UNIVERSIDADE FEDERAL DE PERNAMBUCO

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

Graduação em Engenharia Eletrônica

Vitor Mendes Carvalho

USO DE MÁQUINAS DE APRENDIZADO EXTREMO COM OPERADORES MORFOLÓGICOS NA DETECÇÃO PREVENTIVA DE AMEAÇAS *INSTALLCORE*

Recife
2025

Vitor Mendes Carvalho

USO DE MÁQUINAS DE APRENDIZADO EXTREMO COM OPERADORES MORFOLÓGICOS NA DETECÇÃO PREVENTIVA DE AMEAÇAS *INSTALLCORE*

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

Carvalho, Vitor Mendes.
Uso de máquinas de aprendizado extremo com operadores morfológicos na detecção preventiva de ameaças installcore / Vitor Mendes Carvalho. - Recife, 2025.
80 p. : 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. Antivírus. 2. Detecção de malware. 3. InstallCore. 4. Redes Neurais Artificiais. 5. Processamento Morfológico. I. Lima, Sidney Marlon Lopes de. (Orientação). II. Título.

000 CDD (22.ed.)

Vitor Mendes Carvalho

USO DE MÁQUINAS DE APRENDIZADO EXTREMO COM OPERADORES MORFOLÓGICOS NA DETECÇÃO PREVENTIVA DE AMEAÇAS *INSTALLCORE*

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: 30/01/2025

Banca Examinadora

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

Prof. Andrea Maria Nogueira Cavalcanti Ribeiro, D.Sc.
Universidade Federal de Pernambuco

Prof. Filipe Rolim Cordeiro, D.Sc.
Universidade Federal Rural de Pernambuco

Prof. Lubnnia Morais Florencio de Souza, D.Sc.
Universidade Federal de Pernambuco

Dedico este trabalho aos meus
pais e a minha companheira,
meus maiores incentivadores,
pela apoio que me deram e pelo
amor que me dedicaram.

Agradecimentos

Agradeço aos meus pais, que me apoiaram e abdicaram de tempo e recursos para contribuir na minha formação pessoal e profissional.

Agradeço a minha companheira, que esteve comigo durante toda a minha graduação, me apoiou, incentivou, me suportou e sempre esteve ao meu lado.

Agradeço as amizades que tive nessa fase da minha vida e aquelas que continuarão presentes nessa jornada.

Agradeço a todas as pessoas que fizeram parte dessa etapa decisiva da minha vida.

O medo dá origem ao mal.

O homem coletivo sente a

necessidade de lutar

Chico Science

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.)

USO DE MÁQUINAS DE APRENDIZADO EXTREMO COM OPERADORES MORFOLÓGICOS NA DETECÇÃO PREVENTIVA DE AMEAÇAS *INSTALLCORE*

Vitor Mendes Carvalho

A sofisticação crescente das ameaças cibernéticas, como o *malware InstallCore*, é uma preocupação crescente para a segurança de sistemas e dados. O *InstallCore*, um tipo de *adware*, infiltra-se em dispositivos, exibindo anúncios intrusivos e comprometendo a privacidade do usuário. Este trabalho propõe um método inovador de detecção de *malware* baseado em redes neurais artificiais, visando superar as limitações dos antivírus tradicionais. Utilizando aprendizado profundo e *kernels* de processamento morfológico, desenvolvemos um sistema capaz de identificar com alta acurácia o *malware InstallCore* e suas variantes. Os resultados mostram que o *antimalware* proposto alcança uma acurácia média de 99,35% na detecção do *InstallCore*, com um tempo de treinamento de 1,93 segundos. Comparado com soluções comerciais, nosso modelo apresenta elevada taxa de detecção, menor tempo de treinamento e adaptação a novas ameaças, explorando características intrínsecas dos arquivos executáveis para maior robustez e eficiência. O modelo proposto oferece uma alternativa eficaz às soluções tradicionais, contribuindo para a proteção de sistemas e dados contra ameaças cada vez mais sofisticadas.

Palavras-chave: Antivírus, Detecção de *malware*, *InstallCore*, Redes Neurais Artificiais, Processamento Morfológico.

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.)

USE OF EXTREME LEARNING MACHINES WITH MORPHOLOGICAL OPERATORS IN THE PREVENTIVE DETECTION OF INSTALLCORE THREATS

Vitor Mendes Carvalho

As cyber attacks become more advanced, InstallCore malware poses a serious security risk. This malware shows unwanted ads and steals private information. We created a new malware detection system using artificial neural networks and morphological processing kernels that works better than regular antivirus software. Our system uses deep learning and special data processing to find InstallCore malware with 99.35% accuracy and only needs 1,93 seconds to learn. When compared to regular antivirus programs, our system is faster, more accurate, and better at finding new threats by analyzing key parts of suspicious files. This makes our solution a better choice for protecting computers against modern cyber attacks.

Keywords: Antivirus, Malware detection, InstallCore, Artificial Neural Networks, Morphological Processing.

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviações

SDK	Software Development Kit
PUP	Potentially unwanted program
ELM	Extreme Learning Machine
UC	Unidade de Controle
IoT	Internet of Things
mELM	Morphological Extreme Learning Machine
DL	Deep Learning
GPU	Graphics Processing Units
MLP	Multilayer Perceptron
CUDA	Compute Unified Device Architecture
RAM	Random access memory
TLS	Transport Layer Security
API	Application Programming Interface
GUI	Graphical User Interface
DNS	Domain Name System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol

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
$\dagger$	..... Adjunto de Hermitiano

Sumário

Capítulo 1

Introdução

A Era digital, marcada pela expansão das tecnologias da informação, trouxe consigo uma série de transformações sociais e econômicas. A facilidade de acesso à informação e a automação de processos revolucionaram a forma como a sociedade contemporânea vive e trabalha. Contudo, essa mesma evolução tecnológica também expôs novas vulnerabilidades nos sistemas e redes computacionais, tornando a segurança da informação uma preocupação cada vez mais urgente. A proliferação de ciberataques e a valorização dos dados pessoais como ativos digitais impulsionaram o desenvolvimento de *softwares* maliciosos cada vez mais sofisticados, capazes de comprometer a confidencialidade de informações sensíveis.

Dado esse contexto, os dados pessoais se tornaram um ativo valioso e lamentavelmente negociados em mercados clandestinos. Essa valorização dos dados pessoais expõe os usuários a riscos cada vez maiores, como a proliferação de *adwares*. Esses programas potencialmente indesejados, do inglês *Potentially unwanted program* (PUP), frequentemente adquiridos em sites duvidosos ou junto a softwares piratas, infiltram-se nos dispositivos, coletando informações pessoais e exibindo propagandas invasivas. Além disso, os *adwares* podem abrir portas para outros tipos de *malware*, como vírus e *ransomware*, colocando em risco a segurança e a privacidade dos usuários.

É importante destacar a diferença entre vírus e *malware*. O termo "vírus" refere-se a um tipo específico de *malware* que se propaga infectando outros arquivos ou

programas, semelhante a um vírus biológico. Já "*malware*" (malicioso + software) é um termo mais amplo que engloba qualquer software malicioso, incluindo vírus, mas também *worms, trojans, ransomware, spyware, adware*, entre outros. Dessa forma, os chamados "antivírus" atuam não apenas contra vírus, mas contra uma variedade de ameaças digitais. Por isso, seria mais preciso chamá-los de *antimalwares*, já que sua função vai além da detecção de vírus, abrangendo uma gama mais ampla de códigos maliciosos.

Um exemplo notório de *malware* é o *InstallCore*, que passou por uma transformação perversa ao longo de sua existência. Originalmente, o *InstallCore* foi concebido como uma ferramenta legítima voltada para desenvolvedores, com o propósito de facilitar a criação de instaladores e a entrega de anúncios (IronSource *et al.* 2011). No entanto, em mãos erradas, essa ferramenta tornou-se um instrumento de disseminação de *adwares* e outras ameaças cibernéticas. Empregando técnicas de camuflagem sofisticadas, como criptografia e assinatura com certificados falsos, o *InstallCore* se infiltra nos sistemas, instalando software indesejado e colocando em risco a privacidade e a segurança dos usuários.

Para se ter uma ideia desse tipo de ameaça, basta observar que o ataque *Fireball Adware* em 2017, infectou cerca de 250 milhões de computadores (Andy Greenberg *et al.* 2017), isso serve como um alerta sobre a crescente sofisticação das ameaças digitais. Esse *adware* não se limitava a exibir anúncios indesejados; ele era capaz de manipular navegadores, rastrear atividades online e até mesmo executar comandos remotos nos dispositivos infectados (Andy Greenberg *et al.* 2017). Esse caso demonstra como os *adwares* evoluíram para ameaças mais complexas, capazes de comprometer a segurança de sistemas em escala global, exigindo soluções de segurança cada vez mais robustas e proativas.

Uma vez identificada a demanda de um sistema de segurança efetivo, é necessário analisar de forma crítica o comportamento dos antivírus comercializados atualmente. A dependência dos antivírus tradicionais em *"deny/exclude lists"* de ameaças conhecidas, limita significativamente sua capacidade de detectar novos *malwares*. A falta

de um padrão universal para a classificação das ameaças agrava ainda mais o problema, dificultando a colaboração entre as empresas de segurança e a criação de defesas mais robustas. Por razões comerciais e estratégicas, mesmo os antivírus de uma mesma fabricante podem não compartilhar completamente os mesmos padrões ou classificações de *malware*. (PINHEIRO et al.,2022).

Diante do exposto, nota-se que a dependência dos antivírus tradicionais em listas de assinaturas para identificar *malware* limita sua capacidade de detectar novas ameaças. Essa abordagem, embora eficaz para ameaças conhecidas, torna-se ineficiente diante da rápida evolução das técnicas de ataque. Para superar essas limitações, é preciso explorar novas abordagens, como a inteligência artificial, que permita identificar padrões suspeitos nos programas em tempo real.

Um *antimalware* baseado em inteligência artificial (IA) destaca-se por sua capacidade de detectar ameaças de forma preventiva, superando os métodos tradicionais de antivírus. Durante a etapa de aprendizado, a IA é treinada para reconhecer características maliciosas, permitindo que, após essa fase, seja capaz de identificar até mesmo *malwares* inéditos que explorem vulnerabilidades específicas. Assim, um *antimalware* com IA consegue antecipar padrões maliciosos em aplicativos desconhecidos, permitindo a detecção antes mesmo que sejam executados (clicados) pelo usuário comum.

No contexto de IA, uma alternativa promissora é o uso de redes neurais artificiais para identificar padrões de comportamento de *malware*. Ao analisar grandes volumes de dados, as redes neurais podem aprender a distinguir entre software benigno e malicioso com alta acurácia. Estudos como o de LIMA et al., 2021 demonstraram que as redes neurais podem detectar *malware* com uma taxa de acerto superior a 98%. No entanto, a detecção de *malware* por meio de redes neurais apresenta alguns desafios. O tempo de resposta de um sistema baseado em redes neurais pode ser relativamente alto, o que pode permitir que um novo *malware* cause danos antes de ser detectado. Além disso, os *malwares* estão em constante evolução e podem se adaptar para evitar a detecção, exigindo que os modelos de redes neurais estejam

obsoletos caso não ocorram contínuos retreinamentos.

Como efeito colateral, o treinamento de redes neurais clássicas baseadas em retropropagação de dados envolve a busca por uma configuração ótima de hiperparâmetros que permita à rede aprender as representações subjacentes aos dados de forma eficiente e generalizável (LIMA et al., 2021). entanto, esse processo pode ser desafiador devido à presença de mínimos locais e ao alto custo computacional. A escolha inadequada dos hiperparâmetros pode levar a um desempenho sub ótimo da rede, enquanto um tempo de treinamento excessivo pode inviabilizar a aplicação da rede em cenários reais, por conta de sua obsolescência. Para superar esses desafios, em redes baseadas em retropropagação é fundamental utilizar técnicas de otimização avançadas, como explorar diferentes arquiteturas de redes e investir em hardware especializado (LIMA et al., 2021). Além disso, uma outra particularidade desse formato de rede é a elevada duração do treinamento exigida para torná-la apta a efetuar adequadamente o reconhecimento de padrão de *malware*.

A *Extreme Learning Machine* (ELM) se destaca como uma técnica de aprendizado de máquina eficiente e versátil. Seu funcionamento se baseia em uma arquitetura relativamente simples, com uma única camada oculta onde os pesos sinápticos são inicializados aleatoriamente e a camada de saída é treinada de forma analítica, eliminando a necessidade de um processo iterativo como nas redes neurais baseadas em retropropagação. Essa abordagem resulta em um treinamento significativamente mais rápido e em uma generalização superior em diversas tarefas de classificação e regressão. A versatilidade da ELM, combinada com sua velocidade de treinamento, a torna uma ferramenta atrativa para diversas áreas, como reconhecimento de padrões, processamento de sinais e previsão de séries temporais (LIMA et al., 2014 LIMA et al., 2020 LIMA et al., 2016 PEREIRA, 2020 AZEVEDO and *et al.*, 2015 AZEVEDO and *et al.*, 2020). Redes ELM podem contribuir de forma significativa para o avanço da segurança de computadores.

A pesquisa apresentada utiliza ELMs no campo da segurança da informação, mais precisamente na identificação de padrões de *adwares* gerados a partir do *SDK*

InstallCore. As características auditadas dos arquivos reservados ao aprendizado representam parâmetros de entrada em redes neurais sintéticas extremas. O objetivo é separar arquivos suspeitos em dois tipos de categorias: *apps* inofensivos (benignos) e *apps* maliciosos (*malwares*).

O treinamento extremo é fundamentado em *kernels*. Ao invés de utilizar *kernels* tradicionais, utilizam-se mELMs (*Morphological* ELMs), ELMs com núcleos de camada escondida baseados nos operadores morfológicos do tratamento de imagem digital; Erosão e Dilatação. No que diz respeito aos experimentos, os resultados são contrastados aos dos antivírus de ponta e são empregadas as métricas de avaliação amplamente adotadas. O software antivírus autoral desenvolvido neste trabalho consegue atingir um desempenho médio de 99,35% para diferenciar as aplicações benignas dos programas maliciosos da família *InstallCore*, com um espaço de tempo médio de aprendizado de 1,93 segundos.

1.1 Antimalware

Os programas antivírus, que deveriam ser chamados de *antimalware*, são a referência para a segurança de dispositivos contra perigos virtuais, como vírus, códigos maliciosos, sequestradores de dados, espiões digitais e outras variedades de programas nocivos. A razão para essa sugestão de mudança de nome é que, atualmente, essas ferramentas não combatem apenas vírus, mas uma ampla gama de ameaças digitais, como *malwares, ransomwares, spywares* e outros tipos de software malicioso. O termo "*antimalware*" refletiria de forma mais precisa a abrangência de sua funcionalidade. O desempenho dessas ferramentas apoia-se na capacidade de detecção e comparação com ameaças já catalogadas.

A técnica mais convencional adotada pelos antivírus é a análise de arquivos e ações com uma lista de restrições, que inclui identificadores de códigos maliciosos já conhecidos. Esses identificadores são sequências específicas que caracterizam um arquivo ou ação suspeita. Ao examinar um arquivo, o antivírus confere se ele coincide com algum item na lista de restrições. Caso haja uma coincidência, o arquivo

é reconhecido como prejudicial e é impedido de prosseguir. Entretanto, essa técnica apresenta uma falha significativa: sua eficiência está vinculada à frequente atualização da lista de restrições. Se um código malicioso não estiver registrado, ele pode escapar da detecção, possibilitando a ocorrência de infecções.

1.2 Limitações dos Antivírus Comerciais

Os antivírus comerciais de mercado operam, em grande parte, com base em listas de bloqueio, que são bancos de dados contendo assinaturas hash de *malwares* conhecidos. Essas assinaturas hashes são padrões únicos que identificam arquivos ou comportamentos maliciosos. Quando um arquivo é analisado, o antivírus compara suas características com as entradas na lista de bloqueio. Se houver uma correspondência, o arquivo é identificado como malicioso e bloqueado. No entanto, essa metodologia tem uma limitação significativa: sua eficácia depende da atualização constante da lista de bloqueio. Se um *malware* não estiver catalogado, ele pode passar despercebido, permitindo que infecções ocorram.

Dadas as limitações existentes nos antivírus comerciais, o desenvolvimento e a implantação de variações de aplicativos prejudiciais não é uma tarefa complexa. Esse processo é bastante simples, já que só é necessário introduzir alterações mínimas no *malware* fonte mediante sub-rotinas que não são usuais, como laços de repetição sem conteúdo interno. Em virtude dessas pequenas modificações no aplicativo malicioso, o *hash* do *malware* gerado é distinto do originário. Logo o *malware* modificado se torna irreconhecível para o antivírus que listou o arquivo malicioso original, deixando o dispositivo vulnerável a ataques.

Além disso, é relevante citar a existência de *exploits*, que são encarregados de desenvolver e difundir variações do *malware* original de forma automática. Resumindo, os programas antivírus embasados em listas de bloqueio tornam-se inúteis se expostos a quaisquer variantes de um *malware* já conhecido (SANS, 2017) (LIMA, 2020).

Com o intuito de fornecer uma perspectiva dessa limitação, foi realizada a compa-

ração da performance em 79 antivírus comercializados no mercado. No experimento, adotaram-se duas etapas distintas na plataforma VirusTotal. O primeiro processo consiste em submeter os arquivos à análise nos servidores da ferramenta, e o segundo destina-se a exportar o resultado do diagnóstico dos antivírus comerciais avaliados.

Para esse teste de identificação, foram analisados 9.405 arquivos *malware* do tipo *InstallCore*. Os indicadores de rendimento encontram-se na Tabela 1.1, o resultado do exame tem três tipos de classificações diferentes. A plataforma do VirusTotal disponibiliza uma avaliação de qual *malware* se trata quando o antivírus aponta a existência de atividade maliciosa no arquivo analisado. Em caso de ausência de "*malwares*" ou falso-negativos, o seu resultado é denotado como "benigno" ou "omissão" quando o antivírus não se dispõe a examinar o arquivo suspeito.

Para os arquivos maliciosos, a taxa de identificação de *malware* pelos softwares antivírus disponibilizados pela VirusTotal variou de 0% a 99,79%. De maneira global, essas ferramentas conseguiram revelar adequadamente 66,71% dos *malwares* apresentados, possuindo um desvio padrão de 36,18%. O elevado desvio padrão é uma comprovação de que algumas soluções comerciais de antivírus apresentaram êxito na identificação desse *malware*, porém outras foram absolutamente ineficazes. Ou seja, apenas uma parcela dos softwares antivírus disponíveis no mercado possui listas de bloqueio robustas o bastante para proporcionar uma taxa de detecção considerável em relação aos arquivos nocivos que foram previamente catalogados.

Um outro aspecto relevante a ser destacado é que, em geral, 23,85% dos antivírus erroneamente declararam arquivos *malware* como benignos, ou seja, resultando em falsos negativos com valor de desvio padrão de 31,15% e, adicionalmente, uma média de 9,43% dos antivírus prestaram nenhuma avaliação sobre os arquivos submetidos, se omitindo com uma taxa de desvio padrão de 26,36%. Essa considerável proporção de falsos negativos e omissões acarreta elevado grau de riscos de infecções que poderiam danificar permanentemente os computadores e os dados de seus usuários, órgãos públicos ou corporações. Com esses resultados, pode-se concluir que as soluções antivírus comerciais são ineficientes na defesa dos sistemas contra *malwares*

em tempo real.

Em adição a todos os limites citados anteriormente, os antivírus comerciais tampouco disponibilizam aos usuários dados relevantes sobre os arquivos contaminados e, geralmente, os *malwares* recebem somente denominações genéricas. Dessa forma, parece razoável dizer que os diagnósticos apresentados pelas soluções antivírus disponíveis no mercado não servem de apoio para permitir que o usuário compreenda o grau de periculosidade do *malware* ou suas ações maliciosas no sistema.

Se um arquivo contaminado for submetido à avaliação de dois programas antivírus distintos, é bem provável que ambos informem denominações absolutamente divergentes sobre um mesmo arquivo malicioso. E mais, se forem analisadas variantes do mesmíssimo arquivo malicioso, com alterações irrelevantes no código, as versões poderão receber nomes completamente distintos por um mesmo antivírus comercial. Devido a essa inexistência de padrões na nomenclatura para *malware*, torna-se ainda mais difícil a implantação de novas metodologias de segurança cibernética, uma vez que é preciso lidar com cada categoria de *malware* de maneira específica.

Concluindo, em virtude dessa confusão de análise e denominações aleatórias proporcionadas pelos antivírus, é um desafio enorme colaborar para o avanço da segurança cibernética. De acordo com a Tabela 1.2, não se pode esperar que técnicas de aprendizagem de máquina possam generalizar a identificação de arquivos maliciosos apenas com os dados de detecção disponibilizados pelos antivírus disponíveis no mercado.

Tabela 1.1: Resultados dos antivírus comerciais.

Antivirus	Detecção (%)	Falso Negativo (%)	Omissão (%)
ESET-NOD32	99,79%	0,19%	0,02%
NANO-Antivirus	99,69%	0,26%	0,05%
Comodo	99,63%	0,06%	0,31%
Avira	99,61%	0,23%	0,16%
DrWeb	99,56%	0,29%	0,15%
VBA32	99,27%	0,54%	0,19%
Microsoft	99,18%	0,23%	0,58%
Cyren	99,13%	0,75%	0,12%
SentinelOne	98,94%	1,01%	0,05%
Invincea	98,5%	0,09%	1,41%
BitDefenderTheta	1,84%	70,55%	27,61%
Elastic	1,42%	0,16%	98,42%
Gridinsoft	1,17%	0,23%	98,6%
Zoner	0,93%	98,9%	0%
Kingsoft	0,39%	98,9%	0,17%
Baidu	0,05%	99,57%	0,37%
TheHacker	0,01%	0%	99,99%
Avast-Mobile	0%	98,33%	1,67%
Trustlook	0%	0,12%	99,88%
Babable	0%	0,05%	99,95%

Fonte: Repositório de análise de malwares (InstallCore, 2025).

1.3 Justificativa

A elaboração de um programa antivírus (antimalware) dedicado à detecção preventiva do *malware InstallCore* é fundamental em virtude de sua habilidade de instalação de *adware*, PUPs (programas potencialmente indesejados) e outras ameaças indesejadas sem o consentimento claro dos usuários. Como o *InstallCore* geralmente se disfarça de instaladores legítimos e contorna as defesas tradicionais, um antivírus exclusivo, implementado por meio de redes neurais, pode identificar padrões de comportamento de sua conduta maliciosa antes de sua execução, protegendo o sistema contra alterações indesejadas, perda de dados ou degradação do desempenho. A detecção proativa desse *malware* permite maior segurança para o usuário final, evitando infecções antes de causarem danos significativos ao sistema.

Tabela 1.2: Resultados da submissão de dois malwares para o VirusTotal.

Antivirus	$VirusShare_A$	$VirusShare_B$
MicroWorld-eScan	False Negative	False Negative
NANO-Antivirus	Riskware.Win32. InstallCore.pljgw	Riskware.Win32. InstallCore.nmwea
Avast	FileRepMalware [PUP]	Win32:InstallCore-HF [PUP]
Kaspersky	not-a-virus:HEUR:AdWare. Win32.InstallCore.gen	False Negative
McAfee-GW-Edition	BehavesLike.Win32.Generic.hc	BehavesLike.Win32.Fareit.hc
Microsoft	PUA:Win32/InstallCore	PUA:Win32/InstallCore
AVG	FileRepMalware [PUP]	Win32:InstallCore-HF [PUP]
ESET-NOD32	a variant of Win32/InstallCore.BH potentially unwanted	a variant of Win32/InstallCore.BH potentially unwanted
McAfee	Artemis!0003070A022F	Artemis!000831E751AB
Avira	PUA/InstallCore.Gen	PUA/InstallCore.Gen
Malwarebytes	PUP.Optional.InstallCore	Adware.Agent
Emsisoft	Application.InstallCore (A)	Application.InstallCore (A)
Ikarus	AdWare.Generic	AdWare.InstCore
MAX	malware (ai score=99)	malware (ai score=97)
TrendMicro-HouseCall	TROJ_GEN.R002C0OF221	TSPY_INSTALLCORE _BK082AAB.TOMC
Emsisoft	Application.InstallCore (A)	Application.InstallCore (A)
Ikarus	AdWare.Generic	AdWare.InstCore
Arcabit	False Negative	False Negative
Tencent	Adware.Win32.Installcore.e	Adware.Win32.Installcore.e
VIPRE	Trojan.Win32.Generic!BT	Trojan.Win32.Generic!BT

Fonte: Repositório de análise de malwares (InstallCore, 2025)).

1.4 Objetivo Geral

Elaborar um antivírus dedicado, utilizando redes neurais extremas, que seja apto a identificar um software maligno com base no reconhecimento do padrão de características de arquivos infectados, sem a utilização de tecnologias ultrapassadas como listas de bloqueio.

1.4.1 Objetivos específicos

- Analisar os diversos métodos de enfrentamento a *malware* adotados por antivírus comerciais
- Identificar as diversas falhas e brechas deixadas pelas práticas atuais de defesa

virtual.

- Verificar a possibilidade da utilização de redes neurais extremas para a detecção preventiva desses *malwares*.
- Criar uma base de dados com arquivos malignos e benignos para que seja possível a realização do treinamento.
- Validar a hipótese levantada da utilização de redes neurais extremas a partir da elaboração de um antivírus baseado nessa metodologia.
- Comparar o resultado com tecnologias antivírus pré-existentes para comprovar o sucesso do objeto proposto.

1.5 Organização do TCC

O teor da presente tese divide-se em 6 capítulos e em um apêndice. É possível verificar as referências em suas últimas páginas. A seguir, encontra-se um resumo dos seguintes capítulos.

Capítulo 2. Fundamentação Teórica, capítulo que trata da análise matemática da rede neural empregada, bem como do embasamento teórico e da justificativa.

Capítulo 3. Estado da arte, esse capítulo traça um perfil das produções acadêmicas sobre a confecção de antivírus dotados de inteligência artificial.

Capítulo 4. Metodologia, capítulo que elucida os procedimentos, métodos, equipamentos necessários à criação do antivírus proposto.

Capítulo 5. Resultados, capítulo onde são apresentados os valores obtidos do processo de treinamento e teste do proposto software antivírus baseado em rede neural.

Capítulo 6. Conclusão, parte final onde é apresentado um resumo dos principais pontos discutidos ao longo do trabalho, destacando os resultados mais importantes.

Capítulo 2

Fundamentação Teórica

Apesar de movimentar um mercado bilionário, os antivírus comerciais apresentam técnicas retrógradas baseadas em listas de bloqueio (denúncias prévias). De modo a suprir as limitações dos antivírus comerciais, o estado-da-arte propõe extrair características do arquivo, de maneira preventiva, antes de executá-lo. Torna-se possível investigar a intenção maliciosa do arquivo suspeito preventivamente (LIMA et al.,2021). Dessa forma, é viável determinar a intenção maliciosa do arquivo antes mesmo dele ser executado (clicado) pelo usuário. Através da análise do aplicativo suspeito, o estado-da-arte investiga se o arquivo auditado cria, exclui, altera e faz o *download* de outros arquivos pela internet. Além disso, o estudo do aplicativo suspeito torna viável o rastreio preventivo do tráfego de rede provocado pelo arquivo suspeito.

Ao invés de estratégias remediativas, as técnicas de aprendizado de máquina (*machine learning*) são capazes de detectar preventivamente a intenção maliciosa do *SDK InstallCore*. Técnicas de *machine learning* conseguem automatizar de forma inteligente muitas tarefas, analisando milhares de arquivos, extraindo características deles e os ponderando estatisticamente. O uso de *Machine learning* pode contribuir para o avanço da segurança da informação. Existem iniciativas, mas os antivírus comerciais ainda empregam metodologias retrógradas (LIMA et al.,2021)(PINHEIRO et al.,2022).

O trabalho proposto emprega *machine learning* visando detectar aplicativos ma-

liciosos preventivamente. A meta é identificar, de forma estatística, comportamentos previamente classificados como suspeitos em um tempo de resposta viável a um usuário comum. Então não é necessário aguardar meses ou anos até que o executável suspeito seja denunciado por uma vítima, assim como os antivírus comerciais atuam. No presente capítulo será introduzido o tema de máquinas de aprendizado estatístico. Elas constituem o mecanismo empregado pelo estado-da-arte visando o reconhecimento de padrões dos *malware*.

Quanto ao reconhecimento de padrão de *malware*, uma tarefa essencial diz respeito à atribuição de uma classe (rótulo) a cada arquivo investigado a partir de suas características. Com base em um conjunto de arquivos, chamado de conjunto de treinamento, é possível formular uma hipótese sobre as distintas classes atreladas ao antivírus inteligente proposto. Logo cabe ao classificador estimar a classe de um arquivo inédito através da comparação entre as características do seu comportamento auditado e àquelas captadas durante a sua etapa de treinamento.

O presente trabalho aplica-se às redes neurais ELMs (*Extreme Learning Machine* – Máquinas de Aprendizado Extremo) na área de segurança da informação especificamente no reconhecimento de padrão de *malware*. As redes ELMs têm como principal característica a velocidade de treinamento e a previsão de dados quando comparadas às redes neurais baseadas em retropropagação de dados e *Deep Learning*.

As Extreme Learning Machines (ELMs) mostram-se particularmente adequadas para aplicação na Perícia Forense Digital, considerando o rápido crescimento no número de novos *malwares*. Estima-se que, a cada segundo, são lançados oito novos *malwares* (INTEL, 2022). Nesse contexto, é essencial que o tempo de treinamento dos sistemas de detecção, como os antivírus, acompanhe o ritmo de criação de ameaças. Caso o treinamento seja excessivamente demorado, o antivírus recém-desenvolvido pode já estar obsoleto no momento de seu lançamento, uma vez que novas vulnerabilidades podem já ter surgido, demandando um novo ciclo de treinamento.

2.1 Redes Neurais Extremas

Nesse contexto, é essencial entender que as redes neurais têm-se destacado como uma das soluções mais eficientes para a identificação de padrões. Como efeito colateral, um aspecto-chave a ser considerado é a prevenção da estagnação do treinamento em regiões de baixo interesse no processo de aprendizado das redes neurais clássicas (HUANG, 2000). Para contornar esse desafio, são utilizadas técnicas de gerenciamento de arquitetura de rede neural.

Além disso, uma desvantagem comum das redes neurais clássicas é o extenso período de aprendizado necessário para que a solução consiga realizar uma classificação correta. Apesar da acurácia extremamente alta que esses mecanismos podem oferecer, a fase de aprendizado das redes pode ser demorada, levando dias para ser concluída.

A principal diferença entre as ELM (*Extreme Learning Machine*) e as redes convencionais é sua alta velocidade de aprendizado, mantendo a acurácia no reconhecimento de padrão. Essas redes funcionam com uma única camada oculta, não iterativa e dependem exclusivamente de um processo de análise que permite calcular os pesos da rede de saída a partir de uma inicialização aleatória dos pesos sinápticos.

As ELM têm sido amplamente aplicadas em diversos campos, incluindo a Engenharia Biomédica (AZEVEDO and *et al.*, 2015AZEVEDO and *et al.*, 2020LIMA et al., 2016LIMA et al., 2020LIMA et al., 2014PEREIRA, 2020). Graças ao desempenho excepcional desse modelo, é viável que ele contribua para o avanço na área de segurança em sistemas virtuais. A pesquisa apresentada visa explorar a aplicação de ELMs no setor de segurança da informação, especialmente em relação ao mapeamento de padrões de *malware*.

Do ponto de vista matemático, em um sistema neural 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 exemplares para o aprendizado. A camada oculta, h_j, é composta por m neurônios, descritos pelo conjunto

$\left\{h_j \in \mathbb{R};: j \in \mathbb{N}^*;: j = 1, ...m\right\}$.

Como o ELM possui poucas etapas, seu processo de aprendizado é mais rápido. Inicialmente, os valores dos pesos de entrada w_{ji} e dos vieses b_{jt} são definidos aleatoriamente. Considerando uma função de ativação $f : \mathbb{R} \to \mathbb{R}$, o modo de aprendizado é distribuído da seguinte forma:

- Produção aleatória dos pesos w_{ji}, que ligam as fases de entrada e oculta, e polarização b_{jt}
- Cálculo da matriz H, que representa a resposta dos neurônios da camada oculta.
- Cálculo da matriz de ponderação de resposta $\beta = H^{\dagger}Y$, onde $H^{\dagger}$ é a matriz pseudoinversa de Moore-Penrose da matriz H, e Y representa 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 o conceito de matriz pseudoinversa, é essencial notar como o princípio da matriz inversa está diretamente relacionado à matriz identidade I. Quando uma matriz quadrada H é multiplicada pela sua inversa H^{-1}, o resultado é a matriz identidade I. No entanto, no caso de uma matriz não quadrada, é gerada uma matriz aproximadamente inversa; pseudoinversa, representada por $H^{\dagger}$. Essa matriz pseudoinversa tem a propriedade de ajustar os valores das sinapses entre os neurônios, deslocando os pesos sinápticos para longe do limite de decisão, em direção às bordas da diagonal complementar.

Além disso, a matriz H (que corresponde à saída dos neurônios da camada oculta) é calculada utilizando o *kernel* φ, com as informações e os valores de carga apresentados na matriz da Eq. (2.1). Os pesos de retorno β e a matriz de resultado desejado Y são definidos, 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 mencionado anteriormente, define a relação matemática utilizada para o treinamento da rede neural ELM. Normalmente, uma rede neural aplica um *kernel* linear, que é formulado na Eq. (2.4) e cujos resultados estão apresentados na Tabela 5.2. A função φ é dependente de $f(x_{t,1\ldots n}, ; w_{1\ldots m,1\ldots n}, ; b_{1\ldots m,t})$.

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

Com base no *kernel*, esse método de aprendizado é extremamente relevante, pois permite a criação de um mapeamento não linear das informações sem a necessidade de aumentar o número de variáveis ajustáveis ou a frequência de treinamento normalmente requerida pelas redes neurais baseadas em retropropagação. A Eq. (2.5) define um *kernel* sigmoidal φ para uma rede ELM, cujos resultados estão apresentados na Tabela 5.3.

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

2.2 Estudos Preliminares: Morfologia Matemática

No presente trabalho, são empregadas as mELMs (ELMs morfológicas), ou seja, ELM com núcleos de camada oculta inspirados em operadores morfológicos de processamento de imagem de Erosão e Dilatação. O trabalho proposto estima que os *kernels* morfológicos sejam capazes de se adequar a qualquer fronteira decisão. A Morfologia Matemática diz respeito ao estudo das formas dos corpos presentes nas imagens através do uso da teoria matemática de intersecção e união de conjuntos (PINHEIRO et al.2022). As operações morfológicas lidam naturalmente com a detecção das formas dos corpos presentes nas imagens. Ao interpretar a fronteira de decisão de uma rede neural como uma imagem n-dimensional, onde n diz respeito à quantidade de características extraídas. As mELMs são capazes de naturalmente detectar e modelar as regiões n-dimensionais mapeadas às distintas classes.

Morfologia Matemática é uma teoria completa de processamento não-linear amplamente utilizada no processamento de imagens digitais. Várias aplicações específicas são construídas a partir da Morfologia Matemática como detecção e segmentação de objetos, extração de características, dentre outras. A morfologia é baseada nas transformações de formas, preservando as relações de inclusão dos objetos. Há duas operações morfológicas fundamentais: Erosão e Dilatação (PINHEIRO et al.,2022). A Morfologia Matemática pode ser considerada uma teoria construtiva porque todas as operações são construídas tendo como base as Erosões e Dilatações. Do ponto de vista matemático, no que diz respeito às funções de Erosão e Dilatação, o comportamento se apresenta conforme as equações (2.6) e (2.7), respectivamente.

No contexto de associado de processamento de imagem, a Erosão é uma operação

que reduz o tamanho dos objetos em uma imagem, removendo *pixels* das bordas. Isso faz com que as regiões mais claras diminuam e as regiões mais escuras cresçam. Já a Dilatação, essa corresponde a uma operação que expande os objetos em uma imagem, adicionando *pixels* às bordas. Isso resulta no aumento das regiões mais claras e na redução das regiões mais escuras. Sendo o Pixel a menor unidade de uma imagem digital, representando um ponto único com informações de cor ou intensidade luminosa. No contexto do texto, um pixel é descrito pelo par $(k, f(k))$, onde k é a posição espacial e $f(k)$ é o valor associado ao pixel.

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

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

Conforme as equações supracitadas, as seguintes expressões: $f : S \to \{0.1\}$ e $g : S \to \{0.1\}$ constituem representações normais e em matriz que possuem um formato definido como S, sendo $S \in \mathbb{N}^2$. O valor k corresponde à região que está vinculada com o valor associado a $f(k)$. w consiste na fórmula matricial de $f(k)$, incorporada por g. Por sua vez, a operação máxima encontra-se discriminada na equação (2.6), ao passo em que a operação mínima consta na expressão (2.7). g é designado elemento estruturador de Erosão e Dilatação (SANTOS, 2011). O elemento estruturante é a matriz ou conjunto de *pixels* que define a forma e o tamanho da vizinhança usada para realizar operações morfológicas, como erosão e dilatação. Ele atua como um "molde"aplicado sobre a imagem. $\overline{g}$ é o elemento dual de g.

O processamento apresentado na equação (2.6) inicia-se mediante a negação do elemento estruturante g, seguido pelo emprego da expressão máxima $\vee$, indicada por $f(w) \vee \overline{g}(k - w)$, na qual $f(w)$ diz respeito à matriz da imagem original e se denomina em termos técnicos como região ativa da imagem. Finalmente, o valor $\epsilon_g(f)(k)$, localizado em k, da área erodida da imagem obtém o mínimo entre os máximos valores por meio do operador de interseção. Dessa forma, a erosão leva ao

crescimento de regiões mais escuras, e à atenuação das zonas mais claras.

O procedimento de dilatação está presente na equação (2.7), em que a operação de mínimo $f(w) \wedge g(k-w)$ será primeiramente aplicada, na qual $f(w)$ corresponde à matriz que representa a imagem. Conforme o mecanismo de unificação, o parâmetro $\delta_g(f)(k)$ na região k obtém o máximo dentre os mínimos valores. Dessa forma, a dilatação tenderá a ampliar as zonas claras e atenuar as áreas sombrias.

A Fig. 2.1 (b) e a Fig. 2.1 (c) demonstram o resultado da operação de Erosão e de Dilatação em uma mesma imagem original: Fig. 2.1 (a). Na imagem erodida, o objeto alvo é "murchado". Na imagem dilatada, o objeto alvo é dilatado, como o próprio nome sugere. Ao traçar uma analogia entre a operação de processamento de imagem e as redes mELM autorais, o *kernel* de Dilatação expande a região atrelada à classe alvo (ex .*malware*). Por sua vez, o *kernel* de Erosão expande a região pertencente à contra-classe (ex. benigno).

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

2.3 Redes Neurais Morfológicas

Um dos grandes desafios, em redes neurais artificiais, diz respeito a encontrar um *kernel* de modo que otimize a fronteira de decisão entre as classes de uma dada aplicação. Em redes neurais ELM, um *kernel* Linear, por exemplo, é capaz de resolver um problema linearmente separável, como o visto na Fig. 2.2 (a). Seguindo o mesmo raciocínio, *kernels* Sigmoide, RBF e Senoide são capazes de resolver pro-

blemas separáveis por função Sigmoidal, Radial e Senoidal, vistos na Fig. 2.2 (b), na Fig. 2.2 (c) e na Fig. 2.2 (d), respectivamente.

Uma boa capacidade de generalização da rede neural pode depender de uma escolha ajustada do *kernel*. O melhor *kernel* pode estar subordinado ao problema a ser resolvido. Como efeito colateral, a investigação de diferentes *kernels* é geralmente um processo custoso envolvendo validação cruzada combinada com diferentes condições iniciais aleatórias. A investigação de distintos *kernels* pode ser necessária, caso contrário, a rede neural composta por um *kernel* desajustado pode gerar resultados insatisfatórios.

Como contra-exemplo, observe o emprego do *kernel* Linear aplicado a distribuições Sigmoide e Senoide apresentadas na Fig. 2.3 (a) e na Fig. 2.3 (b), respectivamente. As acurácias das classificações expostas na Fig. 2.3 (a) e na Fig. 2.3 (b) são de 78,71% e 73,00%, respectivamente. Visualmente, é possível observar que o *kernel* Linear não mapeia as fronteiras de decisões das distribuições Sigmoide e Senoide de forma adequada.

A Fig. 2.4 (a), a Fig. 2.4 (b), a Fig. 2.4 (c) e a Fig. 2.4 (d) exibem a atuação do mELM *kernel* Erosão nas distribuições Linear, Sigmoide, Radial e Senoide, com as respectivas acurácias de 100%, 93,07%, 98,18% e 99,50%. A Fig. 2.5 (a), a Fig. 2.5 (b), a Fig. 2.5 (c) e a Fig. 2.5 (d) exibem a atuação do mELM *kernel* Dilatação nas distribuições Linear, Sigmoidal, Radial e Senoidal, com as respectivas acurácias de 100%, 95,05%, 98,18% e 99,50%. Visualmente, é possível observar que as mELMs mapeiam distintas distribuições, referentes a diferentes problemas, de forma satisfatória. Cabe ressaltar que os dois atributos (características) estão normalizados sobre um mesmo limite inferior e superior.

A explicação do sucesso dos *kernels* mELMs diz respeito à sua capacidade de modelar qualquer fronteira de decisão visto que o seu mapeamento não obedece às superfícies geométricas convencionais como elipse e hipérbole. O mapeamento da fronteira de decisão, realizado pelos *kernels* mELMs, emprega as coordenadas no espaço n-dimensional das amostras reservadas ao treinamento, onde n diz respeito

à quantidade de características extraídas. Logo as mELMs são capazes de naturalmente detectar e modelar as regiões n-dimensionais referentes às distintas classes por empregar a Morfologia Matemática a qual lida naturalmente com a detecção das formas dos corpos presentes nas imagens LIMA et al.2014LIMA et al.2021PINHEIRO et al.2022.

2.4 Síntese do Capítulo

O capítulo abordou a evolução da detecção de *malware*, destacando as limitações dos antivírus comerciais, que dependem de listas de bloqueio. Em contraste, o estado da arte propõe análise preventiva, identificando padrões maliciosos pelo comportamento do aplicativo, como manipulação de arquivos e tráfego de rede. *Machine Learning* (ML) surge como alternativa promissora, permitindo a análise de grandes volumes de dados e superando abordagens reativas. O estudo propôs o uso de *Extreme Learning Machines* (ELMs) para detectar *malware* rapidamente, sem depender de denúncias. As ELMs treinam e fazem previsões mais rápido que redes neurais tradicionais, sendo úteis na Perícia Forense Digital diante do crescente número de ameaças. Enquanto redes convencionais enfrentam longos tempos de treinamento e risco de estagnação, as ELMs utilizam uma única camada oculta e um processo não iterativo, tornando a adaptação mais eficiente. A pesquisa incorpora ELMs morfológicas (mELMs), aplicando Morfologia Matemática para otimizar a separação entre classes de *malware* e arquivos benignos. Operadores morfológicos como Erosão e Dilatação são adaptados para melhorar a flexibilidade das regiões decisórias. Experimentos preliminares indicam que essa abordagem pode aprimorar a detecção de *malware* e fortalecer a segurança digital em tempo real.

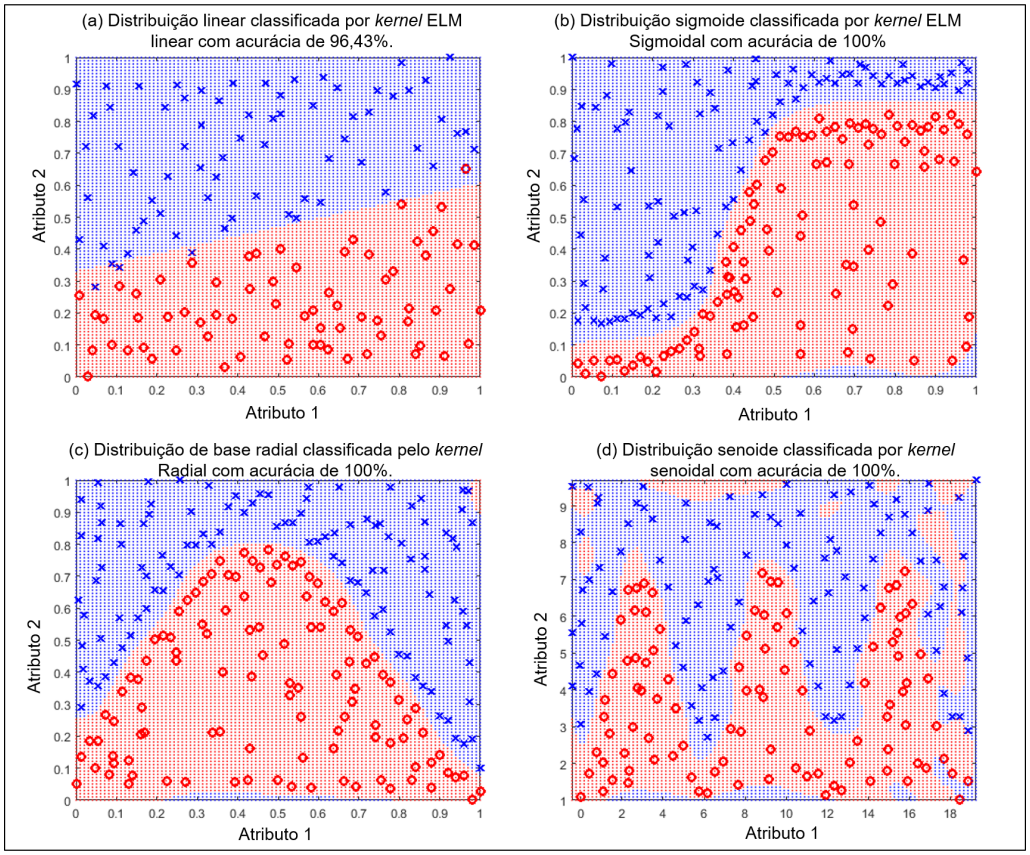


Figura 2.2: Atuações bem-sucedidas dos *kernels* compatíveis com os conjuntos de dados. Fonte: mELM(2025)

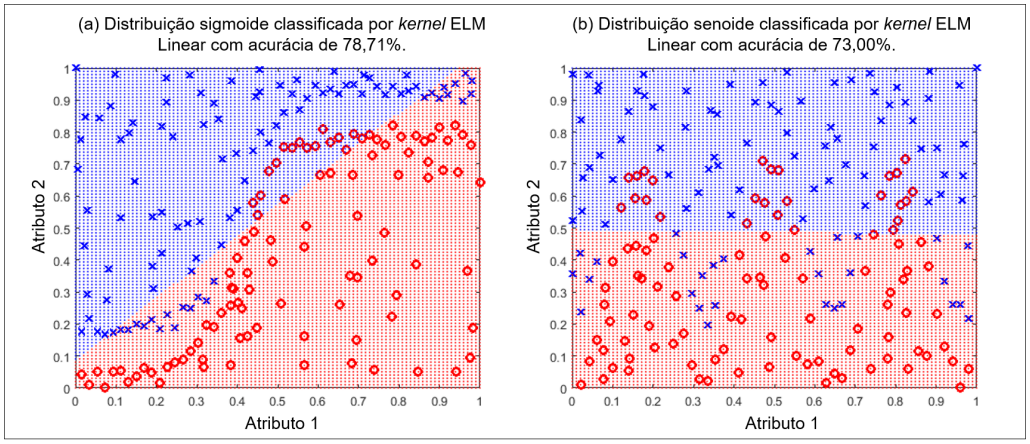


Figura 2.3: Atuações malsucedidas do *kernel* Linear em conjuntos de dados não-linearmente separáveis. Fonte: mELM(2025)

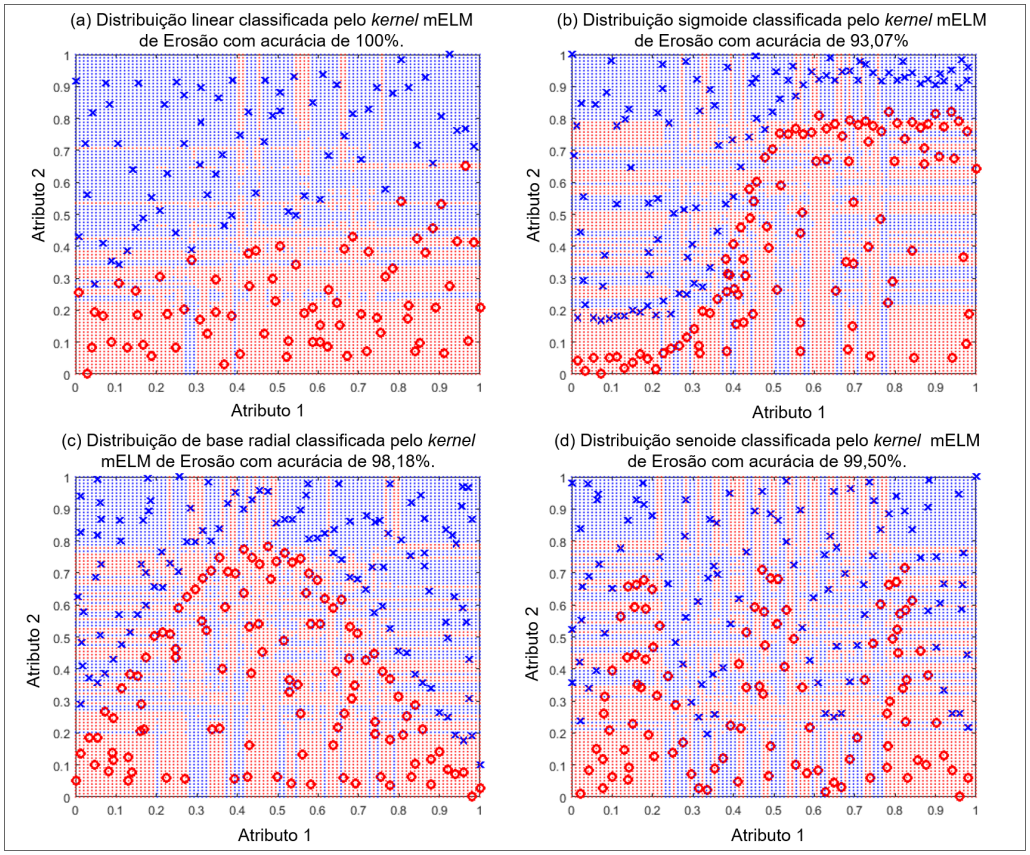


Figura 2.4: Atuações bem-sucedidas do mELM *kernel* Erosão em diversos conjuntos de dados. Fonte: mELM(2025)

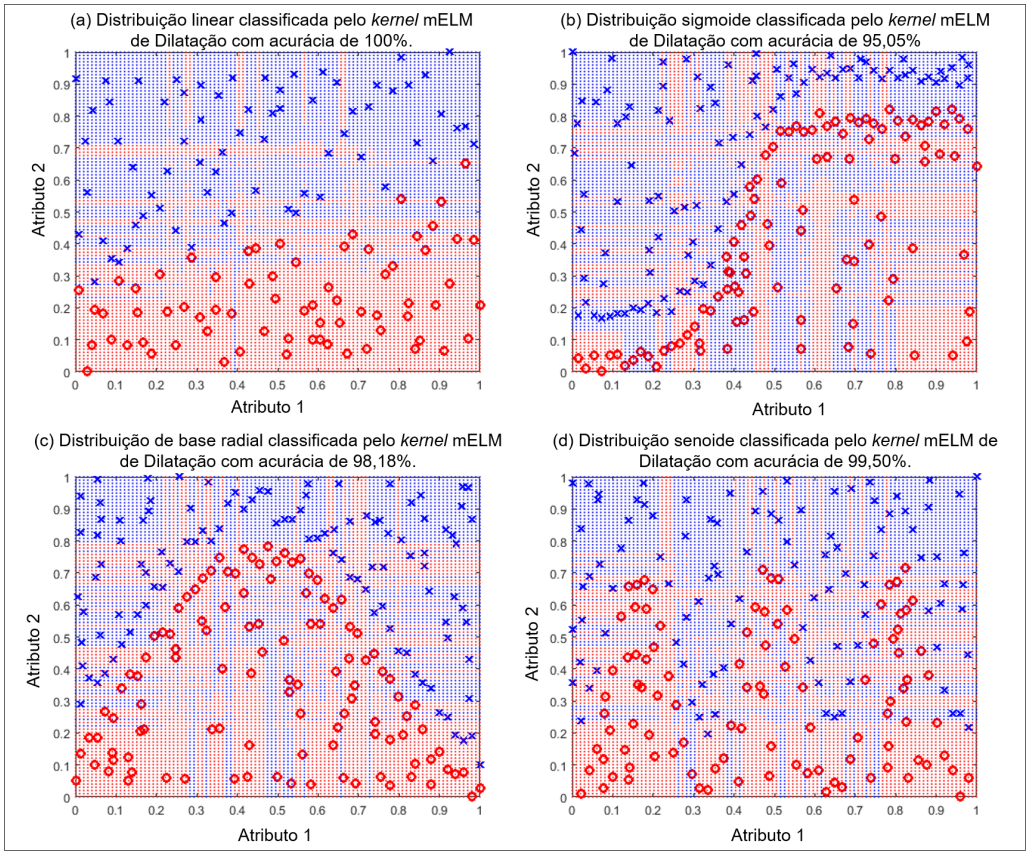


Figura 2.5: Atuações bem-sucedidas do mELM *kernel* Dilatação em diversos conjuntos de dados. Fonte: mELM(2025)

Capítulo 3

Estado da Arte

Mesmo que já se questione essa metodologia por mais de uma década, o modelo de comportamento dos sistemas de antivírus permanece fundamentado em associações entre arquivos suspeitos e registros em bases de dados intituladas listas de bloqueio (LIMA, 2020). Em suma, o objeto inspecionado é confrontado com os *malwares* classificados na lista de proibições; assim, caso essa lista encontre-se defasada, o *malware* não será identificado e provocará um contágio.

A Tabela 3.1 lista antivírus importantes e atualizados (de última geração). Os antivírus sofisticados têm como alvo sistemas específicos (Internet das Coisas, *Windows*). Todos eles, portanto, tiveram suas respectivas técnicas de reconhecimento de padrão replicadas usando nossos materiais e métodos para evitar comparações injustas. Utilizamos essas redes a fim de comparar desempenho entre *antimalware* de redes rasas e de redes profundas como os de LIMA et al. (2021) e o de SU and VASCONCELLOS (2018), respectivamente, em confronto à metodologia proposta.

Os autores utilizaram um critério *ad hoc* para construir o estado da arte, o qual levou em consideração tanto o número de citações quanto a taxa de acurácia. Ademais, artigos em formato de pré-*print* ou que não haviam passado por revisão por pares foram excluídos.

As citações do *Google Scholar* desempenham um papel fundamental na identificação de pesquisas relevantes, especialmente diante da vasta quantidade de artigos publicados por diferentes editoras e conferências. Nesse contexto, considerando a

inviabilidade de se ler todos esses artigos, as citações tornam-se um atalho valioso. Elas indicam que autores respeitáveis endossam o trabalho, além de deixar inferido o impacto da conferência ou do periódico na comunidade científica. Além disso, foi avaliada a possibilidade de execução das redes neurais de acordo com a quantidade de recursos disponíveis; dada essa avaliação, redes profundas como as de LO and YANG, (2019) e KALASH and ROCHAN, (2018) necessitariam de um *datacenter* para serem processadas, o que as torna inviáveis de serem consideradas, dada a urgência de atualização em caso de surgimento de uma nova ameaça.

Tabela 3.1: Antivírus do estado da arte.

Autores	Tipos de redes neurais	Técnica de redes neurais	Dispositivo	Citações	Acurácia
Antivirus Autoral	Redes neurais rasas	ELM Morfológica (mELM)	Computador *Desktop*	–	99,99%
LIMA, *et al.* (2021) [16]	Redes neurais rasas	*Perceptron* Multi-camadas(MLP)	Computador *Desktop*	27	98,32%
SU, *et al.* (2018) [24]	Redes neurais profundas	Treinamento *batch* (CNN)	IoT para x86	423	94,00%

Fonte: O autor (2025).

A academia tem proposto o desenvolvimento de antivírus equipados com inteligência artificial. LIMA et al., 2021 elaborou uma ferramenta antivírus com capacidade de identificar *malwares* (*Windows*) cuja acurácia é, em média, de 98,32%. Nesse sistema em uso, o arquivo executável é submetido a processos de desconstrução, de forma que o objetivo nocivo do executável seja devidamente averiguado. Na implementação do antivírus LIMA et al., 2021, extrai-se 630 características para cada um dos arquivos empregadas, e essas propriedades podem ser inseridas como neurônios de entrada na rede neural artificial. A categorização do software antivírus permite distinguir os arquivos de 32 bits entre duas classificações: software não prejudicial e software potencialmente prejudicial. A rede neural empregada não é profunda.

A partir de uma perspectiva diferente, os programas antivírus que têm como fundamento redes neurais profundas, da mesma forma, atingiram altos níveis de acurácia. SU and VASCONCELLOS, 2018 alcançaram acurácia média por volta de

94,00% para detectar *malwares* de *IoT* (Internet das Coisas) (SU and VASCONCELLOS, 2018). A rede neural profunda apresenta uma arquitetura de seis camadas. Dessas 6, existem 3 níveis de aprendizado que possuem os determinados pesos: 2 nas camadas de desdobramento e 1 camada de conexão plena. O treinamento para essa rede é efetuado em um total de 5.000 iterações, em um lote de treino com tamanho 32 e com índice de aprendizado de 0,0001.

O emprego do *deep learning*, nos mais distintos campos, tem se tornado cada vez mais comum. Porém a réplica dessas soluções enfrenta desafios significativos devido às exigências de hardware e ao alto custo computacional. As redes *deep learning* de propósito geral, por exemplo, demandam grandes *datacenters* e longos períodos de treinamento. Um caso ilustrativo é o sistema de reconhecimento facial *FaceNet* do Google, que levou anos para ser treinado (SCHROFF et al., 2015). Esse processo envolveu a utilização de uma vasta infraestrutura global de *datacenters* e dados de centenas de milhões de usuários de diversas etnias e culturas, permitindo que a rede profunda adquirisse capacidade de generalização.

Nesse contexto, torna-se inviável para a academia, com seus recursos restritos, utilizar redes de *deep learning* de propósito geral em seus projetos. Para contornar esse obstáculo, o presente trabalho optou por reproduzir um antivírus baseado em uma rede de baixa complexidade proposta por LIMA et al., 2021 e uma *deep learning* customizada desenvolvida por SU and VASCONCELLOS, 2018. No capítulo de resultados, o antivírus desenvolvido neste trabalho será comparado com as soluções propostas por LIMA et al., 2021 e SU and VASCONCELLOS, 2018.

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.

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.

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.

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 parâ-

metros 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 teórico 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 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 repositório apresentado na Tabela 3.2.

[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.

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 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 InstallCore*. 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-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* *InstallCore* 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

O presente capítulo tem como objetivo descrever a abordagem utilizada no desenvolvimento do antivírus baseado em ELM e está dividido em quatro subseções principais:

- **Métodos e materiais utilizados**: Onde apresenta-se o conjunto de dados composto por amostras de malware e programas benignos, com foco no malware InstallCore.
- **Metodologia proposta**: Este descreve o desenvolvimento de um antivírus baseado em IA, utilizando um computador com recursos específicos. O processo inclui coleta de dados, refinamento, classificação com a técnica Extreme Learning Machine (ELM) e validação cruzada.
- **Extração de recursos**: Detalha a desmontagem de arquivos executáveis para extrair características. Esses recursos foram usados para identificar padrões de comportamento malicioso.
- **Classificação**: Explica o uso de redes neurais como classificador para distinguir entre arquivos benignos e maliciosos.

4.1 Métodos e materiais utilizados

O presente trabalho apresenta um conjunto de dados destinado à identificação de programas executáveis considerados benignos e *malwares* com características de *InstallCore*. O conjunto é composto por 9.405 amostras de *malware* e 3.135 programas executáveis benignos. Utilizando a técnica de particionamento em blocos, as amostras malignas foram separadas em três conjuntos de 3.135 arquivos cada.

As ameaças virtuais analisadas foram coletadas a partir de fontes de dados disponibilizadas por organizações de pesquisa especializadas no assunto, como o *VirusShare*, que é um repositório de amostras de *malware* destinado a fornecer exemplares a analistas de segurança, peritos forenses e outros interessados. Em relação aos arquivos executáveis inofensivos, estes foram obtidos de repositórios de software benignos, incluindo *SourceForge*, *Sysinternals* e *GitHub*.

Cada um desses conjuntos de amostras malignas foi treinado em conjunto com o único pacote de amostras benignas. Os resultados dessas execuções foram multiplexados, permitindo comparações de desempenho entre os casos de teste e de treino. Dessa forma, o banco de dados foi adequadamente preparado para aprendizado por meio de inteligência artificial, a partir da combinação de pacotes dos dois tipos de executáveis em subpacotes de quantidades equivalentes.

Quando bases de dados apresentam um desequilíbrio significativo entre classes, as redes neurais tendem a favorecer a classe majoritária, comprometendo a acurácia na identificação da classe minoritária. Para mitigar esse problema, uma abordagem amplamente utilizada na engenharia biomédica foi adaptada neste trabalho. Essa técnica é aplicada em contextos onde a incidência de doenças, como o câncer, é muito menor em comparação aos casos saudáveis.

A estratégia consiste em repetir o treinamento da rede neural diversas vezes, alternando os subconjuntos da classe majoritária em cada execução para equilibrar a base de dados. Essa adaptação resultou na criação de subconjuntos balanceados, permitindo que a rede neural seja exposta a um número igual de exemplos de cada classe. A referida abordagem reduz o viés em favor da classe majoritária, ao mesmo

tempo que preserva a diversidade dos dados. Na engenharia biomédica, essa diversidade é fundamental pois pacientes saudáveis podem apresentar características completamente distintas uns dos outros. Uma hipotética eliminação de exemplos da classe majoritária, como pacientes saudáveis, não é viável para balancear as classes, já que isso comprometeria a representatividade e a generalização dos modelos.

É importante destacar que todos os programas executáveis benignos foram verificados no *VirusTotal*, onde sua autenticidade foi confirmada pelos principais sistemas antivírus disponíveis globalmente. Embora possa parecer paradoxal, obter amostras benignas é uma tarefa bastante desafiadora devido ao critério *ad hoc* estabelecido. Um exemplo ilustrativo é o fato de que, frequentemente, um antivírus identifica como *malware* o instalador de um antivírus concorrente, sem fornecer explicações detalhadas. Em síntese, a obtenção de amostras que não tenham sido denunciadas por nenhum dos 79 antivírus comerciais foi a etapa do trabalho que demandou mais esforço. Não há uma entidade global, similar à Interpol ou a uma embaixada, que possa certificar a ausência de alertas sobre executáveis na rede mundial de computadores.

As análises de detecção realizadas pelo *VirusTotal*, que incluem programas executáveis tanto inofensivos quanto nocivos, estão disponíveis no site da base de dados (InstallCore, 2025). O objetivo de criar o conjunto de informações mencionado é facilitar a reprodução da solução metodológica em futuras pesquisas. Com isso, o trabalho desenvolvido, ao disponibilizar gratuitamente seu conjunto de dados, confere um caráter transparente e imparcial à pesquisa, além de evidenciar a acurácia dos resultados obtidos. Assim, espera-se que essa abordagem metodológica sirva como base para outros trabalhos acadêmicos.

4.2 Metodologia proposta

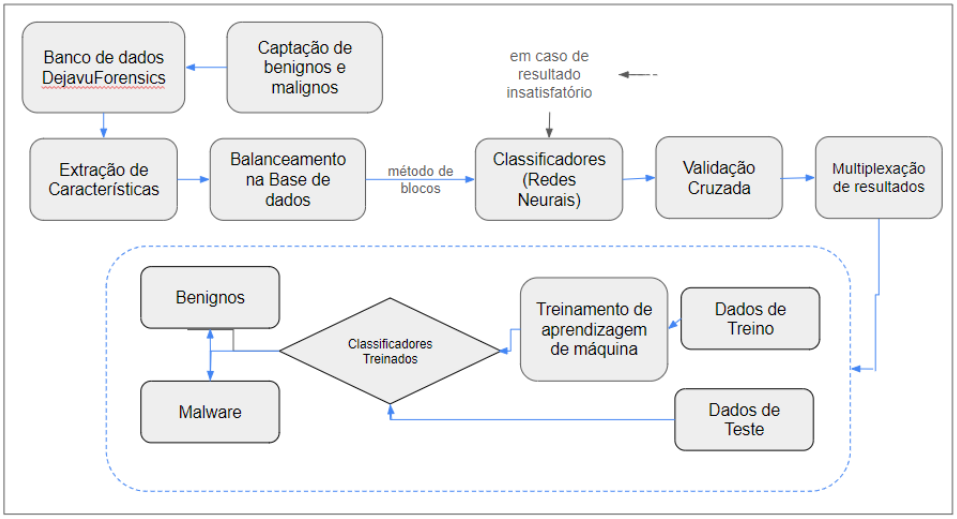
Tendo em vista a constatação das restrições dos programas antivírus disponíveis no mercado, este trabalho visa a elaboração de um programa antivírus, baseado em inteligência artificial, com o intuito de discriminar de maneira antecipada as aplicações maliciosas das inofensivas. É apresentado na Figura 4.1 o diagrama de blocos da proposta metodológica. Em relação ao tratamento das informações, todas as experimentações foram efetuadas em um computador dotado de 24 GB de RAM ("*random access memory*") e de 6 núcleos físicos (processadores) e uma GPU de 2048 núcleos CUDA e 4GB de memória.

O esquema da Figura 4.1 define as partes da metodologia. Inicialmente, a fase da coleta de informações a partir do repositório de dados autoral. Em seguida, depois de obter os recursos, é feito o refinamento adequado desses dados para que sejam processados. Posteriormente, faz-se a classificação desses dados por meio da técnica apresentada (*Extreme Learning Machine*) cujo classificador se encontra disponível no repositório mELM, 2025 e, finalmente, procede-se à validação cruzada a fim de comprovar a efetividade do procedimento elaborado. A ilustração também apresenta a forma de realização do processo de teste e de treinamento desses dados. Além disso, a fim de resguardar em caso de resultados insatisfatórios da rede neural, pode-se retornar para o ponto indicado de treinamento, como denotado na figura, apesar do fato de que a solução proposta possui um desvio padrão insignificantemente baixo.

4.3 Extração de recursos

Ao realizar a coleta de informações dos arquivos executáveis, é aplicado um procedimento de desmontagem. Assim, é possível estudar os mecanismos que constituem o arquivo e, subsequentemente, proceder à respectiva classificação através do sistema neural próprio. A fim de realizar a avaliação, foram extraídos um total de 407 recursos a partir de cada arquivo executável. Ferramentas de origem autoral e o software *Pescanner* foram utilizados de forma a retirar recursos dos executáveis. De

Figura 4.1: Diagrama referente aos procedimentos adotados no presente trabalho.



Fonte: O autor (2025).

modo a simplificar o processo de leitura dos neurônios do nível de entrada, o repositório amplia a definição das características auditadas para antivírus (InstallCore, 2025). Em seguida, são apresentados os grupos de recursos obtidos do executável de pesquisa.

- **Distribuição das Instruções Relacionadas à Importação de Dados**. Gráficos de frequência ilustram as instruções responsáveis pela aquisição de informações (importações).
- **Número de Sub-rotinas que Utilizam *TLS* (Segurança da Camada de Transporte)**. Quantifica-se as rotinas do programa que acionam mecanismos de segurança *TLS* para comunicação segura.
- **Total de Sub-rotinas Envolvidas na Exportação de Dados**. Identifica-se e contabiliza-se as sub-rotinas que desempenham funções de saída de dados (exportações).
- **Histograma das Bibliotecas e APIs Utilizadas pelo Executável**. A análise gráfica apresenta as APIs (Interfaces de Programação de Aplicações) e bibliotecas externas empregadas pelo executável no seu funcionamento.
- **Indícios de Fragmentação e Inicializações Mal-Sucedidas no Disco**

Rígido. Sinais de possíveis fragmentações no sistema de arquivos do disco rígido são investigados, bem como a ocorrência de tentativas de inicialização que não obtiveram sucesso.

- **Modos de Execução de Aplicativos**. Há dois principais métodos de operação para os aplicativos:
  - **Programas com Interface Gráfica (*GUI*)**: Utilizam janelas interativas para interação com o usuário.
  - **Programas Executados no Console**: Operam diretamente em terminais de comando, sem interfaces visuais elaboradas.
- **Características Avaliadas Relacionadas ao Sistema Operacional**. Na investigação forense digital, diversos comportamentos do arquivo testado são verificados, incluindo:
  - Identificação do nome de usuário atualmente logado no sistema.
  - Acesso a funções de API que permitam criar, gerenciar ou manipular perfis de usuários do sistema.
  - Monitoramento do tempo (em milissegundos) desde a inicialização do sistema.
  - Operações específicas em arquivos, como leitura ou modificação.
  - Verificação da versão do sistema operacional *Windows* instalado.
  - Supervisão do tráfego de mensagens internas entre os processos em execução.
  - Alterações na configuração ou nos conteúdos de inicialização do SO (*STARTUPINFO*).
  - Manipulação do *shell* do sistema operacional, incluindo modificações na interface.
  - Alterações nas mensagens exibidas durante o logon no sistema.

  - Configuração de licenciamento e permissões de servidores, incluindo versões específicas como o *Windows Server* 2003.
  - Modificação de configurações de energia ou acessibilidade do sistema.
  - Execução de comandos relacionados a processos, serviços e bibliotecas nativas do sistema.
  - Exclusão de certificados vinculados ao sistema operacional.
  - Operações de manipulação de arquivos, incluindo criação, cópia, exclusão e renomeação.
  - Geração de novos diretórios e buscas por arquivos específicos no sistema.
  - Criação de objetos de serviço para gerenciamento e controle no banco de dados de serviços.
  - Uso de criptografia de dados, prática comum em ataques de *ransomware*, onde os dados são bloqueados e o usuário é coagido a pagar para recuperar o acesso.
  - Acesso a sistemas de arquivos e dispositivos físicos, incluindo gerenciamento de processos, *threads* e tratamento de erros.
  - Ajustes nas propriedades de áudio e dispositivos de som conectados.
  - Monitoramento de portas USB para detectar conexões e interações de dispositivos.
  - Controle de *drivers* de dispositivos e identificação de tipos de unidades de disco (removíveis, fixas, ópticas, etc.).

- **Aspectos Relacionados ao Registro do Sistema Operacional *Windows* (*Regedit*)**. Mesmo após a identificação e remoção de um *malware*, a vítima pode continuar vulnerável devido à permanência de entradas maliciosas no Registro do sistema (*Regedit*). Essas chaves, ao serem ativadas durante a inicialização do SO, reativam o ataque cibernético, explorando vulnerabilidades previamente utilizadas pelo *malware*, como alterar a página inicial do

Internet *Explorer*. Assim, o antivírus desenvolvido verifica se o aplicativo suspeito executa ações como:

  - Obter o nome *NetBIOS* da máquina local, cujo valor é definido durante a inicialização e lido diretamente do *Regedit*.
  - Remover uma chave específica no registro.
  - Criar uma entrada no *Regedit*, com possibilidade de leitura caso já exista.
  - Apagar uma chave de registro juntamente com seus valores associados.
  - Explorar e acessar subchaves vinculadas a uma entrada específica do registro aberta no sistema.

- **Características Ligadas a *Spywares*, como *Keyloggers* e *Screenloggers***. *Spywares* monitoram atividades do usuário com o objetivo de capturar informações sensíveis, como credenciais de acesso e dados financeiros. O antivírus verifica se o arquivo em análise realiza:

  - Identificação de regiões atualizadas na tela da vítima e cópia dessas áreas para outro local.
  - Captura de vídeos em formato AVI usando *webcams* ou outros dispositivos de vídeo conectados.
  - Monitoramento de atividades de votação eletrônica, como em sistemas da empresa *Optical Vote-Trakker*.
  - Registro de estados de teclas do teclado, técnica comumente usada por *keyloggers* para interceptar senhas.
  - Vigilância das atividades online e coleta de informações confidenciais do usuário, incluindo senhas bancárias, com envio desses dados ao invasor.
  - Acesso remoto a computadores para roubo de credenciais e informações pessoais.
  - Criação de *BHO*s (Objetos de Auxílio ao Navegador), que são executados automaticamente com o navegador. Frequentemente, *BHO*s são usados

por *spywares* e *adwares* para registrar atividades de teclado e mouse, escapando de *firewalls* ao serem reconhecidos como parte do navegador.

  - Localização de senhas salvas no sistema.

- **Métodos de Antiforense Digital**. Antiforense refere-se a técnicas destinadas a ocultar, apagar ou manipular evidências digitais, reduzindo a eficácia das análises forenses. O antivírus avalia se o arquivo investigado realiza:

  - Suspensão temporária de sua própria execução até que um período predeterminado seja concluído, prática comum em *malwares* que evitam detecção durante a quarentena de softwares antivírus comerciais.
  - Desativação de ferramentas de segurança da vítima, como *firewalls* e programas antivírus.
  - Bloqueio das atualizações automáticas do sistema *Windows*, comprometendo a segurança do SO.
  - Detecção de ferramentas de depuração ou análise em execução no sistema.
  - Extração de informações relacionadas aos processos armazenados em *snapshots* do SO, prática empregada para danificar backups ou pontos de restauração do sistema.
  - Ocultação de arquivos infectados dentro de outros arquivos, utilizando esteganografia, uma técnica que disfarça o *malware* em aplicativos aparentemente legítimos.
  - Alteração do nome exibido no Gerenciador de Tarefas para evitar identificação.
  - Utilização de bibliotecas documentadas na "Hackers *Encyclopedia* 2002" para manipulações avançadas.
  - Lançamento de ataques *ZeroAccess* por meio da modificação de *firmwares* em dispositivos de hardware, como controladoras de discos rígidos.

- **Aspectos Relacionados à Criação de Interfaces Gráficas (*GUI*) pelo Programa Suspeito**. O antivírus desenvolvido verifica se o arquivo em análise executa ações relacionadas à criação de elementos gráficos interativos, como:

  - Gerar uma interface gráfica durante o tempo de execução do programa.
  - Utilizar o *DirectX*, uma biblioteca que permite que aplicativos multimídia renderizem gráficos bidimensionais (2D).
  - Desenvolver módulos contendo funções para compressão e descompressão de *bitmaps*, usadas em tecnologias como o *Microsoft* Vídeo para *Windows*.
  - Implementar gráficos tridimensionais (3D) por meio de funções utilitárias fornecidas pelo *OpenGL*.
  - Detectar formas ou padrões utilizando técnicas de visão computacional e processamento de imagens digitais.
  - Acessar recursos para criar e gerenciar janelas e controles básicos, como botões, barras de rolagem, e receber entrada de mouse ou teclado. Essas ações também abrangem funcionalidades adicionais de *GUI* do *Windows*, incluindo barras de progresso, ferramentas, status e guias.

- **Características Ligadas à Análise Ilegal da Memória Principal (*RAM*)**. O antivírus também examina se o aplicativo tenta realizar operações na memória principal do sistema local, como:

  - Acessar informações armazenadas em áreas específicas da RAM.
  - Ler dados alocados a um processo específico na memória.
  - Gravar informações em um espaço de memória ocupado por determinado processo.
  - Reservar, confirmar ou modificar o estado de páginas no espaço de endereçamento virtual de um processo em execução.

- **Ações Relacionadas ao Tráfego de Rede**. No que tange à comunicação pela rede, o antivírus avalia se o arquivo em questão tenta:
  - Realizar consultas a servidores *DNS*.
  - Enviar solicitações para servidores *HTTP*.
  - Monitorar cabeçalhos de pacotes de dados associados a solicitações HTTP realizadas pelo sistema.
  - Enviar mensagens *ICMP* de eco para endereços *IPv4*.
  - Utilizar o protocolo *SNMP* para monitorar dispositivos em redes locais.
  - Finalizar a conexão com a Internet.
  - Estabelecer sessões de *FTP* ou *HTTP* durante o funcionamento do aplicativo.
  - Fragmentar endereços URL dinamicamente em tempo de execução.
  - Consultar servidores para verificar a quantidade de dados de tráfego disponíveis.
  - Determinar o status de conectividade com a Internet do sistema local.
  - Inicializar o uso de funções *WinINet* (API do *Windows* para criar e gerenciar aplicações que utilizem a Internet).
  - Capturar dados de pacotes de rede gerados por solicitações anteriores, comportamento comumente associado a *sniffers*.
  - Modificar informações contidas em pacotes de rede do sistema.
  - Administrar redes locais e remotas por meio de ferramentas de gerenciamento.
  - Criar *sockets* de rede no sistema. Diferente de aplicações legítimas, onde os servidores enviam dados aos clientes, *malwares* podem estabelecer *sockets* locais que aguardam conexões de computadores remotos mal-intencionados para enviar informações sigilosas da vítima, como imagens e senhas.

  - Receber informações via *socket*: método frequentemente empregado por *backdoors*, permitindo que o dispositivo comprometido receba instruções e comandos enviados remotamente por cibercriminosos.
  - Transmitir dados através de um *socket*: comportamento típico de *spywares*, que utilizam esse mecanismo para enviar informações confidenciais capturadas, como senhas e dados pessoais, para servidores controlados por invasores.

- **Modificações em Aplicativos e Programas Utilitários** O antivírus também investiga se o arquivo suspeito tenta realizar alterações ou interações com aplicativos amplamente utilizados, como:

  - Reproduzir conteúdos multimídia, incluindo vídeos e áudios, utilizando o reprodutor nativo *Windows Media Player*.
  - Modificar atalhos de ícones e ajustes padrões relacionados à navegação na Internet, especialmente na barra de endereços ou ferramentas do *Explorer*.
  - Alterar configurações no editor de textos padrão *WordPad*, impactando suas funcionalidades ou comportamento esperado.
  - Editar parâmetros de *sockets* controlados pelo navegador *Internet Explorer*, potencialmente comprometendo a segurança das conexões.
  - Realizar mudanças no *Outlook Express*, incluindo acesso à lista de contatos de *e-mails* da vítima, o que pode levar ao roubo de dados ou disseminação de *phishing*.
  - Consultar e manipular dados associados à suíte *Microsoft Office*, incluindo documentos, preferências de usuários e suporte a macros.
  - Alterar configurações de programas da suíte *Adobe Systems*, como *Acrobat Reader* ou *Photoshop*, que podem ser explorados para execução de código malicioso.

- Fazer ajustes nas configurações de ferramentas de manutenção do sistema, como o recurso de limpeza de disco, comprometendo a capacidade de gerenciamento do armazenamento.
- Modificar parâmetros de jogos digitais nativos do sistema operacional ou de títulos populares vinculados às empresas *Tycoon* e *Electronic Arts*, alterando suas funções ou vulnerabilizando o sistema.
- Ajustar as configurações de atualizações automáticas de softwares relacionados ao *Google Inc.*, potencialmente desativando proteções importantes.
- Utilizar o software *Visual Basic*, uma prática comum de *malwares* baseados em macros, visando explorar linguagens de *script* integradas em navegadores, *Microsoft Office* e produtos da *Adobe Systems*.
- Modificar configurações de acesso e interação com a Wikipédia, permitindo manipular informações ou restringir o acesso a conteúdos específicos.

É relevante registrar o fato de que, individualmente, todas essas características não necessariamente constituem um padrão de conduta maliciosa. Assim, a identificação da ameaça precisa ser realizada através do cruzamento de dados e, com isso, da consideração entre todas as características destacadas.

4.4 Classificação

No que diz respeito à detecção e reconhecimento de padrões de *malware*, o objetivo central é classificar (etiquetar) todos os arquivos investigados conforme seus atributos. Posteriormente, a partir de um grupo de arquivos denominado pacote de treinamento, pode-se conceber hipóteses acerca das distintas classificações vinculadas ao proposto antivírus. Depois, o agente classificador avalia a classe de arquivos desconhecidos mediante a comparação das propriedades comportamentais verificadas ao longo do período com as obtidas ao longo da etapa de aprendizado.

No presente trabalho, as redes neurais são empregadas como agente classificador. A finalidade do agente classificador é gerar uma função separadora dentre as classes de software antivírus (*malware*, benigno). Com isso, quando um arquivo executável ainda não publicado é lançado, essa função de separação é ativada e associa uma classe específica para esse arquivo. Em termos matemáticos, $c = f(x)$, onde $x = x_1, x_2, ..., x_t$ representa o vetor extraído da amostra investigada, t refere-se ao número de amostras avaliadas e c à classe, e por fim f corresponde à função matemática do agente classificador.

Quando um agente classificador linear é instituído, o classificador passa a indicar uma determinada linha cujo objetivo é distinguir padrões com classes distintas. Dessa forma, todos os casos examinados terão sua classificação conforme o lado dessa linha onde estão mapeados. A fim de detectar o funcionamento do *malware* da atualidade, é preciso aplicar classificadores com capacidade de desenvolver separação não linear entre classes. Com o objetivo de demonstrar essa fundamentação, o software antivírus autoral adota sistemas de redes neurais não lineares, principalmente redes neurais morfológicas extremas. O classificador utilizado no presente trabalho se encontra, livremente disponível, no repositório (mELM 2025).

4.5 Síntese do Capítulo

O capítulo descreveu o desenvolvimento do antivírus baseado em *Extreme Learning Machine* (ELM), utilizando um conjunto de dados com 9.405 amostras de *malware* (InstallCore) e 3.135 programas benignos, coletados de fontes como *VirusShare, SourceForge e GitHub*, e verificados no VirusTotal. Para equilibrar as classes, foi aplicada uma técnica de particionamento em blocos, alternando subconjuntos da classe majoritária durante o treinamento. O processo inclui coleta de dados, refinamento, classificação com ELM e validação cruzada, executados em um computador com 24 GB de RAM, 6 núcleos físicos e uma GPU de 2048 núcleos CUDA. Foram extraídas 407 características dos arquivos, como distribuição de instruções de importação, uso de TLS, comportamentos relacionados 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 (c=f(x)) para distinguir entre arquivos benignos e maliciosos, garantindo alta eficiência e baixo desvio padrão. A abordagem proposta visa superar as limitações dos antivírus tradicionais, oferecendo uma solução eficiente e adaptável para a detecção de malwares modernos.

Capítulo 5

Resultados

O capítulo apresentará os desempenhos do antivírus desenvolvido, comparando-o com soluções de última geração. Foram testados oito modelos de kernels em redes neurais ELM, com destaque para o kernel de Dilatação, que alcançou a melhor acurácia (99,35% em teste) e um tempo de treinamento rápido (1,93 segundos). O antivírus proposto superou os modelos de LIMA et al. (2021) e SU and VASCONCELLOS (2018) em acurácia e eficiência, com tempos de treinamento significativamente menores. Análises estatísticas apresentadas confirmarão a superioridade do antivírus autoral, que também demonstrou alta sensibilidade e especificidade, com baixas taxas de falsos positivos e negativos.

5.1 Resultados das redes ELM

Foram utilizados 8 modelos distintos de *kernels* em redes neurais. Com relação ao estado da arte, cinco dos *kernels* selecionados já foram descritos por HUANG, 2012 sendo os seguintes: *Wavelet*, Polinomial, *Sigmoid*, *Sine*, *Hard Limit* e *Tribas Transforms* (funções com base trigonométrica) (HUANG, 2012). Ademais, *kernels* autorais são usados de maneira complementar: Erosão e Dilatação.

O *kernel* do tipo *Wavelets* não possui nenhuma camada escondida (HUANG, 2012). Seus cálculos têm como base a conversão das informações de entrada, podendo operar em um modo de operação análogo ao de *kernels* cujas composições

incluem estruturas de camadas ocultas (HUANG, 2012). Para que esses *kernels* apresentem adequada capacidade em termos de generalização, torna-se fundamental que os parâmetros (C, γ) (HUANG, 2012) possam ser selecionados de maneira apropriada. O valor relativo ao custo C diz respeito ao ponto de referência aceitável quanto à dimensão da fronteira do hiperplano com a redução ao mínimo do erro de identificação no que diz respeito ao aprendizado. Já o indicador γ comanda os limiares de deliberação com base nas classes (HUANG, 2012). A escolha desses valores de (C, γ) não segue uma metodologia universal.

O *kernel* do tipo polinomial também não depende de camadas escondidas. Sua operação baseia-se na aplicação de relações de ordem superior entre as variáveis, permitindo capturar padrões mais complexos de maneira eficiente ao expandir as características dos dados de entrada para espaços de maior dimensionalidade (HUANG, 2012). A eficácia dos *kernels* polinomiais depende crucialmente da escolha dos parâmetros (C, γ). O parâmetro C regula a flexibilidade do modelo, equilibrando a margem do hiperplano e o erro de classificação, enquanto o grau γ controla a complexidade do polinômio, determinando o nível de detalhamento das interações entre as características (HUANG, 2012). Essa escolha influencia diretamente a capacidade de generalização do modelo e, embora não exista uma metodologia universal para determinar esses valores, ajustes específicos baseados nos dados e no problema são necessários.

No presente trabalho, foram examinados parâmetros (C, γ) orientados conforme a metodologia apresentada por HUANG, 2012, a qual constitui o aprendizado em sucessões sequenciais progressivas de C e γ, de forma matemática, 2^n, em que $n = \left\{-24, -10, 0, 10, 25\right\}$ (HUANG, 2012). O objetivo dessa verificação é observar em que medida os valores desses parâmetros, quando distintos das predefinições $(C = 1, \gamma = 1)$, proporcionam melhores respostas. Para o *kernel* do tipo linear, apenas o atributo de custo C deve ser averiguado de modo que o atributo γ não é avaliado (HUANG, 2012).

O conjunto é composto por 9.405 amostras de *malware* e 3.135 programas exe-

cutáveis benignos. Utilizando a técnica de particionamento em blocos, as amostras malignas foram separadas em três conjuntos de 3.135 arquivos cada. Na sequência, a validação cruzada foi feita por meio da metodologia *k-fold*, sendo k=10. A finalidade do uso de tal procedimento visa a assegurar a não influência exercida sobre os valores atingidos através dos conjuntos de teste e aprendizado. Por essa razão, o total de informações é subdividido em 10 segmentos.

No processo inicial, é destinada uma parte para o grupo de teste, ao passo que a restante é destinada ao aprendizado . Esse procedimento ocorre de forma intercalada durante dez rodadas para que todas as parcelas possam ter sido submetidas à etapa de teste. O grau de exatidão da ELM é determinado a partir da média aritmética do índice de sucesso alcançado durante todas as dez execuções.

Não existe nenhuma retropropagação das informações contidas no sistema ELM. Dessa forma, o intuito por trás da metodologia de validação cruzada *k-fold* não consiste na definição de um parâmetro para evitar o superajuste. Na verdade, destina-se a constatar a ocorrência de variações abruptas da exatidão da ferramenta de classificação de acordo com os grupos de teste e de aprendizado. Isso posto, a meta é fazer com que os classificadores enviesados na direção de alguma classe particular não apresentem favorecimento nas taxas de acurácia.

Em cada linha das tabelas apresentadas no presente capítulo, há 30 execuções referentes ao balanceamento de classes e ao método *k-fold*. É possível verificar na Tabela 5.1, os resultados alcançados por redes neurais ELM a partir do núcleo *Wavelets* e Polinomiais. Para os *kernels* do tipo *wavelets*, durante a fase de testes, os resultados de desempenho médio alcançaram o valor máximo de 96,07% sob configuração paramétrica $(C, \gamma) = (2^{-10}, 2^{-24})$. Já para os *kernels* do tipo polinomiais, os resultados de desempenho médio alcançaram o valor máximo de 98,24% sob configuração paramétrica $(C, \gamma) = (2^{0}, 2^{-10})$. Os resultados da Tabela 5.1 exibem somente as denominações dos melhores e piores cenários, respectivamente, sobre cada núcleo.

Já a Tabela 5.2 ilustra apenas os resultados encontrados com a rede ELM operando sobre um núcleo linear. É examinado exclusivamente o valor do parâmetro

C, uma vez que é impraticável observar o fator γ para um núcleo linear (HUANG, 2012). Os valores máximo e mínimo para a acurácia correspondem a, respectivamente, 98,23% e 49,99%. Considerando esse resultado, é possível definir que a análise do valor do parâmetro de custo C tem a capacidade de otimizar a acurácia da classificação.

A Tabela 5.3 apresenta os valores alcançados com a rede ELM (*Extreme Learning Machine*) mediante o emprego de diferentes *kernels*, entre eles *Sigmoid*, *Sine*, *Hard Limit*, *Tribas* (funções de base trigonométrica), Dilatação e Erosão, todos aplicados sob um esquema de camada escondida. A avaliação consiste em modificar a quantidade de neurônios existentes na camada oculta a fim de averiguar se a maior complexidade de computação (como quintuplicar o número de neurônios) exerce influência sobre as taxas de acurácia.

Duas arquiteturas foram avaliadas, sendo uma com 100 e a outra com 500 neurônios na camada oculta. Essas configurações contam com um excelente histórico de acurácia, principalmente na aplicação de redes ELM nos setores de Engenharia Biomédica (LIMA et al., 2020). O máximo rendimento médio de acurácia foi de 99,35%, seguido de um desvio padrão de 0,29%, alcançado com o uso do *kernel* de Dilatação com 500 neurônios na camada oculta. Isso sugere que, embora as arquiteturas mais complexas exigem maior capacidade de cálculo, elas podem gerar ganhos de acurácia, conforme evidenciado neste estudo.

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	$(2^{-24}, 2^{10})$	**100,00 ± 0,00**	96,07 ± 0,91	1,89 ± 0,25	0,14 ± 0,05
	$(2^{-10}, 2^{25})$	59,35 ± 0,44	58,32 ± 1,81	2,07 ± 0,28	0,14 ± 0,06
Polinomial	$(2^{-10}, 2^{0})$	98,39 ± 0,07	**98,24 ± 0,64**	1,17 ± 0,14	0,04 ± 0,03
	$(2^{10}, 2^{24})$	50,00 ± 0,01	49,94 ± 0,04	2,67 ± 0,29	0,27 ± 0,08

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^{-10}	**98,41 ± 0,06**	**98,23 ± 0,61**	1,09 ± 0,19	0,03 ± 0,03
	2^{-24}	49,99 ± 0,01	49,99 ± 0,07	1,08 ± 0,17	0,03 ± 0,04

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	75,99 ± 0,14	73,29 ± 1,15	0,32 ± 0,09	0,00 ± 0,01
	100	67,96 ± 0,14	67,57 ± 1,30	0,04 ± 0,03	0,00 ± 0,00
Seno	500	86,43 ± 0,25	81,61 ± 1,25	0,35 ± 0,09	0,01 ± 0,01
	100	74,76 ± 0,47	73,27 ± 1,93	0,05 ± 0,03	0,00 ± 0,00
Hard limit	100	50,01 ± 0,00	49,95 ± 0,04	0,05 ± 0,03	0,00 ± 0,00
	500	50,01 ± 0,00	49,95 ± 0,04	0,39 ± 0,10	0,00 ± 0,01
Tribas	100	50,01 ± 0,01	50,00 ± 0,07	0,04 ± 0,02	0,00 ± 0,01
	500	50,01 ± 0,01	50,00 ± 0,07	0,24 ± 0,05	0,01 ± 0,01
Dilatação	500	**99,71 ± 0,05**	**99,35 ± 0,29**	1,93 ± 0,25	0,16 ± 0,08
	100	98,94 ± 0,16	98,85 ± 0,63	0,38 ± 0,08	0,04 ± 0,02
Erosão	500	98,76 ± 0,09	98,68 ± 0,35	1,98 ± 0,23	0,18 ± 0,06
	100	95,98 ± 0,13	95,77 ± 1,13	0,41 ± 0,10	0,04 ± 0,03

Fonte: O autor (2025).

5.2 Resultados em relação ao estado da arte

Na presente seção, é feita uma avaliação comparativa entre o antivírus proposto e os antivírus contemporâneos. A fim de proporcionar imparcialidade nas comparações, padronizou-se o modo de obtenção de recursos, com o acompanhamento de 407 atributos por parte do executável em suspeita.

A rede neural atrelada ao antivírus autoral apresenta uma estrutura simples, com uma camada oculta de 500 neurônios e um *kernel* de Dilatação , que atua como um filtro não-linear para ponderar características relevantes dos dados. Em contraste, o antivírus proposto por LIMA et al.2021 utiliza uma abordagem mais difusa, explorando onze diferentes configurações de treinamento para redes neurais superficiais baseadas em *backpropagation*. O estudo investiga quatro arquiteturas distintas de camada oculta para cada função de aprendizado, visando otimizar o

desempenho do sistema.

A fim de estabelecer um ponto de referência para a avaliação do nosso modelo, incluímos em nossa análise um sistema antivírus baseado em redes neurais profundas, proposto por SU and VASCONCELLOS, 2018. De modo a viabilizar a imparcialidade da comparação, reproduzimos o modelo em nosso próprio pacote de dados.

As Figuras 5.1 e 5.2 corroboram os resultados numéricos apresentados na Tabela 5.4. A Figura 5.1(a) mostra que o nosso antivírus alcançou uma acurácia média de 99,71% durante o treinamento, superando os demais modelos. O modelo de LIMA et al., 2021 apresentou um desempenho variável, com a melhor performance (99,01%) obtida com uma arquitetura de duas camadas e o pior (50,34%) com uma única camada de 500 neurônios. A escolha da técnica de treinamento também influenciou significativamente os resultados, sendo que o "*Conjugate gradient backpropagation with Fletcher-Reeves updates*" se mostrou mais eficaz. Por outro lado, a função de aprendizado "*Resilient backpropagation*" obteve os piores resultados. O modelo de SU and VASCONCELLOS, 2018, baseado em redes neurais profundas, apresentou uma acurácia média de 99,57%, inferior à do nosso modelo, mas ainda assim muito boa. Essa diferença pode ser atribuída a diversos fatores, como a complexidade da arquitetura e os hiperparâmetros utilizados em sua arquitetura profunda.

A Figura 5.1 (b) ilustra os gráficos referentes à etapa de teste para o antivírus desenvolvido no presente trabalho e o antivírus mais recente. O antivírus desenvolvido neste trabalho obteve uma acurácia média de 99,35%, tendo apresentado desvio padrão de 0,29%. O software de LIMA et al., 2021 registrou média de acurácia de 50,32% na situação mais desfavorável e 99,12% nas condições mais propícias. Tais resultados fortalecem ainda mais a ideia de que as redes neurais com base em *backpropagation* conseguem demonstrar grandes oscilações em termos de acurácia, a depender de suas configurações iniciais. Dessa forma, a opção de LIMA et al., 2021 em explorar distintas configurações de aprendizagem, gradientes e arquiteturas com o intuito de aperfeiçoar as redes neurais apoiadas em *backpropagation* mostrou-se uma decisão correta. O antivírus de SU and VASCONCELLOS, 2018 atingiu a

acurácia média de 99,49% durante a etapa de teste.

Os gráficos 5.2(a) e 5.2(b) ilustram os *boxplots* dos tempos despendidos durante os estágios de aprendizado e de teste, na respectiva ordem. O sistema antivírus apresentado levou apenas 1,95 segundos, em média, para finalizar o treino. Já o antivírus elaborado por SU and VASCONCELLOS, 2018 apresentou maior lentidão com relação à fase de treinamento, gastando 91,06 segundos. No que se diz respeito ao período de teste, as técnicas exibiram valores bastante próximos, com exceção do antivírus autoral que apresentou cerca de 0,1 segundos a mais. Um fato importante a ser ressaltado é que, com o crescimento da base de dados, o tempo de treinamento demandado por uma rede neural profunda tende a aumentar exponencialmente, o que pode inviabilizar o uso dessa rede para gerar soluções atualizadas de forma ágil.

Com o desenvolvimento deste estudo, o software antivírus criado se apresentou com uma superioridade frente às principais ferramentas acadêmicas. Sua acurácia, em teste, atingiu uma taxa média de 99,35%, sendo seu tempo de treinamento médio realizado em apenas 1,93 segundos. Tendo em vista que são lançadas 8 novas versões de *malware* a cada segundo (Intel, 2018), podemos constatar que um antivírus que foi recentemente distribuído se tornará desatualizado com rapidez e demandará um novo período de treinamento a fim de combater as fragilidades detectadas recentemente. Em suma, o intervalo de tempo para treinamento realizado por um software antivírus necessita seguir o ritmo da criação de novos *malwares* numa escala mundial, para evitar grandes defasagens.

O presente trabalho define eficiência como a relação inversa entre a acurácia no teste e o tempo de treinamento. No estudo realizado por LIMA et al., 2021, a pior configuração foi descartada na avaliação da métrica de eficiência, uma vez que adotar uma solução com acurácia em torno de 50% seria equivalente a utilizar um classificador próximo à aleatoriedade. A métrica de eficiência desempenha um papel fundamental ao estabelecer um parâmetro para avaliar o desempenho do modelo. Adicionalmente, ela também reflete a prontidão operacional, aspecto especialmente relevante no contexto de antivírus, onde o fator tempo é crítico. A habilidade

de realizar um novo treinamento rapidamente, em resposta a uma vulnerabilidade recém-descoberta, pode ser decisiva na mitigação de riscos.

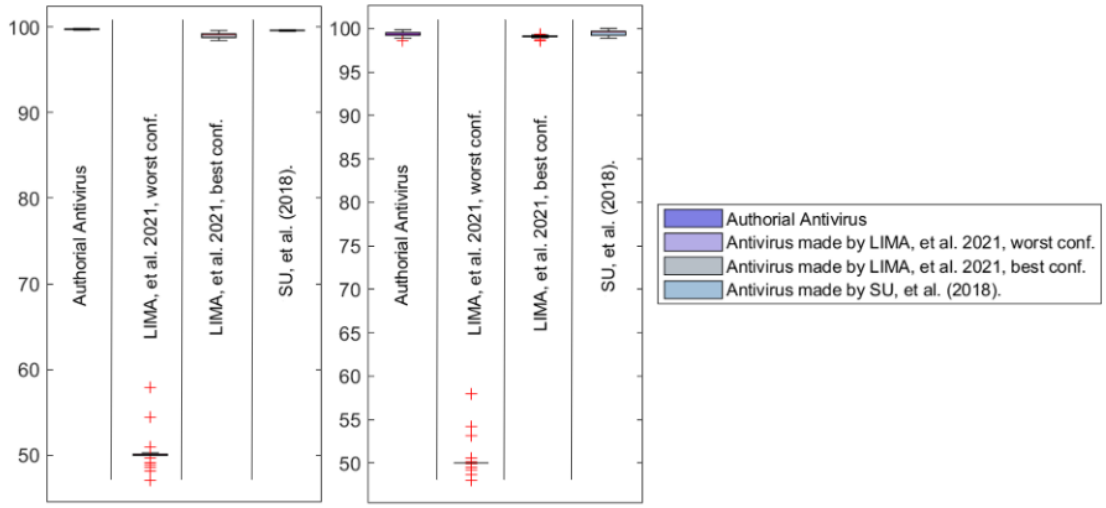
A Tabela 5.5 demonstra as matrizes de confusão para as técnicas mencionadas na Tabela 5.4, descritas de maneira percentual. Essa matriz é essencial na avaliação sobre a qualidade da aprendizagem supervisionada. Ainda nessa tabela, as siglas M. e B. significam *Malware* e Benigno, na ordem respectiva. As classes almejadas encontram-se nas etiquetas verticais, já as classes conseguidas situam-se nas etiquetas horizontais. É importante observar que, na matriz de confusão, a principal diagonal reúne as situações onde a classe esperada converge com a real, ou seja, aquelas denominadas "verdadeiros positivos".

Dessa forma, para um ótimo classificador, a principal linha diagonal precisa conter altos valores, e os demais parâmetros precisam apresentar baixos valores. Na Tabela 5.5, as diagonais principais aparecem realçadas em negrito. Ao longo da etapa de testagem, o software antivírus projetado identificou, erroneamente, em média 0,89% do número total de incidentes de *malware* classificados como benignos (falsos negativos). Do mesmo modo, houve erro de classificação em uma média de 0,42% dos executáveis inofensivos que foram classificados como *malware* (falsos positivos).

Ainda de acordo com a Tabela 5.5, respectivamente, a sensibilidade do antivírus e a sua especificidade correspondem à sua competência em distinguir corretamente *malwares* de aplicativos seguros. A obra aqui abordada expõe a matriz de confusão na forma de porcentagem a fim de simplificar as análises sobre esses indicadores. Em suma, a especificidade e a sua sensibilidade estão embutidas dentro dessa mesma matriz de confusão, tal como exibido na Tabela 5.5. A título de exemplificação, o software autoral de antivírus consegue alcançar uma taxa média de 99,11% em termos no que diz respeito à sensibilidade e a verdadeiros positivos. De forma semelhante, apresenta um valor médio de 99,58% tanto com relação à especificidade quanto com relação aos verdadeiros negativos.

Na Tabela 5.6 constam dados provenientes dos testes de hipótese paramétricos

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



Fonte: O autor (2025).

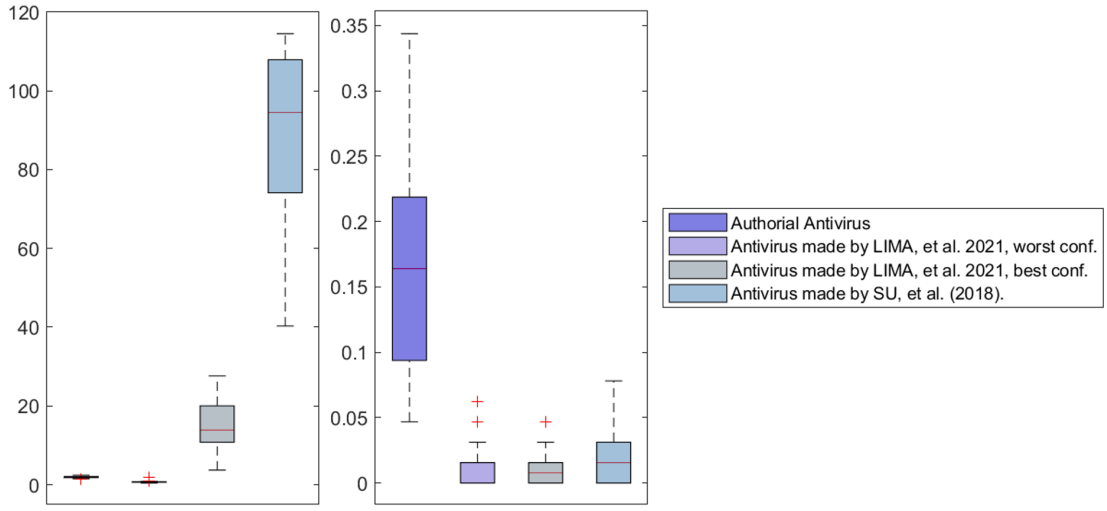
(*t-student*) bem como os não paramétricos (*Wilcoxon*), quando se comparam a proposta de software antivírus juntamente com a última geração de software antivírus. Conclui-se ainda que o rendimento da amostra do software antivírus do autor difere em termos estatísticos das demais amostras. O *p-value* indica a probabilidade de os resultados observados ocorrerem por acaso. Quando se assume o valor 0, há forte evidência de acaso. Já um *p-value* igual a 1 indica que os resultados do teste são altamente consistentes.

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	**99,71 ± 0,05**	**99,35 ± 0,29**	1,93 ± 0,25	0,16 ± 0,08	51,48
LIMA, *et al*, (2021), pior conf.	50,34 ± 1,98	50,32 ± 1,80	0,74 ± 0,23	0,02 ± 0,01	–
LIMA, *et al*, (2021), melhor conf.	99,01 ± 0,28	99,12 ± 0,15	14,90 ± 6,15	0,01 ± 0,01	6,65
SU, R, *et al*, (2018)	99,57 ± 0,05	99,49 ± 0,30	91,06 ± 18,25	0,02 ± 0,02	1,09

Fonte: O autor (2025).

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



Fonte: O autor (2025).

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

		Treino		Teste	
Técnica		M.	B.	M.	B.
Antivírus autoral	M.	**99,74 ± 0,04**	0,31 ± 0,10	**99,11 ± 0,54**	0,42 ± 0,33
	B.	0,26 ± 0,04	**99,69 ± 0,10**	0,89 ± 0,54	**99,58 ± 0,33**
Antivírus de LIMA,	M,	**51,68 ± 49,90**	51,01 ± 48,47	**51,84 ± 49,92**	51,19 ± 48,79
et al. pior conf. (2021)	B,	48,32 ± 49,90	**48,99 ± 48,47**	48,16 ± 49,92	**48,81 ± 48,79**
Antivírus de LIMA,	M,	**98,98 ± 0,42**	0,95 ± 0,28	**99,00 ± 0,34**	0,77 ± 0,16
et al. melhor conf. (2021)	B,	1,02 ± 0,42	**99,05 ± 0,28**	1,00 ± 0,34	**99,23 ± 0,16**
Antivírus de SU,	M,	**99,47 ± 0,06**	0,33 ± 0,07	**99,32 ± 0,46**	0,33 ± 0,31
et al. (2018)	B,	0,53 ± 0,06	**99,67 ± 0,07**	0,68 ± 0,46	**99,67 ± 0,31**

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* Antivírus de LIMA, *et al.* (2021), pior conf.	1	4,41724e-127	1	6,22673e-32
Antivírus autoral *vs* Antivírus de LIMA, *et al.* (2021), melhor conf.	1	6,70717e-11	1	5,73149e-12
Antivírus autoral *vs* Antivírus de SU, *et al.* (2018)	1	0,00104107	1	0,00122687

Fonte: O autor (2025).

Capítulo 6

Conclusão

Em função do constante surgimento de *malwares* inéditos no mundo, faz-se fundamental a oferta, pelas ferramentas de proteção, de eficientes serviços voltados para o monitoramento virtual a fim de suprir de maneira proativa as expectativas de seus usuários. Se isso não estiver acontecendo, nos casos em que a identificação de programas nocivos fracassar, poderá resultar em riscos consideráveis de acesso a dados sigilosos de seus consumidores por parte de terceiros não autorizados.

Dessa forma, é possível constatar como a escolha de um bom software antivírus exerce função primordial na luta contra os perigos do ambiente virtual. No estudo conduzido ao longo deste projeto, observou-se um intervalo de variação quanto à capacidade de identificação de *malwares* relacionados a *InstallCore* entre 0% e 99,79%, variando de acordo com o software antivírus de mercado empregado.

A nossa pesquisa avaliou 79 programas antivírus encontrados oferecendo seus serviços, os quais, por sua vez, apresentaram uma taxa média de identificação de 66,71% dos programas maliciosos. Adicionalmente, a avaliação apontou para uma média de 23,85% de falsos negativos e, em 9,43% de omissão. Durante essa perícia, optou-se pelo recurso da plataforma *VirusTotal*, cuja finalidade foi automatizar o procedimento de transmissão de programas de *malware* para os antivírus do mercado. Convém salientar o fato de que essa plataforma não oferece o recurso de seleção das versões *shareware* dos programas de antivírus, só há a versão *full*. O referido fato torna impraticável a análise comparativa entre as versões comerciais

e sem custo de um mesmo programa. Supõe-se, contudo, que o desempenho das versões de *shareware* são, consideravelmente, piores do que o de versões integrais.

Cabe observar o fato de que, durante os estudos realizados, os softwares maliciosos avaliados são altamente identificados e conhecidos publicamente, além de comumente aplicados à prática de atos ilícitos. Apesar disso, boa parte das soluções antivírus disponíveis no mercado, que foram testadas, não reconheceram a presença do *malware* nos arquivos maliciosos avaliados.

Com o objetivo de contornar tais limitações apresentadas pelos antivírus convencionais, surgiram programas antivírus fundamentados no uso de redes neurais, os quais possuem a habilidade de inspecionar milhares de arquivos maliciosos e de aprender suas características nocivas através de padrões estatísticos. Depois desse procedimento de treinamento, o sistema antivírus inteligente consegue reconhecer e categorizar softwares mal-intencionados futuros, mediante a confrontação das suas propriedades com aquelas previamente adquiridas. Tal procedimento de aprendizado tem a possibilidade de se tornar automatizado, de forma a possibilitar ao antivírus identificar ameaças novas independentemente da denúncia por parte de algum outro usuário anteriormente comprometido, e, com isso, agilizar a reação a novos *malwares*.

Visando auxiliar na prevenção e também no combate ao alastramento de arquivos maliciosos, desenvolveu-se então um antivírus autoral dotado do poder de categorizar os programas executáveis nas categorias malicioso ou inofensivo. A ferramenta monitora e verifica, de forma estatística, condutas relacionadas à execução do conteúdo sob suspeita que podem ser desempenhadas no sistema operacional. Por meio de condições rigorosas de controle, o software antivírus registra as movimentações feitas no sistema de registro (banco de dados) além de supervisionar as solicitações efetuadas em cada um dos processos produzidos pela ameaça. São identificados comportamentos relativos aos 407 processos considerados perigosos mediante o emprego das redes neurais extremas.

No lugar de aplicar núcleos clássicos, o presente trabalho recorre a núcleos au-

torais aplicados a redes ELM (*Extreme Learning Machines*). Optou-se pelo uso da rede ELM em virtude de sua elevada rapidez no processo de treino e de sua exatidão ao predizer informações, quando equiparada às redes tradicionais. O núcleo autoral, denominado Dilatação, é apto a discriminar entre *malwares* tipo *InstallCore* e programas inofensivos, apresentando um índice de êxito de 99,71% e gastando somente 1,93 segundos em seu processo de aprendizado.

Vale ressaltar que o trabalho proposto apresenta algumas limitações, principalmente na comparação com redes neurais profundas. Enquanto as Redes ELM são eficientes e viáveis para ambientes com recursos limitados, as redes profundas exigem infraestruturas complexas, como *datacenters* com GPUs/TPUs, para processamento em larga escala. Isso dificulta a comparação direta de desempenho, já que as redes profundas, quando adequadamente implementadas, oferecem uma boa acurácia e capacidade de generalização. Além disso, a solução proposta é focada especificamente no *InstallCore*, o que pode limitar sua aplicabilidade para outros tipos de *malwares*, e depende de atualizações frequentes para manter sua eficácia. Essas limitações destacam os trocas entre eficiência computacional e capacidade de detecção abrangente.

Como trabalho futuro, propõe-se a integração do antivírus desenvolvido com outros antivírus especializados, criando uma plataforma unificada e abrangente para detecção de *malwares*. Essa plataforma combinaria as forças de diferentes soluções, cada uma focada em tipos específicos de ameaças (como *ransomware*, *spyware* ou *malwares* de IoT e outros *adwares*), permitindo uma cobertura mais ampla e eficiente. Além disso, a plataforma poderia incluir um sistema de gerenciamento centralizado para atualizações em tempo real, garantindo que todos os antivírus especializados estejam sempre sincronizados com as últimas ameaças. Essa abordagem modular e colaborativa poderia oferecer uma solução mais robusta e adaptável, capaz de lidar com a diversidade e a evolução constante dos *malwares*.

Para finalizar, convém salientar ainda que a meta deste trabalho é oferecer à comunidade uma visão diferenciada quanto ao funcionamento dos atuais sistemas de antivírus, propondo também alternativas inovadoras e efetivas a fim de aumentar

a eficiência da identificação de programas maliciosos pertencentes à família *Install-Core*.

Referências

Andy Greenberg *et al.* Hack brief: Dangerous 'fireball' adware infects a quarter billion pcs. *WIRED*, Disponível em `https://www.wired.com/2017/06/hack-brief-dangerous-fireball-adware-infects-quarter-billion-pcs/` 2017.

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.

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.

InstallCore. Retrieval for installcore malware analysis. *O Autor*, Disponível em: `https://github.com/VitorMCarvalho/InstallCore/`. 2025.

Intel. *McAfee Labs*. Disponível em: `https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2018.pdf`, Acesso em Fev 2020. 2018.

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.

IronSource *et al.* Installcore official website. *InstallCore Official WebSite*, Acessado via *WayBackMachine* do Internet Archive pelo url https://web.archive.org/web/20111020040346/http://www.installcore.com/developers/ 2011.

M. KALASH and M. *et al.* ROCHAN. Malware classification with deep convolutional neural networks. *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018. doi: https://doi.org/10.1109/NTMS.2018.8328749.

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.

W. LO and X. *et al.* YANG. An xception convolutional neural network for malware classification with transfer learning. *10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2019. doi: https://doi.org/10.1109/NTMS.2019.8763852.

mELM. Morphological extreme learning machine. Disponível em: `https://github.com/DejavuForensics/mELM`. 2025.

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.

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`.

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

Florian SCHROFF, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

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.