UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

ISAAC NEVES FARIAS

APLICAÇÃO DE MÁQUINAS PSEUDO-MORFOLÓGICAS DE APRENDIZADO EXTREMO À DETECÇÃO DE MALWARES DO TIPO APT

RECIFE

2022

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

ISAAC NEVES FARIAS

APLICAÇÃO DE MÁQUINAS PSEUDO-MORFOLÓGICAS DE APRENDIZADO EXTREMO À DETECÇÃO DE MALWARES DO TIPO APT

TCC apresentado ao Curso de Engenharia Eletrônica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito para a obtenção do título de engenheiro eletrônico.

Orientador: Prof. Dr. Sidney Lima

RECIFE

2022

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

Farias, Isaac Neves .
Aplicação de máquinas pseudo-morfológicas de aprendizado extremo à detecção de malwares do tipo APT / Isaac Neves Farias. - Recife, 2022.
42 : 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, 2022.

1. Antivírus. 2. Detecção de malwares. 3. Antivírus. 4. Ameaças persistentes Avançadas. 5. Máquina de aprendizagem extrema. I. Lima, Sidney Marlon Lopes de . (Orientação). II. Título.

000 CDD (22.ed.)

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

DES
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

ATA DE SESSÃO DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Aos 31 dias do mês de Outubro do ano de Dois Mil e Vinte e Dois, às 14:00, por meio de videoconferência (ONLINE), reuniu-se a banca examinadora para a sessão pública de defesa do Trabalho de Conclusão de Curso em Engenharia Eletrônica da Universidade Federal de Pernambuco, intitulado: "Aplicação de máquinas pseudo-morfológicas de aprendizado extremo à detecção de malwares do tipo APT.", elaborado pelo (a) aluno (a) Isaac Neves Farias, matrícula 125.234.814-21, composta pelos professores Sidney Marlon Lopes de Lima e Hermano Andrade Cabral (membro titular). Após a exposição oral, o (a) candidato (a) foi arguido (a) pelos componentes da banca que em seguida reuniram-se reservadamente e deliberaram pela aprovação do candidato, atribuindo-lhe a média nove (9), julgando-o (a) apto (a) à conclusão do curso de Engenharia Eletrônica. Para constar, redigi a presente ata aprovada por todos os presentes, que vai assinada por mim e pelos demais membros da banca.

Prof.(a)/Membro: Prof. **Sidney Marlon Lopes de Lima** Nota: 9,00

Assinatura

gov.br Documento assinado digitalmente
SIDNEY MARLON LOPES DE LIMA
Data: 31/10/2022 16:49:35-0300
Verifique em https://verificador.iti.br

Prof.(a)/Membro: Prof. **Hermano Andrade Cabral** Nota: 9,00

Assinatura

gov.br Documento assinado digitalmente
HERMANO ANDRADE CABRAL
Data: 31/10/2022 18:57:08-0300
Verifique em https://verificador.iti.br

Recife, 31 de Outubro de 2022.

Prof(a). Guilherme Nunes Melo
Coordenador(a) do Curso de Engenharia Eletrônica

RESUMO

"Malware" é uma junção dos termos "malicioso" e "software". O malware tem como principal objetivo acessar um dispositivo alheio sem permissão explícita de seu proprietário. Dentre os malwares, as Ameaças Persistentes Avançadas, do inglês *Advanced Persistent Threat* (APT), ganharam muito espaço no tópico de roubo de dados e comportamento destrutivo para softwares, principalmente quando se trata de organizações federais e indústrias privadas de grande porte, devido à complexidade e eficiência desse tipo de ataque. Os malwares do tipo APT são direcionado para um alvo pré-definido sendo sempre bem orquestrado com grande precisão e controle, aproveitando os recursos de reconhecimento e vulnerabilidades avançadas. O presente trabalho propõe uma análise crítica acerca do desempenho dos principais antivírus comerciais atuais quanto à detecção de malwares do tipo APT. Em acréscimo, são replicados antivírus do estado da arte dotados de distintas metodologias de detecção baseadas no princípio de redes neurais. Como contribuição principal, um antivírus autoral é criado por meio de uma máquina de aprendizagem extrema e com kernels de processamento pseudo-morfológicos. O referido antivírus autoral tem o intuito de apresentar uma alternativa criativa, eficaz e rápida no controle desse tipo de malware. Por último, são apresentados os resultados da aplicação e sua comparação com os demais antivírus do estado da arte. O antivírus autoral alcança um desempenho médio de 93,62% na distinção entre aplicativos benignos e APT acompanhado de um tempo treinamento de 0,55 segundos, em média. Espera-se que o antivírus inteligente autoral atue de forma preventiva e impeça que os malwares do tipo APT causem prejuízos às instituições privadas e autarquias públicas.

Palavras-chave: Antivírus, Detecção de malwares, Ameaças persistentes Avançadas, Máquina de aprendizagem extrema.

ABSTRACT

"Malware" is a joining of the terms "malicious" and "software." Malware is primarily intended to access another's device without the owner's explicit permission. Among malware, Advanced Persistent Threats (APT) have gained much ground on the topic of data theft and destructive behavior for software, especially when it comes to federal organizations and large private industries, due to the complexity and efficiency of this type of attack. APT-type malware is directed at a predefined target and is always well-orchestrated with great precision and control, taking advantage of advanced vulnerability recognition capabilities. This paper proposes a critical analysis of the performance of the current main commercial antivirus products in detecting APT malware. In addition, state of the art antivirus programs with different detection methodologies based on the neural network principle are replicated. As a main contribution, an authoral antivirus is created by means of an extreme learning machine with pseudo-morphological processing kernels. This authorial antivirus is intended to present a creative, effective and fast alternative in the control of this type of malware. Finally, the results of the application and its comparison with other state of the art antiviruses are presented. The author antivirus reaches an average performance of 93.62% in distinction between benign and APT applications accompanied by a training time of 0.55 seconds on average. The intelligent authoring antivirus is expected to act preventively and stop APT malware from malware from harming private institutions and public autarchies.

Keywords: Antivirus, Malware detection, Advanced persistent threats, Extreme learning machine.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

bpp	Bits por pixel
DFT	Discrete Fourier transform (Transformada de Fourier Discreta)
APT	Advanced Persistent Threat (Ameaça Persistente Avançada)
MLP	Multilayer Perceptron (Perceptron Multicamadas)
ELM	Extreme Learning Machine (Máquinas de aprendizado extremo)
mELMs	Morphological Extreme Learning Machine (Máquina Morfológica de Aprendizagem Extrema)
ReLU	Rectified Liear Unit (Unidade Linear Retificada)
LSTM	Long Short Term Memory (memória de curto prazo longa)
RBM	Restricted Boltzmann Machine (Máquina Boltzmann Restrita)
UC	Unidade de Controle
RAM	Random Acess Memory (Memória de Acesso Aleatório)
TLS	Transport Layer Security (Segurança da Camada de Transporte)
API	Application Programming Interface (Interface de Programação de Aplicativos)
OS	Operational System (Sistema Operacional)
GUI	Graphical User Interface (Interface gráfica do usuário)
DNS	Domain Name System (Sistema de Nomes de Domínio)
FTP	File Transfer Protocol (Protocolo de transferência de arquivos)
HTTP	Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)

SUMÁRIO

1 INTRODUÇÃO

O rápido avanço da tecnologia revolucionou a velocidade de troca de dados no mundo moderno. Esse avanço de forma singular trouxe diversos benefícios para a humanidade, oferecendo serviços de forma bem mais rápida e simplificada. Por outro lado, permitiu que pessoas de má fé conseguissem por meio da tecnologia acessar dados alheios, roubando dados sigilosos que possuem grande importância para o usuário.

Dado esse cenário, as ameaças persistentes avançadas, do inglês Advanced Persistent Threat (APT), ganharam muito espaço no tópico de roubo de dados e comportamento destrutivo para softwares, principalmente quando se trata de organizações federais e indústrias privadas de grande porte, devido à complexidade e eficiência desse tipo de ataque. Esse ataque é direcionado para um alvo pré-definido sendo sempre bem orquestrado com grande precisão e controle, aproveitando os recursos de reconhecimento e vulnerabilidades avançadas.

Para se ter noção do tamanho desse tipo de ameaça, basta reparar em casos de ataques bem conhecidos como por exemplo o caso do grupo *"Equation group"* , em 2010, o qual foi responsável por infectar computadores das centrífugas de enriquecimento de urânio iranianas por meio do malware Stuxnet que danificou as instalações nucleares da cidade de Natanz e acabou atrasando o início da produção da usina de Bushehr. Essa contaminação foi um marco histórico no que se refere a ameaças persistentes avançadas, projetando um futuro sombrio para a segurança da informação em um âmbito internacional o que exige um esforço muito grande no desenvolvimento de sistemas de proteção dos softwares.

Após reconhecer a necessidade de um sistema de proteção eficaz é possível fazer uma análise crítica do desempenho dos atuais antivírus comerciais, os quais apresentam limitações severas no reconhecimento desse tipo de malware. O principal fator que limita a atuação dos atuais antivírus é o seu método de reconhecimento, o qual é baseado em *"blacklists"* onde o sistema possui em seu banco de dados uma lista de vírus, e por meio desses dados o arquivo suspeito é comparado aos vírus presentes na lista. É importante lembrar também que cada antivírus possui um modelo diferente de armazenar essa lista, como é possível verificar na tabela 2, onde os softwares mostram não seguir um padrão na catalogação dos vírus analisados, pois devido a disputas comerciais a lista não é compartilhada entre softwares distintos.

Visto isso, é importante levar em conta outro método de defesa que ao invés de buscar assinaturas em listas negras, deve procurar correspondências de padrões entre os malware através de técnicas de inteligência computacional como as redes neurais artificiais. Tecnicamente, as redes neurais, especialistas em detecção malware, são capazes de identificar comportamentos previamente classificados como suspeitos com uma acurácia média superior a 98% (LIMA; SILVA; LUZ, 2021). Desse modo, o malware poderá ser identificado, de uma forma preventiva,

antes mesmo de ser executado pelo usuário. Como efeito colateral, o tempo de resposta das redes neurais é, costumeiramente, elevado. Caso o tempo de resposta do mecanismo de cyber-vigilância for elevado, um software malicioso, recém-criado, pode gerar malefícios irreversíveis e irrecuperáveis por toda a rede mundial de computadores.

Em grande parte das redes neurais, como a MLP (Multilayer Perceptron - Perceptron com Múltiplas Camadas), é necessário um conhecimento sobre os parâmetros da rede para obter máximo desempenho na solução do problema (LIMA; SILVA; LUZ, 2021). Uma preocupação comum nesse tipo de rede é evitar se ater a mínimos locais de análise, sendo necessário adicionar métodos de controle da rede para desprender-se dessas regiões (LIMA; SILVA; LUZ, 2021). Outra característica comum nesse tipo de rede é o alto tempo de treinamento necessário para tornar a rede apta a realizar classificações corretamente.

A rede ELM (*Extreme Learning Machine* – Máquinas de Aprendizado Extremo) tem como principal característica a velocidade de treinamento e a previsão de dados quando comparada às redes neurais MLP. Os ELMs são máquinas de aprendizagem baseadas em kernel potentes e flexíveis cujas principais características são treinamento rápido e desempenho de classificação robusto (INTEL, 2018). A rede ELM é uma rede de camada escondida única, não recorrente, baseada em um método analítico para estimar os pesos de saída da rede, em qualquer inicialização aleatória de pesos referentes às ligações sinápticas entre os neurônios. As ELMs têm sido largamente aplicadas nas mais diversas áreas como na Engenharia Biomédica (LIMA; SILVA-FILHO; SANTOS, 2014) (LIMA; SILVA-FILHO; SANTOS, 2020) (LIMA; SILVA-FILHO; SANTOS, 2016) (PEREIRA, 2020) (AZEVEDO; *ET AL.*, 2015) (AZEVEDO; *ET AL.*, 2020). As redes ELMs podem contribuir bastante para o avanço da segurança em dispositivos.

O trabalho proposto aplica as ELMs na área de segurança da informação especificamente no reconhecimento de padrão de malware do tipo APT. As características, referente ao arquivo em análise, servem como atributos de entrada das redes neurais artificiais os quais são empregadas como classificadores. O objetivo é classificar os aplicativos suspeitos em duas classes; benignos e malware. O aprendizado das redes ELMs é baseado em kernels.

Ao invés de kernels convencionais, são empregados mELMs (*Morphological* ELMs), ELMs com kernels de camadas escondidas inspirados em operadores morfológicos de processamento de imagens de Erosão e Dilatação. No tocante aos experimentos, os resultados são comparados com antivírus do estado-da-arte e avaliados por meio de métricas de classificação amplamente utilizadas. Esses resultados também são comparados com os melhores cenários obtidos pelo estado-da-arte. O antivírus autoral alcança um desempenho médio de 93,62% na distinção entre aplicativos benignos e APT acompanhado de um tempo treinamento de 0,55 segundos.

1.1 Limitações dos Antivírus Comerciais

Conforme mencionado acima, a metodologia de defesa dos antivírus comerciais é baseada no fato dos arquivos suspeitos serem referenciados ou não em registros chamados listas negras. Portanto, é possível que o hash do arquivo analisado não esteja na lista negra do antivírus para impedir a detecção de um malware. Esse código Hash atua como um identificador exclusivo para um arquivo malicioso específico.

Dadas as limitações dos antivírus comerciais, desenvolver e distribuir variantes de aplicativos maliciosos não é uma tarefa difícil. Fazer isso é muito simples, pois basta fazer pequenas alterações no malware original usando sub-rotinas que são menos comuns, como loops ou condicionais sem escopo. No entanto, por causa dessas mudanças pequenas no arquivo malicioso, o hash do malware modificado é diferente do hash do malware original. Como resultado, o malware modificado não é detectado pelo antivírus que catalogou o arquivo malicioso original.

É importante mencionar também a existência de botnets, as quais são responsáveis pela criação e distribuição de variantes de um malware original de forma automatizada. Em conclusão, antivírus baseados em blacklists são ineficazes quando expostos a qualquer variante de um malware conhecido (SANS, 2017) (LIMA, 2020).

Para apresentar uma análise mais precisa é realizada a identificação do estado da arte da detecção de malwares do tipo “Ameaça Persistente Avançada”, analisando o desempenho de 89 antivírus comerciais e 7 antivírus baseados em inteligência artificial.

Para isso, foram utilizados dois processos diferentes da plataforma VirusTotal, o primeiro é responsável por enviar os arquivos para a análise nos servidores do VirusTotal, e a segunda responsável por emitir o diagnóstico dos antivírus investigados.

Nesse teste de detecção, foram analisadas 1.050 amostras maliciosas para cada software. A representação dos resultados de desempenho é feita na Tabela 1, o diagnóstico pode ser classificado de três maneiras diferentes. A plataforma dá o diagnóstico de “malware” quando software detecta a presença do vírus no arquivo suspeito, “benigno” quando o antivírus erra ao identificar o vírus e “omissão” quando o antivírus não faz um diagnóstico sobre o arquivo suspeito.

A taxa de detecção dos malwares pelos antivírus variaram de 0% a 99,52%. Em uma perspectiva geral,eles foram capazes de detectar corretamente 68,30% com um desvio padrão de 27,83%. O alto desvio padrão é uma evidência de que alguns programas antivírus comerciais tiveram sucesso em detectar os malwares enquanto outros foram absolutamente ineficazes. Em outras palavras, apenas alguns dos antivírus comerciais têm uma lista negra grande o suficiente para proporcionar uma alta taxa de detecção de arquivos maliciosos que foram previamente catalogados. Portanto, o nível de proteção que é fornecido por um antivírus, contra uma invasão cibernética, está diretamente ligado ao tamanho e a variedade de cobertura de sua lista negra.

Outro fator importante a mencionar é que, em média, 17,76% dos antivírus diagnosticaram erroneamente os arquivos malignos como benignos (falso negativos) com desvio padrão de 18,41%, e além disso, em média de 13,94% dos antivírus, não forneceu nenhum diagnóstico sobre os arquivos (omissão) com desvio padrão de 18,37%. Portanto, esta taxa significativa de falsos negativos e omissões levam a um alto risco de infecção que pode trazer danos irreversíveis aos dados e computadores de usuários, governos ou empresas. Esses resultados indicam a ineficácia de antivírus comerciais na proteção de sistemas contra malwares em tempo real.

Além de todas as limitações anteriores, os antivírus comerciais também não fornecem informações úteis sobre os arquivos infecciosos aos usuários, muita das vezes arquivos maliciosos distintos são apresentados apenas com nomes genéricos. Nesse sentido, não seria absurdo afirmar que os diagnósticos fornecidos pelos antivírus são inúteis para o usuário saber o quão prejudicial o malware é, ou quais são suas ações maliciosas sobre o sistema, por exemplo. Isso significa que, quando o mesmo arquivo infectado é enviado para análise por dois programas de antivírus diferentes, esses dois provavelmente fornecerão denominações completamente diferentes para o arquivo malicioso. E além disso, duas variantes do mesmo arquivo infectado, com uma alteração insignificante no código, podem ser denominadas de duas formas completamente diferentes pelo mesmo programa. Essa falta de padrão na nomeação dos vírus faz com que se torne mais difícil implementar estratégias de segurança cibernética, pois cada categoria de malware deve ter um tratamento específico.

Em conclusão, devido a esse emaranhado confuso de erros de diagnóstico e denominações aleatórias fornecidas pelos antivírus. Conforme detalhado na Tabela 2, não é de se esperar que uma técnica de aprendizagem de máquina deva ser capaz de generalizar a detecção de arquivos maliciosos usando apenas as informações de diagnóstico fornecidas pelos antivírus comerciais.

Tabela 1 – Resultados dos antivírus comerciais.

Antivirus	Detecção (%)	Falso Negativo (%)	Omissão (%)
BitDefender	99,52%	0,48%	0%
MicroWorld-eScan	99,33%	0,67%	0%
NANO-Antivirus	98,86%	1,14%	0%
GDataName	98,76%	0,57%	0,67%
ESET-NOD32	98,76%	1,24%	0%
McAfee	98,67%	0,86%	0,48%
Emsisoft	98,48%	0,76%	0,76%
Kaspersky	98,38%	1,24%	0,38%
AVG	98,10%	0,86%	1,05%
MAX	97,52%	0,57%	1,9%
Baidu-International	1,05%	0,48%	98,48%
WhiteArmor	0,76%	1,52%	97,71%
Norman	0,76%	0%	99,24%
AntiVir	0,57%	0%	99,43%
Commtouch	0,48%	0,1%	99,43%
ByteHero	0,1%	1,33%	98,57%
CyrenCloud	0,1%	0%	99,99%
Avast-Mobile	0%	19,71%	80,29%
Trustlook	0%	10%	90%
Babable	0%	11,43%	88,57%

Fonte: Repositório de análise de malwares (APT, 2021).

1.2 Estado da Arte

Embora questionado há mais de uma década, o modus operandi do antivírus permanece baseado em assinaturas de arquivos suspeitos quando pesquisados em bancos de dados chamados Blacklist (LIMA, 2020)(SANS, 2017). Resumindo, o arquivo investigado é comparado aos malwares catalogados na lista negra, logo, se a lista não estiver atualizada o malware não será identificado e causará uma infecção.

LIMA, *et ai.* (2021) desenvolveu um antivírus que tem uma capacidade de detecção de malware (Windows) com uma média de acurácia de 98,32%. No sistema aplicado o executável passa por um processo de desmontagem, desse modo a intenção maliciosa do executável pode ser investigada. Na aplicação do antivírus LIMA *et al.* (2021) são extraídos 630 recursos de cada arquivo executável, esses dados serão utilizados como neurônios de entrada na rede neural artificial. A classificação do antivírus, identifica os arquivos de 32 bits em duas categorias: inofensivo e software malicioso. A rede neural utilizada não é profunda.

Sob outra perspectiva, os antivírus baseados em redes neurais profundas também alcançaram níveis muito elevados de precisão. SU, J. *et al.* (2018) alcançou uma precisão média de

Tabela 2 – Resultados da submissão de dois malwares para o VirusTotal.

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

Fonte: Repositório de análise de malwares (APT, 2021)).

94,00% para detecção malware IoT (Internet das Coisas) (SU; VASCONCELLOS D., 2018). A rede neural profunda possui uma estrutura de 6 camadas. Das 6, existem 3 camadas de aprendizado com os seguintes pesos: 2 para camadas dobráveis e 1 camada totalmente conectada. Essa rede é treinada com um total de 5.000 iterações, lote de treinamento do tamanho de 32 e uma taxa de aprendizagem de 0,0001.

FARUKI, P.*et al.* (2019) alcançou uma média de acerto de 98,65% para detectar malwares Android (FARUKI; BUDDHADEV, 2019). FARUKI, P. *et al.* (2019) utiliza uma rede de aprendizagem profunda ReLU (Rectified Liear Unit), que é equipada com tecnologia dropout. As camadas ReLU executam uma operação de limite em cada elemento da entrada, definindo qualquer valor menor que zero para zero.

MANIATH, S. *et al.* (2017) desenvolveu um antivírus para detectar ransomware usando redes profundas LSTM (Long Memória Curta) (MANIATH; ASHOK, 2017). A rede de formação é composta por 3 camadas com 64 nós LSTM em cada camada. A rede profunda é treinado com treinamento para 500 épocas com um tamanho de lote de 64. MANIATH, S. *et al.* (2016) alcançam uma precisão média de 96,67%.

Além do antivírus, as redes LSTM também foram empregadas em Firewall (WOZNIAK; SILKA, 2015). O objetivo é separar o tráfego das redes malignas das benignas. Um firewall não inteligente tem fórmulas estáticas que bloqueiam portas de usuário selecionadas e formulários.

Se o usuário precisar desta porta para alguma aplicação, deve desabilitar manualmente o bloqueio e isso pode na verdade estar abrindo a porta para um tráfego malicioso . No firewall desenvolvido por WOZNIAK, M. *et al.* (2015), a rede de treinamento consiste de 16 camadas de nós LSTM variando de 256 a 2 neurônios na camada final. A suposição é que a rede profunda é treinada em 1.000 iterações. O firewall feito pela WOZNIAK, M. *et ai.* (2015) alcança uma precisão média excelente de 99,99%.

Já o antivírus desenvolvido por HOU, S. *et al.* (2016) tem o objetivo de combater malwares para sistemas android empregando uma rede de crença profunda (HOU; SAAS, 2016). Toda uma construção de uma pilha de RBM (Restricted Boltzmann Machine) é o que constitui a rede de crença profunda. Sua arquitetura é formada por 3 camadas ocultas com 200 nós em cada camada. HOU, S. et ai. (2016) consegue uma precisão média de 96,66%.

O antivírus desenvolvido por HARDY, W.*et al.* (2016), visa detectar arquivos maliciosos PE (Windows), aplicando uma rede profunda de autoencoders (HARDY; LINGWEI, 2016). O decodificador tenta mapear a representação de saída para a entrada original. Esse modelo de aprendizagem é treinado com 3 camadas ocultas com 100 nós em cada. HARDY, W. *et al.* (2016) alcançou uma precisão média de 96,85%.

A principal desvantagem das redes profundas é o longo tempo de treinamento, uma vez que as camadas são executadas sequencialmente, logo esse tipo de rede possui menos conexões paralelas. Portanto, uma alteração só pode ocorrer após a camada superior ser terminada de executar. Quando isso é levado para aplicações que exigem um treinamento frequente como o antivírus, isso se torna um obstáculo, se for levado em consideração que a cada segundo 8 novos malwares são criados (INTEL, 2018). Em suma, o tempo de aprendizado do software antivírus não deve ser inferior à taxa na qual novos malwares são gerados.

Devido aos excelentes resultados obtidos por técnicas de aprendizagem profunda, foi criado um senso comum de que esse método é capaz de fornecer a melhor precisão em qualquer tipo de aplicação, quando na verdade esse raciocínio não é verdadeiro. As redes de aprendizado profundo, especialmente as redes convolucionais, são baseadas em filtros convolucionais lineares. Tais filtros desempenham um papel fundamental em aplicações computacionais, mas se tornam limitados em aplicações em que dados que têm valores similares possuem representações reais distintas como será exemplificado mais adiante.

Por exemplo, basta analisar imagens de máquina de mamografia, é recorrente que imagens desse tipo possuam muito ruído, o que dificulta a detecção de uma lesão mamária (LIMA; SILVA-FILHO; SANTOS, 2014). Nesse caso, os filtros convolucionais são essenciais para eliminar o ruído, e portanto, descartar as irregularidades no diagnóstico que podem corresponder a uma potencial doença cancerígena. Essas técnicas de redução de ruídos são essenciais, sendo os filtros gaussianos um exemplo.

Como contra-exemplo, basta considerar parte do repositório apresentado na Tabela 3.

Apesar de ficarem na mesma região, os pontos estão completamente separados um do outro. Um aplicativo suspeito de escanear dados WiFi não está relacionado ao acesso da galeria de imagens ou do navegador da vítima. Logo, se o filtro de convolução linear é aplicado acessando o navegador que contém o valor 0, seria tratado como ruído, pois seus dados vizinhos possuem valores positivos. Em resumo, a aplicação suspeita seria acusada de ter acesso ao navegador, o que está incorreto. Dito isto, as técnicas de convolução quando aplicada à detecção de padrões de malware possuem desvantagem se comparada com outras técnicas.

Para provar a base teórica, é proposto um antivírus que utiliza uma rede neural morfológica superficial ao invés de redes convolucionais profundas. Como base experimental, o antivírus possui uma precisão comparada com a próxima geração de redes neurais superficiais e profundas. O antivírus proposto tem como meta combinar uma alta precisão com um tempo de aprendizado reduzido. Evitando comparações injustas, a fase de extração de recursos é padronizada pelo monitoramento de 6.824 indicadores de comportamento que o arquivo suspeito pode executar se forem ativados intencionalmente.

Tabela 3 – Exemplo de parte repositório estatístico baseado em detecção de malware, que realiza a análise a partir dos valores da coluna verificada e de seus próximos.

Atributos		
Acessar Wi-fi	Alterar as configurações do web-navegador	Acessar a Galeria de Imagens
1	0	1

Fonte: Repositório de análise de malwares (APT, 2021).

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Morfologia Matemática

Analisando a limitação dos antivírus baseados em listas negras, foi desenvolvido um antivírus autoral inteligente cuja etapa de aprendizado emprega funções baseadas na Morfologia Matemática. A partir dos operadores morfológicos clássicos de Erosão e Dilatação, são formuladas as conexões sinápticas responsáveis pela identificação preventiva de atividades maliciosas.

Então, houve um estudo dos núcleos de processamento da Morfologia Matemática de maneira a identificar e solucionar possíveis conflitos na execução do algoritmo. Nesse caso existem duas operações morfológicas interessantes no tratamento de dados, Erosão e Dilatação, essas operações foram utilizadas como base para desenvolver a lógica de treinamento do software, Matematicamente, as operações de Erosão e Dilatação se comportam de acordo com as equações (2.1) e (2.2), respectivamente.

$$\epsilon_g(f)(u) = \min(f(v) \ \vee \ \overline{g}(u - v)) \tag{2.1}$$

$$\delta_g(f)(u) = \max(f(v) \ \wedge \ g(u - v)) \tag{2.2}$$

Nas equações acima as representações: $f : S \to \{0, 1\}$ e $g : S \to \{0, 1\}$ são imagens normalizadas na forma matricial com formato nomeado de S, onde $S \in \mathbb{N}^2$. O pixel portanto, é definido pelo par cartesiano $(u, f(u))$ e u é a posição associada ao valor $f(u)$. v é a matriz de $f(u)$, coberta por g. A operação de máxima está relacionada na equação (2.1) , enquanto a operação de mínimo é relacionada na equação (2.2). g é denominado como o elemento estruturante para Erosão e Dilatação (SANTOS, 2011). $\overline{g}$ barrado é a negação de g.

O processo que ocorre na equação (2.1) é iniciado pela negação do elemento estruturante g, só após isso é aplicado a operação de máximo $\vee$, denotado por $f(v) \vee \overline{g}(u - v)$, onde $f(v)$ se refere à matriz da imagem original sendo tecnicamente chamada de região ativa da imagem. Por fim, o valor $\epsilon_g(f)(u)$, na posição u, da imagem erodida recebe o valor mínimo entre os máximos devido ao operador intersecção. Dessa forma a Erosão tende a aumentar as áreas mais escuras e diminuir as áreas mais claras.

O processo da Dilatação ocorre na equação (2.2), nesse caso é aplicada inicialmente a operação de mínimo $f(v) \wedge g(u - v)$, onde $f(v)$ refere-se à matriz da imagem. Dado o operador união, o valor $\delta_g(f)(u)$, na posição u recebe o valor máximo entre os mínimos. Dessa maneira a Dilatação tende a aumentar as áreas claras e diminuir as áreas escuras.

Após a análise matemática do conceito, é importante ressaltar que o processamento dos desvios condicionais existentes na morfologia matemática ainda é um dos grandes obstáculos na execução e otimização do sistema. Tratar algoritmos que variam o seu andamento em função dos valores dos dados de entrada não é algo trivial (PATTERSON, *et al.*, 2014). Mesmo que uma unidade funcional em hardware seja usada para lidar especificamente com desvios condicionais, não há uma solução geral. Cada algoritmo deve ser analisado e proposta uma estratégia (PATTERSON, *et al.*, 2014).

A solução adotada neste caso é criar aproximações dos operadores morfológicos clássicos para que seu tempo e velocidade de execução sejam consistentes independente do valor dos dados de entrada. Portanto, os desvios condicionais são substituídos por operações aritméticas, que além de demandarem menos esforço computacional do que os desvios condicionais, possuem tempo de execução uniforme. Além de ser mais rápida, a aproximação aritmética aplicada ocupa menos espaço, é mais propícia à miniaturização, consome menos energia, e reduz o número de codificações da UC (Unidade de Controle) se comparada à morfologia clássica em ambiente de hardware. Neste trabalho, são propostas aproximações aritméticas para esses dois operadores morfológicos clássicos, substituindo os desvios condicionais, presentes na Morfologia Matemática, por operações aritméticas de somas, subtrações e multiplicações, computacionalmente mais rápidas.

3 ESTUDOS PRELIMINARES

3.1 Operadores Pseudo-morfológicos

A dedução da aproximação aritmética dos operadores morfológicos emprega operações aritméticas e da álgebra booleana como interpretação da teoria dos conjuntos. Inicialmente as operações de conjunto, empregadas pelas aproximações morfológicas propostas, podem ser representadas através da álgebra booleana, vistas na Tabela 4, onde $f_1 : S \rightarrow \{0, 1\}$ e $f_2 : S \rightarrow \{0, 1\}$, onde S está em formato de matriz bidimensional.

A operação booleana AND, graficamente $\wedge$, é exibida na Tabela 4 corresponde à operação de mínimo. A Equação (3.1) mostra a representação da operação $\wedge$ através de operadores aritméticos. O operador '.' significa a instrução aritmética de multiplicação.

$$min\left\{f_1(u),\ f_2(u)\right\} = \wedge\left\{f_1(u),\ f_2(u)\right\} = f_1(u)\ .\ f_2(u) \tag{3.1}$$

Ao se generalizar para *n* funções booleanas, a representação da operação da teoria de conjuntos de mínimos se dá entre o produtório $\prod$ dessas *n* funções, como exibe a Equação (3.2), onde $f_i : S \rightarrow \{0, 1\}\ ;\ \forall\, u \in S$.

Tabela 4 – Operação de mínimo associada às propriedades da álgebra de Boole.

f_1	f_2	$min(f_1,\ f_2) = \wedge\,(f_1,\ f_2)$
0	0	0
0	1	0
1	0	0
1	1	1

Fonte: The mathematical analysis of logic (BOOLE., 1847)

Tabela 5 – Operação de máximo associada às propriedades da álgebra de Boole

f_1	f_2	$max(f_1,\ f_2) = \vee\,(f_1,\ f_2)$
0	0	0
0	1	1
1	0	1
1	1	1

Fonte: O autor (2022).

$$\begin{aligned} min\left\{f_1(u),\ f_2(u),\ f_3(u),\ ...\ ,\ f_n(u)\right\} &= \\ \wedge\left\{f_1(u),\ f_2(u),\ f_3(u),\ ...\ ,\ f_n(u)\right\} &= \\ f_1(u)\ .\ f_2(u)\ .\ f_3(u),\ ...\ ,\ f_n(u) &= \prod_{i=1}^{n} f_i(u) \end{aligned} \tag{3.2}$$

Ao se observar a Tabela 5, é possível notar que a operação booleana OR, graficamente $\vee$, corresponde à operação de máximo, como mostra a Equação (3.3). A última expressão da Equação está de acordo com a propriedade da álgebra de Boole chamada de Teorema da Involução: $A = \bar{\bar{A}}, A \to 0, 1$.

$$\begin{aligned} max\left\{f_1(u),\ f_2(u)\right\} &= \vee\left\{f_1(u),\ f_2(u)\right\} \\ &= \overline{\overline{f_1(u)\ \vee\ f_2(u)}} \end{aligned} \tag{3.3}$$

Entre a Equação (3.4) e a Equação (3.8), a operação de conjunto máximo é desenvolvida sempre a partir dos desdobramentos da Equação imediatamente anterior. A Equação (3.4) segue o Teorema de De Morgan, onde $\overline{A \vee B} = \bar{A} \wedge \bar{B},\ B \to 0, 1$.

$$max\left\{f_1(u),\ f_2(u)\right\} = \overline{\overline{f_1(u)}\ \wedge\ \overline{f_2(u)}} \tag{3.4}$$

Utilizando o complemento padrão das operações $\bar{A} = 1 - A,\ A \to 0, 1$, obtemos

$$max\left\{f_1(u),\ f_2(u)\right\} = 1 - \left(\overline{f_1(u)}\ \wedge\ \overline{f_2(u)}\right) \tag{3.5}$$

A Equação (3.6) emprega novamente o complemento padrão das operações de conjunto, dessa vez para as funções $\overline{f_1(u)}$ e $\overline{f_2(u)}$.

$$max\left\{f_1(u),\ f_2(u)\right\} = 1 - (1 - f_1(u))\ \wedge\ (1 - f_2(u)) \tag{3.6}$$

A Equação (3.7) corresponde à aproximação aritmética da operação booleana de $\wedge$, como foi demonstrado na Equação (3.1).

$$max\left\{f_1(u),\ f_2(u)\right\} = 1 - (1 - f_1(u))\ .\ (1 - f_2(u)) \tag{3.7}$$

Logo, ao se generalizar para n funções booleanas, a operação de máximo está de acordo com a Equação (3.8), $f_i : S \to \{0, 1\}$.

$$max\left\{f_1(u),\ f_2(u),\ ...\ f_n(u)\right\} = 1 - \prod_{i=1}^{n} 1 - f_i(u) \tag{3.8}$$

Desse modo, as operações da teoria de conjunto, podem ser implementadas de maneira aproximada através de operadores aritméticos. Essas aproximações matemáticas por sua vez modificam as formulações clássicas de mínimos e máximos presentes nas operações de Erosão e Dilatação vistas na Equação (2.1) e Equação (2.2), respectivamente. Na Equação (3.9), a aproximação da Erosão ϵ_g inicialmente já modifica a Erosão clássica, descrita na Equação (2.1). A operação de máximo $\vee$, entre f e $\bar{g}$, passa a ser implementada, de forma aproximada, através de operadores aritméticos, conforme Equação (3.7).

$$\epsilon_g(f)(u) = \bigcap_{v \in S} 1 - (1 - f(v))\ .\ (1 - \overline{g}(u - v)) \tag{3.9}$$

A Equação (3.10) emprega o complemento padrão das operações de conjunto para o termo $\bar{g}$

$$\epsilon_g(f)(u) = \bigcap_{v \in S} 1 - (1 - f(v))\ .\ (1 - (1 - g(u - v))) \tag{3.10}$$

A Equação (3.11) simplifica a Equação anterior.

$$\epsilon_g(f)(u) = \bigcap_{v \in S} 1 - (1 - f(v))\ .\ g(u - v) \tag{3.11}$$

Na Equação (3.12) a aproximação da Erosão ϵ_g emprega a Equação (3.1) para a aproximação da operação de mínimo $\bigcap$, através de operações aritméticas. Inicialmente, há o cálculo da expressão $1 - (1 - f(v)).g(u - v)$ entre a região v da imagem original f abrangida (casada) por g. Os 1's estão associados ao branco absoluto e 0's ao preto absoluto. Após isso, ocorre o cálculo do produtório $\prod$ entre os *n*-resultados da expressão.

$$\epsilon_g(f)(u) = \prod_{v \in S} 1 - (1 - f(v))\ .\ g(u - v) \tag{3.12}$$

Entre a Equação (3.13) e a Equação (3.14), é criada a aproximação da Dilatação ϵ_g. A operação clássica de Dilatação, descrita na Equação (2.2), é modificada. A operação de mínimo

$\wedge$, entre f e g, passa a ser implementada, de forma aproximada, através de operadores aritméticos, conforme a Equação (3.12)

$$\delta_g(f)(u) = \bigcup_{v \in S} f(v) \,.\, g(u \text{ - } v) \tag{3.13}$$

Na Equação (3.14), a aproximação da Dilatação δ_g emprega a Equação (3.8) para a aproximação da operação de máximo $\bigcup$ com operações aritméticas. Inicialmente, há o cálculo da expressão $1 - f(v) \,.\, g(u \text{ - } v)$ entre a região v da imagem original f abrangida (conectada) por g. Como mencionado anteriormente os 1's estão associados ao branco absoluto e 0's ao preto absoluto. Após isso, ocorre o cálculo do produtório $\prod$ entre os *n* resultados da expressão. Por fim, o valor 1 é subtraído pelo resultado do produtório $\prod$.

$$\delta_g(f)(u) = 1 - \prod_{v \in S} 1 - f(v) \,.\, g(u \text{ - } v) \tag{3.14}$$

3.2 Máquinas de aprendizagem extrema com operadores pseudo-morfológicos

Após entender o tipo do kernel utilizado no processamento dos atributos de entrada, é necessário discutir sobre como funciona de fato a máquina de aprendizagem que é aplicada ao problema.

Para isso, é necessário entender que uma das principais ferramentas no reconhecimento de padrões são as redes neurais. A característica mais marcante desse tipo de aplicação é a capacidade de generalização dos dados que não foram apresentados na fase de aprendizagem.

Nesse âmbito é de vital importância perceber a necessidade de garantir um bom desempenho da rede, para isso um dos principais pontos a se observar é evitar que o treinamento fique preso em regiões com menor impacto no processo de aprendizagem (HUANG, 2000) para impedir esse tipo de problema são utilizadas estratégias de controle de rede. Outro problema recorrente nesse tipo de aplicação é o alto tempo de treinamento que é necessário para habilitar a rede a fazer uma classificação correta, pois apesar da precisão extremamente alta desses mecanismos, redes de aprendizagem podem levar dias para concluir a fase de treinamento.

As redes do tipo ELM (Extreme Learning Machine), têm como principal diferença das redes convencionais a alta velocidade de treinamento mantendo uma previsão de dados assertiva. Essas redes trabalham com uma única camada oculta que não é recorrente e é baseada em um método analítico para estimar os pesos na rede de saída para qualquer inicialização aleatória dos pesos de entrada.

As ELMs têm sido amplamente aplicadas em diversas áreas como Engenharia Biomédica (AZEVEDO; *ET AL.*, 2015)(AZEVEDO; *ET AL.*, 2020)(LIMA; SILVA-FILHO; SANTOS,

2016)(LIMA; SILVA-FILHO; SANTOS, 2020)(LIMA; SILVA-FILHO; SANTOS, 2014)(PEREIRA, 2020). Visto o ótimo desempenho do método é possível trazer essa abordagem para contribuir no avanço da segurança digital de dispositivos. O estudo proposto é aplicado aos ELMs na área de segurança da informação especificamente no reconhecimento de padrões de malware.

Analisando matematicamente, na rede neural ELM os atributos de entrada x_{ti} correspondem ao conjunto $\left\{x_{it} \in \mathbb{R};\ i = 1, ...n;\ t = 1, ...v\right\}$. Existem n features extraídas do aplicativo e v vetores de dados de treinamento. A camada oculta h_j, é constituída por m neurônios, e é representada pelo conjunto $\left\{h_j \in \mathbb{R};\ j \in \mathbb{N}^*;\ j = 1, ...m\right\}$. O processo de treinamento do ELM é mais rápido porque é composto por apenas alguns passos. Inicialmente os pesos das entradas w_{ji} e $vies\ b_{jt}$ são definidos aleatoriamente. Portanto, dada uma função de ativação f:R→R, o processo de aprendizagem é dividido da seguinte forma:

- Geração aleatória dos pesos w_{ji} correspondente aos pesos entre a entrada e as camadas ocultas, e viés b_{jt}
- Cálculo da matriz H, que corresponde à saída dos neurônios da camada oculta.
- Cálculo da matriz dos pesos de saída $\beta = H^{\dagger}Y$, onde $H^{\dagger}$ é a matriz inversa generalizada de Moore-Penrose da matriz H, e Y corresponde à matriz de saídas desejadas, onde $\left\{Y_{tc} \in \mathbb{R};\ t = 1, ...v;\ c = 1, ...\zeta\right\}$. ζ é a quantidade de classes (ex. benigno, malware).

Para entender um pouco do que é a matriz pseudo-inversa é importante perceber que o conceito de matriz inversa está diretamente relacionado com a matriz identidade I. Quando uma matriz quadrada H é multiplicada por sua inversa H^{-1}, o resultado é a matriz identidade I. No entanto, nos casos de uma matriz não quadrada, uma matriz aproximadamente inversa é gerada $H^{\dagger}$. Essa matriz que foi encontrada é capaz de polarizar os pesos sinápticos entre os neurônios. A matriz pseudo-inversa $H^{\dagger}$ repele os pesos sinápticos da fronteira de decisão em direção aos extremos da diagonal secundária. Importante mencionar também que a matriz H (saída dos neurônios da camada oculta) é calculada pelo kernel φ, e o conjunto de dados e pesos de entrada são apresentados na matriz da Eq.(3.15). Os pesos de saída β e a matriz de saída desejada Y são descritos na Eq. (3.16) na e Eq. (3.17), respectivamente.

$$H_{tj} = \begin{bmatrix} \varphi_1^1 & \varphi_2^1 & \cdots & \varphi_v^1 \\ \varphi_1^2 & \varphi_2^2 & \cdots & \varphi_v^2 \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1^m & \varphi_2^m & \cdots & \varphi_v^m \end{bmatrix}_{m \times v} \tag{3.15}$$

$$\beta_{jc} = \begin{bmatrix} \beta_1^1 & \cdots & \beta_\zeta^1 \\ \beta_1^2 & \cdots & \beta_\zeta^2 \\ \vdots & \ddots & \vdots \\ \beta_1^m & \cdots & \beta_\zeta^m \end{bmatrix}_{m \times \zeta} \tag{3.16}$$

$$Y_{tc} = \begin{bmatrix} Y_1^1 & \cdots & Y_\zeta^1 \\ Y_1^2 & \cdots & Y_\zeta^2 \\ \vdots & \ddots & \vdots \\ Y_1^v & \cdots & Y_\zeta^v \end{bmatrix}_{v \times \zeta} \tag{3.17}$$

Como mencionado anteriormente, o kernel é a função matemática usada como método de aprendizagem da rede neural ELM. Por padrão, as redes neurais usam um kernel linear, que é formalmente dado na Eq. (3.18) e os resultados correspondentes são mostrados na Tabela 8 (seção 6). φ está em função de $f(x_{t,1\ldots n},\ w_{1\ldots m,1\ldots n},\ b_{1\ldots m,t})$.

$$\varphi_t^j(f) = x_{ti}.w_{ji} + b_{jt} \tag{3.18}$$

O aprendizado baseado em kernel é de extrema importância pois oferece a possibilidade de criar um mapeamento não linear de dados sem a necessidade de aumentar o número de parâmetros ajustáveis, como por exemplo a taxa de aprendizado comumente usada em redes neurais, as quais são baseadas em modelos de retropropagação. A Eq. (3.19) descreve um kernel Sigmoidal φ de uma rede ELM e os resultados correspondentes são mostrados na Tabela 8 (seção 6).

$$\begin{aligned} \varphi_t^j(f) &= Sigmoid\Big(x_{ti}.w_{ji} + b_{jt}\Big), \\ where\ Sigmoid(\xi) &= \frac{1}{1+e^{-\xi}} \end{aligned} \tag{3.19}$$

O antivírus proposto emprega mELMs (Morphological Extreme Learning Machines). Como foi apresentado, esse modelo de aprendizagem é inspirado pela matemática morfológica, a qual é baseada em operadores de imagem de Erosão e Dilatação. Os kernels pseudo-morfológicos são responsáveis por fazer a associação entre o processamento da imagem e as redes neurais artificiais. A região ativa da imagem corresponde à entrada dos neurônios nas redes. De acordo com a Eq. (3.12) referente ao operador de imagem pseudo-Erosão, o kernel ELM pode ser definido pela Eq. (3.20), onde $\Big\{i \in \mathbb{N}^*, i = 1, \ldots, n\Big\}$, $\Big\{j \in \mathbb{N}^*, j = 1, \ldots, m\Big\}$, $\Big\{t \in \mathbb{N}^*, t = 1, ..v\Big\}$.

Portanto existem n neurônios na camada de entrada (sem o viés), m neurônios na camada oculta e v vetores de dados de treinamento. O kernel está em função de φ está em função $f(x_{t,1\ldots n},\ w_{1\ldots m,1\ldots n},\ b_{1\ldots m,t})$.

$$\varphi_j^t(f) = \prod_{i=1}^{n} 1 - (1 - x_{ti})\ .\ w_{ji} + b_{jt} \tag{3.20}$$

Semelhante ao kernel de Pseudo-Erosão, a Eq. (3.21) define o kernel Pseudo-Dilatação inspirado na Eq. (3.14) que, logicamente, faz referência ao operador morfológico de Dilatação.

$$\varphi_j^t(f) = 1 - \prod_{i=1}^{n} (1 - x_{ti}\ .\ w_{ji}) + b_{jt} \tag{3.21}$$

Por último é importante ressaltar que os atributos de entrada das redes neurais artificiais ELM são as características extraídas previamente das amostras suspeitas.

4 METODOLOGIA

4.1 Métodos e materiais utilizados

Com relação ao material utilizado, este trabalho propõe um banco de dados que visa a classificação de executáveis benignos e malwares do tipo APT's. Existem 1.050 malwares e 1.050 outros executáveis benignos. Portanto, o conjunto de dados é adequado para aprender com inteligência artificial, já que ambas as classes de executáveis têm a mesma quantidade de executáveis.

Pragas virtuais foram extraídas de bancos de dados fornecidos por grupos de estudo entusiastas como o VirusShare, que é um repositório de amostras de malware para fornecer aos pesquisadores de segurança, analistas forenses e aos interessados em geral acesso a amostras de código malicioso. Quanto aos executáveis benignos, a aquisição veio de repositórios de aplicativos benignos como sourceforge, github e sysinternals. Dito isto, deve-se notar que todos os executáveis benignos foram submetidos ao VirusTotal e todos tiveram sua benignidade atestada pelos principais antivírus comerciais em todo o mundo. O diagnóstico, fornecido pelo VirusTotal, correspondentes aos executáveis benignos e maliciosos estão disponíveis no endereço virtual do banco de dados (APT, 2021).

O objetivo da criação do banco de dados é dar total possibilidade da metodologia proposta ser replicada por terceiros em trabalhos futuros. Assim, o trabalho realizado,disponibilizando gratuitamente a sua base de dados, permite a transparência e imparcialidade à pesquisa, além de demonstrar a veracidade dos resultados alcançados. Portanto, espera-se que a metodologia utilizada sirva de base para a criação de novos trabalhos científicos.

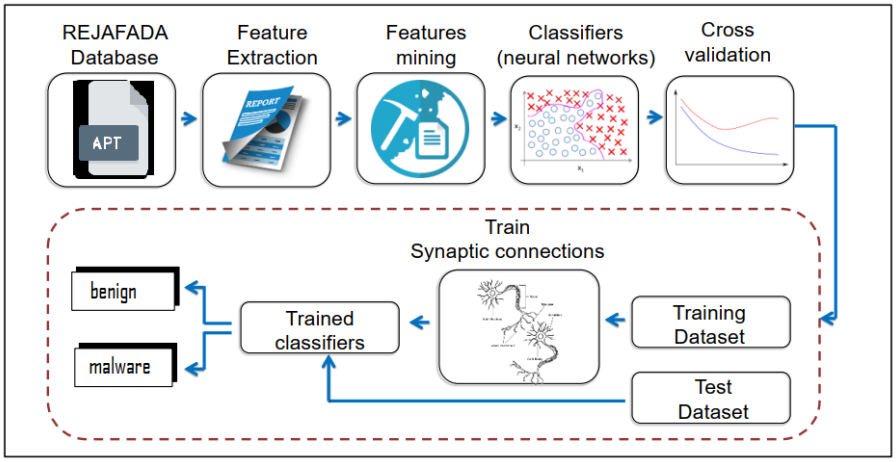
4.2 Metodologia proposta

Após identificar as limitações dos antivírus comerciais, o trabalho proposto visa criar um antivírus, dotado de inteligência artificial, capaz de diferenciar preventivamente aplicações de malware de benignas. A Fig. 1 mostra o diagrama da metodologia proposta em um diagrama de blocos.

Quanto ao processamento de todos os dados, todos os experimentos foram realizados em um computador com 8 GB de RAM (random access memory) e 2 processadores (núcleos físicos).

O diagrama abaixo divide a metodologia em partes lógicas. Primeiramente, a parte de aquisição de dados a partir do banco de dados "REJAFADA", após a extração desses recursos é realizado tratamento desses dados de forma que se tornem adequados para o processamento. Em sequência, é aplicada a classificação dos dados por meio da tecnologia proposta (Máquina de aprendizagem extrema) e por último é feita a validação cruzada com o intuito de garantir a eficácia do processo desenvolvido. O diagrama apresenta também como é realizado o processo de treinamento e testagem dos dados.

Figura 1 – Diagrama da metodologia proposta.



Fonte: O autor (2022).

4.3 Extração de recursos

A extração de recursos de arquivos executáveis utiliza um processo de desmontagem. Dessa forma, os algoritmos do executável de referência podem então ser estudados e posteriormente classificados pela rede neural descrita.Para serem analisados, um total de 409 recursos foram extraídos de cada executável. Ferramentas autorais e o pescanner são usadas para extrair recursos de arquivos executáveis. Para facilitar o entendimento dos neurônios da camada de entrada, o repositório estende a descrição das propriedades auditadas para o antivírus (APT, 2021). A seguir, são detalhados os grupos de recursos extraídos do executável pesquisado.

- Histograma de instruções, em montagem, referente à memória.
- Número de sub-rotinas que invocam TLS (Transport Layer Security).
- Número de sub-rotinas responsáveis pela exportação de dados
- APIs (Application Programming Interface) usadas pelo executável.
- Recursos relacionados a indícios de que o computador sofreu fragmentação em seu disco rígido, bem como tentativas de inicialização inválidas acumuladas.
- Modo de execução do aplicativo, existem duas opções: software com uma interface gráfica e software em execução no console.
- Recursos relacionados ao Sistema Operacional. A forense digital investiga e audita as informações internas (por exemplo, drivers) do Windows O.S.
- Recursos relacionados ao Registro do Windows (Regedit). Vale a pena notar que a vítima pode não estar livre de infecção por malware, mesmo após sua detecção e eliminação. A persistência de malefícios, mesmo após a exclusão do malware, ocorre devido à inserção de entradas maliciosas (keys) no Regedit. Então, quando o sistema operacional inicializa, o ataque cibernético é reiniciado devido à chave maliciosa, invocando a vulnerabilidade explorada pelo malware (por exemplo: redirecionar a página inicial do Internet Explorer).
- Funcionalidades relacionadas a spywares como keyloggers (captura de informações do teclado para roubo de senhas e logins) e screenloggers (captura de tela da vítima). O antivírus visa monitorar se o arquivo suspeito tenta monitorar a atividade do usuário na Internet e informações privadas.
- Funcionalidades relacionadas ao Anti-Forense Digital que são técnicas de remoção, ocultação e subversão de evidências com o objetivo de reduzir as consequências dos resultados das análises forenses.
- Recursos relacionados à criação de GUI (Graphical User Interface) do programa suspeito usado pelo malware.
- Recursos relacionados ao uso ilícito da RAM do sistema local. O antivírus investiga se o aplicativo suspeito tenta reservar, confirmar ou alterar o estado de uma região do endereço virtual de um processo;
- Recursos relacionados a sniffers. O antivírus investiga se o aplicativo suspeito tenta ler dados de pacotes de rede, que são obtidos de solicitações anteriores .
- Recursos relacionados ao soquete. Em uma aplicação convencional, um socket é criado no servidor e aguarda por uma conexão com o usuário(s). Por outro lado, o malware pode

criar soquetes no sistema local esperando que um computador malicioso remoto solicite uma conexão e, assim, receba as informações íntimas (senhas numéricas, fotos) da vítima;

- Recursos relacionados ao tráfego de rede. Checa se o arquivo suspeito tenta consultar servidores DNS e criar uma sessão FTP ou HTTP em tempo de execução.
- Recursos relacionados a programas de aplicativos utilitários.

É importante observar que cada um desses recursos individualmente não representa necessariamente um comportamento malicioso. Em seguida, a detecção de malware deve ocorrer por meio do cruzamento de informações e, consequentemente, ponderação de todos os recursos destacados.

5 CLASSIFICAÇÃO

Quanto ao reconhecimento de padrões de malware, a tarefa principal é relacionada a atribuir a cada arquivo investigado uma classe (rótulo) com base em suas características. Então, com base em um conjunto de arquivos chamado conjunto de treinamento, podem ser feitas hipóteses sobre as diferentes classes associadas ao antivírus proposto. A partir disso o classificador estima a classe de documentos inéditos comparando características comportamentais auditadas a tempo com aquelas capturadas durante a fase de treinamento.

O objetivo do classificador é obter uma função de separação entre as classes do antivírus (malware, benigno). Desta forma, quando é apresentado um executável inédito, a função é aplicada e, em seguida, atribui uma classe à qual este arquivo deve pertencer. Matematicamente, $c = f(x)$, onde $x = x_1, x_2, \ldots, x_t$ é o vetor traçado a partir do arquivo investigado, t corresponde a quantidade de recursos dinâmicos analisados e c a classe, finalmente f é a função do mapeamento do classificador.

Ao estabelecer um classificador linear, o classificador representa uma linha que possui a função de separar os padrões de classes diferentes. Portanto, cada caso investigado será classificado de acordo com o lado da linha em que está mapeado. Visando o reconhecimento comportamental dos malwares modernos, devem ser usados classificadores que possam construir uma separação não linear entre as classes. Para comprovar embasamento teórico, o antivírus autoral emprega redes neurais não lineares, especificamente, redes neurais pseudo-morfológicas extremas.

6 RESULTADOS

6.1 Resultados das redes ELM

Sete tipos diferentes de kernel foram usados nas redes neurais. No estado da arte, cinco desses kernels são descritos por HUANG et al. (2012) e são eles: Wavelet, Sigmoide, Seno, *Hard Limit* e *Tribas Transforms* (funções de base trigonométricas) (HUANG, 2012). Além disso, kernels autorais adicionais são empregados: Pseudo-dilatação e Pseudo-erosão.

O kernel do tipo Wavelets não possui camada oculta (HUANG, 2012). Os cálculos são baseados na transformação dos dados de entrada e podem funcionar de forma similar aos kernels que contém arquiteturas com camadas ocultas (HUANG, 2012). Uma boa capacidade de generalização destes kernels depende de uma escolha ajustada de parâmetros (C, γ) (HUANG, 2012). O parâmetro de custo C refere-se a um ponto de equilíbrio razoável entre a largura da margem do hiperplano e a minimização do erro de classificação em relação ao treinamento. Já o parâmetro do γ controla o limite de decisão em função das classes (HUANG, 2012). Não existe um método universal com relação ao sentido de escolha dos parâmetros (C, γ).

Nesse trabalho, há a investigação dos parâmetros (C, γ) inspirado no método proposto por HUANG, *et al.* (2012), que consiste em treinar sequências crescentes de C e γ, matematicamente, 2^n, onde $n = \left\{-24, -10, 0, 10, 25\right\}$ (HUANG, 2012). A hipótese é verificar se estes parâmetros com valores diferentes dos padrões; $(C = 1, \gamma = 1)$, geram melhores resultados. No núcleo linear, há apenas a investigação do parâmetro de custo C, nesse caso não é possível explorar o parâmetro γ (HUANG, 2012).

Para cada combinação utilizada é empregada validação cruzada através do método k-fold, onde k=10. O objetivo da utilização desse método é que os resultados alcançados não sejam influenciados por conjuntos de treinamento e teste. Por isso, o total de de dados é dividido em dez partes.

Na primeira execução, a primeira parte é destinada ao conjunto de testes, enquanto a outra é reservada para o treinamento. Essa alternância ocorre por dez execuções até que todas as partes tenham sido aplicadas à fase de teste. A acurácia do ELM é dada pela média aritmética da taxa de acerto obtida nas dez iterações.

Como mencionado anteriormente, na rede ELM não há retropropagação de dados. Portanto, o objetivo do método de validação cruzada k-fold não é estabelecer uma parada de critério para evitar overfitting (excesso de treinamento), mas para verificar que o classificador sofre mudanças abruptas em sua acurácia dependendo dos conjuntos destinados a treinamento e teste. Dito isto, o objetivo é que classificadores tendenciosos, em relação a uma determinada classe, não tenham suas taxas de acurácia favorecidas.

A Tabela 6 detalha os resultados obtidos pelas redes neurais ELMs com kernel Wavelets. Cada linha nesta tabela contém 10 execuções referentes à validação cruzada do método k-fold, onde k=10. Em relação à acurácia no teste fase, o desempenho médio máximo foi de 76,24% na distinção entre casos benignos e malignos com a parâmetros $(C, \gamma) = (2^{-10}, 2^{-24})$. Na Tabela 6, há apenas as descrições de melhor e pior caso, nesta ordem, para cada núcleo.

A Tabela 7 apresenta os resultados alcançados pela rede ELM trabalhando com kernel linear. É feita a análise apenas do parâmetro C, pois não é possível explorar o parâmetro γ em um kernel linear (HUANG, 2012). Cada linha na Tabela 7 contém 10 execuções distintas sobre o $k - fold$ na validação cruzada, onde $k = 10$. O máximo e mínimo de acurácia foram de 93,48% e 50,00%, respectivamente. Visto isso, é possível afirmar que a investigação do parâmetro de custo C, é capaz de maximizar a acurácia na identificação .

A Tabela 8 mostra os resultados obtidos pela rede ELM utilizando os kernels Sigmoid, Sine, Hard Limit, Tribas (funções de base trigonométricas), pseudo-dilatação e pseudo-erosão. Eles empregam uma arquitetura de camada oculta. Em seguida, é apresentada a análise em relação à quantidade de neurônios na camada oculta desses núcleos. A hipótese é verificar se as arquiteturas que exigem maior volume de cálculos, como dobrar o número de neurônios na camada oculta, são capazes de gerar melhores taxas de acurácia em comparação com arquiteturas que exigem uma quantidade menor de cálculos. Partindo disso é feíta a avaliação de dois tipo de arquiteturas; são empregados 100 e 500 neurônios em suas respectivas camadas ocultas. Estas arquiteturas possuem antecedentes de excelente acurácia na aplicação de redes ELM na área de Engenharia Biomédica (LIMA; SILVA; LUZ, 2021). Cada linha da Tabela 8 contém 10 execuções distintas referentes ao método k-fold, onde k=10. Em relação à acurácia, o desempenho médio máximo foi de 93,62% com desvio padrão de 1,31% através do kernel de pseudo-diltação dotado de 500 neurônios em sua camada oculta.

Tabela 6 – Resultados das redes neurais ELM. Os parâmetros (C, γ) variam de acordo com o conjunto $\{2^{-24}, 2^{-10}, 2^{0}, 2^{10}, 2^{25}\}$. São exibidas apenas as melhores e piores acurácias.

kernel	(C, γ)	Acurácia de treinamento (%)	Acurácia de teste (%)	Tempo de treino (seg.)	Tempo de teste (seg.)
Wavelets	$(2^{-10}, 2^{-24})$	99,95 ± 0,02	**76,24 ± 2,54**	0,70 ± 0,03	0,09 ± 0,01
	$(2^{-10}, 2^{25})$	**51,86 ± 1,13**	50,95 ± 3,31	0,74 ± 0,03	0,09 ± 0,01

Fonte: O autor (2022).

Tabela 7 – Resultados das redes neurais ELM com o kernel Linear. Os parâmetros C variam de acordo com o conjunto $\left\{2^{-24}, 2^{-10}, 2^{0}, 2^{10}, 2^{25}\right\}$. São exibidas apenas as melhores e piores acurácias.

kernel	C	Acurácia de treinamento (%)	Acurácia de teste (%)	Tempo de treino (seg.)	Tempo de teste (seg.)
Linear	2^{25}	**95,29 ± 0,15**	**93,48 ± 1,27**	0,38 ± 0,02	0,03 ± 0,01
	2^{-24}	50,00 ± 0,00	50,00 ± 0,00	0,38 ± 0,02	0,03 ± 0,00

Fonte: O autor (2022).

Tabela 8 – Resultados das redes ELM. O número de neurônios na camada oculta variam de acordo com os dados 100, 500.

kernel	neurônios	Acurácia de treinamento (%)	Acurácia de teste (%)	Tempo de treino (seg,)	Tempo de teste (seg,)
Sigmoide	500	75,22 ± 0,31	69,90 ± 2,69	0,49 ± 0,04	0,02 ± 0,00
	100	67,11 ± 0,77	65,29 ± 2,86	0,12 ± 0,03	0,00 ± 0,00
Seno	500	83,33 ± 0,50	67,43 ± 2,52	0,49 ± 0,02	0,01 ± 0,00
	100	65,85 ± 1,12	60,00 ± 3,30	0,11 ± 0,02	0,00 ± 0,01
Hard limit	100	50,00 ± 0,00	50,00 ± 0,00	0,13 ± 0,02	0,00 ± 0,00
	500	50,00 ± 0,00	50,00 ± 0,00	0,56 ± 0,03	0,01 ± 0,01
Tribas	100	50,00 ± 0,00	50,00 ± 0,00	0,12 ± 0,02	0,00 ± 0,00
	500	50,00 ± 0,00	50,00 ± 0,00	0,43 ± 0,05	0,01 ± 0,01
Pseudo-Dilatação	500	**95,60 ± 0,20**	**93,62 ± 1,31**	0,55 ± 0,03	0,01 ± 0,01
	100	94,57 ± 0,30	93,19 ± 1,63	0,21 ± 0,02	0,00 ± 0,01
Pseudo-Erosão	500	95,60 ± 0,20	93,57 ± 1,37	0,58 ± 0,03	0,01 ± 0,00
	100	94,49 ± 0,21	93,24 ± 1,66	0,22 ± 0,03	0,00 ± 0,01

Fonte: O autor (2022).

6.2 Resultados em relação ao estado da arte

Nesta seção, o antivírus autoral é comparado com os antivírus atuais. Para evitar comparações injustas, o estágio de extração de recursos é padronizado monitorando 409 comportamentos que o executável suspeito pode fazer quando executado propositalmente. O antivírus autoral aplica redes morfológicas superficiais utilizando um kernel de pseudo-dilatação, além disso sua camada oculta possui 500 neurônios.

Por outro lado, o antivírus por LIMA, *et al.* (2021) emprega redes neurais superficiais baseadas em retropropagação. LIMA,*et ai.* (2021) investiga onze funções distintas de aprendizado para otimizar a acurácia de seu antivírus. Para cada função de aprendizagem, LIMA, *et al.* (2021) explora 4 arquiteturas de camadas ocultas.

O antivírus proposto também é comparado aos antivírus baseados no conceito de rede neural profunda. No presente trabalho, são replicados os antivírus feitos por SU, *et al.* (2018), HOU, S. *et al.* (2016), MANIATH, S. *et al.* (2017), HARDY, W. *et al.* (2016) e FARUKI, P. *et al.*

(2019). Além disso, o firewall desenvolvido por WOZNIAK, M. *et al.* (2015) foi avaliado. O trabalho foi replicado utilizando o conjunto de dados próprio para evitar comparações injustas.

A Figura 2 e a Figura 3 são representações gráficas dos resultados descritos na Tabela 9. A Figura 2 (a) mostra os box-plots para a melhor acurácia no treinamento. O antivírus autoral obteve uma acurácia média de 95,60%. O antivírus feito por LIMA, *et al.* (2021) obteve acurácia média de 49,04% e 95,28%, em seus piores e melhores cenários, respectivamente. Esses resultados foram obtidos usando as funções de aprendizado "Resilient backpropagation" e "Conjugate gradient backpropagation with Fletcher-Reeves updates", respectivamente. A pior arquitetura tem uma única camada oculta contendo 500 neurônios, enquanto a melhor arquitetura possui duas camadas contendo 100 neurônios em cada uma. O trabalho realizado por MANIATH, *et al.* (2017) obteve o melhor desempenho baseado em aprendizagem profunda, alcançando uma acurácia média de 99,77% no estágio de treinamento.

A Figura 2 (b) apresenta os boxplots, na fase de teste, em relação aos antivírus autoral e de última geração. O antivírus autoral obteve desempenho médio de 93,62% com desvio padrão de 1,31%. O antivírus feito por LIMA, *et al.* (2021) obteve acurácia média de 49,11% e 94,86%, em seus piores e melhores cenários, respectivamente. Partindo desse fator, corrobora-se que as redes neurais baseadas em retropropagação podem sofrer grandes variações, em suas acurácias, dependendo de seus parâmetros de configuração. Então, a decisão tomada por LIMA, *et al.* (2021) foi sensata. Este antivírus de última geração explora diferentes funções de aprendizado, gradientes e arquiteturas para otimizar a acurácia de suas redes neurais com base na retropropagação de dados. O trabalho realizado por MANIATH, *et al.*2017) é o melhor antivírus baseado em aprendizado profundo, alcançando uma acurácia média de 96,19% no estágio de teste.

A Figura 3(a) e a Figura 3(b) mostram boxplots envolvendo o tempo gasto nas fases de treinamento e teste, respectivamente. O antivírus autoral consome apenas 0,55 segundos para concluir, em média, seu treinamento. Em relação ao tempo de treinamento, o antivírus feito por HOU, *et al.* (2016) é o mais lento consumindo 6.927,86 segundos. Em relação ao tempo gasto durante a fase de teste, todas as técnicas consumiram tempos muito próximos sem grandes discrepâncias.

A Tabela 10 mostra as matrizes de confusão das técnicas apresentadas na Tabela 9 em termos percentuais. A matriz de confusão é importante para verificar a qualidade do aprendizado supervisionado. Na Tabela 10, B. e M. são abreviações de Benigno e Malware. As classes desejadas estão dispostas na etiqueta vertical enquanto as classes obtidas estão na etiqueta horizontal. Na matriz de confusão, a diagonal principal é ocupada por casos em que a classe obtida coincide com a classe esperada, denominados casos verdadeiros positivos. Logo, um bom classificador tem a diagonal principal ocupada por valores altos e outros elementos possuem valores baixos.

A Tabela 10 mostra as principais diagonais destacadas em negrito. O antivírus proposto,

em fase de teste, classificou erroneamente em média 2,84% dos casos como benignos quando eram casos de malware (falso negativo). Seguindo o mesmo raciocínio, houve uma classificação média de 0,51% dos casos ditos malware quando eram na verdade aplicativos benignos (falso positivo).

Ainda em relação à Tabela 10, a sensibilidade e a especificidade referem-se à capacidade do antivírus em identificar malware e identificar aplicativos benignos, respectivamente. O trabalho proposto apresenta a matriz de confusão em termos percentuais para facilitar a interpretação da sensibilidade e especificidade. Em síntese, a sensibilidade e especificidade são apresentadas na própria matriz de confusão, descrita na Tabela 10. Por exemplo, o antivírus autoral tem uma média de 92,48% em relação tanto à sensibilidade quanto aos verdadeiros positivos. Seguindo o mesmo raciocínio, o antivírus autoral obtém, em média, 94,76% para ambas especificidades e verdadeiros negativos.

A Tabela 11 mostra os valores de t-student paramétricos e não paramétricos do testes de hipótese de Wilcoxon entre o antivírus proposto e o estado da arte. É possível concluir que o antivírus autoral é estatisticamente diferente de todas as outras amostras, com exceção do antivírus feito por SU, *et al.* (2018). A explicação é que tanto no teste t-student paramétrico quanto no teste não paramétrico de Wilcoxon, a hipótese nula foi rejeitada.

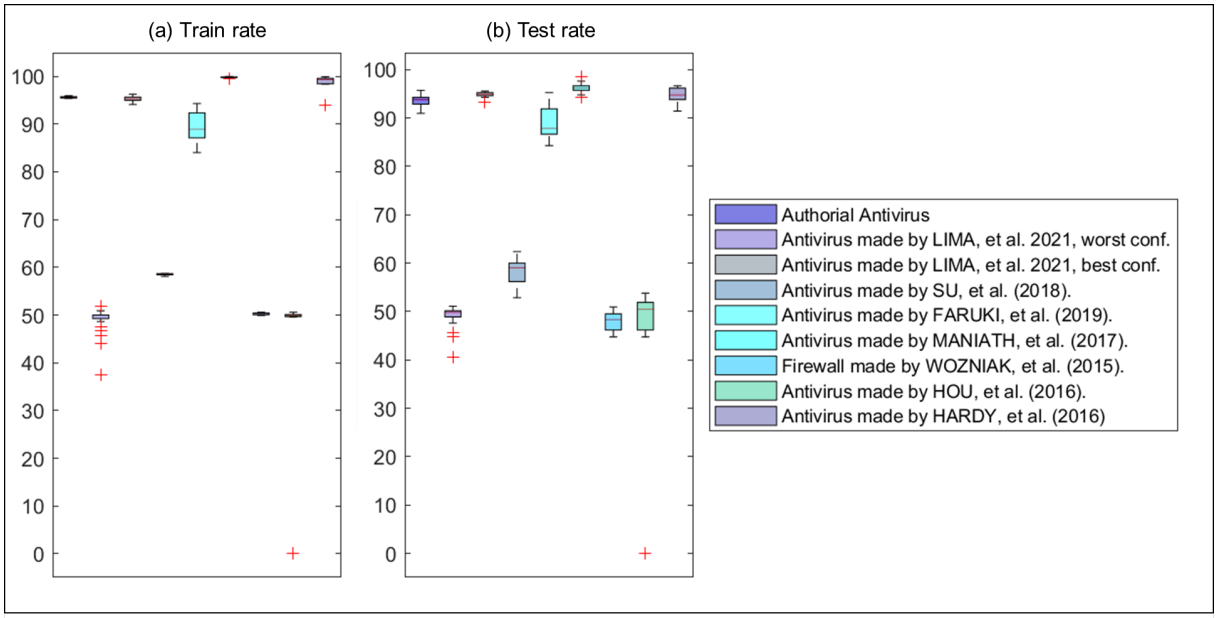
O antivírus autoral demonstrou uma grande vantagem quando comparado ao estado da arte.O antivírus atinge um desempenho médio de 93,62% dentro de um treinamento médio de 0,55 segundos. Sabendo que 8 (oito) novos malwares são lançados a cada segundo (INTEL, 2018), é logicamente coeso que um antivírus recém-lançado pode já estar obsoleto e exigir novo treinamento por meio de uma vulnerabilidade recém-descoberta. Em síntese, o tempo de aprendizado de um antivírus não deve ser discrepante em relação à taxa de criação de novos malwares em todo o mundo.

Tabela 9 – Comparação entre o antivírus autoral e o estado da arte.

Técnica	Acurácia de treino (%)	Acurácia de teste (%)	Tempo de treino (seg.)	Tempo de teste (seg.)
Antivírus autoral	95,60 ± 0,20	93,62 ± 1,31	0,55 ± 0,03	0,01 ± 0,01
LIMA, *et al,* (2021), pior conf.	49,04 ± 2,69	49,11 ± 2,10	0,46 ± 0,06	0,03 ± 0,01
LIMA, *et al,* (2021), melhor conf.	95,28 ± 0,53	94,86 ± 0,47	11,53 ± 2,91	0,03 ± 0,02
SU, R, *et al,* (2018)	58,49 ± 0,20	58,38 ± 2,91	202,42 ± 2,66	0,12 ± 0,02
FARUKI, *et al,* (2019)	89,25 ± 3,28	88,90 ± 3,84	62,90 ± 2,21	0,05 ± 0,01
MANIATH, *et al,* (2017)	**99,77 ± 0,14**	**96,19 ± 1,25**	860,30 ± 32,32	0,05 ± 0,01
WOZNIAK, *et al,* (2015)	50,22 ± 0,21	48,00 ± 1,90	3144,84 ± 192,06	0,11 ± 0,01
HOU, *et al,* (2016)	44,99 ± 15,79	45,10 ± 16,09	6927,86 ± 440,73	0,01 ± 0,01
HARDY, *et al,* (2016)	98,70 ± 1,78	94,71 ± 1,69	373,53 ± 14,72	0,06 ± 0,01

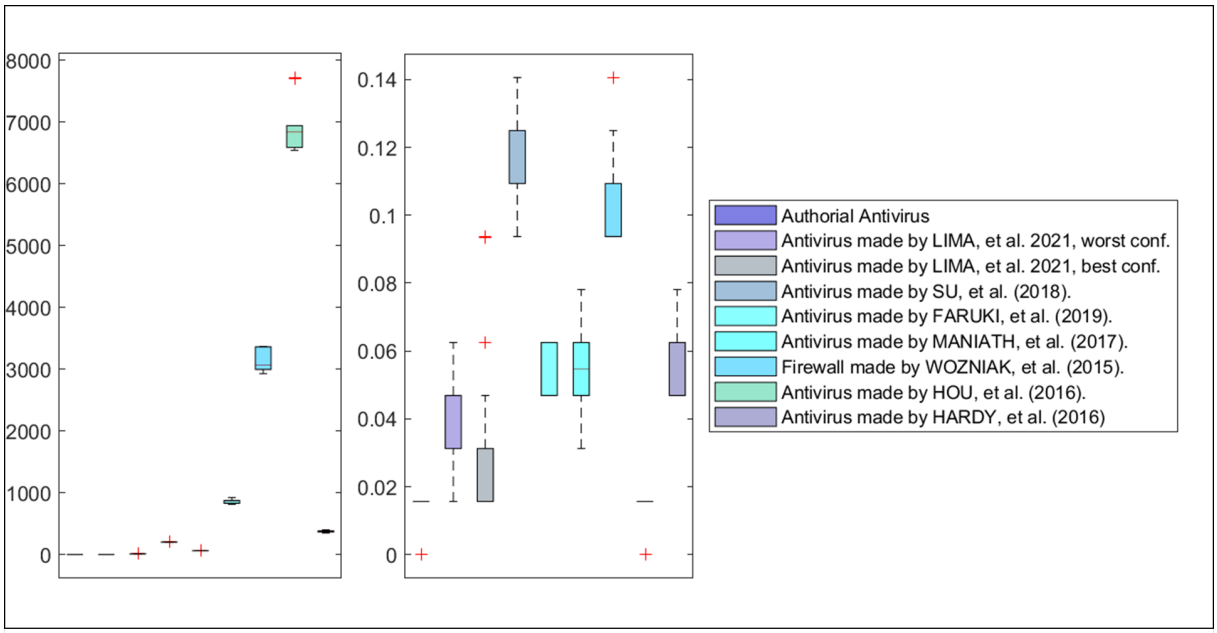
Fonte: O autor (2022).

Figura 2 – Boxplots referente as acurácias do antivírus autoral e do estado da arte.



Fonte: O autor (2022).

Figura 3 – Boxplots dos tempos de processamento do antivírus autoral e do estado da arte.



Fonte: O autor (2022).

Tabela 10 – Matriz de confusão do Antivírus Autoral e Estado da Arte em (%).

		Treino		Teste	
Técnica		M.	B.	M.	B.
Antivírus autoral	M.	**94,18 ± 0,33**	2,98 ± 0,19	**92,48 ± 2,08**	5,24 ± 2,55
	B.	5,82 ± 0,33	**97,02 ± 0,19**	7,52 ± 2,08	**94,76 ± 2,55**
Antivírus de LIMA,	M,	**28,30 ± 24,36**	47,55 ± 21,79	**26,56 ± 24,56**	44,10 ± 24,39
et al. pior conf. (2021)	B,	48,37 ± 32,83	**45,78 ± 21,51**	33,44 ± 29,80	**42,57 ± 23,95**
Antivírus de LIMA,	M,	**96,21 ± 0,33**	5,61 ± 0,84	**94,84 ± 0,56**	5,11 ± 0,82
et al. melhor conf. (2021)	B,	3,79 ± 0,33	**94,39 ± 0,84**	5,16 ± 0,56	**94,89 ± 0,82**
Antivírus de SU,	M,	**83,07 ± 0,41**	66,10 ± 0,57	**83,22 ± 3,18**	66,32 ± 4,38
et al. (2018)	B,	16,93 ± 0,41	**33,90 ± 0,57**	16,78 ± 3,18	**33,68 ± 4,38**
Antivírus de	M,	**82,18 ± 8,37**	3,67 ± 2,48	**81,47 ± 8,75**	3,67 ± 2,85
FARUKI, *et al.* (2019)	B,	17,82 ± 8,37	**96,33 ± 2,48**	18,53 ± 8,75	**96,33 ± 2,85**
Antivírus de	M,	**99,68 ± 0,31**	0,14 ± 0,15	**96,38 ± 1,66**	3,99 ± 1,63
MANIATH, *et al.* (2017)	B,	0,32 ± 0,31	**99,86 ± 0,15**	3,62 ± 1,66	**96,01 ± 1,63**
Firewall de	M,	**35,11 ± 24,23**	14,89 ± 23,97	**34,00 ± 23,50**	16,00 ± 25,78
WOZNIAK, *et al.* (2015)	B,	34,89 ± 24,08	**15,11 ± 24,33**	36,00 ± 24,88	**14,00 ± 22,57**
Antivírus de	M,	**4,99 ± 15,79**	45,01 ± 15,82	**5,10 ± 16,11**	44,90 ± 16,03
HOU, *et al.* (2016)	B,	5,01 ± 15,84	**54,99 ± 15,82**	4,90 ± 15,51	**45,10 ± 16,09**
Antivírus de	M,	**98,36 ± 1,98**	0,95 ± 1,59	**94,88 ± 1,58**	5,46 ± 2,13
HARDY, *et al.* (2016)	B,	1,64 ± 1,98	**99,05 ± 1,59**	5,12 ± 1,58	**94,54 ± 2,13**

Fonte: O autor (2022).

Tabela 11 – 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	2,6677e-38	1	1,94438e-11
Antivírus autoral *vs* Antivírus de LIMA, *et al.* (2021), melhor conf.	1	5,38426e-05	1	4,18126e-06
Antivírus autoral *vs* Antivírus de SU, *et al.* (2018)	1	4,10589e-31	1	2,33755e-11
Antivírus autoral *vs* Antivírus de FARUKI, *et al.* (2019)	1	3,13276e-07	1	2,59838e-05
Antivírus autoral *vs* Antivírus de MANIATH, *et al.* (2017)	1	5,52345e-10	1	1,61456e-09
Antivírus autoral *vs* Firewall de WOZNIAK, *et al.* (2015)	1	1,9662e-38	1	2,33755e-11
Antivírus autoral *vs* Antivírus de HOU, *et al.* (2016)	1	2,07471e-17	1	2,37833e-11
Antivírus autoral *vs* Antivírus de HARDY, *et al.* (2016)	1	0,000682081	1	0,00702735

Fonte:O autor (2022).

7 CONCLUSÃO

Dado o crescente número de novos malwares, é de vital importância que as plataformas de detecção disponibilizem mecanismos de vigilância cibernética que atendam às demandas dos clientes de forma preventiva. Caso contrário, nos cenários em que ocorrem falhas na identificação de aplicativos maliciosos, há iminência de que dados confidenciais de clientes sejam disponibilizados por pessoas não autorizadas.

Assim, infere-se que a seleção do antivírus tem um papel importante no combate às pragas virtuais. Na avaliação apresentada, a variação na detecção de malware do tipo APT foi de 0% a 99,52%, dependendo de qual antivírus comercial foi escolhido. O presente trabalho realizou a análise de 89 antivírus disponíveis comercialmente. Em média, eles conseguiram detectar 68,30% dos malwares. Feita a análise das amostras, foi possível identificar que os antivírus, em média, relataram falsos negativos e foram omitidos em 17,76% e 31,94% dos casos, respectivamente. No presente trabalho, a plataforma VirusTotal foi utilizada para submeter, de forma automatizada, o malware aos antivírus. Deve-se ressaltar que no VirusTotal, não existe a possibilidade de escolher a versão shareware dos antivírus. Então, não foi possível fazer comparações entre os recursos gratuitos e comerciais de um mesmo antivírus. Deduz-se que os serviços oferecidos nas versões shareware apresentam desempenho significativamente inferior ao das versões completas.

Vale ressaltar também que na análise apresentada, os malwares analisados são de domínio público, empregados em atividades maliciosas. Mesmo assim, uma grande quantidade dos antivírus comerciais avaliados não tinham conhecimento sobre a existência dos arquivos infectados investigados.

Para suprir as limitações dos antivírus comerciais, foram desenvolvidos antivírus baseados em inteligência artificial que são capazes de auditar milhares de malwares e aprender, estatisticamente, quais são suas características maliciosas. Portanto, após o aprendizado, o antivírus inteligente pode identificar e classificar o malware recém-criado de acordo com a comparação entre seus recursos e os catalogados durante a fase de aprendizado. Pode-se tornar autônomo o aprendizado do comportamento do malware. Assim, não haveria necessidade de esperar um usuário ser infectado e, posteriormente, denunciar uma atitude suspeita, para só então o antivírus tomar alguma ação em relação à descoberta do malware.

Portanto, com o intuito de contribuir no combate na disseminação de arquivos maliciosos foi desenvolvido um antivírus autoral capaz de classificar os executáveis entre benignos e malware. Ao todo, o antivírus monitora e pondera, estatisticamente, 409 ações que o arquivo suspeito pode realizar quando executado no sistema operacional. Em ambiente controlado, o antivírus monitora alterações no Registro (Banco de Dados) do sistema operacional e os

rastreamentos de chamadas executadas por todos os processos gerados pelo malware. Todo o reconhecimento do padrão, referente às 409 ações suspeitas, é realizado por redes neurais extremas.

Ao invés de kernels convencionais, os kernels autorais são empregados para ELMs. A rede ELM foi escolhida pois tem como principal característica a velocidade de treinamento e previsão de dados assertiva quando comparada às redes neurais convencionais. O kernel pseudo-Dilation autoral é capaz de distinguir malware do tipo APT de aplicativos benignos em 93,62% dos casos, acompanhado por um tempo de treinamento de 0,55 segundos. Por último, é interessante destacar que o intuito deste trabalho é trazer uma visão diferente acerca do desempenho dos antivírus atuais, oferecendo alternativas criativas e eficientes de solucionar o problema de detecção de malwares do tipo APT.

REFERÊNCIAS

APT. Retrieval for apt malware analysis. 2021. Disponível em: <https://github.com/nevesisaac/APT>. Citado 5 vezes nas páginas 14, 15, 17, 27 e 28.

AZEVEDO, W. W.; *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. Citado 2 vezes nas páginas 11 e 23.

AZEVEDO, W. W.; *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.**, p. 1–30, 2020. Citado 2 vezes nas páginas 11 e 23.

BOOLE., G. The mathematical analysis of logic. Philosophical Library, 1847. Citado na página 20.

FARUKI, P.; BUDDHADEV, B. Droiddivesdeep: Android malware classification via low level monitorable features with deep neural networks. **International Conference on Security Privacy**, 2019. Citado na página 15.

HARDY, W.; LINGWEI, C. t. Dl 4 md : A deep learning framework for intelligent malware detection. **In Int'l Conf. Data Mining**, p. 61–67, 2016. Citado na página 16.

HOU, S.; SAAS, A. Droiddelver: An android malware detection system using deep belief network based on api call blocks. **Web-Age Information Management. WAIM 2016 International Workshops, MWDA, SDMMW, and SemiBDMA**, 2016. Citado na página 16.

HUANG, G. B. *et al.*. Classification ability of single hidden layer feedforward neural networks. **The IEEE Transactions on Neural Networks and Learning Systems**, v. 11(3), p. 799–801, 2000. Citado na página 23.

HUANG, G. B. *et al.*. Extreme learning machine for regression and multiclass classification. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 42(2), p. 513–519, 2012. Citado 2 vezes nas páginas 32 e 33.

INTEL. **McAfee Labs**. Accessed on Feb 2020, 2018. Disponível em: <https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2018.pdf>. Citado 3 vezes nas páginas 11, 16 e 36.

LIMA, S. **Limitation of COTS antiviruses: issues, controversies, and problems of COTS antiviruses**. [S.l.]: 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. Citado 2 vezes nas páginas 12 e 14.

LIMA, S.; SILVA-FILHO, A. G.; SANTOS, W. P. Detection and classification of masses in mammographic images in a multi-kernel approach. **Computer Methods and Programs in Biomedicine**, v. 134, p. 11–29, 2016. Citado 2 vezes nas páginas 11 e 24.

LIMA, S.; SILVA, H.; LUZ, J. *et al.*. Artificial intelligence-based antivirus in order to detect malware preventively. **Progress in Artificial Intelligence**, 2021. Citado 3 vezes nas páginas 10, 11 e 33.

LIMA, S. M. L.; SILVA-FILHO; SANTOS, W. P. **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**. [s.n.], 2020. 27-64 p. Disponível em: <https://novapublishers.com/shop/understanding-a-cancer-diagnosis/>. Citado 2 vezes nas páginas 11 e 24.

LIMA, S. M. L.; SILVA-FILHO, A. G.; SANTOS, W. P. D. 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. Citado 3 vezes nas páginas 11, 16 e 24.

MANIATH, S.; ASHOK, A. Deep learning lstm based ransomware detection. **Recent Developments in Control, Automation Power Engineering**, 2017. Citado na página 15.

PEREIRA, J. M. S. *et al.*. **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/: [s.n.], 2020. 117-132 p. Citado 2 vezes nas páginas 11 e 24.

SANS. **SANS Institute InfoSec Reading Room. Out with The Old, In with The New: Replacing Traditional Antivirus**. Accessed on Feb 2020, 2017. Disponível em: <https://www.sans.org/reading-room/whitepapers/analyst/old-new-replacing-traditional-antivirus-37377>. Citado 2 vezes nas páginas 12 e 14.

SANTOS, W. P. **Mathematical Morphology In Digital Document Analysis and Processing**. [S.l.]: New York: Nova Science, 2011. v. 8. 159-192 p. Citado na página 18.

SU, J.; VASCONCELLOS D., t. Lightweight classification of iot malware based on image recognition. **2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)**, 2018. Citado na página 15.

WOZNIAK, M.; SILKA, J. Recurrent neural network model for iot and networking malware threads detection. **IEEE Transactions on Industrial Informatics.**, 2015. Citado na página 15.