Anais do Congresso Brasileiro de Iniciação Científica
(https://periodicos.unisanta.br/COB)

ISSN 2317-1316

DETECÇÃO DE *RANSOMWARE* EM PADRÕES DE ACESSO A ARMAZENAMENTO COM *GRAPH NEURAL NETWORKS*: GENERALIZAÇÃO PARA FAMÍLIAS INÉDITAS E O PAPEL DA REGRA DE DECISÃO

RANSOMWARE DETECTION FROM STORAGE ACCESS PATTERNS WITH GRAPH NEURAL NETWORKS: GENERALIZATION TO UNSEEN FAMILIES AND THE ROLE OF THE DECISION RULE

Lucas Rocha Gaspar, José Augusto Theodosio Pazetti, Juliana da Silva Linhares

Faculdade de Tecnologia (Fatec) Rubens Lara – Baixada Santista, Centro Paula Souza, Curso de Ciência de Dados
E-mail para contato: lucas.gaspar@aluno.cps.sp.gov.br

RESUMO – Detectores de ransomware *baseados em aprendizado profundo são habitualmente avaliados sobre as mesmas famílias usadas no treinamento. Avaliou-se uma representação em grafos de regiões de armazenamento, processada por GraphSAGE, sobre o conjunto RanSMAP, com exclusão de família completa e troca de* hardware*. Em dezoito treinamentos o modelo alcançou área sob a curva (AUC) de 0,999 por janela e acerto integral por execução dentro da distribuição, mas perdeu dois terços das execuções de uma família retirada do treino, sem que a validação acusasse degradação. A falha mostrou-se de calibração e não de discriminação: a AUC por execução permaneceu em 1,000. Trocar o limiar fixo por decisão relativa às execuções benignas locais recuperou sensibilidade integral sem retreinamento.*

Palavras-chave: Detecção de *ransomware*; *Graph Neural Networks*; Padrões de acesso a armazenamento; Generalização; Calibração.

ABSTRACT – Deep learning ransomware detectors are usually evaluated on the same families used for training. A graph representation of storage regions, processed by GraphSAGE, was evaluated on the RanSMAP dataset under complete family exclusion and hardware change. Across eighteen training runs the model reached an area under the curve (AUC) of 0.999 per window and perfect per-execution accuracy within distribution, yet lost two thirds of the executions of a family withheld from training, with no degradation signalled by validation. The failure proved to be one of calibration, not discrimination: the per-execution AUC remained at 1.000. Replacing the fixed threshold by a decision relative to local benign executions restored full sensitivity without retraining.

Keywords: *Ransomware detection; Graph Neural Networks; Storage access patterns; Generalization; Calibration.*

ISSN 2317-1316

1 INTRODUÇÃO

O *ransomware* é um tipo de *malware* que cifra os arquivos da vítima e cobra resgate pela chave de decifragem, sem que o pagamento garanta a recuperação dos dados. Detectores fundados em assinatura estática falham diante de variantes inéditas, razão pela qual a detecção comportamental, que observa o que o programa faz em vez do que ele é, ocupa posição central na literatura recente (Begovic *et al.*, 2023). Entre as camadas de observação disponíveis, como as chamadas de sistema e de interface de programação de aplicações (API) e o monitoramento das atividades do sistema de arquivos, os padrões de acesso ao dispositivo de armazenamento apresentam duas vantagens: situam-se abaixo do sistema operacional, fora do alcance de um processo malicioso, e registram diretamente a manifestação física da cifragem, que é a gravação de conteúdo de entropia elevada sobre regiões lidas momentos antes.

O conjunto RanSMAP (Hirano e Kobayashi, 2024) disponibiliza esses registros para seis famílias de *ransomware* e seis aplicações benignas, coletados por hipervisor forense. A escolha das aplicações benignas é deliberada e torna o problema não trivial: entre elas figuram um utilitário de cifragem simétrica e um compressor, que também gravam conteúdo de entropia elevada. A entropia isolada, portanto, não separa as classes.

A degradação de detectores diante de famílias ausentes do treino já está documentada. Chen *et al.* (2022), classificando *malware* por grafos de chamadas de função, relatam queda de acurácia de 0,920 para 0,704 ao avaliar sobre famílias não observadas em treinamento. O que a literatura raramente distingue, contudo, é a natureza dessa queda: se o modelo deixa de separar as classes ou se apenas desloca a escala com que as pontua. A distinção não é acadêmica, pois a primeira situação exige modelo novo e a segunda exige apenas regra de decisão nova.

Este trabalho avalia uma representação em grafos dessa camada, na qual os nós correspondem a regiões do dispositivo e as arestas à vizinhança entre elas, processada por *Graph Neural Networks* (Hamilton *et al.*, 2017). O objetivo não foi verificar se o modelo detecta *ransomware*, questão que se resolve rapidamente, mas medir o que acontece quando ele encontra uma família que nunca viu e uma plataforma diferente daquela em que foi treinado. Medida a degradação, buscou-se determinar a sua natureza e corrigi-la.

ISSN 2317-1316

2 MATERIAIS E MÉTODOS

2.1 Conjunto de dados

O RanSMAP reúne 1.970 execuções distribuídas em quatro cenários: (i) *original*, com os doze programas isolados; (ii) *extra*, com configurações adicionais de processador e memória; (iii) *mix*, com *ransomware* em execução simultânea a uma aplicação benigna; e (iv) *variants*, com sete configurações do Conti. São seis famílias maliciosas (Conti, Darkside, LockBit, REvil, Ryuk e WannaCry) e seis aplicações benignas (AESCrypt, Firefox, Idle, Office, SDelete e Zip), cada programa executado dez vezes por combinação de *hardware*. Uma execução foi descartada por não conter o arquivo de leituras, restando 1.969.

Os registros de armazenamento trazem marca temporal, endereço lógico de bloco, tamanho da operação e, nas escritas, a entropia de Shannon (Shannon, 1948) do conteúdo gravado, normalizada entre zero e um. Os registros de memória foram empregados apenas como contexto: o instrumento de coleta os captura por violações de *Extended Page Table* e remove as entradas a cada trinta segundos, de modo que revisitas a uma mesma página não são observáveis na escala relevante (Hirano e Kobayashi, 2022).

2.2 Representação em grafos

Cada janela de dez segundos origina um grafo. Os nós correspondem a regiões de 64 KiB do dispositivo e recebem oito atributos: contagens de leitura e de escrita, volume gravado, entropia média e máxima, fração de escritas-chave, intervalo médio entre leitura e reescrita e extensão temporal da atividade na região. Define-se escrita-chave como a gravação de conteúdo com entropia superior a 0,9 em região lida nos sessenta segundos anteriores, janela de memória dimensionada pelos intervalos medidos em estudo piloto, nos quais a mediana entre leitura e reescrita das famílias maliciosas situou-se entre 11,7 e 30,7 segundos, contra milissegundos nas aplicações benignas.

As arestas ligam regiões separadas por no máximo quatro posições na ordenação de endereços, com atributos de distância e de diferença temporal média. Construiu-se também uma variante na qual as arestas seguem a ordem de acesso em vez da adjacência de endereço, empregada como controle.

ISSN 2317-1316

2.3 Modelos

A arquitetura principal combina três camadas GraphSAGE de 64 dimensões, normalização em lote, *readout* por concatenação de média e máximo e um classificador de duas camadas, totalizando 26.305 parâmetros. Cada camada GraphSAGE atualiza o vetor de uma região somando uma transformação dos seus próprios atributos a outra da média dos atributos das regiões vizinhas, de modo que, após três camadas, cada região incorpora informação das que estão a até três passos de distância no grafo (Hamilton *et al.*, 2017). É essa agregação de vizinhança que a ablação descrita a seguir remove. Para isolar a contribuição da estrutura, empregou-se uma ablação idêntica em tudo exceto nas camadas de convolução, que são removidas: os atributos dos nós seguem diretamente ao *readout*. Como referência não aprendida, adotou-se uma regra de limiar sobre a fração de escritas-chave, calibrada em cada partição pelo índice de Youden, que escolhe o limiar maximizando a soma de sensibilidade e especificidade (Youden, 1950).

2.4 Protocolo de avaliação

Como o conjunto reúne doze programas distintos, a unidade de inferência defensável é a execução e não a janela: janelas de uma mesma execução são correlacionadas e distribuí-las entre treino e teste infla o desempenho sem que o efeito apareça nas métricas. Todas as partições agrupam por execução, e a pontuação de uma execução é a média das probabilidades das suas janelas.

Três partições foram empregadas, todas definidas neste trabalho. A primeira separa execuções do cenário *original*, com avaliação adicional sobre *mix* e *variants*. A segunda treina em uma configuração de *hardware* e avalia em outra. A terceira exclui uma família inteira do treino, repetindo o procedimento para cada uma das seis. A divisão é estratificada por classe, de modo que a proporção entre execuções benignas e maliciosas se conserva em treino, validação e teste, e permanece fixa entre as sementes, que variam apenas a inicialização dos pesos e a ordem dos lotes: a primeira partição reserva 216 execuções para teste, e a segunda treina em 1.223 e avalia em 360. Cada configuração foi treinada com três sementes distintas, somando dezoito treinamentos por arquitetura. O ambiente foi Windows 11 com PyTorch 2.9.1 sobre ROCm 7.2.1 e unidade gráfica AMD Radeon RX 9060 XT, com a biblioteca PyTorch Geometric (Fey e Lenssen, 2019). O treinamento empregou o otimizador AdamW

ISSN 2317-1316

com taxa de aprendizado de 0,001 e decaimento de peso de 0,0001, lotes de 32 grafos, redução da taxa por estagnação da validação e parada antecipada por paciência de dez épocas. Os programas de pré-processamento, construção dos grafos, particionamento, treinamento e avaliação, com os hiperparâmetros empregados, serão disponibilizados em repositório aberto, de modo a permitir a reprodução integral do procedimento.

3 RESULTADOS E DISCUSSÃO

3.1 Caracterização dos grafos

A Tabela 1 resume as medianas por programa sobre 960 execuções amostradas do cenário *original*. Dois pontos orientam a leitura dos resultados seguintes. O primeiro é que a fração de escritas-chave separa bem as classes na maioria dos casos, mas é praticamente nula no REvil, cuja mediana é zero e no qual apenas 41,4% das janelas apresentam algum nó com maioria de escritas-chave. O segundo é que os três programas benignos de maior vazão (Zip, SDelete e AESCrypt) produzem grafos tão densos quanto os da família mais densa, de modo que a topologia tampouco separa as classes por si.

Tabela 1: Caracterização dos grafos por programa, medianas sobre 960 execuções

Programa	Tipo	Nós	Grau	Entropia	Escritas-chave
Zip	benigno	3.276	3,507	0,947	0,001
SDelete	benigno	5.224	3,505	0,358	0,030
AESCrypt	benigno	4.611	3,399	0,943	0,002
WannaCry	*ransomware*	5.320	3,369	0,896	0,349
REvil	*ransomware*	3.121	2,814	0,430	0,000
Conti	*ransomware*	4.138	2,489	0,669	0,251
LockBit	*ransomware*	5.571	2,394	0,750	0,344
Darkside	*ransomware*	2.289	2,272	0,616	0,216
Firefox	benigno	304	2,196	0,357	0,000
Ryuk	*ransomware*	937	2,129	0,532	0,055
Idle	benigno	94	1,583	0,355	0,000
Office	benigno	83	1,446	0,424	0,000

3.2 Desempenho dentro da distribuição

Na partição por execução do cenário *original*, o GraphSAGE atingiu área sob a curva (AUC) de 0,999 por janela, com um único falso positivo entre 2.142 janelas benignas, e acerto integral nas 216 execuções de teste. A regra de limiar, na mesma partição, alcançou 0,841. A ablação sem estrutura atingiu 0,996, com 53 falsos positivos. A estrutura, portanto, contribui sobretudo para a precisão, reduzindo o alarme falso em cerca de uma ordem de magnitude.

ISSN 2317-1316

3.3 Controles de capacidade

A ablação sem estrutura possui 1.153 parâmetros com 64 dimensões ocultas, contra 26.305 do GraphSAGE, de modo que a diferença de capacidade constitui explicação concorrente para qualquer discrepância entre as duas. Executaram-se então duas manipulações em direções opostas. Reduzir o GraphSAGE a 16 dimensões e duas camadas piorou a detecção no conjunto de cenários mistos, de 0,7244 para 0,6996. Ampliar a ablação para 256 dimensões, quadruplicando os seus parâmetros para 4.609, não a alterou: 0,8047 contra 0,8026, diferença inferior ao ruído entre sementes.

Nenhuma das duas manipulações reproduz a diferença entre as famílias de modelo, o que afasta a capacidade como explicação. Os parâmetros não ficaram pareados, de modo que o argumento é mais fraco do que um pareamento exato: a capacidade variou dentro de cada família sem explicar a diferença entre elas. A ablação de 256 dimensões foi adotada nos experimentos subsequentes, por ser a melhor da sua família.

3.4 Generalização para *hardware* inédito

Treinando em 1.223 execuções e avaliando em 360 de configurações de processador e memória não observadas, o GraphSAGE obteve AUC de 0,9976 por janela, com oito falsos positivos entre 3.499 janelas benignas, contra 0,9925 e 234 falsos positivos da ablação. Ambos acertaram as 360 execuções de teste. Sob troca de plataforma, portanto, a estrutura não perde: ganha, com vinte e nove vezes menos alarme falso.

O resultado tem consequência sobre a interpretação dos cenários mistos. Se a estrutura prejudicasse a generalização em geral, deveria prejudicá-la também aqui. Não prejudica, o que indica que os dois cenários medem coisas distintas: a troca de plataforma altera a medição de um mesmo comportamento, ao passo que os cenários mistos introduzem composição comportamental inédita.

3.5 Generalização para família inédita

A Tabela 2 apresenta a AUC por janela quando cada família é retirada do treino, em média de três sementes. Em cinco das seis dobras o GraphSAGE supera a ablação. Na sexta, a do WannaCry, ele cai para 0,854, e a queda se repete nas três sementes.

ISSN 2317-1316

Tabela 2: AUC por janela com exclusão de família completa, média de três sementes

Família excluída	GraphSAGE	Sem estrutura	Regra de limiar
Conti	0,9991	0,9981	0,9465
Darkside	0,9914	0,9898	0,9145
LockBit	0,9994	0,9976	0,9392
REvil	0,9942	0,9827	0,5560
Ryuk	0,9916	0,9831	0,8296
WannaCry	0,8539	0,9940	0,9632

A regra de limiar merece registro próprio, por recair sobre o marcador que a literatura da área emprega. Na dobra do REvil ela desaba para 0,556, resultado coerente com a Tabela 1: a fração de escritas-chave é praticamente inexistente naquela família, com mediana nula e apenas 41,4% das janelas apresentando algum nó em que a maioria das escritas seja escrita-chave, contra proporções entre 89% e 96% nas demais. Os dois modelos aprendidos alcançam ali 0,9942 e 0,9827, o que demonstra que detectam aquela família por via diversa da entropia de escrita. O Ryuk, segundo mais fraco nesse marcador com 0,055 de mediana, é a segunda dobra mais difícil para a regra, com 0,830. Uma de cada seis famílias escapa ao marcador isolado, e esse resultado sustenta por si o argumento a favor de aprender sobre o conjunto de atributos em vez de calibrar um corte sobre um deles.

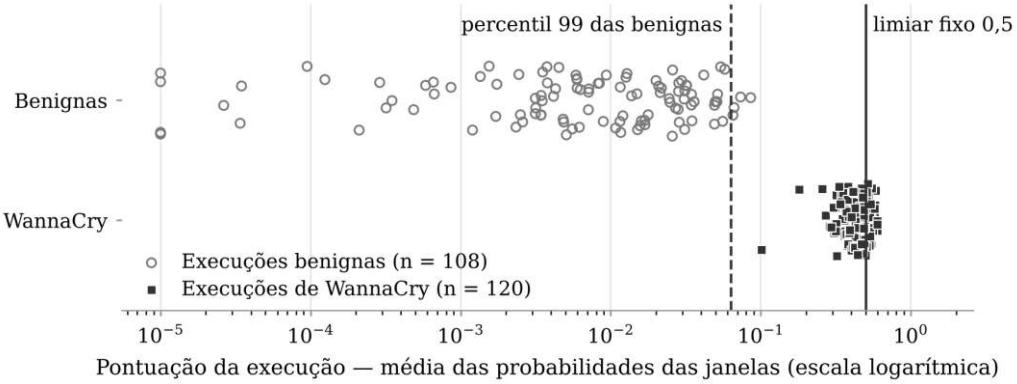
A degradação na dobra do WannaCry não foi anunciada por nenhuma métrica interna. Nas três sementes a AUC de validação situou-se entre 0,9993 e 0,9995, a melhor das seis dobras. Um procedimento que selecionasse modelo por validação entregaria esse detector sem qualquer sinal de alerta, o que converte a avaliação por exclusão de família completa de rigor opcional em requisito.

3.6 Natureza da falha

A AUC por execução nessa mesma dobra permaneceu entre 0,9988 e 1,0000. O modelo, portanto, ordena as execuções corretamente, separando WannaCry de benigno sem erro, e ainda assim lhes atribui probabilidade mediana de 0,47 por execução, abaixo do limiar convencional de 0,5. Trata-se de falha de calibração, não de discriminação, e a Figura 1 exibe a distribuição que a produz.

ISSN 2317-1316

Figura 1: Pontuação por execução na dobra do WannaCry. A separação entre as classes é completa, e o limiar fixo corta o interior do agrupamento malicioso.



A maior pontuação benigna é 0,0856 e a menor pontuação de WannaCry é 0,1014: há um intervalo vazio entre as duas classes, e o limiar de 0,5 não se encontra nele. A mesma assinatura aparece na variância entre sementes. A AUC da ablação reproduz-se com amplitude entre 0,0002 e 0,0081 nas seis dobras, ao passo que a sensibilidade do GraphSAGE no limiar fixo oscila uma ordem de magnitude mais: 0,1065 na dobra do Darkside e 0,1499 na do WannaCry, contra 0,0065 e 0,0072 da ablação. O ordenamento é estável; a régua não é.

3.7 Hipóteses descartadas

Formularam-se e descartaram-se três explicações para o deslocamento, cada uma por medição direta sobre os grafos ou sobre os modelos treinados.

A primeira atribuiu a falha ao agrupamento da evidência na vizinhança: o modelo teria aprendido a confiar em regiões com muitas escritas-chave adjacentes entre si, e o WannaCry distribuiria essa evidência entre regiões isoladas, de modo que a agregação de vizinhança a diluiria. Construiu-se para testá-la a fração de arestas que ligam dois nós com maioria de escritas-chave, normalizada pelo valor esperado caso tais nós estivessem distribuídos ao acaso. A separação entre o WannaCry e as demais famílias resultou em 0,380, isto é, sobreposição quase total e na direção contrária à prevista.

A segunda atribuiu a falha a topologia ausente do treino, apoiada em que o WannaCry apresenta grau médio de 3,369 contra 2,385 das demais famílias maliciosas. A hipótese caiu por erro na sua própria formulação: a comparação fora feita apenas contra as outras cinco famílias, quando o conjunto de treino contém também os seis programas benignos. Ordenados os doze programas por densidade de conexão, conforme a Tabela 1, os três primeiros são

ISSN 2317-1316

benignos (Zip, SDelete e AESCrypt) e o WannaCry é o quarto. O treino visitou aquela região do espaço continuamente, e sempre rotulada como benigna.

A terceira atribuiu a falha a posição atípica no espaço de atributos de nó, dado ser o WannaCry o único programa simultaneamente alto em entropia e em fração de escritas-chave. Também caiu: 94,0% das suas janelas ocupam o quadrante alto-alto definido pelas medianas do treino, e naquela dobra esse quadrante contém 6.282 janelas de treino, das quais 85,9% são de *ransomware*. Os atributos de nó apontam, portanto, fortemente para a classe correta, e o treino observou essa configuração em abundância.

Registre-se ainda uma previsão falhada, formulada antes do experimento que a testaria. Previu-se que trocar o critério de aresta da adjacência de endereço para a ordem de acesso reduziria o colapso, uma vez que tal troca reduz em dois terços a anomalia topológica do WannaCry. O colapso permaneceu inalterado, com sensibilidade de 0,4797 contra 0,4772, o que exclui a vizinhança de endereços como causa.

3.8 Inspeção do vetor agregado

Abandonada a via das hipóteses, passou-se a medir diretamente. Extraiu-se dos modelos treinados o vetor produzido pelo *readout*, anterior ao classificador, e projetou-se a sua distribuição sobre a reta que liga a mediana das execuções benignas de treino à das maliciosas, escala na qual zero corresponde ao centro benigno e um ao centro malicioso. A Tabela 3 apresenta o resultado.

Tabela 3: Posição do WannaCry na reta que separa as classes, decomposição do seu deslocamento e dispersão perpendicular das janelas de treino, em frações da distância entre os centros

Medida	GraphSAGE	Sem estrutura
Benignas de treino	0,055	0,009
Ransomware de treino	1,020	1,005
WannaCry, excluído do treino	0,713	1,478
Deslocamento ao longo do eixo	−0,286	+0,478
Deslocamento perpendicular	0,738	0,117
Dispersão perpendicular típica no treino	0,746	0,380

Sem as camadas de convolução, o WannaCry situa-se em 1,478, além do próprio centro das famílias maliciosas de treino, com deslocamento perpendicular de 0,117. Com as camadas de convolução, situa-se em 0,713, aquém daquele centro, e o deslocamento perpendicular sobe para 0,738.

ISSN 2317-1316

A leitura imediata desses números seria a de que as convoluções convertem uma extrapolação alinhada ao eixo de decisão em outra, ortogonal a ele. Dois controles a desautorizam. O primeiro é a última linha da Tabela 3: o deslocamento perpendicular de 0,738 coincide com a dispersão perpendicular das próprias janelas de *ransomware* de treino, de 0,746, de modo que ele descreve a geometria de um espaço de 128 dimensões e não uma anomalia do WannaCry. O segundo é que 81,3% das janelas de WannaCry situam-se além da posição em que a probabilidade das janelas de treino cruza 0,5 ao longo desse eixo, e ainda assim recebem probabilidade mediana de 0,289 por janela. A projeção descreve, portanto, onde o WannaCry está, mas não explica a sua pontuação: a decisão ocorre em direções que ela não captura. Nenhuma dimensão isolada responde pelo efeito: substituir uma só dimensão do vetor do WannaCry pela mediana das janelas de *ransomware* de treino eleva a probabilidade mediana em no máximo 0,116, de 0,289 para 0,404, aquém do limiar, ao passo que substituir as 128 ao mesmo tempo a leva a 0,9998. O déficit está distribuído, e o classificador responde normalmente quando ele é removido por inteiro. A causa última permanece em aberto, e o texto não lhe atribui explicação que os dados não sustentem.

3.9 Regra de decisão

Qualquer que seja a causa, a ordem das execuções está correta e o corte é que não está, conforme a Seção 3.6. Substituiu-se então o limiar fixo por uma regra relativa: classifica-se como maliciosa a execução cuja pontuação supere o percentil das pontuações das execuções benignas observadas. A calibração não emprega qualquer amostra de *ransomware*, o que corresponde à condição real de implantação, na qual exemplos da próxima família não existem e registros de operação normal existem. A regra opera sobre os modelos já treinados e não exige retreinamento.

A dobra do WannaCry passa de 0,3167 a 1,0000 de sensibilidade, com especificidade de 0,9630, conforme a Figura 2. O ganho persiste no percentil 100%, isto é, com corte acima da maior pontuação benigna já observada, o que indica separação completa e não calibração afortunada. Duas dobras que o limiar fixo não acertava integralmente passam a acertar.

O resultado de maior consequência, contudo, recai sobre o conjunto de cenários mistos, que havia motivado a investigação. Com a ablação sem estrutura, a sensibilidade por execução passa de 0,7647 no limiar fixo para 1,0000 em todos os percentis entre 95% e

ISSN 2317-1316

99,5%, isto é, 170 execuções em 170. Os quatro programas que se mostraram mais difíceis ao longo de todo o estudo apresentam probabilidade mediana entre 0,26 e 0,38, abaixo do limiar convencional e acima do corte benigno de 0,1015, e são integralmente detectados. A falha nos cenários mistos nunca foi de discriminação.

Tabela 4: Sensibilidade e especificidade por execução, limiar fixo e percentil 99% (P99) das execuções benignas, arquitetura GraphSAGE. n.a.: não aplicável, pois o conjunto de cenários mistos não contém execuções benignas isoladas.

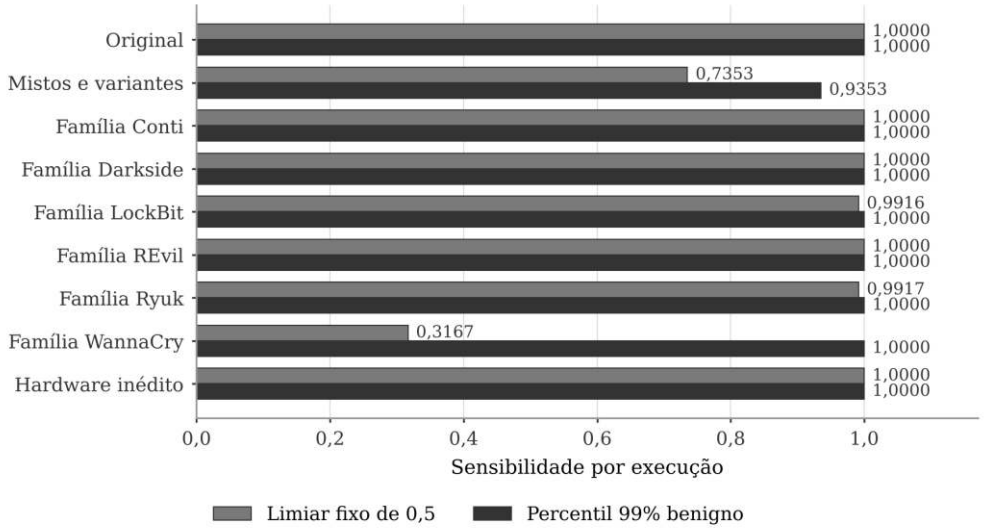
Partição	Fixo: sens.	Fixo: espec.	P99: sens.	P99: espec.
Original	1,0000	1,0000	1,0000	0,9907
Mistos e variantes	0,7353	n.a.	0,9353	n.a.
Família Conti	1,0000	1,0000	1,0000	1,0000
Família Darkside	1,0000	1,0000	1,0000	0,9815
Família LockBit	0,9916	1,0000	1,0000	0,9815
Família REvil	1,0000	1,0000	1,0000	0,9815
Família Ryuk	0,9917	1,0000	1,0000	0,9907
Família WannaCry	0,3167	1,0000	1,0000	0,9630
Hardware inédito	1,0000	1,0000	1,0000	0,9778

Com a regra corrigida, a comparação entre as arquiteturas inverte-se em relação à leitura inicial. Nas seis dobras de família o GraphSAGE acumula onze falsos positivos contra trinta da ablação, em 648 execuções benignas cada, com sensibilidade integral em ambas. O compromisso entre capacidade de representação e generalização, medido ao longo de dezoito treinamentos, era em larga medida artefato da régua empregada para medi-lo.

A regra exige, contudo, que a calibração seja feita no ambiente de destino. Calibrada na plataforma de treino e aplicada à plataforma nova, a especificidade cai para 0,8500 no GraphSAGE e 0,8556 na ablação. A igualdade praticamente exata dos dois valores indica propriedade do método de calibração e não do modelo. Recalibrada com noventa execuções benignas da plataforma de destino, retorna a 0,9778 e 0,9667. O deslocamento ocorre na cauda da distribuição benigna, cujo máximo passa de 0,0265 para 0,0860 entre as duas plataformas, e não no seu centro. Registre-se que, com noventa execuções de calibração, os percentis 99%, 99,5% e 100% produzem resultado idêntico, de modo que nesse tamanho de amostra a regra é, de fato, a de corte acima da maior pontuação benigna observada.

ISSN 2317-1316

Figura 2: Sensibilidade por execução sob limiar fixo de 0,5 e sob o percentil 99% das execuções benignas, por partição, arquitetura GraphSAGE



3.10 Limites observados

Um resíduo sobrevive à correção. No conjunto de cenários mistos o GraphSAGE alcança 0,9353 e não 1,0000, e nove das onze execuções não detectadas são a combinação do REvil com o AESCrypt em execução simultânea. É a conjunção dos dois fatores adversos identificados separadamente: a família cujo marcador de entropia é mais fraco, mascarada pela aplicação benigna de perfil de acesso mais semelhante ao de *ransomware*. A combinação do AESCrypt com o Conti, família de marcador forte, é detectada integralmente, de modo que não se trata de mascaramento genérico. A ablação sem estrutura detecta o caso difícil, o que sugere que a agregação de vizinhança mistura evidência dos dois processos quando ambos gravam em regiões intercaladas.

Registre-se, por fim, uma observação de reprodutibilidade obtida sem família inédita e sem troca de plataforma. Dois treinamentos com a mesma semente, os mesmos dados, os mesmos hiperparâmetros e a mesma partição produziram AUC de 0,9991 e 0,9987, mas detecções de 0,7244 e 0,6838 no conjunto de cenários mistos, quatro pontos percentuais de diferença. A causa provável é o não determinismo das operações de dispersão na unidade gráfica, cuja perturbação é inofensiva para o ordenamento e basta para deslocar as janelas situadas próximas ao limiar. Uma métrica que não se reproduz entre duas execuções idênticas do mesmo treinamento, no mesmo computador, não deveria ser a métrica de decisão de um detector.

ISSN 2317-1316

Os resultados valem para seis famílias de um único conjunto de dados, de modo que família inédita significa inédita dentro do RanSMAP, e as três sementes bastam para estabelecer os efeitos relatados, não para afirmações precisas sobre variância.

4 CONCLUSÃO

A representação em grafos de regiões de armazenamento processada por GraphSAGE supera amplamente a regra de limiar sobre a fração de escritas-chave, sobretudo na família REvil, na qual esse marcador praticamente não existe, e reduz o alarme falso em cerca de uma ordem de magnitude em relação à ablação sem estrutura; sua degradação diante de família inédita revelou-se, contudo, de calibração e não de discriminação, pois a AUC por execução permaneceu em 1,000 enquanto a sensibilidade no limiar convencional caía a 0,3167, e substituir esse limiar por decisão relativa às execuções benignas do próprio ambiente restaurou sensibilidade integral em todas as partições, sem retreinamento e sem exigir amostras da família a detectar, de modo que o compromisso entre representação e generalização medido ao longo de dezoito treinamentos mostrou-se em larga medida artefato da régua empregada, restando como limite efetivo a execução simultânea de *ransomware* de assinatura fraca com aplicação benigna de perfil de acesso semelhante.

5 DECLARAÇÃO DE USO DE IA

Utilizou-se a ferramenta Claude (Anthropic), versão Opus 5, nas seguintes etapas: implementação dos programas de pré-processamento, construção dos grafos, treinamento e avaliação; análise exploratória do conjunto de dados, conferência dos resultados numéricos e verificação das referências citadas; e apoio à redação e à revisão do texto. A concepção da pesquisa, as decisões metodológicas e a interpretação dos resultados são do autor, que revisou e validou todo o conteúdo e assume integral responsabilidade pelo texto final.

6 REFERÊNCIAS

Begovic K, Al-Ali A, Malluhi Q. Cryptographic ransomware encryption detection: Survey. Computers & Security 2023;132:103349.

Chen Y-H, Chen J-L, Deng R-F. Similarity-Based Malware Classification Using Graph Neural Networks. Applied Sciences 2022;12(21):10837.

ISSN 2317-1316

Fey M, Lenssen JE. Fast Graph Representation Learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds; 2019.

Hamilton WL, Ying R, Leskovec J. Inductive Representation Learning on Large Graphs. In: Advances in Neural Information Processing Systems 31; 2017. p. 1025–1035.

Hirano M, Kobayashi R. Machine Learning-based Ransomware Detection Using Low-level Memory Access Patterns Obtained From Live-forensic Hypervisor. In: IEEE International Conference on Cyber Security and Resilience; 2022. p. 323–330.

Hirano M, Kobayashi R. RanSMAP: Open dataset of Ransomware Storage and Memory Access Patterns for creating deep learning based ransomware detectors. Computers & Security 2024;150:104202.

Shannon CE. A Mathematical Theory of Communication. The Bell System Technical Journal 1948;27(3):379–423.

Youden WJ. Index for rating diagnostic tests. Cancer 1950;3(1):32–35.