Modelagem e Avaliação de Aprendizado Estatístico Relacional na Detecção de Falhas de Segurança

Carlos Banjar[1], **Lucas G. Miranda**[1], **Daniel S. Menasché**[1], **Gerson Zaverucha**[1]

[1] Universidade Federal do Rio de Janeiro (UFRJ)

{carlosesb, lucasgm, sadoc}@ic.ufrj.br, gerson@cos.ufrj.br

Resumo. *A avaliação de segurança em redes enfrenta o desafio de superar métricas isoladas que ignoram o risco relacional entre ativos e caminhos de ataque. Este trabalho propõe uma abordagem baseada em Aprendizado Estatístico Relacional (SRL) para modelar probabilisticamente caminhos de ataque via lógica de primeira ordem. Utilizando dados reais de hosts, o modelo integra o MulVAL ao Relational Dependency Network Boosting (RDN-Boost). Avaliado por validação cruzada, o método alcançou médias de 0,9471 em AUC-PR e 0,9678 em AUC-ROC. Experimentos de resiliência indicam que o modelo mantém alto desempenho (AUC-ROC de 0,9067) mesmo sob cenários de dados fragmentados, com omissão parcial de informações de vulnerabilidade e conectividade. Os resultados demonstram que a abordagem infere a explorabilidade de caminhos de ataque a partir de evidências incompletas e da estrutura relacional dos ativos.*

Abstract. *Network security assessment faces the challenge of moving beyond isolated metrics that ignore the relational risk between assets and attack paths. This work proposes an approach based on Statistical Relational Learning (SRL) to probabilistically model attack paths using first-order logic. Utilizing real-world host data, the model integrates MulVAL with Relational Dependency Network Boosting (RDN-Boost). Evaluated through cross-validation, the method achieved average scores of 0.9471 in AUC-PR and 0.9678 in AUC-ROC. Resilience experiments indicate that the model maintains high performance (AUC-ROC of 0.9067) even under fragmented data scenarios with partial omission of vulnerability and connectivity information. The results demonstrate that the approach infers the exploitability of attack paths from incomplete evidence and the underlying relational structure of the assets.*

1. Introdução

À medida que os ambientes de rede se tornam mais complexos, a avaliação de riscos de segurança também se torna uma tarefa progressivamente mais desafiadora. Atribuir pontuações quantitativas a sistemas com base em ameaças específicas é uma estratégia importante para comparar diferentes níveis de exposição e apoiar a tomada de decisão [Yee 2013]. Para que essa quantificação seja significativa, é necessário considerar múltiplos fatores de risco e diferentes componentes da infraestrutura, permitindo identificar quais sistemas estão mais suscetíveis a comprometimentos.

Métricas de segurança representam um mecanismo amplamente utilizado para esse fim. Entretanto, muitas dessas métricas concentram-se em aspectos isolados do

sistema, sem capturar adequadamente o contexto em que as vulnerabilidades estão inseridas. Um exemplo recorrente é o CVSS (*Common Vulnerability Scoring System*) [Mell et al. 2006], que atribui uma pontuação a vulnerabilidades específicas. Apesar de sua ampla adoção, essa abordagem não incorpora elementos relacionais do ambiente [Allodi and Massacci 2017]. Uma vulnerabilidade classificada como crítica pode ter impacto reduzido se estiver em um sistema isolado e fortemente controlado, enquanto uma vulnerabilidade de severidade moderada pode representar risco elevado quando associada a um serviço amplamente exposto e acessível a múltiplos agentes. Diante dessa lacuna, este trabalho propõe incorporar explicitamente tais relações por meio de *Statistical Relational Learning* (SRL), preservando a estrutura lógica do domínio ao mesmo tempo em que introduz uma camada probabilística capaz de lidar com incertezas e generalizações [Pereira et al. 2023].

Além da natureza relacional do risco, a avaliação de segurança prática é frequentemente dificultada pela incompletude dos dados coletados. Scanners de vulnerabilidade e ferramentas de varredura de ativos raramente possuem visibilidade total da infraestrutura, resultando em grafos de ataque fragmentados. Modelos puramente determinísticos, como o MulVAL [Ou et al. 2005], falham nessas condições, pois a ausência de um único fato lógico pode invalidar toda a derivação de um caminho de ataque. Nesse contexto, o SRL destaca-se pela capacidade de realizar inferências latentes, utilizando similaridades estruturais para estimar riscos mesmo sob condições de incerteza informacional.

Este trabalho introduz o SRL como motor central, utilizando uma base de conhecimento para induzir regras de ataque probabilísticas. Conforme ilustrado na Figura 1, o fluxo de análise inicia-se com a planta do sistema, que é traduzida em predicados lógicos que descrevem entidades e conectividade. Este ciclo resulta em recomendações fundamentadas de *patching* e *rezoning*, que permitem o aprimoramento dinâmico da segurança da infraestrutura.[1]

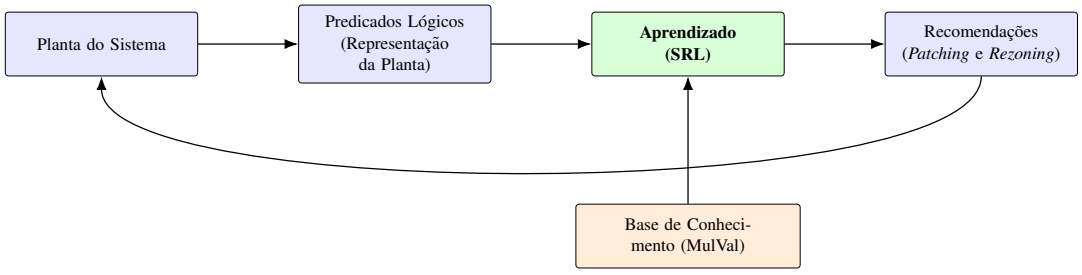


Figura 1. Visão Proposta: O Aprendizado (SRL) utiliza a base de conhecimento para interpretar os predicados da planta.

Nossas principais contribuições são resumidas a seguir:

- **Introdução do SRL no Domínio de Grafos de Ataque:** Exploramos a aplicação de SRL na modelagem de caminhos de ataque. Até onde se tem conhecimento, a aplicação de SRL para quantificação de risco em cenários de segurança cibernética ainda é pouco explorada na literatura no contexto de grafos de ataque práticos, configurando uma contribuição original deste estudo;

[1] *Rezoning* refere-se à reconfiguração das zonas de segurança de uma rede, alterando a segmentação lógica entre ativos (por exemplo, mover um host da rede interna para a DMZ ou ajustar regras de firewall), com o objetivo de reduzir superfícies de ataque e limitar caminhos de exploração.

- **Aprendizado de Regras:** Implementamos um modelo capaz de induzir regras de ataque probabilísticas, permitindo que o sistema aprenda a semântica das relações entre vulnerabilidades, serviços e conectividade de rede;
- **Avaliação de Resiliência:** Validamos a robustez do modelo na identificação de caminhos de ataque em cenários de degradação progressiva de dados. Através de um *stress test* sistemático, ilustramos como o *Relational Dependency Network Boosting* (RDN-Boost) [Pereira et al. 2025] mantém estabilidade preditiva, simulando as limitações do monitoramento em ambientes reais.

A Seção 2 revisa os trabalhos relacionados. A Seção 3 apresenta a metodologia proposta, seguida da Seção 4 que discute os resultados experimentais. A Seção 5 analisa a robustez frente a dados incompletos. Por fim, a Seção 6 traz as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

A modelagem de segurança cibernética tem evoluído de métricas estáticas para análises relacionais e probabilísticas. Esta seção situa o presente trabalho frente ao estado da arte em SRL, grafos de ataque e priorização de vulnerabilidades.

SRL aplicado em Cibersegurança. O uso de SRL em cibersegurança é uma área emergente. Um estudo recente [Pereira et al. 2023] utiliza abordagens relacionais para predizer a existência de *exploits* associados a produtos de software a partir de bases como NVD e ExploitDB. Embora este trabalho compartilhe a premissa de que o risco emerge de interações entre entidades, ele foca no nível produto-vulnerabilidade. Em contraste, nossa pesquisa direciona-se à infraestrutura, incorporando explicitamente topologia de rede, privilégios e condições de conectividade para inferir cenários de comprometimento.

Análise Baseada em Grafos de Ataque. Grafos de ataque representam formalmente caminhos de exploração através de pré-condições e pós-condições [Aksu et al. 2018, Lei 2023]. O sistema MulVAL [Ou et al. 2005] é a referência para derivação lógica de grafos a partir de fatos estruturados. Enquanto o MulVAL adota uma inferência puramente determinística que assume conhecimento total do estado da rede, nossa abordagem estende esse modelo para o paradigma estocástico do SRL. Isso permite lidar com a incerteza intrínseca de redes modernas e informações incompletas, preenchendo a lacuna entre a análise lógica rigorosa e a natureza dinâmica das ameaças reais.

Métricas Probabilísticas em Grafos de Ataque. No escopo de métricas sobre grafos de ataque, Wang et al. [Wang et al. 2008] propõem uma métrica cumulativa para quantificar a probabilidade de um atacante atingir ativos críticos combinando múltiplas vulnerabilidades. A abordagem deles é determinística e depende de caminhos de ataque explícitos e predefinidos para calcular as probabilidades.

Nossa proposta se diferencia ao aplicar SRL via RDN-Boost para induzir regras lógicas generalizáveis em vez de mapear caminhos rígidos. Isso permite realizar inferências probabilísticas sobre a infraestrutura mesmo quando dados de conectividade ou topologia estão fragmentados, cenário em que abordagens baseadas em caminhos tradicionais falham devido à quebra na derivação lógica.

Priorização de Vulnerabilidades. Abordagens tradicionais focam no enriquecimento dinâmico de pontuações CVSS através de dados históricos de *weaponization*

[Banjar et al. 2024]. Contudo, esses modelos priorizam falhas baseando-se em propriedades intrínsecas das vulnerabilidades e tendências globais. Nossa proposta diverge ao argumentar que o risco real é indissociável da acessibilidade do ativo dentro de um grafo de ataque específico. Ao utilizar SRL, permitimos que a priorização de *patching* considere não apenas a severidade isolada, mas o papel do ativo como elo crítico em caminhos de ataque probabilísticos.

3. Metodologia

O processo foi estruturado em três etapas encadeadas: (i) definição abstrata da arquitetura de rede por meio de um template lógico, (ii) instanciação automática de cenários com diferentes configurações de software, privilégios e posicionamento do atacante e (iii) classificação das instâncias com base na derivação formal de regras de segurança, via RDN-Boost. Para garantir a robustez do aprendizado, o treinamento foi realizado utilizando tanto conjuntos de dados completos, com visibilidade total dos ativos, quanto dados truncados, onde informações críticas são omitidas propositalmente.

3.1. Visão Geral da Metodologia

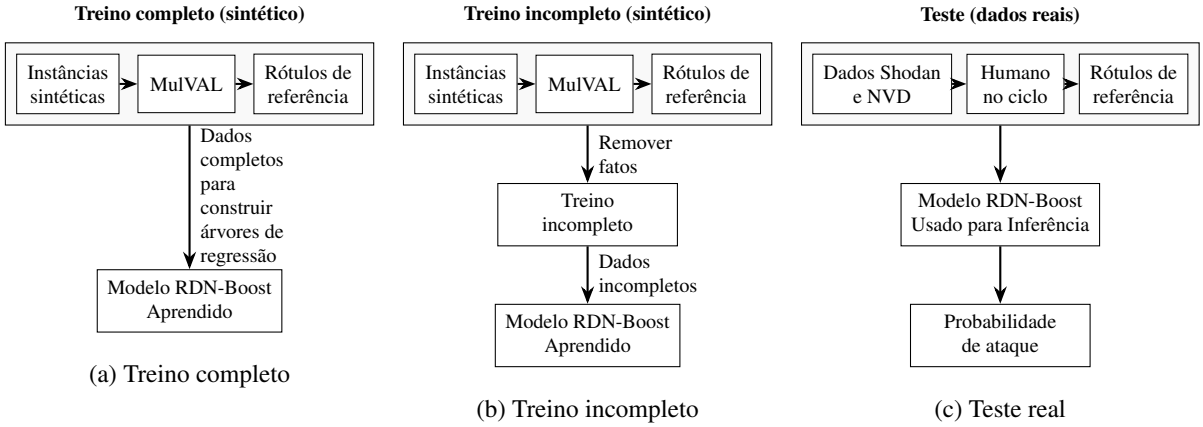
A metodologia deste trabalho é estruturada em um fluxo que integra o rigor da lógica formal com a variabilidade e a incompletude intrínsecas aos dados de redes do mundo real. O processo é organizado em estágios interdependentes, conforme ilustrado na Figura 2. No primeiro estágio, focado na indução de regras lógicas puras, um modelo é treinado exclusivamente com instâncias sintéticas completas. Nesta configuração, o sistema MulVAL atua como um oráculo determinístico, estabelecendo o *ground truth* pela derivação lógica do predicado-alvo `execCode`, que caracteriza se um atacante consegue ou não executar código malicioso em uma máquina alvo. Este modelo serve como *baseline* de desempenho em condições de conhecimento pleno sobre a topologia e vulnerabilidades.

Paralelamente, para adequar a abordagem ao cenário real de segurança cibernética, no qual informações sobre ativos são frequentemente fragmentadas, desenvolveu-se um segundo modelo independente, submetido a um *stress test* de resiliência. Fatos críticos como a localização do atacante (`attackerLocated`) e propriedades de exploração (`vulProperty`) são removidos aleatoriamente durante o treinamento. Esta abordagem permite isolar e quantificar a capacidade do RDN-Boost em inferir probabilidades de risco via similaridade estrutural, mesmo diante de evidências omitidas.

Por fim, ambos os modelos são validados utilizando dados sintéticos (validação cruzada) bem como dados reais (extraídos das APIs Shodan e NVD). Esta distinção metodológica é sumarizada na Tabela 1.

Tabela 1. Comparativo entre os conjuntos de dados e métodos de classificação.

Fonte de Dados	Papel no Experimento	Método de Classificação (Label)
Sintéticos	Treinamento e validação cruzada	**Determinístico:** Rotulado via sucesso de derivação no MulVAL.
Reais (Shodan/NVD)	Estudo de caso qualitativo (teste de generalização)	**Probabilístico:** Saída estimada pelo modelo RDN-Boost treinado.



(a) Treino completo (b) Treino incompleto (c) Teste real

Figura 2. Metodologia proposta para a predição da probabilidade de ataque.

3.2. Modelagem da Arquitetura de Rede

A base do framework é um template de topologia lógica que descreve a arquitetura da rede. O template especifica zonas de segurança, mecanismos de controle de acesso e relações de conectividade, mantendo espaços parametrizáveis para a posterior injeção de ativos, serviços e vulnerabilidades [Hu et al. 2020]. Neste trabalho, utiliza-se o template apresentado na Figura 3, composto por três zonas (*internet*, desmilitarizada (DMZ) e interna), separadas por dois *firewalls*. Os ativos considerados incluem um *webserver*, uma *workstation* e um *fileserver*. A topologia foi inspirada em [Siemens AG 2023], notando que alguns dos caminhos de ataque foram avaliados em [Miranda et al. 2026].

Firewalls. O controle de tráfego entre zonas é formalizado por meio de uma *host access control list* (HACL), representada por predicados lógicos que definem quais comunicações são permitidas, condicionadas a origem, destino, protocolo e porta. Esse conjunto de regras constitui a base para a inferência de acessibilidade na rede (Listing 1).

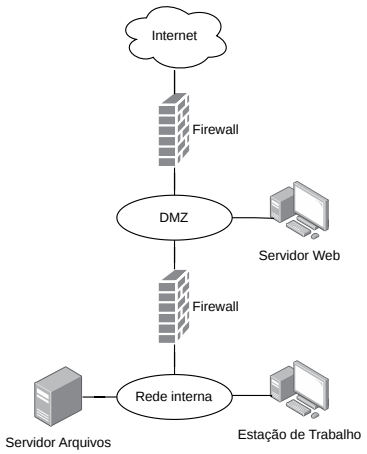


Figura 3. Topologia de rede utilizada para a geração de instâncias.

Listing 1. Regras de definição da topologia.

```
1  hacl(internet, webServer, tcp, 80).
2  hacl(webServer, _, _, _).
3  hacl(fileServer, _, _, _).
4  hacl(workStation, _, _, _).
```

3.3. Inferência Lógica e Classificação

A partir do template definido, criamos automaticamente diferentes instâncias da rede. Em cada execução, são atribuídos perfis distintos de software aos hosts, diferentes vulnerabilidades, níveis de privilégio associados aos serviços e posições iniciais do atacante.

Neste trabalho, o objetivo analisado foi a derivação do predicado `execCode(Host, Priv)`, que representa a capacidade de executar código arbitrário em um host sob determinado nível de privilégio. Em termos semânticos, a regra (em Listing 2) estabelece que a execução remota de código ocorre quando coexistem quatro condições: a presença de uma vulnerabilidade explorável remotamente com impacto de escalonamento de privilégio, a associação dessa vulnerabilidade a um serviço ativo, a execução desse serviço sob determinado privilégio e a possibilidade de acesso de rede à porta correspondente.

Listing 2. Predicado alvo execCode.

```
1  execCode(H, Perm) :-
2     attackerLocated(Zone),
3     hacl(Zone, H, Protocol, Port),
4     vulExists(H, VulID, Software),
5     vulProperty(VulID, remoteExploit, privEscalation),
6     networkServiceInfo(H, Software, Protocol, Port, Perm).
```

A verificação da viabilidade de um ataque é realizada por meio das regras do MulVAL. Com base na derivação ou não do predicado-alvo, cada instância é classificada como positiva ou negativa. Complementarmente, a metodologia introduz dados incompletos, onde o truncamento proposital de evidências (como privilégios ou localização do atacante) desafia o modelo a inferir a viabilidade do ataque por meio de similaridades estruturais, utilizando-se o RDN-Boost [Natarajan et al. 2012] para tal.

3.4. Coleta de Dados e Instanciação do Template com Dados Reais

A instanciação com dados reais do template lógico foi realizada através da coleta de dados de ativos reais. A obtenção das informações relativas a portas abertas, serviços expostos e versões de *software* é realizada por meio da API do *Shodan* [Shodan 2026]. Para cada dispositivo e serviço identificados pelo Shodan são extraídas as versões de *software* e as vulnerabilidades reportadas, identificadas pelo CVE (*Common Vulnerabilities and Exposures*). Essas vulnerabilidades são posteriormente enriquecidas por meio de consultas ao NVD, de onde são obtidas propriedades técnicas relevantes, especialmente aquelas necessárias para instanciar o predicado `vulProperty` (linha 5 do Listing 2).

Exemplos práticos dessa modelagem podem ser observados nos fatos gerados para dois hosts, cujos IPs foram anonimizados para não comprometer a privacidade dos mesmos. No host IPAnon1 (Listing 3), identificou-se o serviço Grafana Open Source 6.7.4 em execução na porta 3000, vulnerável à CVE-2022-21703, classificada como um *remoteExploit* com impacto de *privEscalation*. Essa vulnerabilidade corresponde a uma falha que pode ser explorada para induzir usuários autenticados com altos privilégios a executar ações não intencionais, permitindo que um atacante eleve seus privilégios dentro da plataforma. No restante do trabalho, este host é utilizado como exemplo em que `ExecCode` é verdadeiro.

Já no host IPAnon2 (Listing 4), o serviço Apache HTTPD 2.4.29 foi identificado na porta 80, apresentando múltiplas vulnerabilidades, incluindo a CVE-2024-39573. Em-

bora classificada como *remoteExploit*, seu impacto não foi caracterizado como escalonamento de privilégios. Este host é utilizado como exemplo em que `ExecCode` é falso.

Listing 3. Modelagem do host IPAnon1.

```
vulExists(IPAnon1, CVE-2022-21703, grafana).
vulProperty(CVE-2022-21703, remoteExploit, privEscalation).
networkServiceInfo(IPAnon1, grafana, tcp, 3000, user).
```

Listing 4. Modelagem do host IPAnon2.

```
vulExists(IPAnon2, CVE-2024-39573, apache_httpd).
vulProperty(CVE-2024-39573, remoteExploit, noPrivEscalation).
networkServiceInfo(IPAnon2, apache_httpd, tcp, 80, user).
```

4. Resultados

A seguir, apresentamos os resultados obtidos com o modelo proposto. Após discutirmos a geração do dataset (Seção 4.1), apresentamos o protocolo experimental adotado para o treinamento e validação do modelo sob o cenário de visibilidade total, com dados completos (Seção 4.2). Em seguida, na Seção 4.3, discutimos a inferência qualitativa. Por fim, na Seção 4.4 investigamos a sensibilidade aos hiperparâmetros, avaliando sua influência sobre a capacidade discriminativa do classificador.

4.1. Geração do Dataset

O dataset utilizado neste trabalho foi inspirado em cenários reais de sistemas industriais, em particular arquiteturas similares ao *Siemens PCS7* [Miranda et al. 2026] (Figura 3). A modelagem baseia-se em instâncias representadas por conjuntos de fatos lógicos, compatíveis com a estrutura de entrada do MulVAL, permitindo a avaliação sistemática de inferência e raciocínio probabilístico.

Dataset com dados completos. A construção original do dataset segue um esquema de validação cruzada com 5 *folds*. Em cada *fold*, são geradas 50 instâncias, sendo 25 positivas e 25 negativas. Cada instância corresponde a uma consulta (*query*) acompanhada de um conjunto de fatos (cada fato correspondendo a um dos predicados listados no corpo da regra em Listing 2). As instâncias positivas representam cenários em que um determinado objetivo de ataque é alcançável (derivação bem-sucedida do predicado `execCode`), enquanto as negativas correspondem a cenários em que tal objetivo não pode ser atingido. Cada instância é composta por um conjunto de fatos lógicos correspondentes que descrevem propriedades do sistema, como vulnerabilidades, serviços de rede e condições de exploração. Esse dataset é utilizado no restante da presente seção.

Dataset com dados incompletos. A partir desse dataset base, foram geradas variações destinadas a avaliar a robustez do modelo sob diferentes níveis de incompletude dos dados. Para isso, introduzimos omissão aleatória de fatos com probabilidades $p \in \{0.1, 0.2, \ldots, 0.9\}$. Em cada configuração, mantemos a estrutura de 5 *folds*, com 50 instâncias por *fold*, totalizando 250 instâncias por nível de omissão. Esse dataset é utilizado na Seção 5.

4.2. Treinamento

A avaliação foi conduzida utilizando a técnica de validação cruzada k-fold com $k = 5$. O processo consistiu em cinco rodadas de treinamento e teste. Em cada ciclo, quatro

folds (80% dos dados, ou 200 instâncias) foram utilizados para o treinamento das árvores relacionais, enquanto o *fold* remanescente (20% dos dados, ou 50) foi reservado exclusivamente para o teste. Durante o treinamento, foram adotados os seguintes parâmetros: profundidade máxima das árvores (*max depth*) igual a 5, tamanho mínimo de nó (*minimum node size*) igual a 3 instâncias e número total de estimadores (*n_estimators*) fixado em 20. Uma análise sobre esses hiperparâmetros encontra-se na Seção 4.4.

4.2.1. Treinamento e Desempenho do Modelo

Os resultados obtidos demonstram desempenho consistente do modelo em todos os subconjuntos avaliados. A métrica Area Under the Curve para Precision-Recall (AUC-PR) apresentou valores variando entre 0.8565 e 1.0000, enquanto a Area Under the Curve for Receiver Operating Characteristic (AUC-ROC) variou entre 0.9328 e 1.0000. Observa-se que, em três dos cinco folds, o modelo atingiu valores superiores a 0.95 em ambas as métricas, indicando alta capacidade discriminativa. O desempenho médio obtido foi de 0.9471 para AUC-PR e 0.9678 para AUC-ROC, evidenciando que o método proposto mantém bom poder de generalização.

4.3. Inferência: Análise das Predições do Modelo

Nesta seção, avaliamos a resposta do modelo para os dois hosts coletados via Shodan (o Host IPAnon1, associado a uma vulnerabilidade do tipo *remoteExploit* com impacto de *privEscalation*, e o Host IPAnon2, não).

A Figura 4 apresenta o gráfico resultante, evidenciando a evolução das probabilidades ao longo das iterações de boosting. Observa-se que, à medida que novos estimadores são adicionados, ocorre uma separação progressiva entre as probabilidades associadas aos dois hosts, indicando o refinamento incremental promovido pelo processo de boosting.

Destaca-se que os hosts IPAnon1 e IPAnon2, caracterizados na Seção 3.4, correspondem a instâncias reais e não foram utilizados em nenhuma etapa do treinamento do modelo. Esses hosts apresentam configurações próprias em termos de identificação, serviços ativos, portas expostas e vulnerabilidades associadas.

4.4. Análise de Sensitividade dos Hiperparâmetros

Esta seção apresenta a avaliação comparativa dos principais hiperparâmetros do modelo RDN-Boost. Inicialmente, são analisados os resultados médios obtidos das métricas AUC-ROC e AUC-PR. Em seguida, é apresentado um estudo de caso ilustrativo com dois hosts específicos.

4.4.1. População Completa

A Tabela 2(a) apresenta os valores médios de AUC-PR e AUC-ROC obtidos por validação cruzada com 5 folds. Observa-se que o parâmetro *node_size* exerce influência determinante sobre o desempenho do modelo. Para *node_size* = 2, os resultados permanecem estáveis, com AUC-PR média de 0.5146 e AUC-ROC de 0.5280, independentemente da profundidade da árvore ou do número de estimadores. Por outro lado, ao utilizar

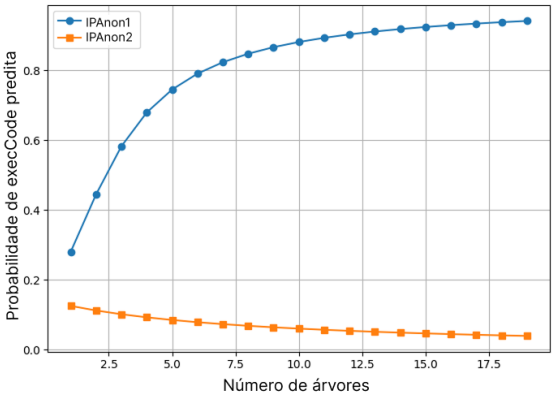


Figura 4. Evolução das probabilidades preditas para os hosts IPAnon1 (azul) e IPAnon2 (laranja), caracterizados na Seção 3.4, em função do número de estimadores.

node_size = 3, verifica-se um salto expressivo de desempenho, alcançando AUC-PR média de 0.9471 e AUC-ROC de 0.9678. Esse comportamento indica que o controle da granularidade mínima dos nós impacta diretamente a capacidade do modelo.

Adicionalmente, nota-se que a variação da profundidade máxima e do número de árvores não produziu alterações nos valores médios obtidos. Por esse motivo, tais variações foram omitidas da tabela, sendo apresentados apenas os resultados representativos. Esse resultado indica saturação estrutural do modelo no cenário avaliado. Como o corpo da regra para o predicado-alvo `execCode` possui apenas cinco literais, uma árvore com `node_size` $= 3$ e profundidade mínima é suficiente para mapear o espaço de estados relacional.

4.4.2. População Restrita

Embora as métricas globais forneçam evidências quantitativas da capacidade discriminativa do modelo, torna-se relevante analisar também seu comportamento probabilístico em um cenário concreto. Para esse fim, foi conduzido um estudo de caso envolvendo dois hosts com características semânticas distintas, caracterizados na Seção 3.4: o Host IPAnon1, associado a uma vulnerabilidade do tipo *remoteExploit* com impacto de *privEscalation*, e o Host IPAnon2, não.

Para cada combinação de parâmetros foram registradas as probabilidades estimadas para ambos os hosts, bem como a diferença entre essas probabilidades (Delta), utilizada como medida de separabilidade do modelo. A Tabela 2(b) apresenta os resultados consolidados. Observa-se que o aumento no número de estimadores refina progressivamente as probabilidades, ampliando o Delta de separabilidade para até 0.7837 quando `node_size` $= 3$, confirmando a sensibilidade do modelo às propriedades semânticas do ataque.

5. Resiliência a Dados Incompletos

Esta seção apresenta os resultados do *stress test* de resiliência, cujo objetivo foi avaliar a robustez do modelo RDN-Boost diante da degradação progressiva da integridade dos

Tabela 2. Resultados experimentais para diferentes configurações de hiperparâmetros. À esquerda: métricas médias de validação cruzada (AUC-PR e AUC-ROC). À direita: probabilidades previstas e diferença entre classes. As árvores podem ser visualizadas em [Banjar et al. 2026].

Depth	Node	Trees	AUC-PR	AUC-ROC
3	2	5	0.5146	0.5280
		10	0.5146	0.5280
		20	0.5146	0.5280
	3	5	0.9471	0.9678
		10	0.9471	0.9678
		20	0.9471	0.9678

(a)

Depth	Node	Trees	Prob IPAnon1	Prob IPAnon2	Delta
3	2	5	0.4742	0.0852	0.3890
		10	0.5885	0.0604	0.5280
		20	0.6226	0.0380	0.5845
	3	5	0.6339	0.0852	0.5487
		10	0.7687	0.0604	0.7083
		20	0.8217	0.0380	0.7837

(b)

dados. O experimento consistiu na variação do parâmetro de omissão uniforme, variando de 0.0 (conhecimento pleno) a 1.0, aplicado em um esquema de validação cruzada com 5 *folds*.

5.1. Análise Quantitativa de Desempenho

A estratégia de treinamento também incluiu a omissão deliberada de predicados essenciais para capacitar o modelo a reconhecer padrões de ataque mesmo em condições de baixa observabilidade. Essa abordagem é motivada pela realidade operacional de redes corporativas, onde a visibilidade completa é raramente alcançada devido a falhas de varredura ou informações incompletas sobre ativos e vulnerabilidades.

Os resultados sumarizados na Tabela 3 revelam que o modelo apresenta uma degradação suave de desempenho, mantendo métricas elevadas mesmo sob condições de incerteza moderada. Com até 20% de omissão de dados ($p = 0.2$), o sistema sustenta uma AUC-ROC de 0.899, o que demonstra a eficácia do aprendizado relacional em compensar a ausência de evidências explícitas por meio de correlações estruturais no grafo de ataque.

Tabela 3. Desempenho (Média de 5 *folds*) em função da probabilidade de omissão (p).

p	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
AUC-PR	0.97	0.94	0.90	0.82	0.82	0.72	0.69	0.75	0.60	0.58	0.50
AUC-ROC	0.98	0.95	0.90	0.85	0.82	0.75	0.67	0.74	0.62	0.57	0.50

Nota-se que o ponto de inflexão ocorre próximo a $p = 0.5$, onde a AUC-ROC declina para 0.751. A partir deste limiar, a escassez de predicados essenciais — como `vulProperty` (características da vulnerabilidade) e `attackerLocated` (origem do ataque) — compromete a capacidade do algoritmo em distinguir caminhos de exploração viáveis de meras falhas de configuração. No limite extremo ($p = 1.0$), o modelo converge para o desempenho de um classificador aleatório (0.5).

5.2. Distribuição de Probabilidades e Confiança

As Figuras 5 e 6 apresentam as Funções de Distribuição Acumulada (CDF) para as instâncias positivas e negativas, respectivamente. Em cenários de baixa omissão ($p \leq$

0.2), a massa da distribuição para instâncias positivas concentra-se fortemente à direita, refletindo alta confiança e baixa incerteza nas predições de risco.

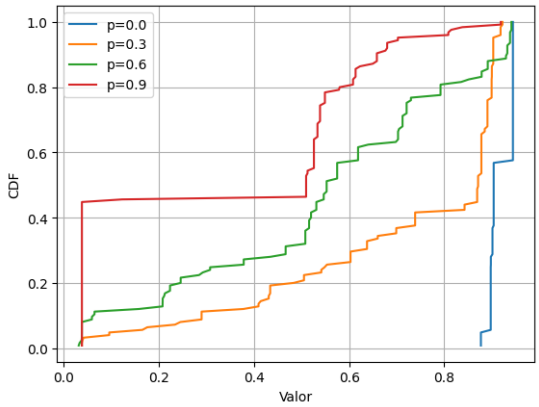


Figura 5. CDF das probabilidades para instâncias positivas (*exec-Code*).

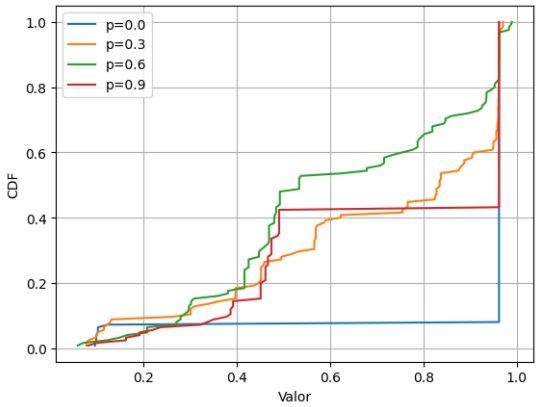


Figura 6. CDF das probabilidades para instâncias negativas (*!exec-Code*).

À medida que a omissão de dados (p) aumenta, observa-se uma transição no perfil das curvas: o formato originalmente convexo, que indicava decisões nítidas e polarizadas, torna-se progressivamente linear. Nas instâncias positivas, essa mudança evidencia que o RDN-Boost passa a atribuir probabilidades intermediárias (em torno de 0.5) devido à fragmentação do caminho de ataque.

6. Conclusão

Este trabalho investigou a aplicação de SRL na modelagem de caminhos de ataque, demonstrando que a integração de lógica de primeira ordem [Ou et al. 2005] com modelos probabilísticos [Miranda et al. 2026] supera a limitação do CVSS de ignorar o contexto organizacional das vulnerabilidades. A metodologia proposta permitiu a construção de uma abordagem capaz de processar dados reais extraídos das APIs Shodan e NVD, convertendo informações técnicas em predicados lógicos estruturados para a inferência de riscos em ambientes de rede.

Os resultados experimentais evidenciaram que o modelo RDN-Boost capturou a semântica das invasões, alcançando um desempenho médio de 0,9471 em AUC-PR e 0,9678 em AUC-ROC. Um diferencial crítico observado foi a resiliência do modelo diante da incerteza: mesmo com a omissão proposital de 20% dos fatos estruturais (p=0,2), o sistema manteve uma AUC-ROC de 0,8998. Essa capacidade de compensar a ausência de evidências explícitas por meio de inferências probabilísticas no grafo de ataque valida a robustez da abordagem para cenários de segurança real, onde a visibilidade dos ativos é frequentemente fragmentada. Em suma, a pesquisa confirmou que o risco em redes é inerentemente relacional e que o uso de SRL é uma via promissora para a automação da segurança baseada em contexto.

Como perspectivas para trabalhos futuros, pretende-se ampliar a validação experimental por meio de estudos em larga escala, incorporando um maior número de instâncias e cenários que reflitam a diversidade e complexidade de redes corporativas reais. Adicionalmente, busca-se a integração de um sistema de recomendação automatizado que utilize

as probabilidades de ataque para sugerir ações dinâmicas de *patching* e *rezoning*, contribuindo para a construção de um framework de defesa ativa e adaptativa.

Agradecimentos. Este trabalho foi financiado pela CAPES, FAPERJ E-26/204.268/2024 e E-26/260.168/2026, CNPq 444956/2024-7, 424622/2021-1, 308376/2021-8 e 315106/2023-9, bem como Finep Clavis PlatCiber.

Referências

Aksu, M. U., Bicakci, K., Dilek, M. H., Ozbayoglu, A. M., and Tatli, E. i. (2018). Automated generation of attack graphs using nvd. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY '18, page 135–142, New York, NY, USA.

Allodi, L. and Massacci, F. (2017). Comparing vulnerability severity and exploits using case–control studies. *ACM Transactions on Information and System Security*, 19(4):1–31.

Banjar, C. E. d. S., Bicudo, M. A. S., Miranda, L., Pereira, C. a. F., Coutinho, L. S., Menasche, D. S., Srivastava, G. K., Lovat, E., Kocheturov, A., Martins, M., and de Aguiar, L. P. (2024). Automated severity driven patch management. In *LADC*, LADC '24, page 179–183, New York, NY, USA.

Banjar, C. E. d. S., Miranda, L. G., Menasché, D. S., and Zaverucha, G. (2026). Modelagem e avaliação de aprendizado estatístico relacional na detecção de falhas de segurança. `https://tinyurl.com/wperf2026`. Material suplementar para o presente artigo.

Hu, Z., Beuran, R., and Tan, Y. (2020). Automated penetration testing using deep reinforcement learning. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 2–10.

Lei, M. (2023). *Risk Score Aggregation for Security Evaluation in Network Environments*. Tese (mestrado em ciência da computação), Carleton University, Ottawa.

Mell, P., Scarfone, K., and Romanosky, S. (2006). Common vulnerability scoring system. *IEEE Security & Privacy*, 4(6):85–89.

Miranda, L., de Schuller Banjar, C. E., Menasché, D. S., Kocheturov, A., Srivastava, G. K., and Limmer, T. (2026). An Automated Approach to Generate Attack Graphs with a Case Study on Siemens PCS7 Blueprint. In *SVM, co-located with ICSE 2026*. IEEE/ACM. arXiv:2603.24888.

Natarajan, S., Khot, T., Kersting, K., et al. (2012). Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 86(1):25–56.

Ou, X., Govindavajhala, S., and Appel, A. W. (2005). Mulval: a logic-based network security analyzer. In *USENIX Security*, SSYM'05, page 8, USA.

Pereira, C., Gabriel Lopes de Oliveira, J., Azevedo Santos, R., Vieira, D., Miranda, L., Zaverucha, G., Pfleger de Aguiar, L., and Sadoc Menasché, D. (2023). A statistical relational learning approach towards products, software vulnerabilities and exploits. *IEEE TNSM*, 20(3):3782–3802.

Pereira, C. F., Menasché, D. S., Zaverucha, G., Paes, A., and Barbosa, V. C. (2025). A utility-driven approach to instance-based transfer learning for relational domains. *Machine Learning*, 114(11):261.

Shodan (2026). Shodan. `https://www.shodan.io/`.

Siemens AG (2023). Secure Guideline Blue Print Water/Waste Water Treatment Plant. `https://cache.industry.siemens.com/dl/files/322/109780322/att_1275024/v1/Secure_Guideline_Blue_Print-WWTP_V04_en.pdf`. Accessed: 2 de junho de 2026.

Wang, L., Islam, T., Long, T., Singhal, A., and Jajodia, S. (2008). An attack graph-based probabilistic security metric. In *DBSEC*, pages 283–296. Springer.

Yee, G. (2013). *Security Metrics: An Introduction and Literature Review*, pages 559–571.