legítimos						Suspeitos					
M	**Precisão ± IC**		**Recall ± IC**		**F1-Score ± IC**		**Precisão ± IC**		**Recall ± IC**		**F1-Score ± IC**	
M1	0,9570	0,0018	0,9590	0,0022	0,9580	0,0014	0,9590	0,0021	0,9569	0,0019	0,9579	0,0013
M4	0,9582	0,0020	0,9580	0,0022	0,9581	0,0015	0,9581	0,0021	0,9581	0,0021	0,9581	0,0015
M5	0,9488	0,0020	0,9488	0,0025	0,9488	0,0015	0,9488	0,0024	0,9488	0,0021	0,9488	0,0015

Durante o período de análise, foram observados 16.107 novos enlaces únicos, dos quais 968 tiveram inferências diferentes entre M1 e M4, e 1.256 entre M1 e M5, correspondendo a 6,0% e 7,8%, respectivamente. Como não há uma verdade absoluta (*ground truth*) para determinar qual modelo é melhor, considerou-se que sequestros de prefixos geralmente ocorrem por curtos períodos, conforme relatado por [Testart et al. 2019] e observado em incidentes noticiados [RIPE NCC RIS 2008, Siddiqui 2022]. Assim, foram analisadas as RIBs das primeiras duas horas de janeiro a maio de 2023 e 2024 para verificar se esses enlaces com divergências nas classificações apareceram nesses períodos, ou seja, posteriormente ao suposto sequestro. Os enlaces que apareceram foram considerados legítimos e os demais como suspeitos. A Figura 3 mostra os resultados para os enlaces com divergências nas classificações entre os modelos considerando o número de meses consecutivos que eles apareceram nas RIBs para que fossem classificados como legítimos. M1(M4) e M1(M5) representam os resultados do modelo M1 para os enlaces que tiveram inferências divergentes entre o modelo M1 e os modelos M4 e M5, respectivamente.

Um sistema de identificação de sequestros deve minimizar falsos positivos para manter sua credibilidade [Holterbach et al. 2024]. Nesse contexto, o modelo M5 apresentou resultados melhores que o modelo M1. Independentemente do número de dias que um enlace é observado para considerá-lo legítimo, o modelo M5 foi mais preciso em ambas as classes, sendo 0,1152 mais preciso que o modelo M1 ao considerar enlaces observados por 5 dias em meses consecutivos. Como algumas *features* são extraídas de



(a) Suspeitos **(b) Legítimos**

Figura 3. Métricas calculadas com base no número mínimo de dias (RIBs) em que um enlace é observado para ser classificado como legítimo. M1(M4) e M1(M5) representam os resultados do modelo M1 para enlaces com inferências divergentes em relação aos modelos M4 e M5, respectivamente.

informações fornecidas pelos operadores e outras estão perdendo a relevância devido às mudanças na topologia da Internet que a tem tornado mais "plana" (*i.e.,* caminhos cada vez mais curtos entre os ASes), elas podem estar contribuindo negativamente para o modelo original, sendo uma das prováveis razões para o melhor desempenho dos modelos reduzidos, o que reforça a necessidade de análises mais detalhadas na seleção de *features*.

7. Conclusão

Modelos do tipo caixa-preta tem apresentado desempenhos melhores que modelos interpretáveis. No entanto, operadores de rede ainda são relutantes em utilizá-los em situações críticas. O uso de técnicas de XAI pode auxiliar a identificar a real relevância de cada *feature* e seu peso nas inferências, trazendo maior confiabilidade nas decisões do modelo. Além disso, elas podem mostrar que uma quantidade maior de *features* em um modelo não garante o seu sucesso, mas sim quão relevantes elas são. Selecionar o melhor conjunto de *features* pode economizar tempo de execução e espaço de armazenamento, além de poder até mesmo melhorar o desempenho de um modelo.

Este artigo mostra, com uma avaliação experimental extensiva, que um modelo com um conjunto de *features* 60,71% menor alcançou resultados sem diferenças estatisticamente significativas aos do modelo completo. Adicionalmente, um modelo com um conjunto de *features* 82,14% menor obteve métricas similares e maior precisão com dados reais que o modelo original. Os modelos reduzidos também diminuíram o tempo de execução em mais de 30% e o espaço de armazenamento em mais de 59%. Por serem computacionalmente mais leves, eles podem ser retreinados com maior frequência e assim capturar mais rapidamente mudanças nas distribuições dos dados.

O uso de modelos caixas-pretas para solucionar problemas de redes de computadores é uma tendência atual, mas este trabalho mostra que a correta seleção de *features* é essencial para se obter melhores resultados com economia de recursos computacionais.

Agradecimentos

O presente trabalho foi realizado com apoio da CAPES – Código de Financiamento 001, do CNPq – Procs. 420934/2023-5, 308101/2022-7 e 465446/2014-0, e da FAPESP – Procs. 2023/00812-7, 2023/00811-0 e 2020/05183-0.

Referências

Ahmed, M., Seraj, R., and Islam, S. M. S. (2020). The K-Means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics*, 9(8):1295.

Alfroy, T., Holterbach, T., and Pelsser, C. (2022). MVP: Measuring Internet Routing from the Most Valuable Points. In *Proceedings of the 22nd ACM IMC 2022*, page 770–771.

Arai, T., Nakano, K., and Chakraborty, B. (2019). Selection of effective features for bgp anomaly detection. In *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6.

Beltiukov, R., Guo, W., Gupta, A., and Willinger, W. (2023). In Search of netUnicorn: A Data-Collection Platform to Develop Generalizable ML Models for Network Security Problems. In *Proc. of the 2023 ACM CCS*, CCS '23, page 2217–2231.

Birge-Lee, H., Sun, Y., Edmundson, A., Rexford, J., and Mittal, P. (2018). Bamboozling Certificate Authorities with BGP. In *Proc. of the 27th USENIX Security'18*, pages 833–849.

Bühler, T., Milolidakis, A., Jacob, R., Chiesa, M., Vissicchio, S., and Vanbever, L. (2023). Oscilloscope: Detecting BGP Hijacks in the Data Plane. *arXiv preprint arXiv:2301.12843*.

Bush, R. and Austein, R. (2017). The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1. RFC 8210.

CAIDA (2001). CAIDA AS Rank. `http://as-rank.caida.org/`.

CAIDA (2015). AS Relationships (Serial-2). `https://catalog.caida.org/dataset/as_relationships_serial_2`.

Carvalho, A. B., da Silva Jr, B. A., da Silva, C. A., and Ferreira, R. A. (2024). Material suplementar. https://github.com/Bastos-abc/blackbox-explainable-xai-hijack.

Cho, S., Fontugne, R., Cho, K., Dainotti, A., and Gill, P. (2019). BGP Hijacking Classification. In *2019 Network Traffic Measurement and Analysis Conference*, pages 25–32.

Du, B., Izhikevich, K., Rao, S., Akiwate, G., Testart, C., Snoeren, A. C., and claffy, k. (2023). IRRegularities in the Internet Routing Registry. In *Proc. of the ACM IMC 2023*, page 104–110.

Freedman, D., Foust, B., Greene, B., Maddison, B., Robachevsky, A., Snijders, J., and Steffann, S. (2019). Mutually Agreed Norms for Routing Security (MANRS) Implementation Guide.

Hammood, N. H. and Al-Musawi, B. (2021). Using BGP Features Towards Identifying Type of BGP Anomaly. In *Proc. of the 2021 ICOTEN*, pages 1–10.

Holterbach, T., Alfroy, T., Phokeer, A. D., Dainotti, A., and Pelsser, C. (2024). A System to Detect Forged-Origin Hijacks. In *Proc. of the 21th USENIX NSDI*.

Jacobs, A. S., Beltiukov, R., Willinger, W., Ferreira, R. A., Gupta, A., and Granville, L. Z. (2022). AI/ML for Network Security: The Emperor Has No Clothes. In *Proc. of the 2022 ACM Conf. on Computer and Comm. Security*, CCS '22, page 1537–1551.

Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., and Zhang, L. (2006). PHAS: A Prefix Hijack Alert System. In *USENIX Security Symposium*, volume 1, page 3.

Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). Interpretable Decision Sets: A Joint Framework for Description and Prediction. In *Proc. of the 22nd ACM KDD*.

Lepinski, M. and Sriram, K. (2017). BGPsec Protocol Specification. RFC 8205.

Liu, Y., Su, J., and Chang, R. K. (2012). LDC: Detecting BGP Prefix Hijacking by Load Distribution Change. In *2012 IEEE 26th IPDPS Workshops*, pages 1197–1203.

Lychev, R., Schapira, M., and Goldberg, S. (2016). Rethinking Security for Internet Routing. *Commun. ACM*, 59(10):48–57.

Mcgregor, T., Alcock, S., and Karrenberg, D. (2010). The RIPE NCC internet measurement data repository. In *Int. Conf. on Passive and Active Network Measurement*.

Merit Network, Inc (2024). Internet Routing Registry. https://irr.net/.

Meyer, D. (1997). University of Oregon Route Views Archive Project.

Milolidakis, A. and et al. (2023). On the Effectiveness of BGP Hijackers That Evade Public Route Collectors. In *IEEE Access*, volume 11, pages 31092–31124.

PeeringDB (2010). `https://catalog.caida.org/dataset/peeringdb`.

Qin, L., Li, D., Li, R., and Wang, K. (2022). Themis: Accelerating the Detection of Route Origin Hijacking by Distinguishing Legitimate and Illegitimate MOAS. In *Proc. of the 31st USENIX Security Symposium (USENIX Security 22)*, pages 4509–4524.

Rekhter, Y. and et al. (2006). A Border Gateway Protocol 4 (BGP-4). RFC 4271.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144.

RIPE NCC RIS (2008). YouTube Hijacking: A RIPE NCC RIS Case Study. `https://www.ripe.net/publications/news/youtube-hijacking-a-ripe-ncc-ris-case-study/`.

Sermpezis, P., Kotronis, V., Gigis, P., Dimitropoulos, X., Cicalese, D., King, A., and Dainotti, A. (2018). ARTEMIS: Neutralizing BGP Hijacking Within a Minute. In *IEEE/ACM Transactions on Networking*, volume 26, pages 2471–2486.

Shapira, T. and Shavitt, Y. (2022). AP2Vec: An Unsupervised Approach for BGP Hijacking Detection. *IEEE Trans. on Network and Service Management*, 19(3):2255–2268.

Shi, X., Xiang, Y., Wang, Z., Yin, X., and Wu, J. (2012). Detecting Prefix Hijackings in the Internet with Argus. In *Proc. of the 2012 ACM IMC*, page 15–28.

Siddiqui, A. (2022). KlaySwap – Another BGP Hijack Targeting Crypto Wallets. `https://manrs.org/2022/02/klayswap-another-bgp-hijack-targeting-crypto-wallets/`.

Testart, C., Richter, P., King, A., Dainotti, A., and Clark, D. (2019). Profiling BGP Serial Hijackers: Capturing Persistent Misbehavior in the Global Routing Table. In *Proc. of the 2019 ACM Internet Measurement Conference*, IMC '19, page 420–434.

Willinger, W., Gupta, A., Jacobs, A. S., Beltiukov, R., Ferreira, R. A., and Granville, L. (2023). A NetAI Manifesto (Part I): Less Explorimentation, More Science. *SIGMETRICS Perform. Eval. Rev.*, 51(2):106–108.