UNIVERSIDADE FEDERAL DE PERNAMBUCO

Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Eletrônica e Sistemas

Graduação em Engenharia Eletrônica

Lucas Rattes Lima Caldas

ANTIVÍRUS BASEADO EM APRENDIZADO EXTREMO VISANDO A DETECÇÃO PREVENTIVA DE *RANSOMWARE*

Recife

2025

Lucas Rattes Lima Caldas

ANTIVÍRUS BASEADO EM APRENDIZADO EXTREMO VISANDO A DETECÇÃO PREVENTIVA DE *RANSOMWARE*

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica, do Departamento de Eletrônica e Sistemas, da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientador(a): Prof. Sidney Marlon Lopes de Lima, D.Sc.

Recife
2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Caldas, Lucas Rattes Lima.
Antivírus Baseado em Aprendizado Extremo Visando a Detecção Preventiva de Ransomware / Lucas Rattes Lima Caldas. - Recife, 2025.
65 : il., tab.

Orientador(a): Sidney Marlon Lopes de Lima
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Eletrônica - Bacharelado, 2025.
Inclui referências.

1. Malware. 2. Ransomware. 3. Máquinas Morfológicas de Aprendizado Extremo. 4. Segurança Cibernética. I. Lima, Sidney Marlon Lopes de. (Orientação). II. Título.

000 CDD (22.ed.)

Lucas Rattes Lima Caldas

ANTIVÍRUS BASEADO EM APRENDIZADO EXTREMO VISANDO A DETECÇÃO PREVENTIVA DE *RANSOMWARE*

Trabalho de Conclusão apresentado ao Curso de Graduação em Engenharia Eletrônica, do Departamento de Eletrônica e Sistemas, da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.

Aprovado em: 05/09/2025

Banca Examinadora

Documento assinado digitalmente
gov.br
SIDNEY MARLON LOPES DE LIMA
Data: 26/09/2025 13:47:35-0300
Verifique em https://validar.iti.gov.br

Prof. Sidney Marlon Lopes de Lima, D.Sc.
Universidade Federal de Pernambuco

Documento assinado digitalmente
gov.br
GUILHERME NUNES MELO
Data: 27/09/2025 17:32:38-0300
Verifique em https://validar.iti.gov.br

Prof. Guilherme Nunes Melo, D.Sc.
Universidade Federal de Pernambuco

Resumo do Trabalho de Conclusão de Curso apresentado ao Departamento de Eletrônica e Sistemas, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Eletrônica(Eng.)

ANTIVÍRUS BASEADO EM APRENDIZADO EXTREMO VISANDO A DETECÇÃO PREVENTIVA DE *RANSOMWARE*

Lucas Rattes Lima Caldas

Ransomware são programas maliciosos que bloqueiam ou criptografam dados do usuário, exigindo pagamento para restaurar o acesso às informações comprometidas. Essa família de *malware* (malicioso + *software*) tem se destacado pela crescente sofisticação, como o uso de técnicas de criptografia robustas, mecanismos de persistência avançados e vetores de disseminação altamente eficientes. A proposta apresentada neste trabalho consiste na criação de uma solução antivírus própria, baseada em uma arquitetura de aprendizado extremo, voltada para a detecção e contenção preventiva de *ransomware*. O antivírus autoral é capaz de identificar *ransomware* antes de ser iniciado pelo usuário. Os experimentos demonstraram uma taxa média de assertividade de 99,87% na distinção entre *softwares* legítimos e ameaças do tipo *ransomware*, com tempo médio de treinamento de apenas 3,75 segundos, evidenciando alto desempenho e resposta rápida. Assim, a solução contribui de forma significativa para o fortalecimento da segurança cibernética.

Palavras-chave: *Malware*, *Ransomware*, Máquinas Morfológicas de Aprendizado Extremo, Segurança Cibernética.

Abstract of Course Conclusion Work, presented to Departament of Eletronic and Systems, as a partial fulfillment of the requirements for the degree of Bachelor of Electronic Engineering(Eng.)

EXTREME LEARNING-BASED ANTIVIRUS AIMED AT PREVENTIVE DETECTION OF *RANSOMWARE*

Lucas Rattes Lima Caldas

Ransomware is a type of malicious software. It blocks or encrypts user data. Then, it demands payment to unlock the compromised information. This family of malware (malicious + software) is known for its growing sophistication. It uses strong encryption methods, clever persistence techniques, and effective ways to spread. This proposal suggests developing a unique antivirus solution. It will use an extreme learning architecture. The goal is to prevent and contain ransomware. The proprietary antivirus can spot ransomware before the user launches it. The experiments found an average accuracy of 99.87% in distinguishing real software from ransomware threats. The average training time was just 3.75 seconds, which shows quick performance. Thus, the solution contributes significantly to strengthening cybersecurity.

Keywords: Malware, Ransomware, Morphological Extreme Learning Machines, Cybersecurity.

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$\vee$ . . Operação lógica de "OU"(disjunção)

$\wedge$ . . Operação lógica de "E"(conjunção)

$\in$ . . . Simbolo de "pertence a"Algébrico

$\mathbb{N}$ . . . . Conjunto dos números naturais

$\prod$ . . . . . Produtório matemático

$\bigcap$ . . . . interseção entre dois conjuntos

$\bigcup$ . . . . . União entre dois conjuntos

$\mathbb{R}$ . . . . . Conjunto dos números reais

$\dagger$ Matriz pseudoinversa de Moore-Penrose

Lista de Abreviações

SDK	Software Development Kit
ELM	Extreme Learning Machine
IoT	Internet of Things
mELM	Morphological Extreme Learning Machine
DL	Deep Learning
MLP	Multilayer Perceptron
CUDA	Compute Unified Device Architecture
RAM	Random access memory
TLS	Transport Layer Security
API	Application Programming Interface
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
RBF	Radial Bases Function
LTSM	Long Short-Term Memory
RBM	Restricted Boltzmann Machine
PE	Portable Executable
ROM	Read-Only Memory

Sumário

Capítulo 1

Introdução

A Evolução tecnológica - transformou profundamente a forma como dados circulam no mundo atual. Esse avanço trouxe inúmeros benefícios, tornando processos mais ágeis e acessíveis. Contudo também abriu espaço para que agentes maliciosos explorassem vulnerabilidades, acessando informações confidenciais e causando prejuízos significativos à sociedade.

Dentro desse contexto, os ataques classificados como *ransomware* têm se destacado cada vez mais, especialmente devido à sua capacidade de extorquir vítimas e interromper serviços essenciais, incluindo órgãos públicos e grandes corporações. Trata-se de ameaças direcionadas, planejadas antecipadamente, que aproveitam brechas críticas nos sistemas e utilizam métodos sofisticados para alcançar seus objetivos.

Um exemplo alarmante ocorreu em setembro de 2024, quando um hospital localizado em Recife sofreu um ataque de *ransomware*. Como consequência, pacientes ficaram impossibilitados de acessar resultados de exames e históricos médicos, impactando diretamente a vida das pessoas e dos serviços de saúde. O incidente reforça a necessidade urgente de investimentos constantes em segurança da informação para proteger dados sensíveis e viabilizar a continuidade operacional das instituições [1].

É importante diferenciar dois conceitos fundamentais: vírus e *malware*. O termo

[1] Fonte: https://g1.globo.com/pe/pernambuco/noticia/2024/09/04/hospital-portugues-e-alvo-de-invasao-hacker-e-pacientes-ficam-sem-acesso-a-resultados-de-exames-ghtml

"vírus" se refere a uma categoria específica de *malware* que se replica infectando outros arquivos, de maneira semelhante a organismos biológicos. Já o termo "*malware*" engloba uma gama mais ampla de *softwares* maliciosos, incluindo *worms*, *trojans*, *ransomware*, *spyware*, *adware*, entre outros. Nesse sentido, os chamados "antivírus" atuam não apenas contra vírus, mas contra diferentes tipos de ameaças digitais. Portanto o termo mais apropriado seria *antimalware*, já que essas ferramentas têm um alcance bem mais abrangente.

Reconhecendo a urgência de mecanismos mais eficientes, torna-se possível realizar uma análise crítica sobre o desempenho das soluções tradicionais de proteção digital. Muitos antivírus comerciais apresentam limitações relevantes na detecção de ameaças avançadas, pois se baseiam em bancos de dados com assinaturas de executáveis infectados já conhecidos; as chamadas listas sujas [2].

Cada fabricante de antivírus utiliza sua própria lista suja para armazenar essas assinaturas, e por questões comerciais, não há compartilhamento entre diferentes produtos. Essa limitação exige que se explorem abordagens mais modernas, como o uso de inteligência artificial para identificar padrões suspeitos, de maneira antecipada. Um exemplo promissor está no uso de redes neurais, que podem alcançar índices de acerto superiores a 98% (LIMA et al., 2020).

Embora as redes neurais sejam ferramentas poderosas para a detecção de ameaças, enfrentam desafios, como o tempo necessário para gerar respostas. Em cenários onde cada segundo conta, uma alta latência pode permitir que o *malware* cause danos antes que a intervenção ocorra. No caso específico das redes Perceptron Multicamadas (MLP), a obtenção de um bom desempenho depende de uma calibragem cuidadosa dos parâmetros, além de estratégias para evitar mínimos locais, que prejudicam a capacidade de generalização do modelo (LIMA et al., 2020). Outro fator relevante é o tempo de treinamento, que pode ser considerável para alcançar a acurácia desejada.

Uma alternativa interessante está nas redes ELM (*Extreme Learning Machine*),

[2]Lista suja: anteriormente chamada de lista negra, termo substituído para evitar conotações pejorativas étnico-raciais.

conhecidas pelo treinamento rápido e boa performance, especialmente em comparação com as MLPs. Essas redes possuem apenas uma camada oculta e calculam os pesos da saída de forma analítica, reduzindo a dependência da aleatoriedade na inicialização. As redes extremas trabalham com *kernels* e apresentam alta adaptabilidade em tarefas classificatórias (LIMA et al., 2021). As ELMs têm sido amplamente empregadas em áreas como engenharia biomédica (LIMA et al., 2014 LIMA et al., 2020 LIMA et al., 2016 PEREIRA, 2020 AZEVEDO and *et al.*, 2015 AZEVEDO and *et al.*, 2020).

Neste trabalho, propõe-se empregar as ELMs para reforçar a segurança digital, utilizando características extraídas de arquivos suspeitos como entradas para o modelo, que então os classifica como legítimos ou maliciosos. A técnica explorada utiliza *morphological* ELMs (mELMs), uma variante que incorpora operações matemáticas inspiradas na morfologia, como erosão e dilatação, em substituição aos *kernels* tradicionais. O modelo de mELM autoral se encontra disponível no repositório autoral (DejavuForensics, 2025).

Os experimentos realizados permitiram comparar o desempenho do modelo proposto com antivírus comerciais líderes. Utilizando métricas consolidadas, os resultados mostraram uma impressionante taxa média de acerto de 99,87% na distinção entre *softwares* legítimos e *ransomware*, com tempo médio de treinamento de apenas 3,75 segundos.

1.1 Limitações das Soluções Tradicionais

Parte significativa dos antivírus disponíveis no mercado baseia sua detecção na análise de assinaturas digitais armazenadas em bases internas. Isso significa que, se o *hash* de um novo *malware* não estiver registrado, ele pode não ser detectado. Pequenas modificações no código malicioso são suficientes para alterar seu *hash*, tornando-o invisível para esses sistemas (Tavares-Silva et al., 2025), (Pinheiro Henriques de Araújo et al., 2024).

Essa vulnerabilidade é frequentemente explorada por pessoas mal-intencionadas,

que geram novas versões de *malware* alterando apenas detalhes superficiais do código. Adicionalmente, o uso de *exploit* automatiza a criação e propagação dessas variantes, tornando os ataques ainda mais perigosos (Santos and Lopes-Lima, 2024).

No presente estudo, 1.174 *ransomware*, de domínio público, foram empregados de modo a analisar os 86 principais antivírus comerciais mundiais, por intermédio da plataforma *VirusTotal*. Na referida plataforma, os antivírus comerciais podem emitir três tipos distintos de diagnóstico para cada arquivo suspeito investigado. No VirusTotal, os três diagnósticos são:

- *malware*: no presente contexto é quando o *ransomware* é identificado corretamente
- benigno: trata-se de falso negativo no presente experimento, visto que todoas as 1.174 são *ransomware* de domínio público.
- "sem diagnóstico": nenhuma classificação atribuída, dá-se quando o antivírus se omite quanto ao arquivo suspeito.

A Tabela 1.1 apresenta os 10 melhores e os 10 piores antivírus comerciais no enfrentamento aos *ransomware* presentes na base de dados autoral. Os resultados completos podem ser consultados no repositório autoral (ransomware, 2025). As taxas de acerto variaram de 0% a 97,63%, com média de 55,22% e desvio padrão de 30,05%, evidenciando um desempenho altamente desigual. Adicionalmente, verificou-se uma média de 13,15% de falsos negativos (desvio padrão de 11,49%) e 31,71% de omissões (desvio de 31,88%). Esses índices representam riscos consideráveis, tanto para organizações quanto para usuários individuais.

Outro ponto crítico é a falta de padronização na nomenclatura das ameaças detectadas, conforme visto na Tabela 1.2. Diversos antivírus rotulam códigos semelhantes com nomes distintos ou genéricos, dificultando a criação de estratégias unificadas de resposta. Até mesmo pequenas alterações podem resultar em classificações totalmente diferentes, dependendo do produto utilizado.

Diante desse cenário, torna-se essencial adotar soluções mais avançadas e padronizadas, capazes de superar essas limitações e oferecer uma defesa mais eficaz diante dos desafios crescentes da segurança cibernética. Isso inclui, por exemplo, a incorporação de abordagens baseadas em inteligência artificial, aprendizado contínuo e análise comportamental, que permitem detectar ameaças mesmo quando elas não são reconhecidas por métodos tradicionais baseados em assinaturas.

Tabela 1.1: Resultados sintetizados dos antivírus comerciais. Os resultados completos podem ser consultados no repositório autoral (ransomware, 2025).

Antivirus	Detecção (%)	Falso Negativo (%)	Omissão (%)
McAfee	97.63%	2.29%	0.07%
McAfee-GW-Edition	97.26%	2.43%	0.29%
Kaspersky	95.04%	4.50%	0.44%
BitDefender	94.53%	5.09%	0.36%
GData	94.38%	5.02%	0.59%
Avast	94.16%	5.46%	0.36%
AVG	94.16%	5.61%	0.22%
Symantec	93.05%	6.43%	0.51%
Sophos	92.90%	6.06%	1.03%
Panda	92.75%	6.87%	0.36%
Prevx	0.14%	1.10%	98.74%
Authentium	0.14%	1.10%	98.74%
a-squared	0.14%	0%	99.85%
Alibaba	0.07%	22.46%	77.45%
ahnlab	0%	0%	100%
Command	0%	0%	100%
SAVMail	0%	0%	100%
FileAdvisor	0%	0%	100%
Ewido	0%	0%	100%
Webwasher-Gateway	0%	0%	100%

Fonte: Repositório autoral de análise de *ransomware* (ransomware, 2025).

Tabela 1.2: Nomenclatura dada pelos antivírus comercias em relação à 2 dos 1.174 *ransomware* usando no experimento autoral.

Antivirus	$VirusShare_A$	$VirusShare_B$
McAfee	PWS-Zbot.gen.ahc	Artemis!000ADC56F2D8
McAfee-GW-Edition	PWS-Zbot.gen.ahc	BehavesLike.Win32.Virus.tc
Kaspersky	HEUR:Trojan.Win32.Generic	Win32:Trojan-gen
Kaspersky	Trojan-Downloader.Win32.Agent.stsm	HEUR:Trojan.Win32.Generic
McAfee-GW-Edition	BehavesLike.Win32.Generic.nh	RDN/Generic BackDoor.km
Microsoft	Trojan:Win32/Sluegot.D	Backdoor:Win32/Stradatu
AVG	Win32:Trojan-gen	Win32:Trojan-gen
ESET-NOD32	a variant of Win32/Agent.ONL	a variant of Win32/Agent.UAX
McAfee	BackDoor-FALR!001DD76872D8	RDN/Generic BackDoor.km
Avira	HEUR/AGEN.1109847	HEUR/AGEN.1122860
Malwarebytes	Malware.AI.224237819	false negative
Emsisoft	Android.Gen:Variant.Zusy.334368 (B)	Gen:Variant.Zusy.329358 (B)
IkarusV	Trojan.AndroidOS.FakeInst	Trojan.AndroidOS.FakeInst
MAX	Malware	malware
TrendMicro-HouseCall	Suspicious_GEN.F47V0322	AndroidOS_OPFAKE.A,
Emsisoft	Android.Trojan.FakeInst.CB	Android.Trojan.FakeInst.CB
Ikarus	Trojan-Dropper.Agent	Backdoor.Win32.Dalbot
Arcabit	Trojan.Zusy.D51A20	Trojan.Zusy.D5068E
Tencent	Malware.Win32.Gencirc.114cb474	Win32.Trojan.Kookimon.Fie
VIPRE	Trojan.Win32.Generic!BT	Trojan.Win32.Generic!BT

Fonte: Repositório autoral de análise de *ransomware* (ransomware, 2025).

1.2 Objetivo Geral

Desenvolver um sistema antivírus especializado, baseado em redes neurais extremas, capaz de identificar *softwares* maliciosos, em específico variantes de *ransomware*. A detecção deve ocorrer por meio da análise de padrões de características presentes em arquivos comprometidos, eliminando a dependência de mecanismos obsoletos como listas de bloqueio tradicionais.

1.2.1 Objetivos específicos

- Investigar as estratégias e abordagens utilizadas por antivírus comerciais no combate aos *malware*, com ênfase nos *ransomware*.
- Mapear falhas, vulnerabilidades e lacunas presentes nas práticas correntes de proteção digital.
- Explorar a viabilidade do uso de redes neurais extremas como ferramenta para detecção proativa de *malware*, especialmente *ransomware*.
- Montar um conjunto de dados balanceado, contendo exemplos reais de arquivos infectados e não infectados, para treinamento e validação do modelo proposto.
- Comprovar, através do desenvolvimento de um protótipo funcional, a hipótese de que redes neurais extremas podem ser empregadas com sucesso na criação de antivírus inteligentes.
- Realizar uma análise comparativa entre o desempenho do antivírus desenvolvido e as soluções de antivírus já estabelecidas no mercado, destacando pontos fortes e ganhos obtidos.

1.3 Organização do TCC

O trabalho está organizado em 6 capítulos e conta ainda com um apêndice. As referências utilizadas podem ser consultadas ao final do documento. A seguir,

apresenta-se um breve panorama do conteúdo de cada capítulo.

Capítulo 2. Fundamentação Teórica, capítulo dedicado à análise matemática da rede neural aplicada, bem como aos fundamentos teóricos e justificativas metodológicas.

Capítulo 3. Estado da Arte, capítulo que realiza um levantamento das produções acadêmicas e pesquisas voltadas ao desenvolvimento de antivírus com inteligência artificial.

Capítulo 4. Metodologia, seção que descreve em detalhes os métodos, os procedimentos e os equipamentos empregados na criação do antivírus proposto.

Capítulo 5. Resultados, capítulo destinado à apresentação e análise dos dados obtidos durante o treinamento e teste do *software* antivírus baseado em rede neural.

Capítulo 6. Conclusão, capítulo final que sintetiza os principais tópicos tratados ao longo do trabalho, ressaltando os resultados alcançados e suas implicações.

Capítulo 2

Fundamentação Teórica

Apesar de movimentar cifras bilionárias, as soluções antivírus comerciais ainda se baseiam em abordagens ultrapassadas, como o uso de listas sujas (ou denúncias prévias). Ao invés de atuar apenas de forma corretiva, as técnicas de *machine learning* são com capacidade de reconhecer previamente a intenção nociva de *softwares* como o *ransomware*. Enfatiza-se que não existem "vacinas" para os ataques de *ransomware*. Tentativas de descriptografar os arquivos sequestrados através de métodos de tentativa e erro podem ser computacionalmente inviáveis, chegando a demandar séculos de processamento (Oz et al., 2021).

Para superar essas limitações, o estado da arte explora estratégias de análise preventiva, nas quais as características do arquivo são extraídas logo antes de sua execução. Essa abordagem torna possível identificar intenções maliciosas de um arquivo suspeito sem precisar que ele seja aberto pelo usuário (LIMA et al., 2021). A partir da inspeção detalhada de um aplicativo, é possível prever se ele tenta criar, excluir, modificar ou realizar *downloads* de arquivos pela internet, além de rastrear preventivamente o tráfego de rede gerado.

Esses métodos inteligentes analisam milhares de arquivos, extraindo e ponderando estatisticamente suas características. O uso de aprendizado de máquina representa, portanto, um avanço crucial para o campo da segurança digital. Embora existam iniciativas na área, os antivírus comerciais ainda dependem fortemente de abordagens antiquadas (LIMA et al., 2021).

O projeto atual emprega *machine learning* com a finalidade de detectar aplicativos maliciosos antes que causem danos, identificando comportamentos suspeitos logo antes de ser aberto pelo usuário. Assim, não é necessário aguardar que o arquivo seja denunciado por vítimas, como ocorre nos métodos tradicionais. Este capítulo introduz o conceito das máquinas de aprendizado estatístico, mecanismo fundamental na detecção de *malware*.

No contexto do reconhecimento de padrões, uma tarefa essencial consiste em atribuir uma classe (ou rótulo) a cada arquivo analisado, com base em suas características. A partir de um conjunto de treinamento, o classificador aprende a formular hipóteses sobre as classes envolvidas no antivírus inteligente proposto. Quando confrontado com um arquivo novo, o classificador estima sua classe comparando as características auditadas com aquelas aprendidas anteriormente.

A pesquisa apresentada neste trabalho aplica redes neurais do tipo ELM (*Extreme Learning Machine*) para detecção de padrões maliciosos no campo da segurança digital. Essas redes são reconhecidas por sua velocidade de treinamento e capacidade preditiva quando comparadas a modelos clássicos baseados em retropropagação ou *Deep Learning*.

As ELMs são particularmente adequadas para aplicações em Perícia Forense Digital, especialmente frente ao ritmo acelerado de criação de novos *malwares*, estimado em oito novos casos por segundo (INTEL, 2022). Esse cenário exige que o tempo de treinamento das ferramentas de detecção acompanhe a escalada das ameaças.

Estudo científico, conduzido por JOHNSON et al., 2016), demonstrava que os melhores cibervigilantes mundiais detectam, em média, 153 novas vulnerabilidades a cada 20 (vinte) dias, há cerca de uma década atrás. Os cibervigilantes, em questão, estavam vinculados às companhias; Google, Cert, Mozilla Firefox, Palo Alto, dentre outras (JOHNSON et al., 2016). Em condições ideais, o tempo de aprendizado do antivírus não deve ser discrepante da taxa de surgimento de *malware "Zero-Day"*. Portanto deveria haver 153 retreinamentos a cada 20 dias, cerca de 7 retreinamentos

diários. Paradoxalmente, um antivírus recém-lançado já pode se encontrar obsoleto devido à lentidão do tempo de treinamento de sua *deep learning*.

2.1 Redes Neurais Extremas

Entender como as redes neurais funcionam é essencial, já que elas se destacam na identificação de padrões complexos. Um dos desafios, entretanto, é evitar que o treinamento fique estagnado em mínimos locais, comprometendo o processo de aprendizado (HUANG, 2000). Para contornar esse problema, técnicas específicas de gerenciamento de arquitetura são aplicadas.

Outro fator que limita as redes tradicionais é o longo tempo necessário para que alcancem níveis adequados de acurácia. Mesmo sendo altamente precisas, essas redes podem demandar dias de treinamento até fornecerem classificações confiáveis.

A principal vantagem das ELMs (*Extreme Learning Machine*) está justamente na alta velocidade de aprendizado, sem abrir mão da qualidade preditiva. Essas redes operam com uma única camada oculta, não iterativa, utilizando um processo analítico para calcular os pesos de saída a partir de uma inicialização aleatória.

As ELMs têm sido amplamente exploradas em diversas áreas, incluindo Engenharia Biomédica (AZEVEDO and *et al.*, 2015AZEVEDO and *et al.*, 2020LIMA et al., 2016LIMA et al., 2020LIMA et al., 2014PEREIRA, 2020). Seu desempenho excepcional as torna candidatas promissoras para aplicações em segurança digital, especialmente no mapeamento de padrões de *malware*.

Do ponto de vista matemático, em um sistema ELM, os parâmetros de entrada x_{ti} pertencem ao conjunto $\left\{x_{it} \in \mathbb{R}; : i = 1, ...n; : t = 1, ...v\right\}$, onde n representa os atributos extraídos do programa e v os vetores de amostras usados no aprendizado. A camada oculta, h_j, contém m neurônios, descritos pelo conjunto $\left\{h_j \in \mathbb{R}; : j \in \mathbb{N}^*; : j = 1, ...m\right\}$.

Como o ELM envolve poucas etapas, seu aprendizado é ágil. Inicialmente, os pesos de entrada w_{ji} e os bias b_{jt} são gerados aleatoriamente. Dada uma função de ativação $f : \mathbb{R} \to \mathbb{R}$, o processo segue os seguintes passos:

- Definição aleatória das ligações sinápticas w_{ji} (entre as camadas de entrada e oculta) e dos vieses b_{jt}.
- Cálculo da matriz H, que representa a resposta dos neurônios ocultos.
- Determinação da matriz das ligações sinápticas de saída $\beta = H^{\dagger}Y$, onde $H^{\dagger}$ é a matriz pseudoinversa de Moore-Penrose e Y é a matriz dos resultados desejados, com $\left\{Y_{tc} \in \mathbb{R}; : t = 1, ...v; : c = 1, ...\zeta\right\}$. Aqui, ζ indica o número de classes (por exemplo, benigno ou *malware*).

Para compreender a matriz pseudo-inversa, é útil lembrar que a matriz inversa está ligada à matriz identidade I. Quando uma matriz quadrada H é multiplicada por sua inversa H^{-1}, o resultado é I. Já no caso de matrizes não quadradas, obtém-se uma matriz aproximadamente inversa: a pseudo-inversa, representada por $H^{\dagger}$, que ajusta os valores das sinapses. A intenção é afastar as regiões próximas ao limite de decisão, em direção às bordas da diagonal secundária.

A matriz H (saída da camada oculta) é calculada usando o *kernel* φ expresso na matriz da Eq. (2.1). As ligações sinápticas de saída β e a matriz de resultados desejados Y são apresentados, respectivamente, nas Eqs. (2.2) e (2.3).

$$H_{tj} = \begin{bmatrix} \varphi_1^1 & \varphi_2^1 & \cdots & \varphi_v^1 \\ \varphi_1^2 & \varphi_2^2 & \cdots & \varphi_v^2 \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1^m & \varphi_2^m & \cdots & \varphi_v^m \end{bmatrix}_{m \times v} \tag{2.1}$$

$$\beta_{jc} = \begin{bmatrix} \beta_1^1 & \cdots & \beta_\zeta^1 \\ \beta_1^2 & \cdots & \beta_\zeta^2 \\ \vdots & \ddots & \vdots \\ \beta_1^m & \cdots & \beta_\zeta^m \end{bmatrix}_{m \times \zeta} \tag{2.2}$$

$$Y_{tc} = \begin{bmatrix} Y_1^1 & \cdots & Y_\zeta^1 \\ Y_1^2 & \cdots & Y_\zeta^2 \\ \vdots & \ddots & \vdots \\ Y_1^v & \cdots & Y_\zeta^v \end{bmatrix}_{v \times \zeta} \tag{2.3}$$

O *kernel*, como já mencionado, é responsável por estabelecer a relação matemática que rege o treinamento da rede neural ELM. Em geral, essas redes utilizam um *kernel* linear, descrito pela Eq. (2.4), cujos resultados podem ser consultados na Tabela 5.2. A função φ depende diretamente de $f(x_{t,1\ldots n}, ; w_{1\ldots m,1\ldots n}, ; b_{1\ldots m,t})$.

$$\varphi_t^j(f) = x_{ti}.w_{ji} + b_{jt} \tag{2.4}$$

Esse tipo de mapeamento matemático é extremamente relevante porque possibilita a criação de representações não lineares sem aumentar a complexidade ajustável ou a quantidade de treinamento normalmente exigida por redes neurais baseadas em retropropagação. A Eq. (2.5) define um *kernel* sigmoidal φ utilizado em redes ELM, cujos resultados são apresentados na Tabela 5.3.

$$\begin{aligned} \varphi_t^j(f) &= Sigmoid\Big(x_{ti}.w_{ji} + b_{jt}\Big), \\ onde\ Sigmoid(\xi) &= \frac{1}{1+e^{-\xi}} \end{aligned} \tag{2.5}$$

2.2 Estudos Preliminares: Morfologia Matemática

Neste trabalho, são exploradas mELMs (ELMs morfológicas), ou seja, redes ELM cujos núcleos de camada oculta se baseiam em operadores morfológicos inspirados nas operações de Erosão e Dilatação, típicas do processamento de imagens. A hipótese é que esses *kernels* sejam capazes de adaptar-se a qualquer fronteira decisória.

A Morfologia Matemática, por sua vez, refere-se ao estudo das formas e estruturas em imagens usando conceitos matemáticos de interseção e união de conjuntos (SANTOS, 2011). Essas operações lidam naturalmente com a detecção e modelagem de regiões distintas. Ao interpretar a fronteira de decisão de uma rede neural como uma imagem n-dimensional (onde n representa o número de atributos extraídos), as mELMs conseguem mapear de forma natural essas regiões para diferentes classes.

A Morfologia Matemática é amplamente reconhecida como uma abordagem não linear eficaz no processamento de imagens digitais, sendo aplicada em tarefas como segmentação, detecção de objetos e extração de características. Suas operações centrais são a Erosão e a Dilatação (SANTOS, 2011), que servem como base para a construção de operações mais complexas. Matematicamente, as funções de Erosão e Dilatação são descritas pelas Eqs. (2.6) e (2.7).

No contexto do processamento de imagens, a Erosão reduz o tamanho dos objetos, removendo *pixels* das bordas, o que resulta no encolhimento das regiões claras e crescimento das escuras. Já a Dilatação faz o oposto: expande objetos, adicionando *pixels* às bordas, ampliando áreas claras e reduzindo regiões escuras. Aqui, cada *pixel* representa a menor unidade de uma imagem digital, definido pelo par $(k, f(k))$, onde k é a posição espacial e $f(k)$ é o valor associado.

$$\epsilon_g(f)(k) = \min(f(w) \ \vee \ \overline{g}(k \text{ - } w)) \tag{2.6}$$

$$\delta_g(f)(k) = \max(f(w) \ \wedge \ g(k \text{ - } w)) \tag{2.7}$$

Nas expressões acima, $f : S \to \{0, 1\}$ e $g : S \to \{0, 1\}$ representam funções em uma matriz de formato S, onde $S \in \mathbb{N}^2$. O valor k corresponde à posição no espaço, e w indica a fórmula matricial, integrada ao elemento estruturante g. A operação máxima é destacada na Eq. (2.6), enquanto a operação mínima aparece na Eq. (2.7). O elemento estruturante g define a forma e a extensão da vizinhança considerada nas operações morfológicas, atuando como um molde. Já $\overline{g}$ representa

o elemento estruturante negado.

O processamento descrito na Eq. (2.6) começa pela negação do elemento estruturante g, seguido pelo uso da operação máxima $\vee$; representada por $f(w) \vee \overline{g}(k-w)$, onde $f(w)$ é a matriz da imagem original, também chamada de região ativa. O valor $\epsilon_g(f)(k)$ obtido em k resulta do mínimo entre os máximos calculados. Esse processo leva ao crescimento das regiões escuras e à redução das zonas claras.

Na Eq. (2.7), o processo de Dilatação envolve primeiro a operação mínima $f(w) \wedge g(k-w)$, onde $f(w)$ representa a matriz de imagem. Após isso, o parâmetro $\delta_g(f)(k)$, na posição k, alcança o máximo entre os valores mínimos, resultando na ampliação das áreas claras e diminuição das escuras.

A Fig. 2.1 (b) e a Fig. 2.1 (c) ilustram os efeitos da Erosão e da Dilatação sobre a mesma imagem original (Fig. 2.1 (a)). Na imagem erodida, o objeto-alvo aparece "encolhido", enquanto na dilatada, ele é expandido. Fazendo uma analogia com as redes mELM, o *kernel* de Dilatação aumenta a área associada à classe alvo (como *malware*), enquanto o de Erosão amplia a região da contra-classe (como benigno).

Figura 2.1: a) Imagem original. b) Imagem erodida. c) Imagem dilatada. Figura obtida da biblioteca gráfica OpenCV.

Fonte: Figura obtida da biblioteca gráfica OpenCV.

2.3 Redes Neurais Morfológicas

Um dos principais desafios em redes neurais artificiais é encontrar o *kernel* ideal para otimizar a separação entre classes. Em ELMs, por exemplo, um *kernel* linear resolve problemas linearmente separáveis (Fig. 2.2 (a)), enquanto *kernels* sigmoide,

RBF e senoide lidam com problemas separáveis por funções correspondentes (Fig. 2.2 (b)-(d)).

A capacidade de generalização de uma rede depende fortemente da escolha correta do *kernel*, e encontrar o melhor ajuste pode ser um processo custoso, envolvendo validação cruzada e múltiplas condições iniciais. Quando o *kernel* não é bem ajustado, a rede tende a apresentar resultados insatisfatórios.

Como exemplo negativo, observa-se que um *kernel* linear aplicado a distribuições sigmoide e senoide (Fig. 2.3 (a)-(b)) produz acurácias de apenas 78,71% e 73,00%, respectivamente, mostrando que ele não modela bem essas fronteiras.

As Fig. 2.4 (a)-(d) mostram o desempenho do mELM com *kernel* Erosão em distribuições linear, sigmoidal, radial e senoide, alcançando acurácias de 100%, 93,07%, 98,18% e 99,50%. Já as Fig. 2.5 (a)-(d) apresentam os resultados do mELM com *kernel* Dilatação, com acurácias de 100%, 95,05%, 98,18% e 99,50%. Esses resultados destacam a capacidade das mELMs de mapear corretamente diferentes problemas, mesmo com os atributos normalizados no mesmo intervalo.

O sucesso dos *kernels* mELMs está na habilidade de modelar qualquer fronteira decisória, já que seu mapeamento não está restrito a superfícies geométricas convencionais, como elipses ou hipérboles. Utilizando as coordenadas no espaço n-dimensional das amostras de treinamento, as mELMs conseguem identificar e modelar com precisão as regiões correspondentes a cada classe, graças à flexibilidade oferecida pela Morfologia Matemática (LIMA et al., 2021).

2.4 Síntese do Capítulo

Este capítulo apresentou a evolução das técnicas de detecção de *malware*, evidenciando as limitações das soluções comerciais baseadas em listas de bloqueio. Como alternativa, destacou-se a análise preventiva, que identifica padrões maliciosos a partir do comportamento do aplicativo. O uso de *machine learning* permite analisar grandes volumes de dados, superando abordagens reativas. Neste contexto, as ELMs surgem como solução rápida e eficiente, especialmente úteis na Perícia Forense Di-

gital. Enquanto redes tradicionais sofrem com longos tempos de treinamento, as ELMs, e especialmente as mELMs, aplicam Morfologia Matemática para otimizar a separação entre classes, empregando operadores como Erosão e Dilatação. Os resultados preliminares indicam que essa abordagem pode fortalecer significativamente a detecção de *malware* e aprimorar a segurança digital em tempo real.

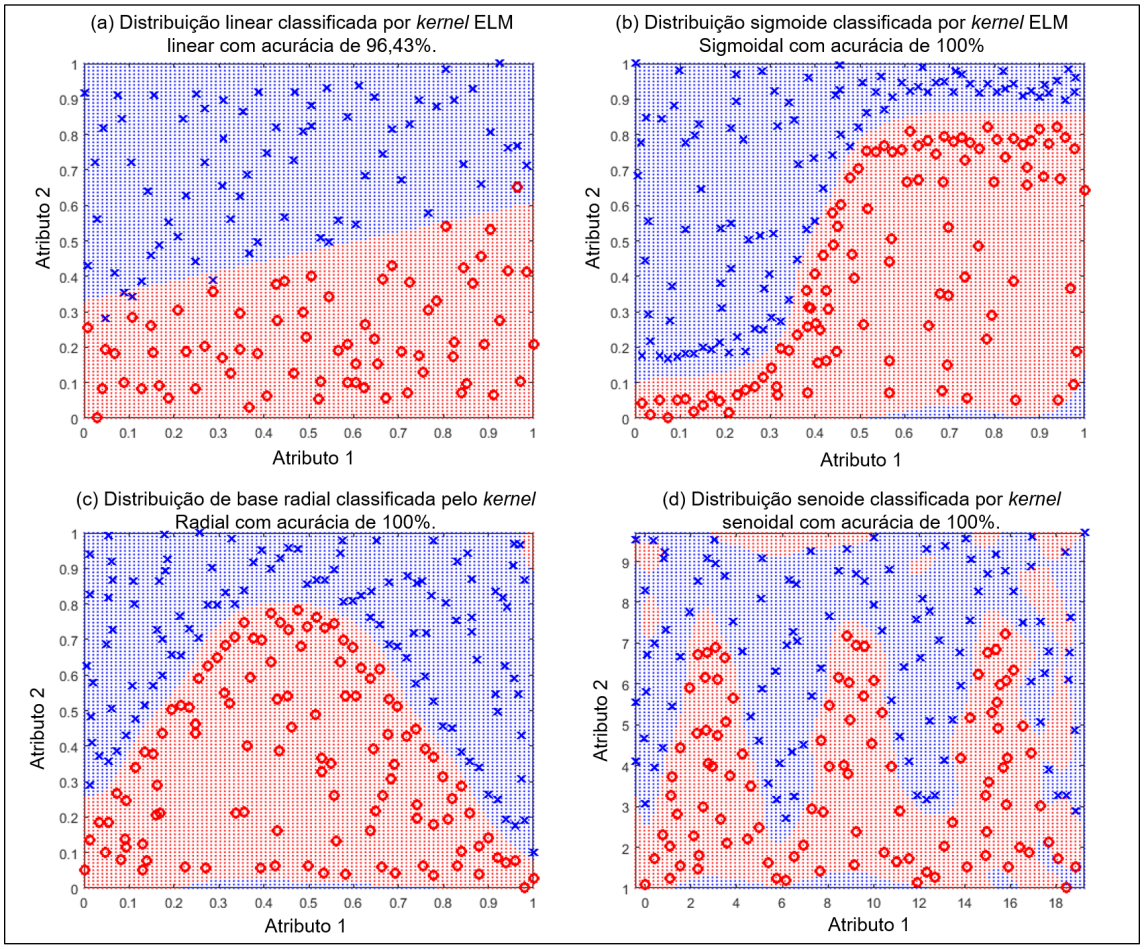
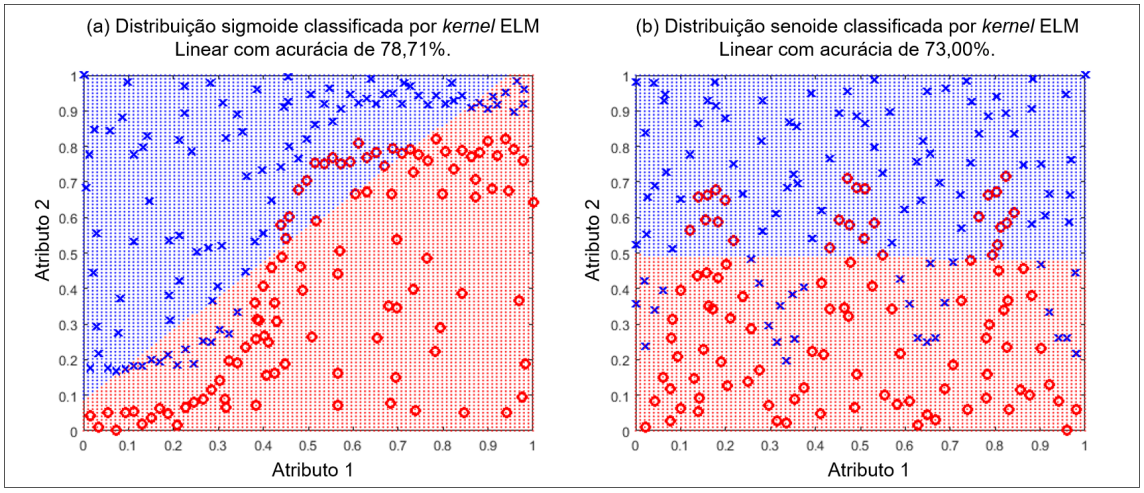


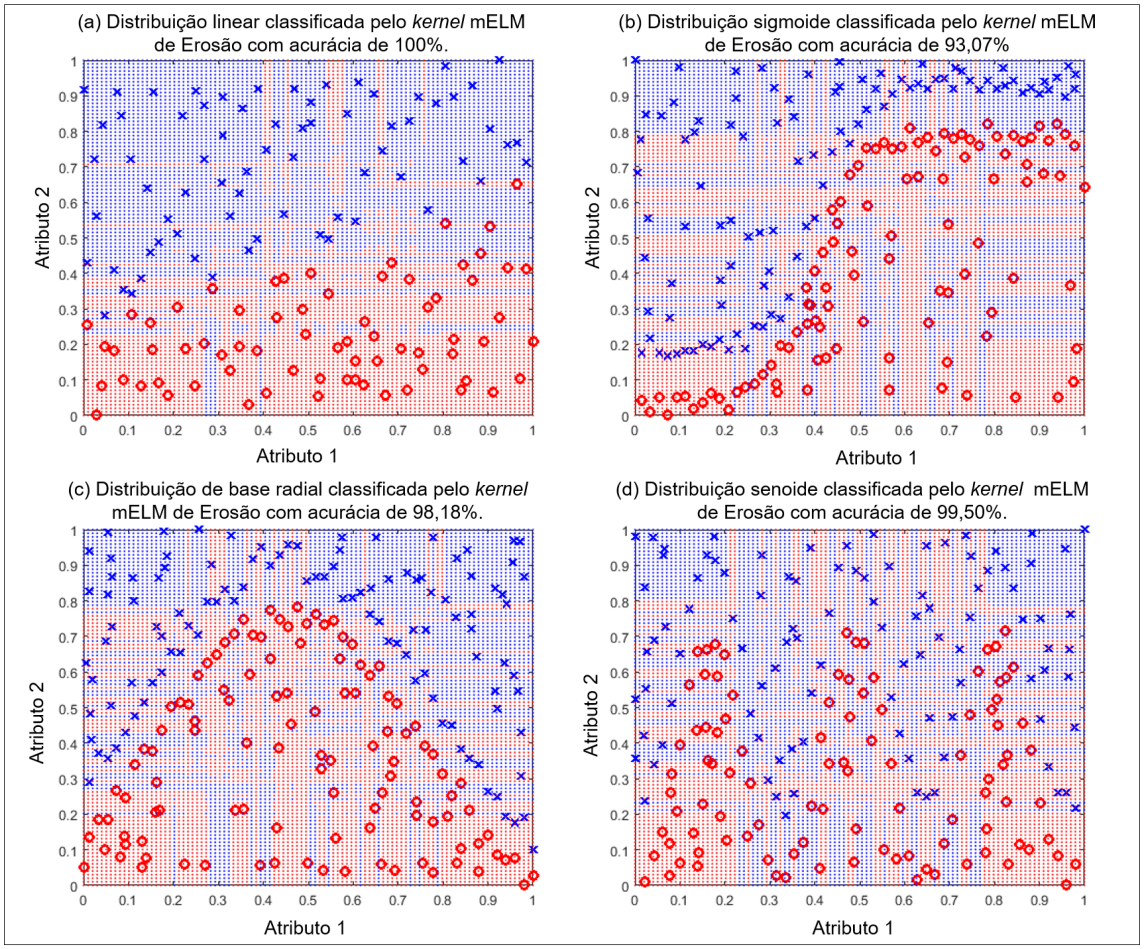
Figura 2.2: Atuações bem-sucedidas dos *kernels* compatíveis com os conjuntos de dados.
Fonte: O autor (2025).

Figura 2.3: Atuações malsucedidas do *kernel* Linear em conjuntos de dados não-linearmente separáveis.



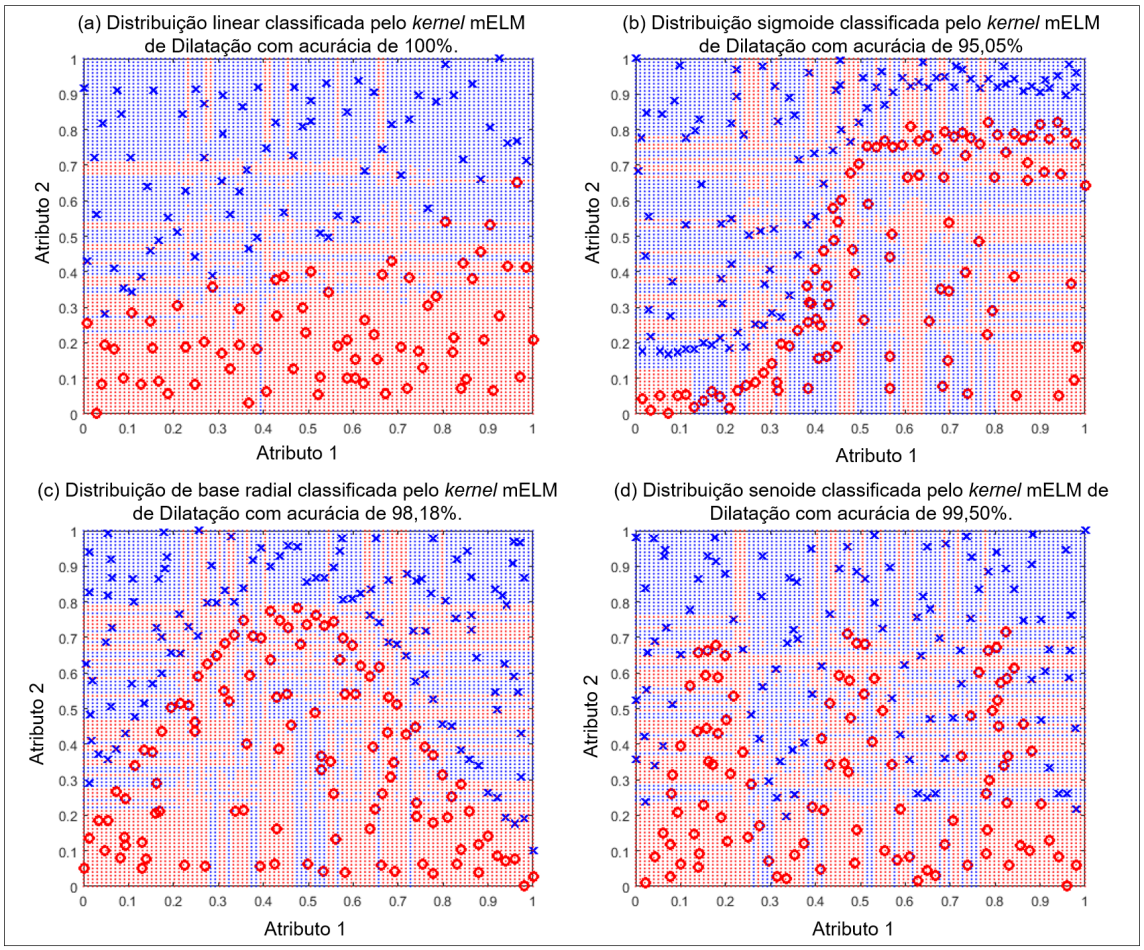
Fonte: O autor (2025).

Figura 2.4: Atuações bem-sucedidas do mELM *kernel* Erosão em diversos conjuntos de dados



Fonte: O autor (2025).

Figura 2.5: Atuações bem-sucedidas do mELM *kernel* Dilatação em diversos conjuntos de dados.



Fonte: O autor (2025).

Capítulo 3

Estado da Arte

Por mais que já seja alvo de críticas há mais de uma década, a forma de ação dos antivírus permanece baseada na comparação de assinaturas de arquivos suspeitos com registros armazenados em bancos de dados conhecidos como listas sujas (LIMA, 2020; SANS, 2017). Em suma, o arquivo analisado é confrontado com os *malwares* previamente registrados nessas listas; então, se o banco de dados não estiver atualizado, o código malicioso pode não ser identificado, ocasionando uma infecção no sistema.

Por outro lado, os antivírus acadêmicos são capazes de detectar a intenção maliciosa mesmo antes da execução pelo usuário. A tabela 3.1 apresenta um panorama do estado da arte, com soluções que alcançam acurácias superiores a 90% na detecção de *malware* em diversos ambientes.

Tabela 3.1: Resumo das principais técnicas para antivírus de última geração.

Autores	Tipo de Rede Neural	Técnica da Rede Neural	Dispositivo alvo	Citações	Acurácia
Antivírus Autoral	Rede Rasa	Rede extrema)	Computador *Desktop*	–	99.87%
FARUKI and BUDDHADEV, 2019	Deep Learning	Rectified Linear Unit (ReLU)	Android	18	98.65%
LIMA et al., 2021	Rede Rasa	*Multi-Layer Perceptron* (MLP)	Computador *Desktop*	24	98.32%
HARDY and LINGWEI, 2016	*Deep Learning*	*Stacked Autoencoders*	Computador *Desktop*	271	96.85%
MANIATH and ASHOK, 2017	*Deep Learning*	*Long-Short Term Memory* (LSTM)	Computador *Desktop*	105	96.67%
HOU and SAAS, 2016	*Deep Learning*	*Deep Belief*	Android *Smartphone*	111	96.66%
SU and VASCONCELLOS, 2018	*Deep Learning*	Training batch (CNN)	IoT for x86	398	94.00%

LIMA et al., 2021 trouxe o desenvolvimento de um antivírus capaz de identificar *malwares* no sistema Windows, alcançando uma acurácia média de 98,32%. No modelo proposto, os arquivos executáveis são submetidos a um processo de desmontagem, possibilitando a análise de seu comportamento malicioso. Na implementação, são extraídas 630 características de cada arquivo executável, que são utilizadas como entradas para a rede neural artificial implementada. A classificação realizada divide os arquivos de 32 *bits* em duas classes: sérios e *malware*. Vale ressaltar que a rede neural, por LIMA et al., 2021, utilizada não é do tipo profunda.

SU, J. *et al.* (2018) Obteve uma precisão média de 94,00% na identificação de *malware* da Internet das Coisas (IoT) (SU and VASCONCELLOS, 2018). A arquitetura de rede profunda possui 6 camadas, sendo que 3 delas contêm pesos apreensíveis: 2 camadas convolucionais e 1 camada totalmente conectada. O treinamento da rede é realizado por meio de 5.000 iterações, utilizando um tamanho de lote de 32 e uma taxa de aprendizado de 0,0001.

O antivírus feito por HOU, S. *et al.* (2016) detectou *malware* Android empregando uma rede de crenças profundas(*deep belief*) HOU and SAAS 2016 e construído a partir de um conjunto de Máquinas de Boltzmann Restritas (RBM), que constituem a profunda crença rede, onde as ativações treinadas de um RBM são utilizadas como entradas para o próximo RBM. A estrutura da rede é composta por 3 camadas, cada uma com 200 nós ocultos. HOU, S. *et al.* (2016) obteve uma acurácia média de 96,66%.

O antivírus feito por HARDY, W. *et al.* (2016) detecta *malware* de arquivo PE (Windows) empregando redes profundas de autoencoder empilhado HARDY and LINGWEI 2016. O decodificador busca reverter essa representação para a entrada original. O modelo de aprendizado profundo é treinado com 3 camadas ocultas, sendo que cada camada oculta possui 100 neurônios. O codificador realiza o mapeamento da entrada para uma representação oculta. HARDY, W. *et al.* (2016) obteve uma acurácia média de 96,85%.

3.1 Redes neurais rasas *versus* Redes neurais profundas

Indubitavelmente, modelos de aprendizado aprofundado (*DL*) apresentam excepcionais habilidades de generalização. Em virtude desses extraordinários valores atingidos mediante o uso das técnicas de aprendizagem profunda, desenvolveu-se uma ideia predominante segundo a qual a tecnologia de aprendizado profundo é capaz de oferecer acurácia mais elevada em diversos tipos de aplicações. Quando comparados a modelos de sistema neural superficial, as redes profundas têm a possibilidade de detectar uma quantidade consideravelmente mais alta de classes (LIMA, 2020).

O modelo *Inception-V3*, a título de exemplificação, é capaz de operar com 1.000 neurônios em sua última camada, permitindo-lhe reconhecer a mesma quantidade de elementos. Dessa forma, ao ser apresentado a um novo item, o *Inception-V3* é capaz de reconhecer e classificar perfeitamente os elementos apresentados nele (CHOLLET, 2017).

Inúmeros *softwares* antivírus que conseguem identificar centenas de classes de ameaças se baseiam nos modelos DL com múltiplos neurônios no seu nível de saída. Adicionalmente, a rede profunda consegue fazer o reconhecimento de uma determinada amostra, baseando-se em uma classe pré-existente (por exemplo, Zeus, *Citadel*, *Ransomware*, etc.). Devido a essa capacidade de generalização volumosa, os modelos de aprendizagem profunda são capazes de obter elevadas acurácias em classificações de *softwares* mal-intencionados (LIMA, 2020)..

Quando comparado a um sistema de antivírus dedicado, um programa antivírus de propósito genérico e baseado em aprendizagem profunda pode se sobressair ao classificar diferentes tipos de ameaças, mas ainda assim pode falhar em detectar *malwares* específicos dentro de uma categoria particular. O último, elaborado especificamente tendo como alvo um determinado grupo de ameaças, pode demonstrar um melhor resultado em relação aos antivírus de uso geral, conseguindo níveis de

desempenho superiores, mas somente na identificação de uma classe específica de *malware*.

Dentro desse contexto, é possível que o aprendizado profundo não represente a solução mais viável. A fim de exemplificar essa situação, consideremos o caso da rede neural profunda *Inception*-V3. Essa rede é complexa, contendo 23,6 milhões de parâmetros reguláveis. Tais parâmetros necessitam de um conjunto de dados amplo que permita o pleno treinamento do modelo. Em outras palavras, um modelo de aprendizagem profunda carece muito de dados. Uma grande quantidade de parâmetros ajustáveis exige a utilização de inúmeros valores de entrada. No setor de segurança cibernética, é preciso ressaltar que não é todo tipo de ameaça que apresenta amostras compatíveis com uma arquitetura profunda. Em algumas categorias de *malware*, as amostras podem ser escassas.

O aprendizado profundo trabalha melhor com um grande volume de dados de treinamento. Dessa forma, a rede adquire experiência em lidar com cenários distintos. Porém, caso o modelo só possua milhares, em vez de milhões, de exemplares dentro de seu pacote de dados, a capacidade do aprendizado profundo de generalizar pode ser prejudicada.

Implementar um antivírus com *Deep Learning* pode parecer uma opção atrativa, mas é essencial considerar alguns fatores. A aprendizagem profunda exige grandes quantidades de dados para funcionar de maneira eficaz, pois depende significativamente do volume e da qualidade desses dados. No contexto da cibersegurança, onde os cenários podem ser complexos e imprevisíveis, essa dependência de dados extensos pode se tornar uma limitação, especialmente se os dados disponíveis forem insuficientes ou inadequados para o treinamento do modelo. Portanto, é crucial avaliar se essa abordagem é a mais adequada para o ambiente em questão. Caso não tenhamos disponível uma vasta quantidade de dados, o desempenho da aprendizagem profunda pode ser prejudicado.

A aprendizagem profunda se mostra extremamente efetiva com uma grande quantidade de dados. Todavia, no caso de um *software* antivírus dedicado a ameaças

determinadas, é possível que existam outras metodologias mais eficazes. Por exemplo, a aprendizagem de baixa complexidade supervisionada baseada em pequenos conjuntos de dados direcionados pode resultar em mais eficiência. Essas técnicas possibilitam uma melhor abordagem para desenvolver soluções de cibersegurança.

No presente trabalho, advogamos a favor do desenvolvimento de múltiplos programas antivírus dedicados e de baixa complexidade computacional. O intuito é alcançar a maior acurácia possível sem desconsiderar as diferentes categorias de *malware*. Os antivírus especializados podem também ser vistos como uma maneira de impedir o elevado desperdício de tempo de treinamento em modelos imensos de *deep leanings*. Nos antivírus dedicados, várias unidades de processamento funcionam de forma independente e permitem que sejam treinadas em paralelo, diminuindo ainda mais o período de treino da solução geral.

O longo período de aprendizado é uma desvantagem ao empregar as redes de aprendizagem profunda. Levando em consideração que em média oito novas instâncias de *malware* surgem a cada segundo (Intel, 2018), é nítido que existe uma forte demanda de retreinamento constante de modelos de redes neurais de antivírus. Isso se torna uma barreira significativa. Em termos ideais, o período requerido no treinamento do *software* antivírus deveria alinhar-se ao ritmo de surgimento de novas versões de *malware* em escala global.

Esse tempo de treinamento excessivo das *Deep Learning* pode ser estendido ainda mais pelo fato delas oferecerem o desafio de serem treinadas paralelamente em função de sua organização sequencial de camadas. Portanto, a operação da camada seguinte só pode ser iniciada depois da conclusão da camada anterior.

Adicionalmente, um outro problema relevante das redes profundas é o fato delas apresentarem diversas definições passíveis de configuração. É o caso, por exemplo, das redes profundas do tipo *Inception*-V3, as quais apresentam 23,6 milhões dessas configurações reguláveis ao longo de 48 camadas sequenciais (CHOLLET, 2017). Com esse volume de processamento sequencial, é inviável obter a máxima otimização dos períodos de treinamento sequer com a utilização de um supercomputador teó-

rico, que possua milhões de processadores. É que o princípio produtor-consumidor inviabiliza o trabalho simultâneo em todas as camadas. O sistema sequencial em cascata de redes profundas impõe um considerável obstáculo no que se refere ao paralelismo computacional. Em um processamento sequencial, como o próprio termo indica, só é possível iniciar a operação de uma camada, uma vez que a camada precedente esteja concluída.

Todavia já é possível encontrar modelos de *Deep Learning* na fase inicial que são aptos para processar informações de forma paralela. No momento, a acurácia desses modelos é insuficiente para algumas aplicações existentes (PINHEIRO et al., 2022). Na prática, os resultados obtidos pelo treinamento de tais redes seguem sendo estatisticamente insatisfatórios em contraste aos modelos profundos sequenciais (PINHEIRO et al., 2022). Consequentemente, não existem provas suficientes que indiquem a possibilidade do uso de redes paralelas profundas para alcançar valores aceitáveis, quando aplicadas a um antivírus. Embora o processo de treinamento seja demorado, o sistema baseado em aprendizagem profunda permite obter altas acurácias.

Redes profundas consistem em sistemas de computação que envolvem a utilização de grafos de profundidade. Tradicionalmente, a construção da estrutura se dá a partir de múltiplos níveis consecutivos. As mais avançadas arquiteturas de redes profundas, adotam níveis contendo diversos perfis de processamento. Em geral, a discriminação existente com relação às camadas, envolve funções de ativação, de normalização, de convolução e de redução de dimensionalidade [1].

Do ponto de vista matemático, no que se refere às redes neurais profundas, sobretudo as redes convolutivas, essas têm como base a convolução de filtros lineares. Esse filtro, por mais que exerça uma função essencial em aplicações computacionais, restringe-se a aplicativos nos quais é estabelecido um gradiente de fluxo de vetores.

A título de ilustração, ao analisarmos cuidadosamente algumas imagens da área

[1] Exemplo de arquitetura de rede neural profunda. Disponível em: https://se.mathworks.com/help/deeplearning/gs/create-simple-image-classification-network-using-deep-network-designer.html. Acessado em setembro de 2024.

biomédica de aparelhos de mamografia, é evidente o fato de que estão carregadas de traços ruidosos, de forma que se torna extremamente complexa a percepção de lesões na mama. Dessa forma, a convolução desses filtros é crucial a fim de que seja possível retirar o ruído. Assim, é possível descartar as pequenas inconsistências no diagnóstico que correspondem aos achados, possivelmente câncer. De modo a minimizar o fenômeno ruidoso nas imagens biomédicas, técnicas de convolução tais como o uso de filtros Gaussianos se fazem imprescindíveis (LIMA et al., 2020).

De forma oposta ao exemplo, observe o exemplo hipotético apresentado na Tabela tab:3. A despeito de fazerem parte dos mesmos círculos de vizinhança, tais elementos não estão interligados. No caso de aplicativos suspeitos que fazem a varredura em dados relacionados a *Wi-Fi*, não há conexão direta ao uso de dados de navegação ou ao acesso ao banco de imagens do usuário. Daí, ao convoluir linearmente os filtros, o acesso ao *browser* passaria a ser encarado como ruído, simplesmente pelo fato de os arredores possuírem detecção igual a 1. Em suma, o programa suspeito poderia vir a ser incriminado da invasão ao mecanismo de navegação da vítima. Sendo assim, técnicas de convolução apresentam um aspecto negativo em sua aplicação à detecção de padrões de *software* mal-intencionados.

Tabela 3.2: Exemplo hipotético de repositório de estatísticas baseado na detecção de atividades maliciosas.

Funções		
Checagem de dados de *Wi-fi*	Acesso ao *Browser*	Acesso a galeria de Imagem
1	0	1

Com o intuito de fundamentar a estrutura teórica proposta, projetamos uma solução antivírus dedicada que é capaz de identificar o *malware Ransomware*. O resultado servirá de alicerce para a criação de antivírus especialistas em determinadas classes de *malware*, com o uso de nosso próprio padrão de redes neurais.

No antivírus de nossa autoria, são utilizadas redes superficiais ao invés de redes profundas convolucionais. Na forma de experimento, nosso programa antivírus é capaz de conciliar uma acurácia elevada a um período de treinamento reduzido.

Seu nível de eficiência também foi comparado com o de outros antivírus modernos baseados no uso de redes neurais superficiais e profundas. De modo a prevenir a ocorrência de comparativos equivocados, o estágio de extração de dados é normalizado mediante acompanhamento dos 407 processos que o arquivo sob suspeita pode desempenhar. O *software* antivírus criado pelo autor demora pouco mais de 1 segundo na conclusão de sua aprendizagem. No capítulo 5 são exibidos resultados da replicação do antivírus de ponta e de nosso *software* antivírus.

3.2 Síntese do Capítulo

O capítulo "Estado da Arte" analisa a evolução dos sistemas antivírus, criticando a metodologia tradicional baseada em listas de bloqueio, que depende de atualizações constantes para identificar *malwares*, e destacando a transição para abordagens modernas que utilizam redes neurais. A Tabela 3.1 compara antivírus de última geração, como o de LIMA et al. (2021), que usa redes neurais rasas para detectar *malwares* em sistemas *Windows* com 98,32% de acurácia, e o de SU and VASCONCELLOS (2018), que emprega redes profundas para identificar *malwares* em dispositivos IoT, alcançando 94% de acurácia com a metodologia proposta que atinge 99,99%. Redes profundas, como o modelo $Inception\text{-}V$, exigem grandes volumes de dados e recursos computacionais, além de tempo prolongado de treinamento, o que é problemático diante da rápida evolução das ameaças. Em contraste, redes rasas podem ser mais eficientes em recursos e tempo, porém podem depender muito da configuração, fazendo com que as ELM morfológicas se sobressaiam em detrimento das redes neurais genéricas. O capítulo propõe antivírus especializados baseados em redes ELM morfológicas, como o desenvolvido pelos autores, que identifica o *malware Ransomware* com alta acurácia e rapidez, demonstrando que soluções dedicadas podem ser mais viáveis do que abordagens genéricas, baseadas em redes profundas e redes rasas convencionais.

Capítulo 4

Metodologia

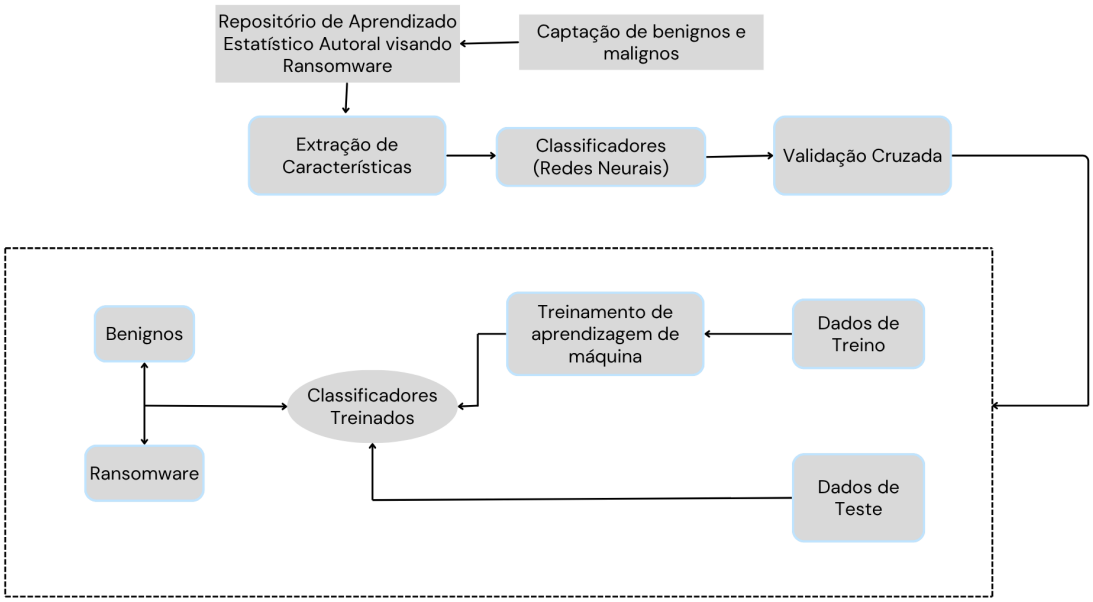
Após identificar as limitações observadas nos antivírus comerciais, esta pesquisa busca desenvolver uma ferramenta, baseada em inteligência artificial, capaz de distinguir de forma preventiva aplicações maliciosas de benignas. A Fig. 4.1 ilustra o diagrama em blocos que representa a metodologia proposta.

O diagrama apresentado divide a metodologia em etapas lógicas. Inicialmente, ocorre a fase de aquisição de dados, a partir de um repositório de aprendizado estatístico autoral (ransomware, 2025). Em seguida, os recursos extraídos passam por uma etapa de pré-processamento para viabilizar que estejam adequados para análise. Posteriormente, os dados são classificados utilizando a técnica proposta (Máquina de Aprendizagem Extrema — ELM), e por fim é aplicada a validação cruzada, com o objetivo de viabilizar a eficácia do processo desenvolvido. O diagrama também descreve o fluxo das etapas de treinamento e teste das amostras.

4.1 Materiais e Métodos Utilizados

Com relação ao material empregado, esta pesquisa apresenta um banco de dados autoral, voltado à classificação de executáveis benignos e *malwares* do tipo *ransomware* (ransomware, 2025). Foram incluídas 1.174 amostras maliciosas e 1.174 executáveis benignos, garantindo um conjunto de dados balanceado e apropriado para técnicas de aprendizado de máquina, já que ambas as classes possuem a mesma

Figura 4.1: Diagrama da metodologia proposta.



Fonte: O autor (2025).

quantidade de elementos.

As amostras de código malicioso foram coletadas de repositórios especializados, como o *VirusShare*, uma plataforma que disponibiliza amostras de *malware* para pesquisadores em segurança, analistas forenses e interessados na área. Por outro lado, os executáveis benignos foram obtidos de repositórios confiáveis, como *SourceForge*, *GitHub* e *Sysinternals*. Importante destacar que todos os arquivos benignos foram submetidos ao serviço VirusTotal, tendo sua integridade e ausência de código malicioso confirmadas pelos principais antivírus comerciais. Os diagnósticos gerados pelo VirusTotal, referentes às amostras benignas e maliciosas, estão acessíveis no endereço virtual do banco de dados (ransomware, 2025).

A criação desse banco de dados tem como objetivo permitir que a metodologia proposta seja replicada por outros pesquisadores em investigações futuras. Assim, ao disponibilizar gratuitamente os dados utilizados, o trabalho assegura transparência, imparcialidade e reforça a credibilidade dos resultados apresentados. Espera-se que a abordagem adotada sirva de referência para o desenvolvimento de novos estudos científicos.

4.2 Ambiente Experimental

Em relação ao ambiente computacional utilizado para o processamento dos dados, todos os experimentos foram conduzidos em uma máquina equipada com 32 GB de RAM (*random access memory*). O processador utilizado foi um Intel(R) Xeon(R) CPU E5-2680 v4 @ 2,40 GHz. A máquina também contava com uma placa de vídeo GeForce GT610 de 2 GB e um total de 16 núcleos físicos e 16 núcleos lógicos. Porém foram utilizados apenas 10 núcleos, pois o método de avaliação adotado foi o *k-Fold* com $k = 10$. Por isso, optou-se por alinhar o número de núcleos à quantidade de partições definidas pelo método.

Cabe enfatizar que o antivírus autoral apresenta baixa complexidade computacional, o que possibilita sua execução eficiente mesmo em computadores convencionais, sem a necessidade de *hardware* especializado. Essa característica o torna acessível e viável para ampla adoção, especialmente em ambientes com recursos limitados. Ressalta-se, contudo, que a aquisição de um servidor durante os experimentos se fez necessária, não por exigência do antivírus autoral em si, mas devido à demanda computacional elevada dos métodos concorrentes, utilizados para fins de comparação. Em particular, os antivírus do estado da arte baseados em *Deep Learning* requerem infraestrutura mais robusta, com maior capacidade de processamento e memória. Justifica-se, assim, o uso de um ambiente de alto desempenho para viabilizar que os antivírus concorrentes do estado-da-arte pudessem ser replicados.

4.3 Extração de recursos

A extração de características dos executáveis é um passo essencial para a análise dos arquivos binários. Esse processo começa com a etapa de engenharia reversa, que visa converter o arquivo binário de volta para seu código de montagem. Após essa conversão, o algoritmo que descreve o comportamento do executável pode ser examinado e posteriormente classificado com o uso de redes neurais artificiais, conforme detalhado na seção seguinte.

A partir dos executáveis analisados, são extraídas 649 características distribuídas entre os diversos grupos de interesse. Essas características são extraídas utilizando *scripts* personalizados e a ferramenta *pescanner*, que facilita a coleta dos dados necessários. As principais características extraídas podem ser divididas nos seguintes grupos:

- Histograma das instruções responsáveis pela aquisição de dados (*imports*): este grupo de características envolve a análise das instruções do executável que fazem chamadas para funções de importação de dados.
- Quantidade de sub-rotinas que invocam o TLS (*Transport Layer Security*): este item avalia a quantidade de sub-rotinas dentro do executável, que interagem com o protocolo de segurança TLS.
- Quantidade de sub-rotinas responsáveis pela exportação de dados (*exports*): são contabilizadas as sub-rotinas que lidam com a exportação de dados para outros sistemas ou aplicações.
- Histograma das importações das API (Interface de Programação de Aplicações - API): este histograma captura as diferentes funções de API que o executável utiliza, fornecendo informações sobre as interações do *software* com o sistema operacional e outros componentes externos.
- Características relacionadas à fragmentação de disco e inicializações inválidas acumuladas: essas características são indicativas de possíveis falhas ou tentativas de ataque relacionadas à integridade do sistema de arquivos.
- Modo de execução do aplicativo: O modo de execução é classificado em duas categorias:
  - *Softwares* com interface gráfica (GUI);
  - *Softwares* que operam diretamente no console;

- Características relacionadas ao Sistema Operacional: aqui, o antivírus busca identificar se o arquivo suspeito tenta interagir com o sistema operacional da seguinte maneira:
  - Detectar o nome do usuário atual no sistema operacional;
  - Acessar funções de API para gerenciar perfis de usuários no sistema;
  - Determinar o tempo de inicialização do sistema, medido em milissegundos desde o *boot*;
  - Realizar operações em arquivos específicos;
  - Identificar a versão do sistema operacional Windows em uso;
  - Monitorar o tráfego de mensagens entre processos;
  - Alterar configurações de inicialização do sistema operacional Windows;
  - Permitir que aplicativos acessem e alterem a funcionalidade fornecida pelo *shell* do sistema operacional;
  - Modificar as mensagens de *logon* na inicialização do Windows.

Além das características relacionadas ao sistema operacional, outras informações são extraídas a respeito de interações específicas com os componentes do sistema, como a alteração de configurações de energia, manipulação de arquivos e criação de novos processos e diretórios.

- Configurações de energia e manipulação de arquivos: o antivírus busca identificar ações como a alteração das configurações de energia do sistema, a cópia de arquivos para novas localizações e a criação, abertura, exclusão ou alteração de arquivos.
- Criação e execução de processos: a identificação de criação e execução de novos processos é uma característica importante, já que *malwares* frequentemente geram novos processos para realizar suas atividades.

- Criação de diretórios e busca por arquivos específicos: características que envolvem a criação de novos diretórios ou a busca por arquivos específicos também são analisadas.

- Operações com objetos de serviço: algumas operações que envolvem a criação de objetos de serviço e sua adição ao banco de dados de gerenciamento do sistema operacional são monitoradas.

- Criptografia de dados: características associadas à criptografia de dados são especialmente relevantes, pois são típicas de *ransomwares*. Tais *malwares* sequestram os dados da vítima, criptografando-os e exigindo um resgate para que os dados sejam devolvidos.

- Acesso ao sistema de arquivos e dispositivos: este grupo de características refere-se a interações com o sistema de arquivos, dispositivos, processos, *threads* e o tratamento de erros.

- Ajustes em som e dispositivos de áudio: o antivírus verifica se o executável tenta alterar as propriedades de som ou interagir com dispositivos de áudio do sistema.

- Acesso a informações gráficas: o antivírus também monitora interações do executável com sistemas gráficos, como impressoras, monitores e outros dispositivos de saída.

- Uso da porta USB e controle de *drivers*: o acesso e monitoramento da porta USB, bem como o controle sobre *drivers* de dispositivos, também são características investigadas.

- Investigações sobre unidades de disco: o antivírus verifica se o executável tenta determinar se uma unidade de disco é removível, fixa, de CD/DVD-ROM, RAM ou de rede.

Além disso, o antivírus também monitora interações com o Registro do Sistema Operacional (Regedit), uma área crítica que pode ser usada para persistência mali-

ciosa. Mesmo após a remoção de um *malware*, ele pode deixar entradas maliciosas no registro, que são usadas para reiniciar o ataque quando o sistema é inicializado. O antivírus analisa tentativas do executável de:

- Detectar o nome NetBIOS do computador local;
- Criar, excluir ou modificar entradas no Regedit;
- Encerar ou abrir chaves específicas do registro.

Por fim, são analisadas características relacionadas a *spywares*, como *keyloggers* e *screenloggers*. Estes *malwares* visam capturar informações confidenciais, como senhas e transações bancárias, através da monitoração das entradas do teclado ou da filmagem da tela da vítima. O antivírus audita tentativas do executável de realizar as seguintes ações:

- Detectar áreas da tela que foram atualizadas;
- Capturar informações sobre votação eletrônica;
- Monitorar a atividade da Internet e capturar dados sensíveis como senhas bancárias.

Essas características são essenciais para identificar atividades maliciosas e prevenir ataques cibernéticos. A extração e análise detalhada dessas informações permite que o sistema de segurança seja mais eficiente na detecção e neutralização de ameaças.

4.4 Classificação

No que diz respeito à identificação de padrões de *malware*, a atividade central consiste em atribuir a cada arquivo analisado uma categoria (ou rótulo) com base em suas propriedades. A partir de um conjunto de dados denominado conjunto de treinamento, é possível formular suposições sobre as diferentes categorias que o

sistema antivírus desenvolvido deve reconhecer. Com isso, o classificador realiza a predição da categoria de arquivos ainda não vistos, por meio da comparação entre comportamentos observados em tempo real e os registrados durante a fase de treinamento.

O propósito do classificador é definir uma função que discrimine entre as diferentes categorias do antivírus (malicioso, legítimo). Assim, quando um novo executável é submetido à análise, essa função é utilizada para determinar a qual grupo o arquivo provavelmente pertence. Em termos matemáticos, temos $c = f(x)$, onde $x = x_1, x_2, \ldots, x_t$ representa o vetor de características extraídas do arquivo examinado, t é a quantidade de atributos dinâmicos observados, c indica a classe prevista, e f corresponde à função de mapeamento definida pelo classificador.

Ao utilizar um classificador linear, assume-se que há uma linha ou fronteira que separa os padrões correspondentes a diferentes classes. Dessa forma, cada exemplo será rotulado com base na posição em que se encontra em relação a essa divisão. No entanto, para a detecção comportamental de ameaças modernas, é necessário empregar classificadores capazes de estabelecer uma fronteira não linear entre as categorias. Para garantir fundamentação teórica, o antivírus proposto adota redes neurais de natureza não linear, mais precisamente, redes neurais pseudo-morfológicas extremas.

4.5 Síntese do Capítulo

O capítulo descreveu o desenvolvimento do antivírus baseado em *Extreme Learning Machine* (ELM), utilizando um agrupamentos de dados com 1.174 amostras de *malware* (*ransomware*) e 1.,174 programas benignos, coletados de fontes como *VirusShare, SourceForge e GitHub*, e verificados no VirusTotal. Foram extraídas 649 características dos arquivos, como distribuição de instruções de importação, uso de TLS, comportamentos vinculados ao sistema operacional, técnicas de antiforense, tráfego de rede e modificações em aplicativos, para identificar padrões maliciosos. O classificador, baseado em redes neurais morfológicas extremas, gera uma função

separadora não linear para distinguir entre arquivos benignos e maliciosos, permitindo alta eficiência e baixo desvio padrão. A abordagem proposta visa superar as restrições dos antivírus tradicionais, oferecendo uma solução eficiente e adaptável para a detecção de *malware* modernos.

Capítulo 5

Resultados

O presente capítulo detalha o desempenho do antivírus autoral. É realizada uma comparação com soluções baseadas em aprendizado de máquina, ciência dos dados e redes neurais artificiais. Foram avaliados oito tipos diferentes de *kernels* em redes neurais extremas. Destaca-se o *kernel* de Dilatação, que obteve a maior acurácia (99,87% nos testes) e um tempo de treinamento reduzido (3,75 segundos). De tal maneira que o antivírus proposto adotou o *kernel* de Dilatação em seu modelo, para reconhecimento de padrão dos *ransomware*.

5.1 Resultados das redes ELM

Foram avaliados oito modelos diferentes de *kernels* em redes neurais. No contexto da literatura, seis desses *kernels* já são abordados por HUANG, 2012: *Wavelets*, *Linear*, *Sigmoide*, Seno, *Hard Limit* e *Tribas* (estas últimas funções fundamentadas em operações trigonométricas) (HUANG, 2012). Além desses, também foram explorados *kernels* inéditos, criados de forma autoral: Erosão e Dilatação.

O *kernel* do tipo *Wavelets* é caracterizado pela ausência de camadas ocultas (HUANG, 2012). Seu funcionamento baseia-se na conversão dos dados de entrada, podendo funcionar de maneira similar a *kernels* que integram camadas escondidas em suas estruturas (HUANG, 2012). Para que esses modelos alcancem uma boa capacidade de generalização, é essencial que os parâmetros (C, γ) sejam ajustados

de forma adequada (HUANG, 2012). O parâmetro de custo C se refere ao controle da largura da margem do hiperplano, buscando minimizar o erro de classificação durante o treinamento. Por sua vez, γ atua nos limites da concavidade da função geométrica empregada na separação das classes (HUANG, 2012). A definição desses valores de (C, γ) não segue uma regra universal.

Neste estudo, os parâmetros (C, γ) foram explorados conforme a abordagem de HUANG, 2012, que propõe uma varredura progressiva de valores para C e γ na sequência matemática 2^n, onde $n = \left\{-24, -10, 0, 10, 25\right\}$ (HUANG, 2012). O intuito dessa análise é investigar em que medida a escolha de parâmetros fora dos padrões usuais $(C = 1, \gamma = 1)$ pode resultar em desempenhos superiores. Para o *kernel* Linear, apenas o parâmetro de custo C é considerado, sendo γ desconsiderado neste caso (HUANG, 2012).

O conjunto experimental é formado por 1.174 amostras de *ransomware* e 1.174 arquivos benignos. Adotou-se a validação cruzada pelo método *k-fold*, fixando *k=10*. Tal procedimento visa permitir que os resultados obtidos não sejam influenciados por partições específicas dos dados de teste ou treino. Para tanto, o conjunto total de dados é dividido em dez partes iguais.

Inicialmente, uma dessas partes é reservada para teste, enquanto as demais compõem o grupo de treinamento, com essa rotação se repetindo ao longo de dez execuções. O objetivo é que todas as partições assumam a função de teste ao menos uma vez. O índice de acurácia das redes extremas é calculado a partir da média aritmética dos sucessos verificados em todas as execuções.

Nas redes extremas, não há retropropagação dos dados. Portanto o uso da validação cruzada *k-fold* não tem como objetivo evitar *overfitting*, mas sim identificar possíveis oscilações nos resultados de acurácia, conforme as divisões dos grupos de treinamento e teste. Dessa forma, o objetivo é evitar que o classificador apresente viés em relação a alguma classe específica.

Em cada linha das tabelas deste capítulo, são reportados os resultados de 10 execuções relativas à metodologia *k-fold*. A Tabela 5.1 apresenta os resultados da

rede extrema com *kernels* do tipo *Wavelets*. Nos testes, o *kernel Wavelets* atingiu desempenho médio máximo de 67.62% com parâmetros $(C, \gamma) = (2^{25}, 2^{0})$.

A Tabela 5.2 mostra os dados do desempenho da ELM usando o *kernel* Linear. Nessa configuração, apenas o parâmetro C é avaliado, já que γ não é aplicável a este tipo (HUANG, 2012). Não cabe estudo de concavidade em uma reta. As taxas de acurácia máxima e mínima são, respectivamente, 99.83% e 99.32%. Isso indica que o ajuste do custo C pode otimizar a performance do classificador linear.

A Tabela 5.3 exibe os resultados da rede ELM com diferentes *kernels*, entre eles *Sigmoid*, *Sine*, *Hard Limit*, *Tribas* (base trigonométrica), Dilatação e Erosão, todos empregados em arquitetura com camada escondida. O objetivo da análise foi variar a quantidade de neurônios ocultos para verificar se o aumento da complexidade computacional (por exemplo, quintuplicando o número de neurônios) impacta a acurácia.

Foram testadas duas configurações principais: uma com 100 e outra com 500 neurônios na camada oculta, ambas historicamente eficazes, especialmente em aplicações de ELM em Engenharia Biomédica (LIMA et al., 2020). A maior acurácia média foi de 99,87%, com desvio padrão de 0,20%. Esse resultado ocorreu com o *kernel* de Dilatação dotado de 500 neurônios artificiais ocultos. Apesar do maior custo computacional, leia-se tempo de aprendizado, essa configuração apresentou melhor desempenho entre as arquiteturas ELM. O uso de *kernels* morfológicos pode, portanto, gerar ganhos de acurácia, como apresentado na presente pesquisa.

Tabela 5.1: Valores obtidos de sistemas neurais ELM. A variação dos parâmetros (C, γ) é determinada conforme o conjunto $\{2^{-24}, 2^{-10}, 2^{0}, 2^{10}, 2^{25}\}$. Estão expostas apenas as melhores e piores acurácias.

kernel	(C, γ)	Acurácia de treinamento (%)	Acurácia de teste (%)	Tempo de treino (seg.)	Tempo de teste (seg.)
Wavelets	$(2^{25}, 2^{0})$	**100,00 ± 0,00**	67,62 ± 2,61	1,51 ± 0,04	0,09 ± 0,02
	$(2^{-10}, 2^{25})$	50,56 ± 0,76	50,06 ± 3,22	0,81 ± 0,20	0,10 ± 0,02

Fonte: O autor (2025).

Tabela 5.2: Valores obtidos de sistemas neurais ELM de núcleo linear. A variação de parâmetros de C depende da definição do conjunto $\{2^{-24}, 2^{-10}, 2^{0}, 2^{10}, 2^{25}\}$. Estão sendo exibidas apenas a melhor e a pior acurácia.

kernel	C	Acurácia de treinamento (%)	Acurácia de teste (%)	Tempo de treino (seg.)	Tempo de teste (seg.)
Linear	2^0	**100,00 ± 0,00**	**99,83 ± 0,22**	0,39 ± 0,03	0,03 ± 0,01
	2^{10}	100,00 ± 0,00	99,32 ± 0,41	0,46 ± 0,13	0,04 ± 0,02

Fonte: O autor (2025).

Tabela 5.3: Valores obtidos do sistema ELM. A quantidade de neurônios no nível oculto muda conforme os dados 100, 500.

kernel	neurônios	Acurácia de treinamento (%)	Acurácia de teste (%)	Tempo de treino (seg.)	Tempo de teste (seg.)
Sigmoide	500	74,25 ± 0,21	69,29 ± 2,05	0,39 ± 0,05	0,01 ± 0,01
	100	63,92 ± 0,16	62,66 ± 1,83	0,08 ± 0,01	0,00 ± 0,00
Seno	500	82,17 ± 0,43	66,56 ± 2,99	0,44 ± 0,07	0,01 ± 0,01
	100	65,59 ± 0,71	60,85 ± 3,11	0,10 ± 0,03	0,00 ± 0,00
Hard limit	100	50,00 ± 0,00	50,00 ± 0,00	0,11 ± 0,03	0,00 ± 0,00
	500	50,00 ± 0,00	50,00 ± 0,00	50,00 ± 0,00	0,01 ± 0,01
Tribas	100	50,00 ± 0,00	50,00 ± 0,00	0,10 ± 0,02	0,00 ± 0,00
	500	50,00 ± 0,00	50,00 ± 0,00	0,35 ± 0,09	0,01 ± 0,01
Dilatação	500	**100,00 ± 0,00**	**99,87 ± 0,20**	3,75 ± 0,27	0,32 ± 0,01
	100	99,76 ± 0,12	99,53 ± 0,47	0,88 ± 0,04	0,06 ± 0,01
Erosão	500	99,99 ± 0,03	98,00 ± 1,07	4,12 ± 0,18	0,37 ± 0,01
	100	88,61 ± 0,20	87,51 ± 1,96	0,93 ± 0,04	0,07 ± 0,01

Fonte: O autor (2025).

5.2 Resultados em relação ao estado da arte

Nesta seção, realiza-se uma análise comparativa entre o antivírus desenvolvido neste estudo e outras soluções antivírus atuais. Para viabilizar a isenção no confronto dos resultados, foi adotado um procedimento padronizado de extração de características, monitorando 649 atributos de cada executável avaliado.

A rede neural integrada ao antivírus proposto neste trabalho conta com uma arquitetura simples, composta por uma única camada oculta com 500 neurônios e utilizando um *kernel* de Dilatação, que funciona como um filtro não linear para selecionar e ponderar as propriedades relevantes dos dados. Por outro lado, a solução apresentada por LIMA et al.2021 emprega uma abordagem mais variada, utilizando onze diferentes métodos de treinamento em redes neurais rasas, baseadas em *back-propagation*. Neste caso, são testadas quatro arquiteturas distintas de camada oculta

para cada algoritmo de aprendizado, buscando encontrar a configuração de melhor desempenho para o sistema.

Para fins de comparação e referência, também foi incluído na avaliação, um antivírus baseado em redes neurais profundas, conforme os modelos de: SU and VASCONCELLOS, 2018, FARUKI and BUDDHADEV, 2019, MANIATH and ASHOK, 2017, HOU and SAAS, 2016 e HARDY and LINGWEI, 2016 . Com o objetivo de viabilizar a equidade nas comparações, todos os modelos do estado da arte foram reimplementados utilizando o mesmo conjunto de dados empregado neste trabalho. Os resultados apresentados na Tabela 5.4 referem-se às réplicas dos modelos de redes neurais profundas, que foram retreinadas especificamente para a detecção de *ransomware*. Desta vez, empregando o repositório de aprendizado estatístico desenvolvido por este estudo.

As Figuras 5.1 e 5.2 ilustram os resultados obtidos, que também podem ser conferidos nos valores numéricos da Tabela 5.4. A Figura 5.1(a) revela que o antivírus desenvolvido obteve acurácia média de 100,00% durante o treinamento. O modelo de LIMA et al., 2021 apresentou variação considerável, com desempenho máximo de 100% e mínimo de 49,83%. Ambas as arquiteturas possuíram duas camadas ocultas com 100 neurônios cada. Destaca-se ainda que a função de aprendizado utilizada influenciou bastante o desempenho, sendo o "*Conjugate gradient backpropagation with Fletcher-Reeves updates*" o mais eficiente, enquanto o "*Resilient backpropagation*" resultou nos menores índices de acurácia.

A Figura 5.1(b) exibe os dados de desempenho na fase de teste para o antivírus deste trabalho e para o concorrente mais recente. O modelo desenvolvido apresentou acurácia média de 99,87% e desvio padrão de 0,20%. Já o sistema de LIMA et al., 2021 apresentou acurácia variando entre 49,83% (cenário menos favorável) e 99,98% (melhor cenário). Esses resultados reforçam que redes neurais baseadas em *backpropagation* podem sofrer grandes flutuações de desempenho, dependendo de suas parametrizações iniciais. Assim, a opção feita por LIMA et al., 2021 de experimentar diversas arquiteturas e estratégias de otimização se mostrou pertinente.

Os gráficos 5.2(a) e 5.2(b) mostram os *boxplots* dos tempos gastos no treinamento e na validação, respectivamente. O antivírus proposto precisou, em média, de apenas 3,75 segundos para treinar, enquanto o sistema de HOU and SAAS, 2016 demandou 11975.77 segundos para a mesma etapa. No período de teste, os tempos foram próximos, com exceção do antivírus autoral, que apresentou um leve acréscimo. Ressalta-se que, à medida que o volume de dados aumenta, o tempo de treinamento de uma rede profunda pode crescer exponencialmente, o que dificulta sua aplicação em cenários que exigem rápida atualização.

O antivírus apresentado destacou-se em relação às principais soluções acadêmicas disponíveis. No teste, o sistema alcançou média de acurácia de 99,87%, e tempo médio de treinamento de 3,75 segundos. Um antivírus lançado, em tempos contemporâneos, se tornará rapidamente obsoleto, exigindo re-treinamento contínuo para mitigar as novas vulnerabilidades detectadas. Assim, o tempo necessário para atualizar um antivírus deve acompanhar o ritmo de surgimento de novas vulnerabilidades globalmente, a fim de minimizar defasagens de proteção.

Neste estudo, a eficiência é definida como o inverso do produto entre a acurácia no teste e o tempo de treinamento. No trabalho de LIMA et al., 2021, a configuração menos eficiente foi excluída da análise, pois uma acurácia próxima a 50% equivale a classificações aleatórias. Pelo mesmo critério, a obra de HOU and SAAS, 2016 também foi descartada.

A métrica de eficiência se mostra crucial ao definir critérios de desempenho do modelo e também por indicar sua prontidão para operação, atributo particularmente relevante para soluções antivírus, onde a rapidez na atualização é um diferencial. A capacidade de executar um novo treinamento de forma ágil, diante de ameaças recém-identificadas, pode ser decisiva para a redução de riscos.

A Tabela 5.5 traz as matrizes de confusão das técnicas comparadas na Tabela 5.4, apresentadas em porcentagem. Essas matrizes são essenciais para avaliar a qualidade do aprendizado supervisionado. As siglas M. e B. referem-se, respectivamente, a *Malware* e Benigno. As etiquetas verticais indicam as classes reais, enquanto as

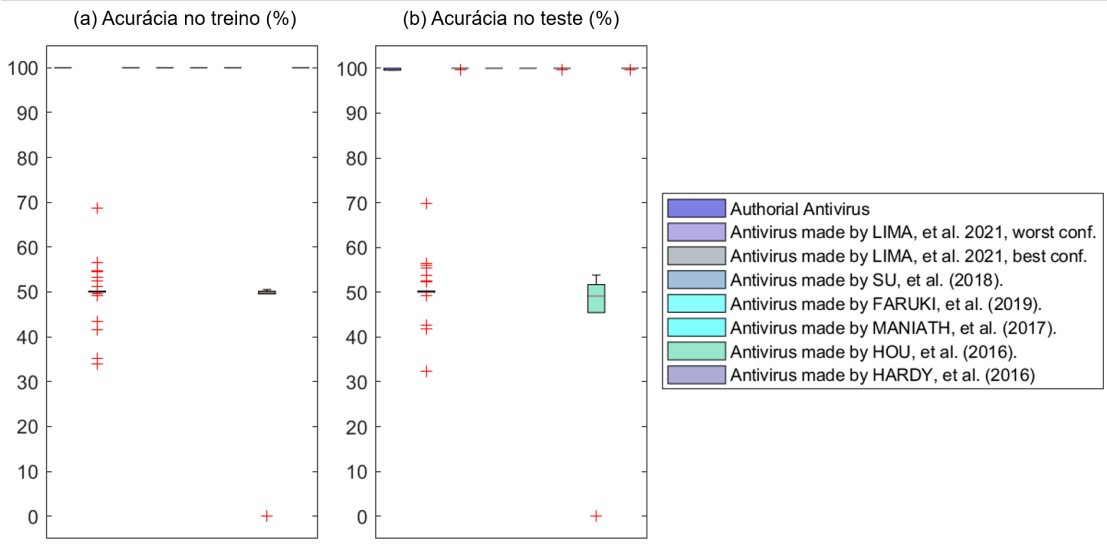
horizontais representam as classes previstas pelo modelo. Na matriz, a diagonal principal reúne os casos em que a predição corresponde à classe verdadeira, os chamados “verdadeiros positivos”.

Para que um classificador atinja ótimo desempenho, espera-se que os valores ao longo da diagonal principal sejam elevados, enquanto os demais elementos da matriz permaneçam baixos. Na Tabela 5.5, as diagonais principais são destacadas em negrito. Durante a etapa de testes, o antivírus desenvolvido classificou equivocadamente, em média, 0,17% dos incidentes de *malware* como benignos (falsos negativos), e 0,08% dos arquivos benignos foram incorretamente classificados como *malware* (falsos positivos).

Ainda segundo a Tabela 5.5, a sensibilidade e especificidade do antivírus representam, respectivamente, sua capacidade de identificar corretamente ameaças e distinguir aplicações seguras. A matriz de confusão apresentada em formato percentual visa facilitar a análise desses indicadores. Portanto, tanto sensibilidade quanto especificidade podem ser extraídas diretamente dessa matriz, conforme evidenciado na Tabela 5.5. Por exemplo, o antivírus aqui desenvolvido atingiu sensibilidade média de 99,92% (verdadeiros positivos) e especificidade média de 99,83% (verdadeiros negativos).

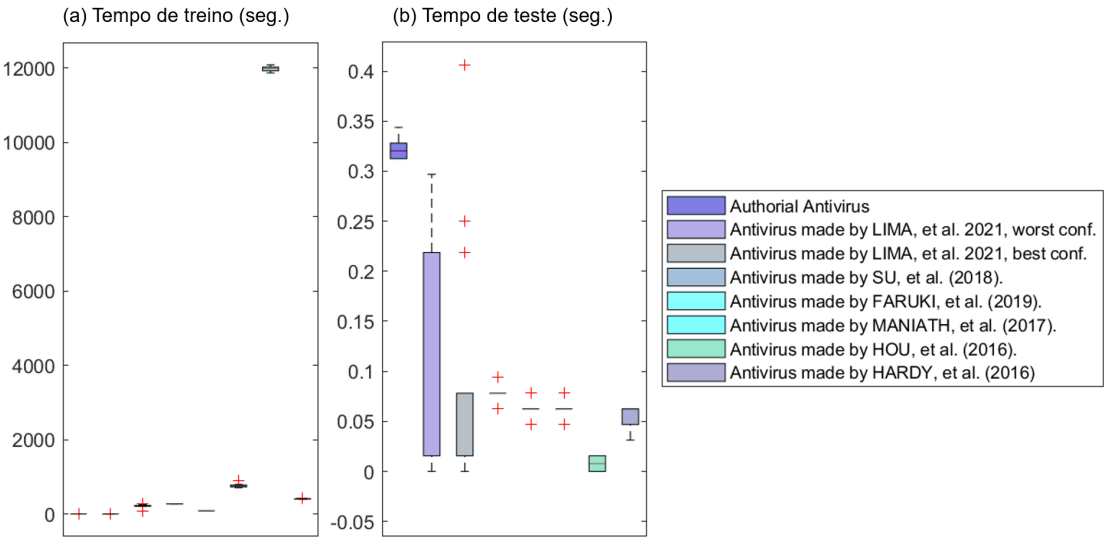
Por fim, a Tabela 5.6 apresenta os resultados dos testes de hipótese paramétricos (*t-student*) e não paramétricos (*Wilcoxon*) para a comparação do antivírus proposto com as demais soluções de última geração. O *p-value* informa a probabilidade de os resultados observados serem atribuídos ao acaso: valores próximos de 0 sugerem baixa probabilidade de acaso, enquanto valores próximos de 1 indicam alta consistência dos resultados. Em termos didáticos, quando o *p-value* é igual a 1, isso significa que os antivírus são estatisticamente diferentes entre si.

Figura 5.1: *Boxplots* a respeito da acurácia de antivírus autorais e de última geração.



Fonte: O autor (2025).

Figura 5.2: *Boxplots* de tempos de processamento de antivírus autorais e de última geração.



Fonte: O autor (2025).

Tabela 5.4: Comparação entre o antivírus autoral e os mais recentes.

Técnica	Acurácia de treino (%)	Acurácia de teste (%)	Tempo de treino (seg,)	Tempo de teste (seg,)	Eficiência
Antivírus autoral	**100,00 ± 0,00**	99,87 ± 0,20	3,75 ± 0,27	0,32 ± 0,01	**26,63**
LIMA et al., 2021, pior conf.	49,83 ± 6,01	49,83 ± 6,62	1,90 ± 0,33	0,08 ± 0,10	–
LIMA et al., 2021, melhor conf.	**100,00 ± 0,00**	99,98 ± 0,07	218,72 ± 31,92	0,08 ± 0,10	0,46
SU and VASCONCELLOS, 2018	**100,00 ± 0,00**	**99,99 ± 0,01**	273,35 ± 2,09	0,08 ± 0,01	0,36
FARUKI and BUDDHADEV, 2019	**100,00 ± 0,00**	**99,99 ± 0,01**	88,34 ± 0,75	0,06 ± 0,01	1,13
MANIATH and ASHOK, 2017	**100,00 ± 0,00**	99,91 ± 0,18	767,88 ± 53,65	0,06 ± 0,01	0,13
HOU and SAAS, 2016	40,00 ± 21,09	39,97 ± 21,20	11975,77 ± 67,41	0,01 ± 0,01	–
HARDY and LINGWEI, 2016	**100,00 ± 0,00**	99,96 ± 0,14	410,03 ± 11,23	0,05 ± 0,01	0,24

Fonte: O autor (2025).

Tabela 5.5: Matriz de confusão do Antivírus Autoral e dos Antivírus de Última Geração (%).

		Treino		Teste	
Técnica		M.	B.	M.	B.
Antivírus autoral	M.	**100,00 ± 0,00**	0,00 ± 0,00	**99,92 ± 0,27**	0,17 ± 0,35
	B.	0,00 ± 0,00	**100,00 ± 0,00**	0,08 ± 0,27	**99,83 ± 0,35**
LIMA et al., 2021 pior conf.	M,	**44,09 ± 44,70**	44,43 ± 49,35	**44,03 ± 44,40**	44,38 ± 49,44
	B,	55,91 ± 44,70	**55,57 ± 49,35**	55,97 ± 44,40	**55,62 ± 49,44**
LIMA et al., 2021 melhor conf.	M,	**100,00 ± 0,00**	0,00 ± 0,00	**99.96 ± 0.14**	0.00 ± 0.00
	B,	0,00 ± 0,00	**100,00 ± 0,00**	0.04 ± 0.14	**100.00 ± 0.00**
SU and VASCONCELLOS, 2018	M,	**100,00 ± 0,00**	0,00 ± 0,00	**99.97 ± 0.10**	0.00 ± 0.00
	B,	0,00 ± 0,00	**100,00 ± 0,00**	0.03 ± 0.10	**100.00 ± 0.00**
FARUKI and BUDDHADEV, 2019	M,	**100,00 ± 0,00**	0,00 ± 0,00	**99.97 ± 0.10**	0.00 ± 0.00
	B,	0,00 ± 0,00	**100,00 ± 0,00**	0.03 ± 0.10	**100.00 ± 0.00**
MANIATH and ASHOK, 2017	M,	**100,00 ± 0,00**	0,00 ± 0,00	**99.92 ± 0.27**	0.08 ± 0.26
	B,	0,00 ± 0,00	**100,00 ± 0,00**	0.08 ± 0.27	**99.92 ± 0.26**
HOU and SAAS, 2016	M,	**20.00 ± 20.16**	20.00 ± 20.16	**20.00 ± 20.16**	20.00 ± 20.16
	B,	80.00 ± 20.16	**80.00 ± 20.16**	80.00 ± 20.16	**80.00 ± 20.16**
HARDY and LINGWEI, 2016	M,	**100,00 ± 0,00**	0,00 ± 0,00	**99,92 ± 0,27**	0,00 ± 0,00
	B,	0,00 ± 0,00	**100,00 ± 0,00**	0,08 ± 0,27	**100,00 ± 0,00**

Fonte: O autor (2025).

Tabela 5.6: T-students e Wilcoxon testam as hipóteses do antivírus autoral e do estado da arte.

	t-students (teste paramétrico)		Wilcoxon (teste não-paramétrico)	
Comparação	Hipóteses	*p*-value	Hipóteses .	*p*-valor
Antivírus autoral *vs* LIMA et al., 2021, pior conf.	1	2,18643e-27	1	6,04829e-12
Antivírus autoral *vs* LIMA et al., 2021, melhor conf.	1	0,012324	1	0,0127881
Antivírus autoral *vs* Antivírus de SU and VASCONCELLOS, 2018	1	0,00142818	1	0,00134893
Antivírus autoral *vs* Antivírus de FARUKI and BUDDHADEV, 2019	1	0,00142818	1	0,00134893
Antivírus autoral *vs* Antivírus de MANIATH and ASHOK, 2017	0	0,46027	0	0,487005
Antivírus autoral *vs* Antivírus de HOU and SAAS, 2016	1	5,53269e-16	1	1,02472e-11
Antivírus autoral *vs* Antivírus de HARDY and LINGWEI, 2016	1	0,011668	0	0,0710494

Fonte: O autor (2025).

Capítulo 6

Conclusão

Diante do aumento constante na criação de novos códigos maliciosos, torna-se essencial que os sistemas de detecção implementem ferramentas de monitoramento digital, capazes de atuar de forma preventiva, atendendo às necessidades dos usuários. Caso isso não ocorra, em situações onde haja falhas na identificação de *softwares* nocivos, existe o risco iminente de que informações sigilosas sejam acessadas por indivíduos não autorizados.

Dessa forma, conclui-se que a escolha do *software* antivírus desempenha um papel crucial no enfrentamento das ameaças digitais. Na análise realizada, a taxa de detecção de *malwares* do tipo *ransomware* variou de 0% a 97,63%, dependendo do antivírus comercial selecionado. Este estudo examinou 86 soluções antivírus disponíveis no mercado. Em média, esses programas conseguiram identificar corretamente 55,22% das ameaças. A partir da análise das amostras, observou-se que, em média, os antivírus apresentaram 13,15% de falsos negativos e deixaram de emitir diagnóstico em 31,71% dos casos. Para a realização dos testes, foi utilizada a plataforma VirusTotal, que submeteu automaticamente os arquivos maliciosos aos programas antivírus.

É importante destacar que o VirusTotal não permite a seleção entre versões gratuitas e pagas dos antivírus, o que impossibilitou uma comparação entre os desempenhos de ambas. Infere-se que os recursos oferecidos nas versões gratuitas possuem eficácia consideravelmente inferior à das versões completas. Outro ponto relevante

é que, mesmo sendo de domínio público e utilizados em ataques reais, muitos dos *malwares* analisados não eram reconhecidos por uma parcela significativa dos antivírus testados.

Para superar essas falhas nos antivírus tradicionais, foi desenvolvido um sistema baseado em inteligência artificial, capazes de examinar milhares de amostras e aprender, de maneira estatística, as características associadas a comportamentos maliciosos. Após esse processo de aprendizado, o sistema inteligente autoral pôde identificar e classificar novas ameaças, comparando seus atributos com os aprendidos anteriormente. Assim, o processo de reconhecimento de comportamentos maliciosos pôde se tornar automatizado, eliminando a dependência de relatos de usuários, após já terem sido infectados, para só então se iniciar alguma resposta por parte do antivírus.

Com o objetivo de colaborar na contenção da propagação de arquivos nocivos, foi desenvolvido um antivírus próprio, apto a classificar executáveis como inofensivos ou maliciosos. O sistema analisa, de forma estatística, 649 comportamentos que um arquivo pode executar dentro do ambiente operacional. Em laboratório, o antivírus acompanha modificações no Registro do sistema e rastreia chamadas executadas por todos os processos iniciados pelo arquivo analisado. Todo o processo de identificação de padrões dessas 649 ações é realizado por redes neurais extremas.

Em vez de utilizar kernels tradicionais, foram aplicados kernels proprietários nas redes extremas. A escolha pelas ELMs se deve à sua alta velocidade de treinamento e à eficácia na previsão de resultados, quando comparadas a redes neurais convencionais. O *kernel* próprio, denominado Dilatação, demonstrou capacidade de distinguir com acurácia *malware* do tipo *ransomware*, de *softwares* legítimos em 99.87% dos testes, com um tempo médio de treinamento de apenas 3,75 segundos.

Por fim, é válido destacar que a proposta deste trabalho é oferecer uma nova perspectiva sobre a eficiência dos antivírus atuais. O trabalho propõe soluções criativas e eficazes para melhorar a identificação de ameaças do tipo *ransomware*.

6.1 Trabalhos Futuros

Os resultados alcançados neste estudo, com destaque para a elevada acurácia e o rápido tempo de resposta das redes neurais extremas com núcleos morfológicos, abrem diversas possibilidades para investigações futuras. A intenção é explorar os seguintes caminhos para dar continuidade e aprimoramento à pesquisa:

- Integração de novos tipos de *malware*: ampliar a base de dados para contemplar não apenas *ransomware*, mas também outras categorias sofisticadas de ameaças digitais, como *spyware, keyloggers* e *rootkits*, permitindo uma avaliação mais abrangente da eficácia do modelo;

- Aprimoramento dos *kernels* morfológicos: investigar variantes avançadas de *kernels* morfológicos, bem como combinações híbridas com *kernels* tradicionais, de modo a potencializar a capacidade do sistema em mapear fronteiras de decisão complexas e aumentar sua robustez;

- Implementação em ambientes reais: desenvolver protótipos que possam ser incorporados em sistemas reais de monitoramento, como *gateways* de rede e estações de trabalho corporativas, para validar o desempenho do modelo em situações práticas e em tempo real;

- Otimização do consumo computacional: embora o modelo já apresente tempos de treinamento bastante competitivos, recomenda-se investigar estratégias de paralelização e aceleração por GPU para reduzir ainda mais os custos computacionais, especialmente em ambientes com volumes massivos de dados.

Espera-se que estas direções contribuam para a evolução do campo de detecção automática de *malware*. A intenção é ampliar as defesas cibernéticas e permitir a segurança de dados sensíveis em diferentes contextos de aplicação.

6.2 Dificuldades Encontradas

Durante o desenvolvimento deste trabalho, foram identificados diversos desafios técnicos e metodológicos que impactaram o andamento da pesquisa. Um dos principais obstáculos foi relacionado ao elevado tempo de treinamento exigido por modelos de aprendizado profundo, como as redes *Deep Learning*, que apresentam arquitetura sequencial e complexa. Esse fator limitou a possibilidade de exploração de redes mais profundas, uma vez que, mesmo com alto poder computacional, a organização em cascata das camadas, inviabiliza o paralelismo completo e prolonga significativamente os ciclos de treinamento.

Adicionalmente, a coleta de dados representativos para alimentar o modelo, representou um desafio prático. A obtenção de amostras reais de *malware* foi uma etapa especialmente delicada. Variantes maliciosas sofisticadas envolvem riscos operacionais significativos. Por isso, foi necessário adotar cuidados específicos para permitir o isolamento e o controle rigoroso do ambiente de testes.

Por fim, destaca-se a dificuldade em validar os resultados em ambientes reais, dado que muitos antivírus comerciais não compartilham seus bancos de assinaturas, nem padronizam as nomenclaturas de ameaças, o que dificulta comparações diretas. Esse cenário obrigou o projeto a focar em métricas laboratoriais, que, apesar de robustas, não capturam todas as variáveis presentes em contextos operacionais do mundo real.

Esses desafios reforçam a complexidade envolvida na criação de soluções de segurança digital baseadas em inteligência artificial. Foram necessários esforços contínuos de pesquisa e aprimoramento para alcançar resultados cada vez mais eficientes e aplicáveis.

Referências

W. W. AZEVEDO and *et al.* Morphological extreme learning machines applied to detect and classify masses in mammograms. *In: 2015 International Joint Conference on Neural Networks (IJCNN), Killarney.*, 2015. doi: https://doi.org/10.1109/IJCNN.2015.7280774.

W. W. AZEVEDO and *et al.* Morphological extreme learning machines applied to the detection and classification of mammary lesions. *In: Tapan K Gandhi; Siddhartha Bhattacharyya; Sourav De; Debanjan Konar; Sandip Dey. (Org.). Advanced Machine Vision Paradigms for Medical Image Analysis. 1ed.Londres: Elsevier Science.*, pages 1–30, 2020. doi: https://doi.org/10.1016/B978-0-12-819295-5.00003-2.

François CHOLLET. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.1109/CVPR.2017.195.

DejavuForensics. melm. *GitHub repository*, 2025. URL `https://github.com/DejavuForensics/melm`.

P. FARUKI and B. *et al.* BUDDHADEV. Droiddivesdeep: Android malware classification via low level monitorable features with deep neural networks. *International Conference on Security Privacy*, 2019. doi: https://doi.org/10.1007/978-981-13-7561-3_10.

W. HARDY and C. *et al.* LINGWEI. Dl 4 md : A deep learning framework for intelligent malware detection. *In Int'l Conf. Data Mining*, pages 61–67, 2016.

S. HOU and A. *et al.* SAAS. Droiddelver: An android malware detection system using deep belief network based on api call blocks. *Web-Age Information Management. WAIM 2016 International Workshops, MWDA, SDMMW, and SemiBDMA*, 2016. doi: https://doi.org/10.1007/978-3-319-47121-1_5.

G. B. *et al.* HUANG. Classification ability of single hidden layer feedforward neural networks. *The IEEE Transactions on Neural Networks and Learning Systems*, 11(3):799–801, 2000. doi: https://doi.org/10.1109/72.846750.

G. B. *et al.* HUANG. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 42(2): 513–519, 2012. doi: https://doi.org/10.1109/TSMCB.2011.2168604.

Intel. *McAfee Labs*. Accessed on Feb 2020, 2018. URL https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2018.pdf.

INTEL. Mcafee labs: Threat report. , Disponível em: https://secure.mcafee.com/us/resources/reports/rp-quarterly-threats-mar-2017.pdf?cid=BHP-075. Acesso em 20 janeiro 2022. 2022.

PONTUS JOHNSON, DAN GORTON, ROBERT LAGERSTRÖM, and MATHIAS EKSTEDT. Time between vulnerability disclosures: A measure of software product vulnerability. *Computers Security*, 62:278–295, 2016. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2016.08.004. URL https://www.sciencedirect.com/science/article/pii/S0167404816300955.

S. M. L. LIMA, A. G. SILVA-FILHO, and W. P. DOS SANTOS. A methodology for classification of lesions in mammographies using zernike moments, elm and svm neural networks in a multi-kernel approach. *In: 2014 IEEE International Conference on Systems, Man and Cybernetics SMC, San Diego*, 2014. doi: https://doi.org/10.1109/SMC.2014.6974041.

S. M. L. LIMA, SILVA-FILHO, and W. P. SANTOS. *Morphological Decomposition to Detect and Classify Lesions in Mammograms.In: Wellington Pinheiro dos Santos; Maíra Araújo de Santana; Washington Wagner Azevedo da Silva. (Org.). Understanding a Cancer Diagnosis*. 2020. URL https://novapublishers.com/shop/understanding-a-cancer-diagnosis/.

S.M.L LIMA. *Limitation of COTS antiviruses: issues, controversies, and problems of COTS antiviruses*. In: Cruz-Cunha, M.M., Mateus-Coelho, N.R. (eds.) Handbook of Research on Cyber Crime and Information Privacy, vol. 1, 1st edn. IGI Global, Hershey, 2020. doi: http://dx.doi.org/10.4018/978-1-7998-5728-0.ch020.

S.M.L. LIMA, A. G. SILVA-FILHO, and W. P. SANTOS. Detection and classification of masses in mammographic images in a multi-kernel approach. *Computer Methods and Programs in Biomedicine*, 134:11–29, 2016. doi: https://doi.org/10.1016/j.cmpb.2016.04.029.

S.M.L. LIMA, H.K.L. SILVA, and J.H.S. *et al.* LUZ. Artificial intelligence-based antivirus in order to detect malware preventively. *Progress in Artificial Intelligence*, 2021. doi: https://doi.org/10.1007/s13748-020-00220-4.

S. MANIATH and A. *et al.* ASHOK. Deep learning lstm based ransomware detection. *Recent Developments in Control, Automation Power Engineering*, 2017. doi: https://doi.org/10.1109/RDCAPE.2017.8358312.

Harun Oz, Ahmet Aris, Albert Levi, and A. Selcuk Uluagac. A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Computing Surveys*, 53(6):1–42, 2021.

J. M. S. *et al.* PEREIRA. *Method for Classification of Breast Lesions in Thermographic Images Using ELM Classifiers. In: Wellington Pinheiro dos Santos; Maíra Araújo de Santana; Washington Wagner Azevedo da Silva. (Org.). Understanding a Cancer Diagnosis.* https://novapublishers.com/shop/understanding-a-cancer-diagnosis/, 2020.

R.P. PINHEIRO, S.M.L. LIMA, D.M. SOUZA, and *et al.* Antivirus applied to jar malware detection based on runtime behaviors. *Scientific Reports - Nature: 12, 1945 (2022)*, 2022. doi: https://doi.org/10.1038/s41598-022-05921-5.

Igor Pinheiro Henriques de Araújo, Liosvaldo Mariano Santiago de Abreu, Sthéfano Henrique Mendes Tavares Silva, Ricardo Paranhos Pinheiro, and Sidney Marlon Lopes de Lima. Antimalware applied to iot malware detection based on softcore processor endowed with authorial sandbox. *Journal of Computer Virology and Hacking Techniques*, 1:1, 2024.

ransomware. Retrieval for ransoware malware analysis. 2025. URL https://github.com/DejavuForensics/ransomware.

SANS. *SANS Institute InfoSec Reading Room. Out with The Old, In with The New: Replacing Traditional Antivirus.* Accessed on Feb 2020, 2017. URL https://www.sans.org/reading-room/whitepapers/analyst/old-new-replacing-traditional-antivirus-37377.

C. H. M. Santos and S. M. Lopes-Lima. Xai-driven antivirus in pattern identification of citadel malware. *Journal of Computational Science*, 1:1–21, 2024.

W. P. SANTOS. *Mathematical Morphology In Digital Document Analysis and Processing*, volume 8. New York: Nova Science, 2011.

J. SU and *et al.* VASCONCELLOS, D. Lightweight classification of iot malware based on image recognition. *2018 IEEE 42nd Annual Computer Software and*

Applications Conference (COMPSAC), 2018. doi: https://doi.org/10.1109/COMPSAC.2018.10315.

Sthéfano Henrique Mendes Tavares-Silva, Sidney Marlon Lopes-Lima, Ricardo Paranhos-Pinheiro, Liosvaldo Mariano Santiago-Abreu, Rafael Diniz Toscano-Lima, and Sérgio Murilo Maciel Fernandes. Antivirus solution to iot malware detection with authorial next-generation sandbox. *Journal of Supercomputing*, 81:81–151, 2025. ISSN 0920-8542. doi: 10.1007/s11227-024-06506-x.