Malware Classification from Binary Images Using Machine Learning: A Review

Gustavo S. B. Lima[1,2,3], **João Gondim**[1]

[1]Universidade de Brasília (UNB)

[2]Mestre em Engenharia Aeronáutica (ITA)

[3]Ministério da Defesa (CENSIPAM)

gustavoblima@bsd.com.br, joao.gondim@gmail.com

Abstract. *The proliferation of malware has reached alarming levels, demanding detection techniques that go beyond traditional signature-based antivirus systems. One of the most promising approaches is the conversion of malicious binaries into grayscale or alternative visual representations, enabling the use of computer vision and deep learning for malware detection and family classification. This paper presents a systematic literature review of 69 studies published between 2019 and 2026, analyzing a wide range of models — from convolutional neural networks (CNNs) and Vision Transformers to self-supervised, few-shot, and generative approaches — applied to diverse benchmark datasets.*
The main findings are: (1) the highest reported accuracy on the Malimg benchmark is 99.97%, achieved by a hybrid ensemble of transfer learning models [1]; (2) lightweight architectures such as MalShuffleNet (99.03%, 2.5M parameters) and γ-M2I with MobileNetV2 (99.82%, 1.25 ms inference) demonstrate that compact models approach top performance; (3) standard grayscale classifiers are vulnerable to obfuscation, while Markov-image and multi-image strategies significantly improve robustness; and (4) cross-dataset evaluations reveal accuracy drops of up to 19 percentage points, evidencing the limitation of single-benchmark evaluations.

Resumo. *A proliferação de malwares exige técnicas de detecção que vão além dos sistemas antivírus tradicionais baseados em assinaturas. Uma das abordagens mais promissoras é a conversão de binários maliciosos em representações visuais — como imagens em tons de cinza ou matrizes de Markov — que permitem aplicar visão computacional e aprendizado profundo à detecção e classificação de malware por família. Este artigo apresenta uma revisão sistemática da literatura de 69 estudos publicados entre 2019 e 2026, analisando modelos que vão de CNNs e Vision Transformers a abordagens autossupervisionadas, de few-shot e generativas.*
Os principais resultados são: (1) a maior acurácia reportada no benchmark Malimg é de 99,97%, alcançada por um ensemble híbrido de modelos de transfer learning [1]; (2) arquiteturas leves como MalShuffleNet (99,03%, 2,5M parâmetros) e γ-M2I com MobileNetV2 (99,82%, 1,25 ms de inferência) demonstram que modelos compactos se aproximam do desempenho máximo; (3)

classificadores padrão em grayscale são vulneráveis à ofuscação, enquanto estratégias baseadas em Markov e multi-imagem melhoram significativamente a robustez; e (4) avaliações cross-dataset revelam quedas de até 19 pontos percentuais na acurácia, evidenciando a limitação de avaliações em dataset único.

1. Introdução

Malware é um software desenvolvido para danificar sistemas, roubar dados ou espionar seus alvos. Constitui um dos problemas mais antigos e persistentes da segurança computacional. Sistemas tradicionais de detecção, que comparam arquivos com bases de padrões de ameaças conhecidas, tornam-se insuficientes diante de novas variantes obtidas por recompilação, compressão ou criptografia do código original.

Para superar essa limitação, pesquisadores passaram a tratar o binário de malware como imagem: os bytes são organizados em uma grade e mapeados em tonalidades de cinza, produzindo uma representação visual cujos padrões de textura são caracteristicamente semelhantes entre amostras de uma mesma família — mesmo após modificações superficiais. Essa observação, formalizada inicialmente por Nataraj et al. [2], deu origem à linha de pesquisa de *classificação de malware como imagem*, que possibilita aplicar diretamente toda a potência da visão computacional moderna ao problema de detecção.

Esta revisão sistemática de 69 estudos (período de busca: abril de 2026) está organizada em torno de quatro Questões de Pesquisa (RQ):

RQ1 — Arquiteturas: Quais modelos de aprendizado de máquina têm sido utilizados com maior frequência e quais apresentam os melhores resultados quantitativos?

RQ2 — Datasets: Quais conjuntos de dados são mais comumente empregados, quais são suas características e limitações?

RQ3 — Representações: Como um arquivo binário é convertido em imagem e em que medida o método de conversão influencia os resultados?

RQ4 — Lacunas: Quais problemas em aberto ainda precisam ser enfrentados por pesquisas futuras e quais são as limitações da própria revisão?

O artigo está organizado como segue: Seção 2 apresenta a fundamentação teórica; Seção 3 descreve o protocolo PRISMA (RQ1–RQ4 são respondidas nas Seções 4–7); Seção 9 discute as limitações da revisão; e Seção 11 apresenta as conclusões quantitativas consolidadas.

1.1. Motivação e objetivos

O malware afeta múltiplos ambientes: Windows PE, Android, IoT, memória de processos e plataformas de criptomineração. Nesse contexto, a detecção baseada em imagens é atraente porque: (a) dispensa engenharia de características manual; (b) é compatível com pipelines de visão computacional de alta performance; e (c) é relativamente robusta a modificações superficiais que alterariam a assinatura binária mas preservariam a textura visual. O objetivo central desta revisão é sintetizar o estado da arte, identificar tendências metodológicas e apontar as lacunas que comprometem a transferência dos resultados de laboratório para implantações reais.

2. Fundamentação Teórica

2.1. De arquivos binários para imagens

Todo programa armazenado em um computador é, em última instância, uma sequência de bytes — números de 0 a 255. A ideia central desta linha de pesquisa é tratar esses bytes como valores de pixels: organizá-los linha por linha em um retângulo e atribuir a cada pixel uma tonalidade do preto (valor 0) ao branco (valor 255). Programas pertencentes à mesma família de malware tendem a compartilhar grandes blocos de código semelhante, que aparecem como regiões de textura visualmente consistentes. Os pesquisadores têm expandido essa ideia básica em diferentes direções:

- **Imagens coloridas (RGB):** o conteúdo binário é codificado em três canais ou categorizado semanticamente [3, 4].
- **Imagens de Markov:** registram com que frequência um valor de byte é seguido por outro, produzindo uma matriz 256×256 que captura padrões estatísticos de coocorrência [5, 6, 7].
- **Imagens de entropia:** dividem o binário em blocos e medem a aleatoriedade de cada bloco, destacando regiões empacotadas ou criptografadas [5, 8].
- **Domínio da frequência (DCT):** estatísticas de coocorrência de pares de bytes são transformadas, capturando periodicidades estruturais do código [9].
- **Curvas de preenchimento de espaço (SFC):** bytes são mapeados em coordenadas espaciais, preservando melhor a localidade do que a organização linha por linha [10].

A Figura 1 ilustra o pipeline genérico de detecção baseada em imagem.

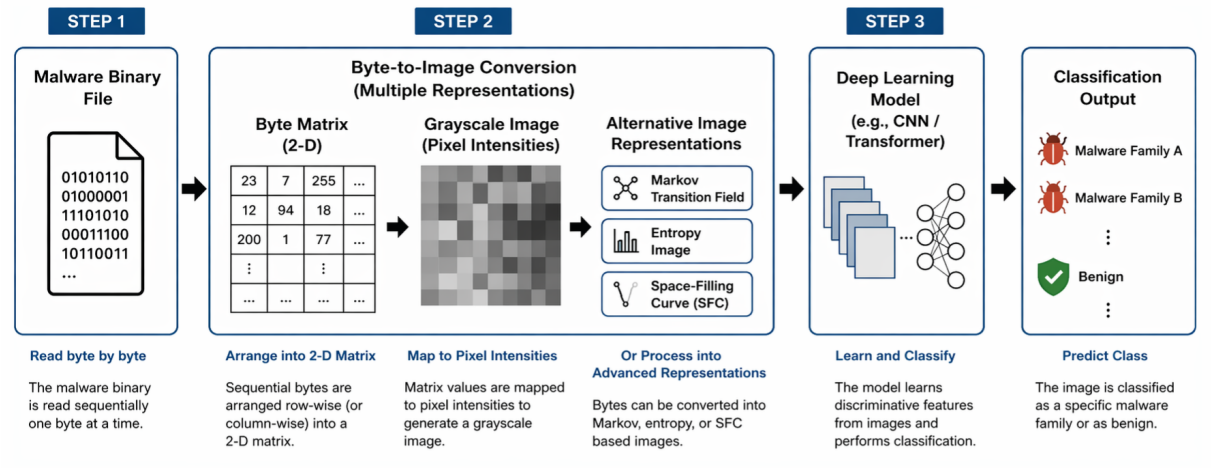

Figura 1. Pipeline genérico para detecção de malware baseada em imagens. Etapa 1: leitura byte a byte do binário. Etapa 2: organização em matriz 2D e mapeamento em intensidade de pixel. Etapa 3: classificação por modelo de aprendizado profundo.

A Figura 2 apresenta uma taxonomia geral da área, organizando as representações visuais, as famílias de arquiteturas e as aplicações discutidas nesta revisão.

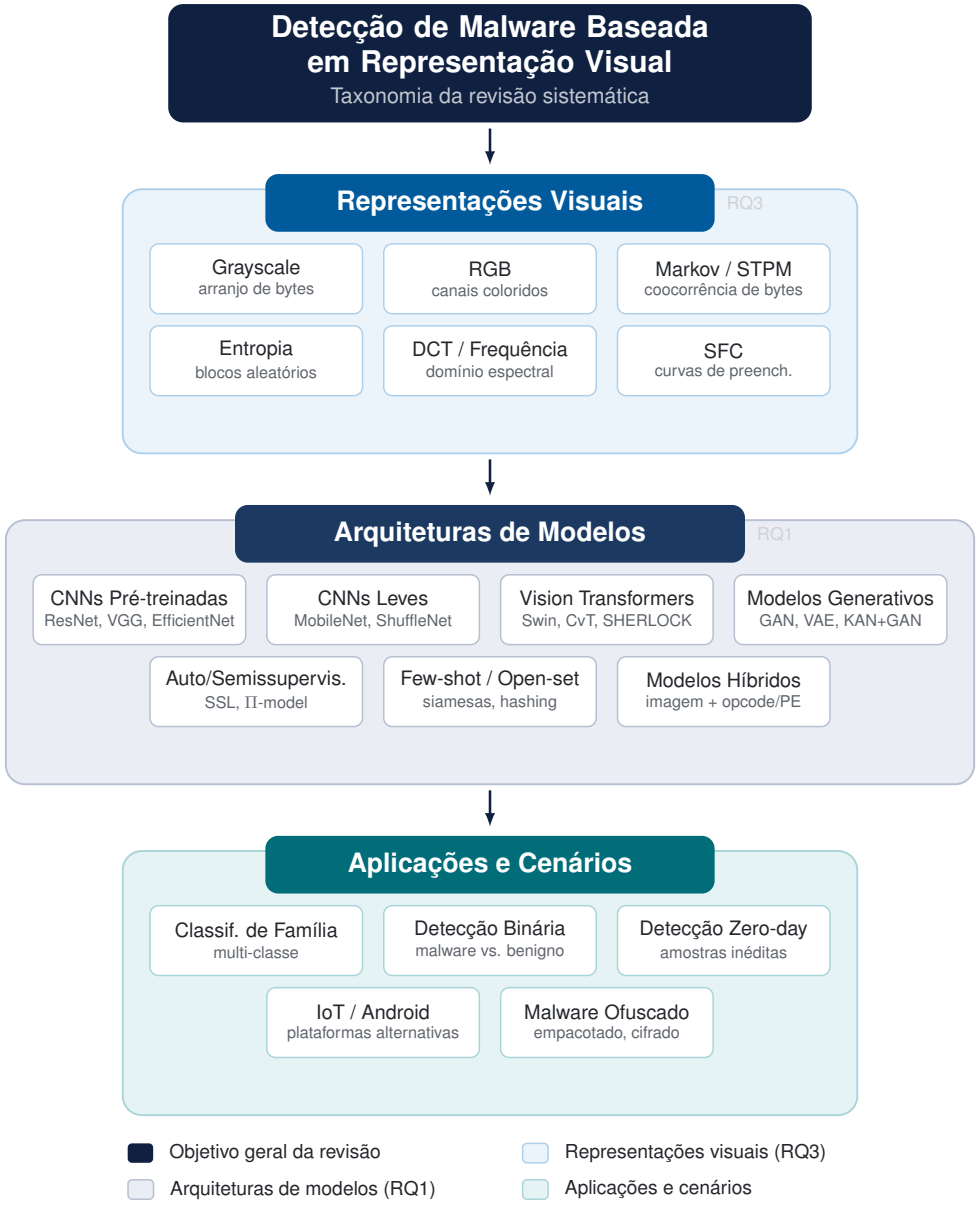

Figura 2. Taxonomia geral da detecção de malware baseada em imagem: representações visuais (grayscale, Markov, entropia, SFC, RGB), famílias de arquiteturas (CNNs pré-treinadas, CNNs leves, Vision Transformers, modelos generativos, few-shot) e aplicações (classificação de família, detecção binária, detecção zero-day, detecção em IoT/Android).

2.2. Principais famílias de modelos

CNNs pré-treinadas (VGG16, VGG19, ResNet, Inception, DenseNet, EfficientNet) aplicam transfer learning a partir do ImageNet. São responsáveis por 24–20% dos artigos revisados (ResNet: 42 artigos; VGG: 35 artigos). O custo computacional varia amplamente: ResNet-50 tem 25,6M parâmetros; EfficientNet-B0 tem 5,3M [1, 11].

CNNs leves (MobileNet, ShuffleNet, EfficientNet-B0) sacrificam parte da acurácia em favor de menor uso de memória e latência. MalShuffleNet tem apenas 2.5M parâmetros; γ-M2I com MobileNetV2 atinge 1.25 ms de inferência por amostra — valores

relevantes para implantação em dispositivos de borda [12, 13, 7].

Vision Transformers (ViT) dividem a imagem em *patches* e aplicam autoatenção. Demandam mais dados e maior custo de treinamento que CNNs de porte equivalente; estão presentes em 7 artigos do corpus [14, 15, 16].

Modelos autossupervisionados aprendem sem rótulos via tarefas auxiliares, reduzindo o custo de anotação. MalSSL usa ResNet-18 com objetivo contrastivo similar ao SimCLR [17].

Modelos de few-shot (redes siamesas, prototípicas) permitem classificar novas famílias a partir de poucos exemplos. Importantes para cenários operacionais onde novas variantes surgem continuamente [18, 19, 20].

Modelos generativos (GAN, VAE) sintetizam amostras para balanceamento de classes e simulação de *zero-day* [21, 22, 23].

Modelos híbridos combinam extração visual com outros modalidades (opcodes, cabeçalhos PE, tráfego de rede) [19, 24].

3. Metodologia da Revisão

Esta revisão seguiu as diretrizes PRISMA 2020 [25]. A Figura 3 apresenta o fluxograma de seleção.

3.1. Estratégia de busca

As buscas foram realizadas em **abril de 2026** nas bases IEEE Xplore, ACM Digital Library, Scopus e arXiv, cobrindo publicações de **2019 a 2026**. As expressões de busca combinaram termos como: *"Malware binary image detection"*, *"Malware binary image-based using machine learning"*, *"Malware binary image and CNN"*, *"Malware visualization"*, *"Malware binary image and deep learning"*, *"Malware binary image and transfer learning"*, *"few-shot and malware"*, *"self-supervised malware"*, *"lightweight malware"*, *"Markov malware image"*, *"zero-day malware detection"* e *"obfuscated malware classification"*.

3.2. Processo de seleção dos estudos

A seleção ocorreu em três etapas:

Etapa 1 — Remoção de duplicidades: 180 registros das bases de dados e 61 de outras fontes (busca manual, citações cruzadas) totalizaram 241 registros; 67 duplicatas foram removidas, restando 174 registros únicos.

Etapa 2 — Triagem por título e resumo: dos 174 registros, foram descartados aqueles sem relação com malware ou sem uso de representação visual, restando 166 para recuperação de texto completo.

Etapa 3 — Avaliação de elegibilidade: os 166 textos completos foram avaliados; 97 foram excluídos pelos critérios detalhados na Seção 3.3. **Total incluído na síntese: 69 artigos.**

3.3. Critérios de elegibilidade

Critérios de inclusão (todos devem ser atendidos):

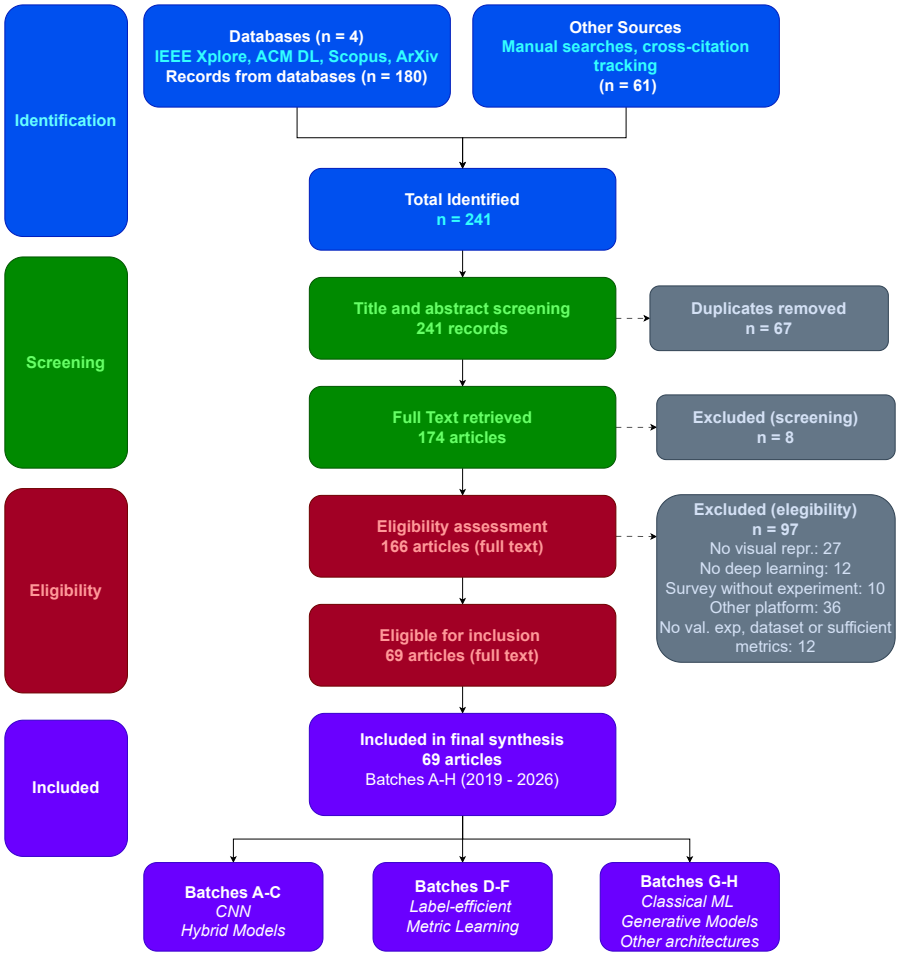

Figura 3. Fluxograma PRISMA 2020. Busca realizada em Abril de 2026. Total de 174 registros triados; 69 artigos incluídos na síntese final.

1. Objetivo principal: detecção ou classificação de famílias de malware.
2. Uso de representação visual 2D derivada do binário (grayscale, RGB, Markov/STPM, entropia, DCT, SFC ou equivalente).
3. Aplicação de classificador de ML ou DL (CNN, Transformer, GAN, modelo semissupervisionado ou híbrido).
4. Relato de ao menos uma métrica quantitativa (acurácia, F1-score, AUC) ou contribuição metodológica substantiva.

Critérios de exclusão:

- Artigos de revisão sem experimento original próprio ($n = 10$).
- Detecção sem representação binária visual (grafos de API, sequências de opcode puras, tráfego de rede) ($n = 27$).
- Classificação de *arquivos maliciosos* (PDFs, JPEGs) em vez de imagens *de binários de malware* ($n = 36$).
- Plataforma ou domínio fora do escopo (canal lateral de hardware, similaridade de firmware) ($n = 12$).
- Ausência de modelo de ML/DL ($n = 12$).

3.4. Avaliação de qualidade e risco de viés

Cada artigo foi avaliado em três dimensões: (1) *validade do dataset* — disponibilidade pública e descrição suficiente; (2) *rigor da avaliação* — uso de divisão treino/teste ou validação cruzada com múltiplas métricas; e (3) *reprodutibilidade* — disponibilidade de código-fonte e hiperparâmetros. Classificação: *alta*, *média* ou *baixa*. A maioria recebeu classificação *média* por omissão de hiperparâmetros ou uso de datasets privados. Apenas [18] realizou validação explícita de reprodutibilidade (r=0.97 com o baseline original).

3.5. Extração de dados

Para cada artigo foram extraídos: autor(es), ano, local de publicação, estratégia de representação, arquitetura de ML, dataset(s), acurácia, F1-score e métricas adicionais. Campos não informados no artigo foram registrados com hífen (–). A extração foi realizada em oito lotes sequenciais (A–H).

3.6. Distribuição temporal

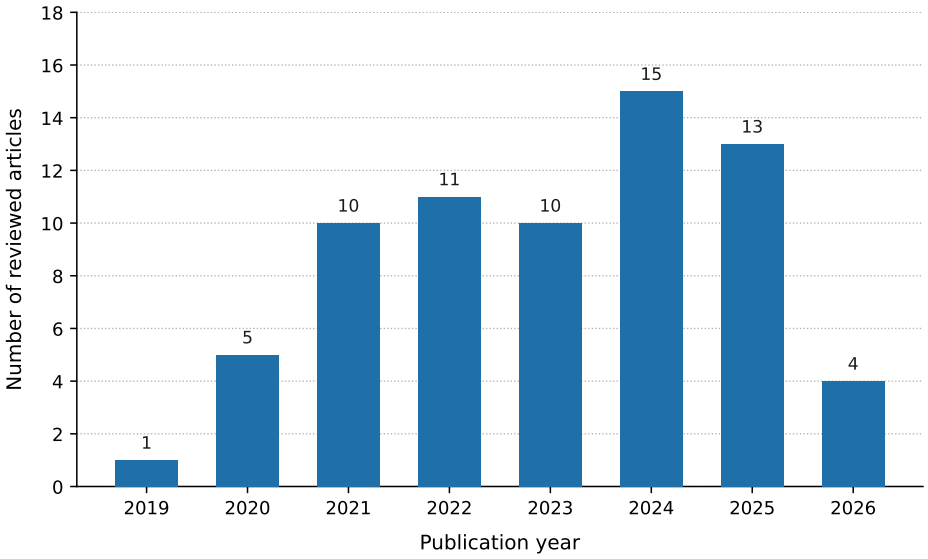

Figura 4. Número de artigos incluídos por ano (2019–2026).

A Figura 4 mostra a distribuição anual. A concentração em 2022–2026 reflete a maior disponibilidade de datasets públicos e de modelos pré-treinados.

4. Datasets — RQ2

A Tabela 1 apresenta os principais datasets identificados nos artigos estudados durante o processo de desenvolvimento da revisão sistemática.

4.1. Predominância do Malimg e suas implicações

O Malimg aparece em mais de 60% dos artigos revisados. Foi construído a partir de amostras coletadas antes de 2012, o que o torna potencialmente desatualizado em relação ao malware contemporâneo. O artigo [26] quantifica o problema diretamente: o

Tabela 1. Principais Datasets Utilizados nos Artigos Revisados

Dataset	Amostras	Fam.	Formato
Malimg	9,339	25	PE/grayscale
MS BIG 2015/MMCC	10,868	9	PE/bytes
MaleVis	14,226	26	PE/RGB
MalNet-Image	1.26M	696	DEX/grayscale
Dumpware10	4,294	11	Memory/RGB
BODMAS	—	—	PE/grayscale
DikeDataset	—	—	PE/grayscale
CICMalDroid 2020	—	—	APK
Maldeb	—	—	Binary
Virus-MNIST	—	—	Binary
Kaspersky (internal)	7,162	—	Binary/RGB
VirusShare	—	—	Various
vx underground	—	20+	PE/grayscale
REWEMA	—	—	Various
Own (60k PE)	60,000	2	PE/grayscale

mesmo modelo ResNet50 alcança 98,21% no BODMAS, cai para 93,22% no DikeDataset e reduz para 79,58% no theZoo — diferença de 18,6 pontos percentuais. Variação similar (18,96 p.p.) é reportada em [27]. Esses resultados evidenciam que acurácia em dataset único não é indicador confiável de desempenho em produção.

4.2. MalNet-Image: um passo em direção à escala

O MalNet-Image [28] contém mais de 1 milhão de imagens em 696 famílias. O Vision Transformer autossupervisionado SHERLOCK [16], avaliado nesse dataset, reporta 97% de acurácia binária e F1 de 0,491 no nível de famílias — resultado mais realista sobre a dificuldade efetiva de classificação em grande escala.

Síntese — RQ2: O corpus é dominado pelo Malimg, cuja escala reduzida e idade comprometem a validade externa dos resultados. Avaliações em múltiplos datasets são realizadas por apenas 8 dos 69 artigos, constituindo a principal lacuna metodológica do campo.

5. Arquiteturas e Modelos — RQ1

Esta seção responde à RQ1 ao descrever as principais famílias de modelos identificadas na revisão, organizadas de acordo com o paradigma de aprendizado que representam. Sempre que os estudos reportam dados de eficiência ou implantação, a análise também contribui para a RQ4. Na Figura 5 é possível visualizar a frequência das principais familias de arquiteturas encontradas nos artigos revisados.

A Tabela 2 resume a distribuição de arquiteturas.

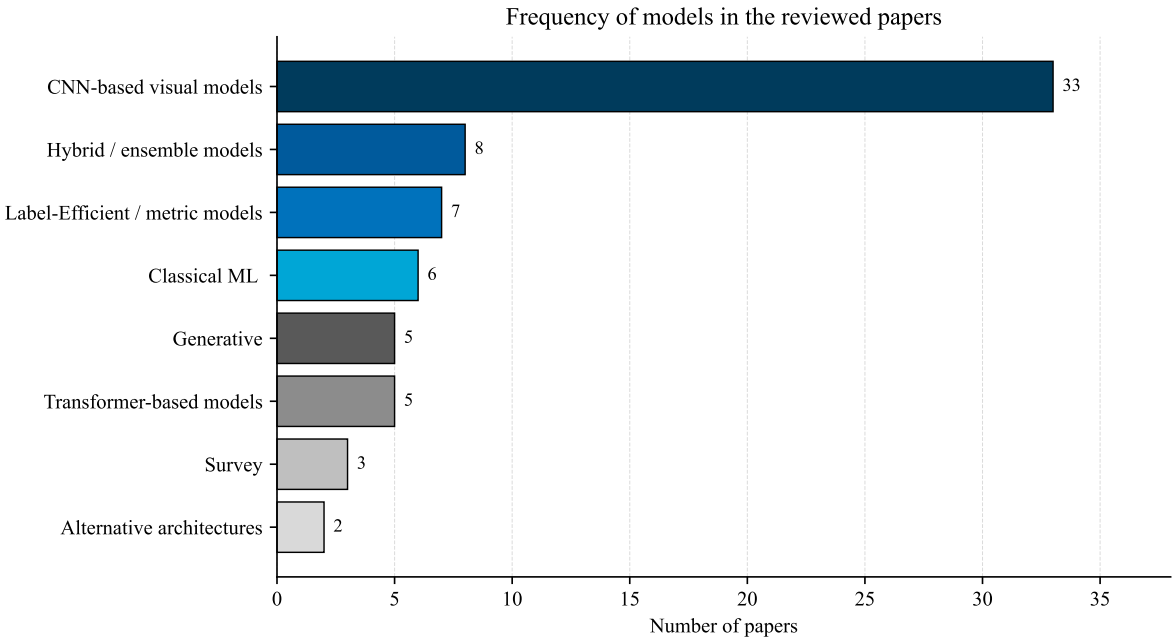

Figura 5. Frequência das principais famílias de arquitetura nos 69 artigos revisados.

Tabela 2: Frequência das Famílias de Modelos nos 69 Artigos Revisados

Família	**Arts.**	**%**	**Trabalhos representativos**
Transfer learning e CNN			
ResNet (variantes)	42	61%	[11, 26, 1, 7]
VGG-16 / VGG-19	35	51%	[29, 27, 30, 31]
EfficientNet (variantes)	22	32%	[32, 33, 1]
Inception V3 / V4	18	26%	[34, 35]
DenseNet	12	17%	[30, 28]
MobileNet (variantes)	18	26%	[36, 12, 7]
AlexNet / LeNet	8	12%	[34, 11]
CNNs leves / orientadas à borda			
MalShuffleNet / ShuffleNet V2	3	4%	[12]
MalEdgeNeXt / EdgeNeXt	2	3%	[13]
CNNs compactas customizadas	28	41%	[37, 38]
Vision Transformers			
Swin Transformer	3	4%	[14]

Continua na próxima página

Família	Arts.	%	Trabalhos representativos
CvT / ViT / CCT / EA-Net	7	10%	[15, 39]
SHERLOCK / ViT autossupervisionado	1	1%	[16]
Modelos generativos			
GAN / DCGAN / WGAN	18	26%	[21, 22, 40]
VAE	3	4%	[23]
KAN + GAN	1	1%	[41]
Aprendizado autossupervisionado / semissupervisionado			
Contrastivo / autossupervisionado	3	4%	[17, 16]
Semissupervisionado / Π-model	4	6%	[42, 43]
Few-shot / aprendizado métrico			
Redes siamesas / prototípicas	5	7%	[18, 20, 19]
ML clássico sobre características visuais			
SVM / RF / XGBoost (sobre CNN)	22	32%	[27, 29]
Outros / fronteira			
Quantum CNN	1	1%	[44]
Multimodal / híbrido	12	17%	[19, 45, 24]

Nota: percentuais excedem 100% porque alguns artigos usam múltiplas arquiteturas.

5.1. CNNs pré-treinadas — análise crítica

ResNet domina o corpus (42 artigos, 61%), justificado pelo equilíbrio entre profundidade, eficiência de gradiente (conexões residuais) e disponibilidade de pesos pré-treinados no ImageNet. O melhor resultado individual é 99,78% no Malimg [11]. No entanto, ResNet-50 exige 25,6M parâmetros e tempo de inferência tipicamente acima de 10 ms em CPU — inadequado para endpoints de tempo real.

A VGG16 é o *backbone* mais versátil: aparece como classificador independente, extrator de características para ML clássico, componente de ensembles e codificador de imagens no MalwareCLIP [46]. Sua popularidade se deve à arquitetura simples e bem documentada, mas seu custo (138M parâmetros) é proibitivo em dispositivos de borda.

EfficientNet-B0 (5,3M parâmetros) oferece um compromisso melhor: 99,70% no Malimg com o ensemble EffiCBNet [32] e latência substancialmente menor que ResNet-50.

5.2. CNNs leves — análise crítica

MalShuffleNet [12] (2,5M parâmetros, 99,03% no Malimg) e MalEdgeNeXt [13] (98,81%) demonstram que modelos compactos aproximam-se do teto de performance com fração dos parâmetros. O γ-M2I com MobileNetV2 [7] (2,24M parâmetros, 1,62 ms de treinamento/amostra, 1,25 ms de inferência) é o único artigo do corpus a reportar todos os três indicadores (acurácia, parâmetros e tempo) de forma sistemática — tornando-se a referência mais completa para implantação em borda.

Razão para o desempenho: a eficiência das CNNs leves em imagens de malware se deve ao fato de que os padrões discriminativos dessas imagens são predominantemente locais (texturas em regiões de bytes semelhantes), o que favorece filtros de pequena receptividade — exatamente o que convoluções separáveis em profundidade capturam a baixo custo.

5.3. Vision Transformers — análise crítica

ViTs dividem a imagem em *patches* e aprendem relações globais por autoatenção. A comparação sistemática em [39] mostra que uma CNN proposta (99,44%) supera ViT, CCT e EANet no Malimg. *A razão* é a escala: ViTs precisam de grandes volumes de dados para superar os vieses indutivos locais das CNNs; com apenas 9.339 amostras (Malimg), esse volume não é atingido. No MalNet-Image (1,26M amostras), o SHERLOCK [16] (ViT autossupervisionado) atinge 97% binário — confirmando que a vantagem dos Transformers emerge em maior escala.

5.4. Modelos autossupervisionados e semissupervisionados

MalSSL [17] (ResNet-18, contrastivo) atinge 98,4% no Malimg com fração dos rótulos necessários no treinamento supervisionado completo. O modelo Π aprimorado [42] demonstra ganho de 6% ao usar apenas 20% de dados rotulados. *A razão:* rotular amostras de malware exige análise reversa especializada; métodos que reduzem essa dependência têm impacto operacional direto.

5.5. Few-shot e open-set learning

[18] (CNN siamesa, 56 famílias, 89,2% em 2-way) e [19] (imagem + opcodes, 92,95% em 5-way 10-shot) estabelecem baselines para cenários onde novas famílias de malware emergem continuamente. Para detecção em conjunto aberto, o RNDPN [47] (hashing profundo) atinge 96,5% em famílias conhecidas e 85,7% em amostras desconhecidas do VX-Heaven.

5.6. Modelos generativos e aumento de dados

PlausMal-GAN [21] (98,65% em zero-day análogo) e KOL-4-GEN [41] (KAN+GAN, 99,36% no Malimg, 95,44% no MaleVis, 92,12% no Virus-MNIST — único artigo avaliado em três benchmarks) representam o estado da arte em geração sintética. VAE+CNN [23] demonstra salto de 90% para 98% no Malimg após balanceamento baseado em VAE.

5.7. Interpretabilidade e robustez

Grad-CAM [48, 30], LIME [39] e HiResCAM/SHAP [31] são os métodos de interpretabilidade mais utilizados. O estudo mais abrangente é [31]: acurácia em malware empacotado cai para 35,7% no modelo baseline, recuperando-se para apenas 51,1% com aprimoramento. A robustez adversarial é avaliada em [49] (ataque PGD, 96,93% no BIG-2015).

Síntese — RQ1: Transfer learning com ResNet domina o corpus (61% dos artigos). CNNs leves com 2–5M parâmetros atingem acurácia comparável com latência inferior a 2 ms, viabilizando implantação em borda. Vision Transformers são competitivos apenas em datasets de grande escala ($>$100k amostras). Métricas de custo computacional (parâmetros, tempo de inferência, custo de treinamento) são raramente reportadas de forma sistemática — lacuna que compromete a comparabilidade dos resultados.

6. Representações Visuais — RQ3

Grayscale é dominante, mas nem sempre é a melhor opção. Entre artigos que comparam grayscale com alternativas: RGB melhora 0,56 p.p. no Malimg [50]; imagens de Markov superam grayscale na classificação no REWEMA [6]; a transformação γ aplicada às imagens de Markov eleva a acurácia tanto no Malimg quanto no BIG-2015 [7].

A fusão de múltiplas imagens enfrenta o problema do malware ofuscado. Combinando quatro tipos (valor de byte, entropia, semântica e frequência de bigramas) com EfficientNet-B0 e Random Forest, [51] atinge 88,66% em amostras PE empacotadas, onde abordagens puramente grayscale falham. [5] reporta 99,41% em malware Android ofuscado combinando Markov + entropia + características estáticas.

O redimensionamento progressivo preserva a estrutura espacial. O P-DCNN [52] agrupa arquivos por tamanho e aplica redimensionamento para o quadrado mais próximo dentro de cada grupo, reportando 98,6% no MalImg e 98,1% no BIG-2015.

Síntese — RQ3: Grayscale é a estratégia padrão, presente em mais de 60% dos artigos. Representações alternativas (Markov, entropia, SFC, RGB) oferecem ganhos documentados, especialmente em cenários com malware ofuscado ou empacotado. A escolha da representação é frequentemente tratada como decisão de design sem avaliação comparativa sistemática — lacuna relevante para pesquisas futuras.

7. Resultados Experimentais — RQ1/RQ2/RQ3

Estatísticas descritivas — Malimg: dentre os artigos que reportam acurácia no Malimg, o valor mínimo é 87,0%, o máximo é 99,97%, a mediana é aproximadamente 98,5%, e a média ponderada pelo número de modelos avaliados é $\approx$98,1%. A distribuição na Figura 6 é fortemente assimétrica à direita, com mais de 75% dos resultados acima de 97% — evidência de saturação do benchmark.

Estatísticas descritivas — BIG-2015: resultados no Microsoft BIG-2015 variam de 93,07% a 99,68%, com mediana $\approx$98,2% — ligeiramente inferior ao Malimg, refletindo a maior diversidade estrutural das 9 famílias.

Acurácia média por família de arquitetura (Malimg): ResNet: $\approx$99,1%; EfficientNet: $\approx$98,9%; VGG: $\approx$98,3%; Vision Transformers: $\approx$97,5%; abordagens semissupervisionadas: $\approx$96,0%; few-shot: $\approx$90,6%.

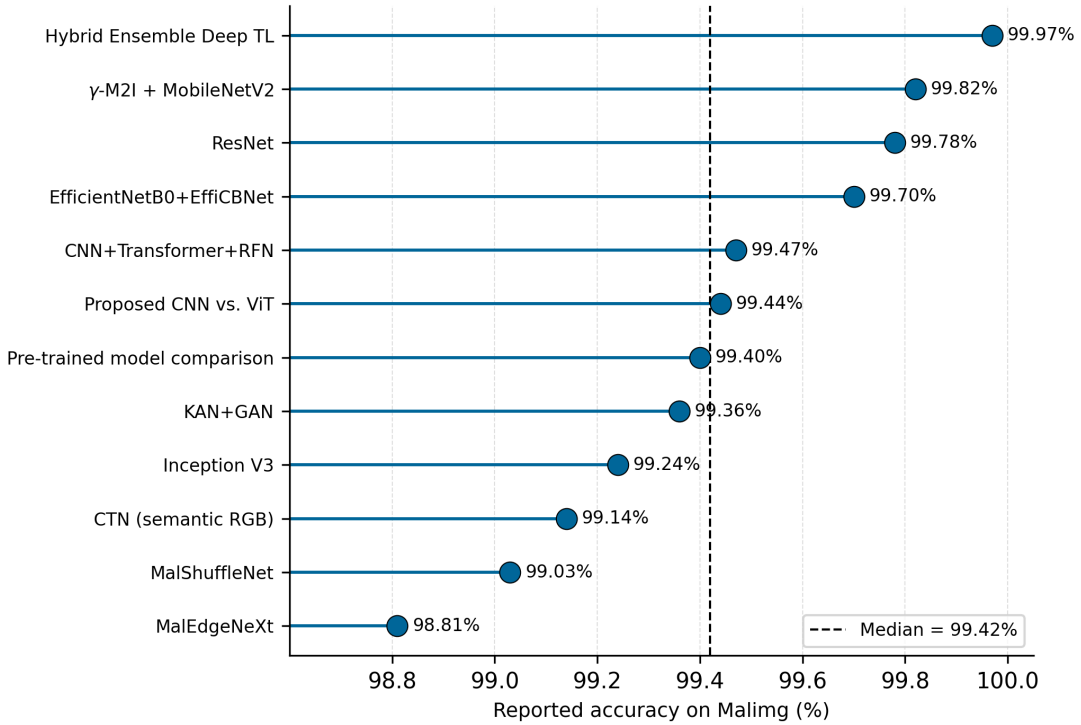

Figura 6. Distribuição dos valores de acurácia reportados no Malimg (histograma, intervalos de 5 p.p.).

A Tabela 3 apresenta uma síntese comparativa dos principais trabalhos, incluindo representação, arquitetura, dataset, robustez e interpretabilidade.

Tabela 3: Síntese comparativa de trabalhos representativos da revisão.

Trabalho	Representação	Arquitetura	Dataset	Cenário	Robustez	Interpretabilidade
[1]	Imagem de malware	Ensemble de transfer learning + autoencoder	MaleVis; Malimg	Classificação benchmark	Detecção de outliers	Não identificado
[7]	Markov image + transformação γ	MobileNetV2 e outros backbones CNN	Malimg; BIG-2015	Eficiência e comparação multi-dataset	Avaliação em dois datasets	Não identificado
[12]	Grayscale	MalShuffleNet / ShuffleNet V2	Malimg	Modelo leve	Não identificado	Não identificado
[13]	Grayscale	MalEdgeNeXt	Malimg	Modelo leve	Não identificado	Não identificado
[31]	Imagem de malware com aprimoramento	CNN inspirada em VGG16	Malimg; BIG-2015; VXZoo	Malware empacotado	Avalia queda e recuperação em amostras empacotadas	HiResCAM, SHAP e mapas de oclusão
[51]	Byte, entropia, semântica e bigramas	EfficientNet-B0 + Random Forest	PE ofuscado/empacotado	Robustez à ofuscação	Avaliação específica em PE empacotado	Não identificado
[17]	Imagem de malware	ResNet-18 com aprendizado contrastivo	Malimg; Maldeb	Rotulação limitada	Não identificado	Não identificado
[16]	Grayscale em larga escala	Vision Transformer autossupervisionado	MalNet-Image	Escala e classificação por família	Avaliação em dataset grande	Não identificado
[18]	Imagem de malware	CNN siamesa / one-shot	VirusShare; VirusTotal	Few-shot e reprodutibilidade	Reexecução de baseline, r=0.97	Não identificado

Continua na próxima página

Trabalho	Representação	Arquitetura	Dataset	Cenário	Robustez	Interpretabilidade
[46]	Imagem + texto	Modelo visão-linguagem contrastivo	vx-underground	Similaridade e explicabilidade	Não identificado	Rótulos em linguagem natural

A Tabela 4 apresenta os resultados de todos os 69 artigos incluídos.

Tabela 4: Resultados de desempenho nos 69 artigos analisados.

Nº	Método / algoritmo	Dataset	Acurácia	Precision	Recall	F1
1	Markov image + γ mapping [7]	Malimg; BIG-2015	99,82% / 99,46%	–	–	–
2	Convolutional Transformation Network [4]	–	99,14%	–	–	–
3	CNN models for malware images [34]	Malimg	99,24%	–	–	–
4	Frequency-domain visualization [9]	–	96,15%	–	–	–
5	Large-scale classification [28]	MalNet	–	–	–	0,86 / 0,49 / 0,45
6	Few-shot Siamese [20]	–	–	85,3%	88,7%	86,2%
7	Self-supervised ViT [16]	MalNet	97,0% / 83,7% / 80,2%	–	–	0,491
8	Robustness / XAI CNN [31]	MalImg; BIG-2015	98% / 95%	39,5–63,5%	–	35,7–51,1%
9	Comparative CNN architectures [11]	Malimg	99,78%	99,8%	99,65%	99,72%
10	Encoded byte-image corpus [26]	BODMAS / Dike / Zoo	98,21% / 93,22% / 79,58%	89,42%	93,00%	91,18%
11	Visualization + feature fusion [53]	Malimg; BIG	99,47% / 99,02%	99,27% / 98,06%	99,47% / 98,89%	99,36% / 99,19%
12	Swin Transformer + Mona [14]	MaleVis; BIG; Fusion	98,52% / 98,35% / 98,01%	–	–	–
13	Lightweight CNN [37]	Malimg	98,86%	–	–	–
14	Memory image + attention [54]	Memory dataset	98,84% / 88,29%	–	–	–
15	EDR + image-based [55]	–	91,87%	98,78%	92,14%	95,35%
16	Weakly coupled SGAN [40]	–	98,94% / 98,01%	–	$>$93,75%	–
17	CNN vs Transformer [39]	Malimg	99,44%	99,44%	99,44%	99,44%
18	CNN + LSTM / BiLSTM [2]	–	97,4%	–	–	–
19	Co-training image-based [43]	Malimg; BIG	92,09%	–	92,09%	94,96%
20	ViT / CvT / EfficientNet [15]	MaleVis	95,934%	–	–	96,054%
21	TL on memory images [56]	Dumpware10	98%	–	–	–
22	Explainable deep ensemble [30]	Malimg + MaleVis	96%	97%	96%	96%
23	Deep hashing RNDPN [47]	Malimg; CIC; VX-Heaven	96,5% / 85,7%	96,01% / 86,8%	95,71% / 85,3%	–
24	RGB / grayscale streams [38]	VirusShare + Windows	97,35%	–	–	–
25	GA-designed CNN [57]	Malimg; MS Malware	98,50% / 93,07%	–	–	–
26	Hybrid CNN + GRU [58]	Malimg; BIG	98,34% / 95,53%	98% / 95,54%	98% / 95,54%	98% / 95,54%
27	Fine-tuned CNN [50]	Malimg	98,81%	98,84%	98,80%	97,85%
28	EfficientNet-B1 [33]	Malimg	98,57%	0,96	0,97	0,97

Nº	Método / algoritmo	Dataset	Acurácia	Precision	Recall	F1
29	EfficientNet + EffiCBNet [32]	–	98,93%	–	–	–
30	Few-shot image + opcode [19]	VirusShare; LargePE	92,95% / 91,34%	–	–	–
31	Enhanced Π-model SSL [42]	Malimg	96,35% / 95,97%	–	–	–
32	VAE augmentation + CNN [23]	Malimg	98%	–	–	–
33	CNN + Grad-CAM analysis [48]	AMD repository	98,43%	–	–	–
34	Voting + Siamese one-shot [18]	VirusShare / VT curated	89,2% / 70,8% / 53,2%	–	–	–
35	Butterfly ViT global-local [59]	Malimg; BIG; PE imports	99,49% / 99,99%	–	–	–
36	KAN + GAN [41]	Malimg; MaleVis; V-MNIST	99,88% / 96,10% / 92,12%	–	–	–
37	MalEdgeNeXt [13]	Malimg	98,81%	0,9883	0,9879	0,9885
38	JPEG structural + LightGBM [60]	JPEG dataset	–	–	TPR:0,951	–
39	MalSSL self-supervised [17]	Imagenette; Malimg; Maldeb	98,4% / 96,2%	–	–	–
40	2MFI + RepVGG [61]	BIG 2015	99,68%	99,68%	99,68%	99,67%
41	VisMal: CLAHE + CNN [62]	Malimg	96,0%	95,3%	96,0%	95,2%
42	MalwareCLIP [46]	vx-underground	93,77% / 91,76% / 88,93%	–	–	–
43	MalShuffleNet [12]	Malimg	99,03%	–	–	–
44	MobileNet + LSTM [36]	Malimg	99,26%	99,31%	99,29%	99,28%
45	VGG16 + ML classifiers [29]	Malimg	98,51%	–	–	–
46	SFC + VGG19 / SVM [10]	VX Heavens; Win PE	98,24% / 99,02%	–	–	–
47	Markov + VGG19 TL [6]	BIG; REWEMA	98,7% / 97,3%	–	–	–
48	Markov + entropy + GLMI [8]	CICMalDroid 2020	94,79%	94,81%	94,79%	94,79%
49	OMD audio + image + static [45]	Malimg; MaleX; Win	97% / 99,57% / 91,21%	–	–	–
50	Markov + CNN obfuscated Android [5]	Drebin; CIC	99,74%	–	–	–
51	Opcode BiLSTM vs image CNN [24]	BIG 2015; Malimg	99,17% / 98,74%	–	–	–
52	PH-CNN perceptual hash [63]	MS Malware	98,98%	98,27%	96,17%	97,21%
53	Pixel-level + classical ML [64]	MaleViz	92%	–	–	–
54	Obfuscation attack + defense [65]	VirusShare IoT ELF	87% / 100%	–	–	–
55	PlausMal-GAN [21]	MS Malware; Malimg	96,35% / 99,21% / 98,74%	–	–	–
56	ResNet50 / VGG16 / Xception [35]	Malimg	98,63%	–	–	–
57	Quantum CNN + TL [44]	Custom smart grid	97,78%	1,0	0,988	0,994
58	PE-section imaging [66]	VirusShare; Malicia	94%	–	–	–
59	GIST texture + KNN [67]	Malimg	97,27%	–	–	–
60	MobileNetV3 + MLP [49]	BIG 2015	96,93%	–	–	97%
61	P-DCNN progressive resizing [52]	Malimg; MMD	98,6% / 98,0%	–	–	98,1% / 97,5%
62	EfficientNet-B0 + RF fusion [51]	Obfuscated PE	88,66%	–	96,27%	91,22%
63	Survey DL malware [68]	Multiple	–	–	–	–
64	WGAN / Diffusion synthetic [69]	VirusShare; Malicia	–	–	–	–
65	VGG16 + SVM [27]	Malimg; real-world	99,25% / 80,29%	–	–	–

Nº	Método / algoritmo	Dataset	Acurácia	Precision	Recall	F1
66	Assembly-to-RGB + CNN [3]	BIG 2015	97,73%	–	–	–
67	HEDTL ensemble [1]	MaleVis; Malimg	99,94% / 99,97%	–	–	–
68	DL malware review [70]	Multiple	–	–	–	–
69	GAN unseen malware [22]	MaleVis / Hacettepe	99,20%	–	–	–

Síntese — RQ1/RQ2/RQ3: Os melhores resultados no Malimg são consistentemente obtidos por ensembles de transfer learning (99,97%) e arquiteturas ResNet (99,78%). A mediana de 98,5% com mais de 75% dos artigos acima de 97% indica saturação do benchmark. Avaliações cross-dataset revelam quedas de 18–19 p.p., confirmando que o desempenho no Malimg não prediz generalização.

8. Desafios e Problemas em Aberto — RQ4

8.1. Saturação do benchmark Malimg

Com mais de 35 arquiteturas avaliadas no Malimg e diferenças de acurácia dentro da margem de variação por inicialização aleatória, a comunidade não pode mais reivindicar "novo estado da arte" no Malimg sem avaliação mais ampla. *Razão:* o dataset tem apenas 9.339 amostras e 25 famílias de malware pré-2012 — insuficiente para discriminar arquiteturas modernas.

8.2. Ofuscação: a maior lacuna prática

[65] mostra que classificadores VGG16 padrão classificam incorretamente 100% do malware IoT ofuscado com oLLVM. O retreinamento com amostras ofuscadas restaura a acurácia para 100% na ofuscação vista durante treinamento, mas a generalização para ferramentas de ofuscação não vistas permanece em aberto. *Razão:* a ofuscação altera radicalmente a textura da imagem grayscale (bytes são reordenados ou criptografados), mas deixa intactos os padrões estatísticos de coocorrência — por isso imagens de Markov são mais robustas.

8.3. Generalização entre datasets

As diferenças de 18–19 p.p. reportadas em [26, 27] indicam que os modelos parcialmente memorizam propriedades específicas do Malimg em vez de aprender características gerais de malware. *Razão:* o Malimg foi coletado antes de 2012 e não reflete a diversidade estrutural do malware atual; modelos treinados nele sofrem viés de distribuição (*dataset shift*) ao serem testados em corpora mais recentes.

8.4. Validade temporal e drift conceitual

Nenhum artigo revisado avalia explicitamente se um modelo treinado em amostras antigas detecta ameaças mais recentes. *Razão:* a maioria dos benchmarks disponíveis são estáticos; não há protocolos padronizados de avaliação temporal.

8.5. Reprodutibilidade e custo computacional

A maioria dos artigos omite hiperparâmetros, sementes aleatórias e código-fonte. Métricas de custo computacional (número de parâmetros, tempo de inferência, custo de treinamento em GPU) são reportadas por menos de 10% dos artigos. Sem essas informações, é impossível avaliar a viabilidade prática de implantação.

8.6. Desbalanceamento de classes

Benchmarks como Malimg possuem famílias com centenas de amostras e outras com dezenas. Modelos que ignoram famílias raras podem apresentar altas métricas agregadas enquanto falham nas ameaças mais novas.

9. Limitações desta Revisão

9.1. Bases consultadas e viés de publicação

As buscas foram realizadas em quatro bases (IEEE Xplore, ACM DL, Scopus, arXiv). Artigos publicados em periódicos não indexados nessas bases, em idiomas além do inglês ou em conferências de menor visibilidade podem ter sido omitidos. O viés de publicação favorece resultados positivos: artigos com acurácia abaixo de 90% raramente chegam às bases consultadas.

9.2. Restrições do período de busca

A busca foi realizada durante o mês de abril de 2026 e cobre publicações de 2019 a 2026. Trabalhos fundacionais anteriores a 2019 (como o artigo original de Nataraj et al., 2011) são citados apenas como referências de contexto.

9.3. Ausência de meta-análise formal

A heterogeneidade dos experimentos revisados — diferentes datasets, divisões treino/teste, versões de modelos e protocolos de avaliação — impossibilita a realização de meta-análise estatística formal (e.g., cálculo de effect size combinado). As estatísticas descritivas apresentadas na Seção 7 são agregações informais sujeitas a vieses de comparabilidade.

9.4. Restrições metodológicas nos artigos revisados

A maioria dos artigos não reporta custo computacional, usa dataset único e omite avaliação de robustez. As conclusões desta revisão sobre generalização e robustez são portanto baseadas em um subconjunto pequeno dos 69 artigos que realizaram avaliações mais abrangentes.

Síntese — RQ4: As principais lacunas são: (a) ausência de avaliação multi-dataset (apenas 8/69 artigos); (b) não reporte de custo computacional ($<$10%); (c) ausência de avaliação temporal (0/69); e (d) cobertura limitada de cenários de ofuscação real (3/69). Essas lacunas limitam a transferência dos resultados de laboratório para implantações operacionais.

10. Direções Futuras de Pesquisa

Avaliação obrigatória em múltiplos datasets. Todo artigo que propuser uma nova arquitetura deve incluir ao menos dois datasets: um benchmark padrão (Malimg ou MaleVis) e um teste com distribuição distinta (BIG-2015, CICMalDroid para Android ou dataset temporal).

Reporte de custo computacional. Número de parâmetros, tempo de inferência por amostra (em CPU e GPU) e custo de treinamento devem ser reportados sistematicamente, especialmente para arquiteturas destinadas a borda.

Treinamento e teste com malware ofuscado. Seguindo [65, 5], artigos futuros devem reportar acurácia separadamente para amostras limpas, empacotadas e ofuscadas com ferramentas distintas.

Protocolos de avaliação temporal. Dividir os dados por data de coleta das amostras permitiria quantificar a taxa de degradação dos modelos ao longo do tempo.

Benchmarks de larga escala como co-primários. O MalNet-Image [28] (1,26M amostras, 696 famílias) fornece avaliações muito mais discriminativas do que o Malimg e deve ser adotado como co-primário pela comunidade.

Interpretabilidade como critério padrão. Grad-CAM, LIME e SHAP devem ser reportados em conjunto com a acurácia em todo novo artigo, seguindo [31, 30, 46].

11. Conclusão

Esta revisão sistemática analisou **69 artigos** publicados entre 2019 e 2026 sobre detecção e classificação de malware por meio de visualização de binários como imagem. As principais conclusões quantitativas consolidadas são:

1. **Dominância do transfer learning com ResNet.** ResNet aparece em 42 artigos (61% do corpus); VGG em 35 (51%); EfficientNet em 22 (32%). Transfer learning é a estratégia em $>$70% dos artigos.
2. **Acurácias no Malimg: mediana 98,5%, máximo 99,97%.** Mais de 75% dos artigos reportam acurácia $>$97% no Malimg — evidência de saturação do benchmark. O melhor resultado é 99,97% [1].
3. **Modelos leves são viáveis.** MalShuffleNet (99,03%, 2,5M parâmetros) e γ-M2I com MobileNetV2 (99,82%, 1,25 ms de inferência) demonstram que arquiteturas compactas aproximam-se do teto de performance em uma fração do custo.
4. **GANs são a representação generativa dominante.** 18 artigos (26%) usam GAN; VAE aparece em 3 (4%). Modelos generativos melhoram acurácia em 8–10 p.p. quando usados para balanceamento.
5. **Ofuscação e generalização são as maiores lacunas práticas.** Apenas 3/69 artigos avaliam robustez à ofuscação; apenas 8/69 realizam avaliação cross-dataset. A queda mediana cross-dataset é de 18–19 p.p.
6. **Custo computacional raramente reportado.** Menos de 10% dos artigos reportam número de parâmetros e tempo de inferência — impedindo comparação de viabilidade prática.
7. **Tendências emergentes.** Aprendizado autossupervisionado [17], modelos visão-linguagem [46] e few-shot learning [18, 19] representam os paradigmas com maior potencial para reduzir o custo de rotulação e melhorar a generalização.

A abordagem baseada em imagens para detecção de malware evoluiu de uma ideia promissora para um campo maduro — com mais de 35 arquiteturas benchmarkadas em Malimg e resultados consistentemente acima de 97%. O próximo desafio é reduzir a distância entre benchmarks de laboratório e implantações reais, o que exige tratar robustez à ofuscação, validade temporal, reporte de custo computacional e avaliação multi-dataset como critérios mínimos de publicação.

Referências

[1] Abdullah Sheneamer. Visualized malware images using hybrid ensemble deep transfer learning. In *2024 7th International Conference on Information and Computer Technologies (ICICT)*, pages 7–12, 2024.

[2] Pooja Bagane, Susheel George Joseph, and Abhishek Singh. Classification of malware using deep learning techniques. In *2021 9th International Conference on Cyber and IT Service Management (CITSM)*, 2021.

[3] Jinrong Chen, Xiaoqi Jia, Chongming Zhao, Weijuan Zhang, and Qingjia Huang. Using the rgb image of machine code to classify the malware. In *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pages 542–549, 2020.

[4] Duc-Ly Vu, Trong-Kha Nguyen, Tam V. Nguyen, Tu N. Nguyen, Fabio Massacci, and Phu H. Phung. A convolutional transformation network for malware classification. *arXiv*, pages 234–239, 2019.

[5] Dhanya K. A., Vinod P., Suleiman Y. Yerima, Abul Bashar, Anwin David, Abhiram T., Alan Antony, Ashil K. Shavanas, and Gireesh Kumar T. Obfuscated malware detection in iot android applications using markov images and cnn. *IEEE Systems Journal*, 17(2):2756–2766, 2023.

[6] Lok Man Kwan. Markov image with transfer learning for malware detection and classification. In *TENCON 2022 - 2022 IEEE Region 10 Conference (TENCON)*, pages 1–6, 2022.

[7] Wanhu Nie and Changsheng Zhu. y-M2I: Image-based malware classification via feature spatial transformation. *Journal of Information Security and Applications*, 97:104355, 2026.

[8] Pham Nhat Duy and Nguyen Tan Cam. Multi-feature image analysis for android malware classification using convolutional neural networks. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, pages 1425–1430, 2024.

[9] Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Satish Chikkagoudar, Shivkumar Chandrasekaran, and B. S. Manjunath. Malware detection using frequency domain-based image visualization and deep learning. *arXiv preprint arXiv:2101.10578*, 2021.

[10] Zhuojun Ren and Ting Bai. Malware visualization based on deep learning. In *2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, 2021.

[11] Anisha Mahato, Rana Majumdar, and Swarup Kr Ghosh. A comparative analysis of different cnn architectures for malware classification. In *2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, pages 1–7, 2024.

[12] Lingfeng Qiu, Shuo Wang, Jian Wang, Yifei Wang, and Wei Huang. Malware classification based on a light-weight architecture of cnn: Malshufflenet, 2022.

[13] Zepeng Wu, Wenyin Yang, Linjiu Guo, Dong Yuan, Ziliang Liu, and Li Ma. Maledgenext: A lightweight malware family classification method. In *2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 492–496, 2023.

[14] Linjun Yu and Fuyong Zhang. A malware detection model based on the fusion of swin transformer and mona modules. In *2025 6th International Conference on Electronic Communication and Artificial Intelligence (ICECAI)*, pages 796–801, 2025.

[15] Sarath Jayan Nair and Sreelakshmi R Syam. Comparing transformers and cnn approaches for malware detection: A comprehensive analysis. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6, 2024.

[16] Sachith Seneviratne, Ridwan Shariffdeen, Sanka Rasnayaka, and Nuran Kasthuriarachchi. Self-supervised vision transformers for malware detection. *IEEE Access*, 10:103121–103135, 2022.

[17] Setia Juli Irzal Ismail, Hendrawan, Budi Rahardjo, Tutun Juhana, and Yasuo Musashi. Malssl—self-supervised learning for accurate and label-efficient malware classification. *IEEE Access*, 12:58823–58835, 2024.

[18] Sahil Singh and Stones Dalitso Chindipha. Evaluating a novel voting system for malware dataset curation with siamese networks. In *2026 28th International Conference on Advanced Communications Technology (ICACT)*, pages 1–11, 2026.

[19] Hanjun Li, Shuhong Chen, Guojun Wang, Liu Cheng, Haojie Yin, and Zhenkun Luo. Enhancing few-shot malware classification through joint learning of malware images and opcode sequences. In *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, pages 2060–2068, 2024.

[20] Jinting Zhu, Julian Jang-Jaccard, Amardeep Singh, Ian Welch, Harith Al-Sahaf, and Seyit Camtepe. A few-shot meta-learning based siamese neural network using entropy features for ransomware classification. *arXiv preprint arXiv:2112.00668*, 2022.

[21] Dong-Ok Won, Yong-Nam Jang, and Seong-Whan Lee. Plausmal-gan: Plausible malware training based on generative adversarial networks for analogous zero-day malware detection. *IEEE Transactions on Emerging Topics in Computing*, 11(1):82–94, 2023.

[22] Chirag Joshi, Jashvant Kumar, and Gaurav Kumawat. Detection of unseen malware threats using generative adversarial networks and deep learning models. *Scientific Reports*, 15:34804, 2025.

[23] Hayat Hussain Reshi and Karan Singh. Enhancing malware detection using deep learning approach. In *2024 International Conference on Automation and Computation (AUTOCOM)*, pages 497–501, 2024.

[24] Shymala Gowri S and Savari Rhea Charis L. Opcode sequence-based malware detection using bi-lstm with attention mechanism: A comparative study with image-based analysis. In *2025 International Conference on Next Generation Computing Systems (ICNGCS)*, pages 1–6, 2025.

[25] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372:n71, 2021.

[26] Ryan Frederick, Joseph Shapiro, and Ricardo A. Calix. A corpus of encoded malware byte information as images for efficient classification. In *2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 32–36, 2022.

[27] Harman Saggu, Er. Karandeep Singh, and T. N. Suresh Babu. Transfer learning based multi-class malware classification using vgg16 feature extractor and svm classifier, 2024.

[28] Scott Freitas, Rahul Duggal, and Duen Horng Chau. Malnet: A large-scale image database of malicious software. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22)*, 2022.

[29] Deepa K, Adithyakumar K S, and Vinod P. Malware image classification using vgg16. In *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, 2022.

[30] Malak Alamri, Zulfiqar Ahmad, Mamoona Humayun, Menwa Alshammeri, Amr Munshi, and Jawad Rasheed. Cybersecurity framework for proactive detection of image-based malware in healthcare networks using explainable deep ensemble learning and edge analytics. *IEEE Transactions on Consumer Electronics*, pages 1–1, 2026.

[31] Matteo Brosolo, P Vinod, and Mauro Conti. The road less traveled: Investigating robustness and explainability in cnn malware detection. *arXiv preprint arXiv:2503.01391*, 2025.

[32] Eshwar Kotha, Lakshmi Harika Palivela, and Afruza Begum. Enhanced malware image classification through ensemble model. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 1421–1425, 2024.

[33] Vasundhara Acharya, Vinayakumar Ravi, and Nazeeruddin Mohammad. Efficientnet-based convolutional neural networks for malware classification. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6, 2021.

[34] Ahmed Bensaoud, Nawaf Abudawaood, and Jugal Kalita. Classifying malware images with convolutional neural network models. *arXiv preprint arXiv:2010.16108*, 2020.

[35] Nadia El ghabri, Elmostafa Belmekki, and Mostafa Bellafkih. Pre-trained deep learning models for malware image based classification and detection. In *2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS)*, pages 1–7, 2024.

[36] Hayet Benbrahim and Ali Behloul. Malware classification on malimg using mobilenet and lstm for efficient detection. In *2024 1st International Conference on Innovative and Intelligent Information Technologies (IC3IT)*, pages 1–6, 2024.

[37] Hong-Yi Chang, Yu-Cheng Huang, Tzu-Shan Chang, and Yu-Chieh Chang. A malware detection and classification system based on lightweight cnn model. In *2025 IEEE International Conference on Consumer Electronics (ICCE)*, 2025.

[38] Alvina Liaqat, Ukasha Shahid, Izza Shah, Ahmed Kaleem, and Aman Riaz. Deep learning-based malware detection using independent stream analysis of rgb and grayscale images. In *2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, pages 1–5, 2025.

[39] Mohammad Muhibur Rahman, Anushua Ahmed, Mutasim Husain Khan, Mohammad Rakibul Hasan Mahin, Fahmid Bin Kibria, Dewan Ziaul Karim, and Mohammad Kaykobad. Cnn vs transformer variants: Malware classification using binary malware images. In *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, pages 308–315, 2023.

[40] Shuwei Wang, Qiuyun Wang, Zhengwei Jiang, Xuren Wang, and Rongqi Jing. A weak coupling of semi-supervised learning with generative adversarial networks for malware classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3775–3782, 2021.

[41] Anurag Dutta, Satya Prakash Nayak, Ruchira Naskar, and Rajat Subhra Chakraborty. Kol-4-gen: Stacked kolmogorov-arnold and generative adversarial networks for malware binary classification through visual analysis. *IEEE Embedded Systems Letters*, 17(4):268–271, 2025.

[42] Salma Abdelmonem, Shahd Seddik, Rania El-Sayed, and Ahmed S. Kaseb. Enhancing image-based malware classification using semi-supervised learning. In *2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 125–128, 2021.

[43] Tan Gao, Xudong Li, and Wen Chen. Co-training for image-based malware classification. In *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pages 568–572, 2021.

[44] Alve Rahman Akash, BoHyun Ahn, Alycia Jenkins, Ameya Khot, Lauren Silva, Hugo Tavares-Vengas, and Taesic Kim. Quantum convolutional neural network-based online malware file detection for smart grid devices. In *2023 IEEE Design Methodologies Conference (DMC)*, pages 1–5, 2023.

[45] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, Tejaswi Nanjundaswamy, Satish Chikkagoudar, Shivkumar Chandrasekaran, and B. S. Manjunath. Omd: Orthogonal malware detection using audio, image, and static features. In *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 2021.

[46] Yi Xiang Marcus Tan, Koh Ting Yew, and Lee Joon Sern. Malwareclip: Towards a scalable and explainable image-based malware classification. In *2023 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pages 1–8, 2023.

[47] Yunchun Zhang, Zikun Liao, Ning Zhang, Shaohui Min, Qi Wang, Tony Q. S. Quek, and Mingxiong Zhao. Deep hashing for malware family classification and new malware identification. *IEEE Internet of Things Journal*, 11(16):26837–26851, 2024.

[48] Giacomo Iadarola, Fabio Martinelli, Francesco Mercaldo, and Antonella Santone. Evaluating deep learning classification reliability in android malware family detection. In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 255–260, 2020.

[49] Sanjeev Kumar, Yudhishthira Sapru, and Sugandha Sapru. Robust image based malware detection using mobilenet and multilayer perceptron model. In *2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA)*, pages 1–6, 2025.

[50] L. Sasikala and C. Shanmuganathan. Effective malware classification using fine-tuned cnn architecture: An image-based approach. In *2024 2nd International Conference on Networking and Communications (ICNWC)*, pages 1–6, 2024.

[51] Tetsuro Takahashi, Rikima Mitsuhashi, Masakatsu Nishigaki, and Tetsushi Ohki. Study of appropriate information combination in image-based obfuscated malware detection. In *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, pages 267–268, 2025.

[52] Raveena V and Kaja Mohideen A. Scalable malware image classification using

progressive resizing in deep neural networks. In *2026 9th International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–6, 2026.

[53] Sijie Wang, Faqun Jiang, Li Wang, and Jiangnan Zhang. A malware classification framework based on visualization and feature fusion. In *2025 18th International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pages 374–379, 2025.

[54] Wenjie Liu and Liming Wang. A novel malware classification method based on memory image representation. In *2023 IEEE Symposium on Computers and Communications (ISCC)*, pages 1404–1409, 2023.

[55] Tran Hoang Hai, Vu Van Thieu, Tran Thai Duong, Hong Hoa Nguyen, and Eui-Nam Huh. A proposed new endpoint detection and response with image-based malware detection system. *IEEE Access*, 11:122859–122875, 2023.

[56] Febri Putra Sandhya Prawiranata and Raden Budiarto Hadiprakoso. Comparison of transfer learning performance in image-based malware file classification on the dumpware10 dataset. In *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)*, pages 252–257, 2023.

[57] Cornelius Paardekooper, Nasimul Noman, Raymond Chiong, and Vijay Varadharajan. Designing deep convolutional neural networks using a genetic algorithm for image-based malware classification. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2022.

[58] Sanjeev Kumar, Amit Singh, and Aniket Abhishek Soni. Detection of advance malware threats using hybrid deep learning model and image analysis. In *2025 Seventeenth International Conference on Contemporary Computing (IC3)*, pages 1–6, 2025.

[59] Mohamad Mulham Belal and Divya Meena Sundaram. Global-local attention-based butterfly vision transformer for visualization-based malware classification. *IEEE Access*, 11:69337–69355, 2023.

[60] Aviad Cohen, Nir Nissim, and Yuval Elovici. Maljpeg: Machine learning based solution for the detection of malicious jpeg images. *IEEE Access*, 8:19997–20011, 2020.

[61] Chenghua Tang, Chen Zhou, Min Hu, Mengmeng Yang, and Baohua Qiang. Malicious family identify combining multi-channel mapping feature image and fine-tuned cnn. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 9–19, 2022.

[62] Fangtian Zhong, Zekai Chen, Minghui Xu, Guoming Zhang, Dongxiao Yu, and Xiuzhen Cheng. Malware-on-the-brain: Illuminating malware byte codes with images for malware classification. *IEEE Transactions on Computers*, 2022.

[63] Hongjiao Li, Jiabei Wu, and Fan Gu. Ph-cnn for pe malware classification by means of enhanced images. In *2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR)*, pages 104–109, 2024.

[64] Mohd Zamri Osman, Ahmad Firdaus Zainal Abidin, Rahiwan Nazar Romli, and Mohd Faaizie Darmawan. Pixel-based feature for android malware family classification using machine learning algorithms. In *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, pages 552–555, 2021.

[65] Hayato Sato, Hiroshi Inamura, Shigemi Ishida, and Yoshitaka Nakamura. Plain source code obfuscation as an effective attack method on iot malware image classification. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 940–945, 2023.

[66] MinSu Kim. Research on malware detection system using artificial intelligence. In *2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, pages 211–213, 2022.

[67] Hui Guo, Jiang-tao Wu, Shu-guang Huang, Zu-lie Pan, Fan Shi, and Zhi-hao Yan. Research on malware variant detection based on global texture features. In *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pages 1443–1446, 2020.

[68] Kshatriya Vinaya Sree Bai and M. Thirumaran. Survey of deep learning techniques for malware detection: Insights, challenges, and future directions. In *2024 4th International Conference on Soft Computing for Security Applications (ICSCSA)*, pages 320–324, 2024.

[69] Arjun Sudheer, Ayesha Ahmed, Fabio Di Troia, and Younghee Park. Synthetic malware image generation based on generative models against zero-day attacks. In *2025 Silicon Valley Cybersecurity Conference (SVCC)*, pages 1–9, 2025.

[70] Yafei Song, Dandan Zhang, Jian Wang, Yanan Wang, Yang Wang, and Peng Ding. Application of deep learning in malware detection: a review. *Journal of Big Data*, 12:99, 2025.