Revista de Sistemas de Informação da FSMA
n. 26 (2020) pp. 31-50

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

Antivírus dotado de Máquina Morfológica de Aprendizado Extremo

Sidney Marlon Lopes de Lima, *Departamento de Eletrônica e Sistemas (DES) - Universidade Federal de Pernambuco (UFPE)*
Heverton Kleidson de Lima Silva, *Empresa Bidweb Security IT – Recife, Brasil*
João Henrique da Silva Luz. *Empresa Bidweb Security IT – Recife, Brasil*
Samuel Lopes de Paula, *Curso de Segurança da Informação, UniSãoMiguel– Recife, Brasil*
Hercília Juliana do Nascimento Lima, *Curso de Segurança da Informação, UniSãoMiguel – Recife, Brasil*
Anna Beatriz Augusta de Andrade, *Curso de Segurança da Informação, UniSãoMiguel – Recife, Brasil*
Alisson Marques da Silva, *Curso de Segurança da Informação, UniSãoMiguel – Recife, Brasil*
Washington Wagner Azevedo da Silva, *Departamento de Engenharia Biomédica - Universidade Federal de Pernambuco (UFPE)*
Wellington Pinheiro dos Santos, *Departamento de Engenharia Biomédica - Universidade Federal de Pernambuco(UFPE)*

Resumo. **A cada segundo, são lançados 8 (oito) novos *malwares* (malicioso + software). Baseado nessa constatação, nosso objetivo é propor um antivírus dotado de inteligência artificial, capaz de identificar *malwares* por meio de modelos baseados em redes neurais de treinamento rápido e de alta acurácia. O antivírus proposto é dotado de máquinas morfológicas de aprendizado extremo (mELMs) inspiradas na teoria de processamento de imagem da Morfologia Matemática. Os resultados obtidos são comparados com abordagens clássicas e avaliadas através de métricas de classificação amplamente usadas. Em média, o antivírus proposto consegue distinguir aplicativos *malwares* dos benignos em 99,80% dos casos após um tempo de treinamento de 9,32 segundos. Estabelecemos a existência de uma relação média entre a acurácia percentual e tempo de treinamento em ordem inversa. As análises feitas demonstram que o antivírus proposto é 96 vezes melhor do que *Deep Learning* (Aprendizado Profundo) de última geração.**

Palavras-Chave—Malwares; Antivírus; Redes Neurais Artificiais; Detecção de Malwares em tempo real.

***Abstract*—Each second, 8 (eight) new *malwares* (malicious + software) are launched. Based on this finding, our goal is to propose an antivirus endowed with artificial intelligence, which can identify malware through models based on neural networks of rapid training and high accuracy. The proposed antivirus is equipped with morphological extreme learning machines (mELMs), which are inspired by the image processing theory of Mathematical Morphology. The results shown here are compared with classical approaches and evaluated through widely used classification metrics. On average, the antivirus proposed can distinguish malware from benign applications in 99.80% of cases, with a training time of 9.32 seconds. We found an average ratio between percentage accuracy and training time in reverse order. The proposed antivirus is 96 times better than state-of-the-art Deep Learning.**

***Index Terms—Malwares*; Antivirus; Artificial Neural Networks; Real-time malware detection.**

Autor correspondente: Dr. Sidney Marlon Lopes de Lima sidney.lima@ufpe.br

I. INTRODUÇÃO

A sociedade contemporânea tem como uma das suas principais fontes de riqueza o conhecimento. Segundo predições de TOFLER (1981), a informação se tornaria até mais importante do que a terra, o trabalho, o capital e a matéria-prima. Nota-se, em tempos contemporâneos, que a informação assumiu papel relevante, visto que ela é compartilhada em tempo real e em qualquer parte do mundo [24]. Os grandes *datacenters*, o armazenamento em nuvem e a mineração de dados são marcos que concretizam esse avanço do compartilhamento. Assim, pode-se dizer que o controle da informação é de suma importância no ambiente corporativo, acadêmico e científico.

Pesquisa do MCSI (Índice Global de Segurança na Computação da *Microsoft*), realizada em 2014, revela que 15% dos internautas ouvidos já foram vítimas de ataques virtuais. Em média, cada uma dessas vítimas perdeu U$ 158 dólares [18]. O relatório contempla 20 países e, com base nos dados coletados, estima-se que o roubo de dados digitais ultrapassa US$ 23 bilhões por ano mundialmente [18].

Cada vez mais as empresas vêm investindo na segurança digital através da implantação de antivírus, *firewall* e biometria [18]. Apesar dos esforços e investimentos, os ataques cibernéticos vêm causando prejuízos bilionários e continuam crescendo [18]. Uma das razões desse insucesso diz respeito ao *modus operandi* retrógado dos antivírus comerciais, que somente reagem a vírus e *malwares* já existentes, precisando de alterações de suas bases quando novas ameaças surgem, alterações estas que têm que ser feitas de forma centralizada e levam tempo, durante o qual o malware pode realizar seu estrago [13].

As limitações dos mecanismos de vigilância digital podem ser supridas por técnicas de inteligência artificial baseadas em aprendizado de máquina. Inteligência artificial consegue automatizar muitas tarefas, analisando milhares de dados, extraindo características deles e os classificando. Logo, a

inteligência artificial é capaz de reconhecer padrões de comportamento previamente classificados como suspeitos em tempo real.

O estado da arte propõe extrair características do aplicativo suspeito, de maneira preventiva, antes de executá-lo [13]. O executável passa por um processo de *disassembling*. Então, as características extraídas do aplicativo servem como atributos de entrada das redes neurais artificiais empregadas como classificadores cujo objetivo é agrupar os executáveis em duas classes: benignos e *malwares* [13].

Redes neurais são modelos de inteligência computacional frequentemente utilizados para resolver problemas de reconhecimento de padrões tendo como principal característica o poder de generalização diante de dados não apresentados à rede. Em grande parte das redes neurais, como a MLP (*Multilayer Perceptron* - Perceptron com Múltiplas Camadas) [25], é necessário um conhecimento sobre os parâmetros da rede para obter máximo desempenho na solução do problema. Uma preocupação comum nesse tipo de rede é evitar se ater a mínimos locais [10], sendo necessário adicionar métodos de controle que permitam à rede desprender-se dessas regiões. Outra característica comum nesse tipo de rede é a grande quantidade de tempo de treinamento necessária para torná-la apta a realizar classificações corretamente. Apesar de excelentes acurácias, as redes neurais de última geração, especificamente *Deep Learning* (Aprendizado Profundo), podem requerer uma duração excessiva até a conclusão do seu treinamento.

Tecnicamente, em termos de inteligência artificial, quando uma nova vulnerabilidade for detectada, deve haver uma nova etapa de aprendizado (treinamento) das redes neurais artificiais empregadas pelos mecanismos de segurança digital. As novas características referentes à vulnerabilidade recém-detectada devem ser agrupadas às características convencionais, previamente estabelecidas. Dessa forma, é possível proteger os demais computadores, ainda não infectados, dos *malwares* que explorem essa falha recém-encontrada. Então, quanto mais rápido for o tempo de treinamento do modelo de inteligência computacional maiores são as chances de prevenção da infecção dos computadores pessoais e das organizações. Caso o tempo de treinamento do antivírus seja elevado, a exploração da vulnerabilidade recém-descoberta pode gerar malefícios irreversíveis e irrecuperáveis para toda a rede mundial de computadores.

O trabalho proposto aplica as redes neurais ELMs (*Extreme Learning Machine* – Máquinas de Aprendizado Extremo) na área de segurança da informação, especificamente no reconhecimento de padrão de *malwares*. As redes ELMs têm como principal característica a velocidade de treinamento e a previsão de dados com qualidade semelhante àquela das redes neurais baseadas em retropropagação de dados e *Deep Learning*. As ELMs são adequadas à Perícia Forense Digital visto que são lançados 8 (oito) novos *malwares* por segundo [11], o que inviabiliza o uso de técnicas retrógradas, descritas anteriormente nesta seção.

A rede ELM é uma rede de camada oculta única, não recorrente, com base em um método analítico para estimar os pesos de saída da rede, na qual pode ser utilizada qualquer inicialização aleatória dos pesos de entrada. As ELMs têm sido largamente aplicadas nas mais diversas áreas como na Engenharia Biomédica [3][4][5][14][15] e podem contribuir bastante para o avanço da segurança em dispositivos.

As ELMs são máquinas de aprendizado poderosas e flexíveis baseadas em *kernel* [10]. Ao invés de *kernels* convencionais, este trabalho cria *kernels* autorais para as ELMs. Os *kernels* são funções matemáticas utilizadas como método de aprendizado das redes neurais. O aprendizado baseado em *kernel* oferece a possibilidade da criação de um mapeamento não-linear de dados sem que haja a necessidade do aumento do número de parâmetros ajustáveis como, por exemplo, taxa de aprendizagem comumente empregada em redes neurais com retropropagação.

No entanto, os *kernels* podem apresentar limitações. Um *kernel* linear, por exemplo, não é capaz de resolver um problema não linearmente separável, como uma distribuição senoidal, enquanto um *kernel* senoidal pode ser capaz de resolver um problema, desde que ele seja separável por uma função senoidal. Logo, um dos grandes desafios, em redes neurais artificiais diz respeito a encontrar um *kernel* que otimize a fronteira de decisão entre as classes de uma dada aplicação.

No presente artigo são criadas 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 de 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. Logo, as operações morfológicas lidam naturalmente com a detecção das formas dos corpos presentes nas imagens [23] ao se interpretar a fronteira de decisão de uma rede neural como uma imagem *n*-dimensional, onde *n* diz respeito à quantidade de características extraídas. Assim, pode-se afirmar que as mELMs são capazes de naturalmente detectar e modelar as regiões *n*-dimensionais mapeadas às distintas classes.

No tocante aos experimentos, os nossos resultados são comparados com ELMs clássicos e avaliados por meio de métricas de classificação amplamente usadas na literatura da área. Nossos resultados também são comparados com antivírus do estado-da-arte e com *Deep Learning* (Aprendizado Profundo) de última geração. Nosso antivírus pode combinar alta taxa de acurácia com reduzido tempo de aprendizado. O antivírus autoral alcança um desempenho médio de 99,80% na distinção entre executáveis benignos e *malwares* acompanhado de um tempo de treinamento médio de 9,32 segundos.

Este trabalho está organizado da seguinte forma: na seção 2, apresentamos o estado-da-arte quanto aos antivírus dotados de inteligência artificial; na seção 3, introduzimos as características das redes neurais extremas; na seção 4, explicamos a morfologia matemática empregada em nossa rede neural extrema autoral; na seção 5, apresentamos a

metodologia proposta; na seção 6, fazemos um comparativo entre a rede ELM autoral e as redes ELMs clássicas; na seção 7, mostramos os resultados e algumas discussões. Por fim, na seção 8, fazemos as conclusões gerais e discutimos as perspectivas do nosso trabalho.

II. Estado-da-Arte

Apesar de ser questionado há mais de uma década, o *modus operandi* dos antivírus é baseado em assinaturas relativas ao arquivo suspeito quando consultado em bases de dados denominadas "lista negra" [20]. Em síntese, o executável suspeito é comparado a uma lista negra confeccionada a partir de denúncias prévias. Logo, caso a base de dados do antivírus não esteja atualizada o malware não é identificado e a infecção acontece.

Visando demonstrar a ineficiência dos antivírus comerciais, LIMA*, et al.* (2019) investigou os 86 principais antivírus comerciais mundiais. Em média, os principais antivírus mundiais foram capazes de detectar 54,84%, dos *malwares*. No experimento, quase a quinta parte dos antivírus não tinha qualquer conhecimento sobre os *malwares*, de domínio público, os quais já infectaram milhões de vítimas. Conclui-se que os antivírus comerciais apresentam severas limitações quanto à detecção de *malwares* em larga escala e em tempo real [13].

Admite-se que ao invés de buscar assinaturas em listas negras, deve-se procurar impedir atuações de *cyber*-ataques recém-criados e não apenas de *malwares* já conhecidos [20]. Atualmente, as organizações buscam suprir as deficiências dos antivírus tradicionais através de ciência dos dados, máquinas de aprendizado estatístico e inteligência artificial [20]. Os trabalhos, listados na Tabela 1, empregam o processo de *disassembling* visando reverter o arquivo PE [1] binário em seu código de montagem. Então, as características extraídas do aplicativo servem como atributos de entrada para as redes neurais artificiais empregadas como classificadores. O objetivo é agrupar os executáveis em duas classes: benignos e *malwares*.

BAI, *et al* (2014) emprega 8592 executáveis benignos nativos do Windows, além de softwares de origem não relatada pelo trabalho. O trabalho também utiliza outros 10521 executáveis *malwares* extraídos dos VX *Heaven* [2]. Na etapa de classificação, o artigo emprega quatro classificadores; Árvore de Decisão, *Random Forest*, *Bagging* e *Adaboost*. Não há variação dos parâmetros de configuração dos classificadores, e a suposição razoável é que os parâmetros usados estejam relacionados às configurações padrão do Weka [3]. A obra cria três cenários de experimentos visando a validação dos classificadores empregados. No melhor cenário, os autores alcançam um desempenho médio de 99,10% através do classificador *Random Forest* [6].

KUMAR, *et al* (2017) emprega 2488 arquivos executáveis portáveis benignos nativos do Windows XP e Windows 7. A obra também emprega outros 2722 executáveis portáveis *malwares* oriundos do VirusShare [4]. Na etapa de classificação, a validação cruzada é empregada através do método *k-fold,* onde k=10. O artigo emprega seis classificadores; KNN [5], Regressão logística, Análise Discriminante Linear, *Random Forest*, Árvore de Decisão e Gaussiano. No melhor cenário, a obra de KUMAR, *et al* (2017) alcança um desempenho médio de 98,40% através do classificador Árvore de Decisão [12].

LIMA, *et al* (2019) cria o repositório de dados REWEMA contendo 3136 executáveis malignos que atacam aplicativos de arquiteturas de 32 *bits*. As análises dos malwares são emitidas pelos principais antivírus comerciais e extraídas pela plataforma VirusTotal [19]. A REWEMA disponibiliza outros 3136 executáveis benignos os quais, em conjunto com as amostragens malignas, são empregados no aprendizado das redes neurais artificiais. Quanto à etapa de classificação, LIMA, *et al* (2019) emprega redes neurais artificiais do tipo MLP [6] dotadas de retropropagação visando o reconhecimento de *malwares*. Com o objetivo de otimizar a precisão do seu antivírus, LIMA, *et al* (2019) averigua onze diferentes funções de aprendizado f. Para cada função de aprendizado, LIMA, *et al* (2019) explora 90 (3 arquiteturas * 30 pesos iniciais) modelos.

ALAZAB, M. (2015) cataloga 15480 executáveis benignos de origem não divulgada. O trabalho também utiliza outros 51223 executáveis *malwares* extraídos dos VX Heaven. O trabalho relata a presença de executáveis *malwares* empacotados contidos em suas bases de dados. Observa-se que a base de dados, empregada por ALAZAB, M. (2015), está desbalanceada. Na etapa de classificação, a validação cruzada é empregada através do método *k-fold,* onde k=10. O artigo emprega um único classificador KNN visando a separação entre as classes: benigno e *malware*. Quanto aos resultados, não há o relato do desempenho médio da metodologia, a obra utiliza a porcentagem de verdadeiros positivos e falsos positivos como métrica. A obra de ALAZAB, M. (2015) alcança uma acurácia de 94,80% e 5,01% para verdadeiros positivos e falsos positivos, respectivamente [1].

[1] PE (*Portable Executable* – Portáveis Executáveis): arquivos com as extensões; .exe, .sys e .dll e . referentes a processos, bibliotecas e serviços, respectivamente.

[2] VX Heaven: base de dados descontinuada de *malwares* do tipo PE (*Portable Executable* – Executáveis Portáveis).

[3] Weka: Software com disponibilidade de redes neurais artificiais através de interface gráfica. Disponível em: https://www.cs.waikato.ac.nz/ml/weka/. Acesso em março de 2020.

[4] VirusShare: base de dados de *malwares*. Disponível em: https://virusshare.com/. Acesso em março de 2020.

[5] KNN: *K Nearest Neighbors* – K Vizinhos mais Próximos.

Tabela 1. Sumário dos principais antivírus visando a classificação dos executáveis entre benigno e *malware*.

	Possibilidade de replicar	Quantidade de benignos	Quantidade de malwares	Balanceamento da base de dados	Classificadores avaliados	Exploração dos Parâmetros dos Classificadores	Melhor Resultado (%)
Antivírus proposto	Sim	3136	3136	Sim	11	Sim	99,80
BAI, *et al* (2014)	Não	8592	10521	Não	4	Não	99,10
KUMAR, *et al* (2017)	Não	2488	2722	Não	6	Não	98,40
LIMA, *et al* (2019)	Sim	3136	3136	Sim	11	Sim	98,13
ALAZAB, M. (2015)	Não	15480	51223	Não	1	Não	94,80
SANTOS, *et al* (2013)	Não	1000	1000	Sim	11	Sim	94,73
DING, *et al* (2014)	Não	650	650	Sim	3	Não	94,00

SANTOS, *et al* (2013) cataloga 1000 executáveis benignos atestados pelos antivírus comercial *Eset*. O trabalho também utiliza outros 1000 executáveis *malwares* extraídos dos VX Heaven. Observa-se que a base de dados, empregada por SANTOS, *et al.* (2013), está igualitariamente distribuída entre casos benignos e malware. Portanto, a base possui a excelente característica de ser balanceada. Na etapa de classificação, o artigo emprega onze classificadores: KNN, Árvore de Decisão, *Random Forest*, *Naïve Bayes*, além das redes Bayesianas dotadas de três *kernels* (K2, Hill Climber e TAN). Também há o emprego do classificador SVM [7] dotados de quatro *kernels*: RBF, Polinomial, Polinomial normalizado e Pearson VII. A obra de SANTOS, *et al* (2013) explora distintos parâmetros das funções de aprendizado dos classificadores. O objetivo é obter taxas de acertos superiores em comparação às configurações padrões. No melhor cenário, a obra SANTOS, *et al.* (2013) alcança um desempenho médio de 94,73% através do classificador KNN configurado com um único vizinho mais próximo a cada predição [21].

DING, *et al* (2014) cataloga 650 executáveis benignos nativos do Windows XP. O trabalho também utiliza outros 650 executáveis *malwares* extraídos do *Netlux* e *Offensive Computing*. Na etapa de treinamento, a validação cruzada é empregada através do método *k-fold,* onde *k*=5. O artigo emprega três classificadores; Árvore de Decisão, KNN e SVM. Não há variação dos parâmetros de configuração dos classificadores, logo a suposição é que os parâmetros usados estão relacionados às configurações padrão do Weka. No melhor cenário, a obra de DING, *et al* (2014) alcança um desempenho médio de 94,00% através do classificador DT [7].

Quanto à extração de características dos executáveis, todos os trabalhos listados na Tabela 1 empregam o processo de *disassembling*, visando reverter os executáveis em seus respectivos códigos de montagem. As ferramentas, responsáveis pelo *disassembling*, empregadas pelo estado-da-arte são PEScanner, IDA Pro, Ollydbg e WinDbg. A partir dos códigos de montagem são extraídos os repertórios de instruções e suas APIs (*Application Programming Interface* - Interface de Programação de Aplicações) referentes aos executáveis investigados. Então, as características pertencentes aos aplicativos servem como atributos de entrada das redes neurais artificiais de modo a haver a separação dos arquivos em duas classes; benigno e *malware*. O Weka costuma ser a ferramenta empregada na etapa de classificação na maior parte dos trabalhos analisados.

Com exceção da obra de LIMA, *et al* (2019), os trabalhos do estado da arte, listados na Tabela 1, apenas informam as fontes de obtenção dos executáveis. Não há, no entanto, a descrição de quais executáveis são empregados nos experimentos. Logo, torna-se inviável a repetição dos experimentos das obras de KUMAR, *et al* (2017), ALAZAB, M. (2015), DING, *et al* (2014), BAI, *et al* (2014) e SANTOS, *et al* (2013). Isso posto, o trabalho proposto emprega a REWEMA visto que tal base de dados se encontra livremente disponível. Logo, nossa metodologia pode ser replicada por terceiros em trabalhos futuros, além de demostrar a veracidade dos resultados alcançados.

Acrescido a isso, os trabalhos do estado da arte listados na Tabela 1 geralmente não têm a preocupação metodológica em balancear as bases de dados. Então, bases de dados desbalanceadas podem proporcionar uma tendência de acertos maiores nas classes majoritárias e uma elevada taxa erros nas classes minoritárias [2]. A REWEMA, criação da obra de LIMA, *et al* (2019), possui quantidades balanceadas pertencentes à classe benigna e maligna, cada uma com 3136 exemplares. O objetivo é que classificadores tendenciosos em relação a uma determinada classe não tenham suas taxas de acerto favorecidas.

Com exceção da obra de LIMA, *et al* (2019), os artigos listados na Tabela 1 não relatam se houve o cuidado metodológico de reservar quantidades relevantes de exemplares benignos e *malwares* nas amostras separadas para o treinamento e teste. Então, supondo uma amostra reservada para teste com pouca ou nenhuma instância da classe *malware*, a classificação seria tendenciosa à classe benigna. Na prática forense digital, no entanto, a errônea absolvição de um malware poderia acarretar prejuízos irreversíveis. Logo, o trabalho proposto apresenta o cuidado metodológico de selecionar equitativamente, de forma randômica, exemplares benignos e *malwares* para as amostras destinadas ao treinamento e teste.

O antivírus proposto é comparado com o antivírus de LIMA, *et al* (2019). Os demais antivírus do estado-da-arte não foram replicados devido à impossibilidade de suas reprodutibilidades, pois suas bases de dados não estão

[7] *Support Vector Machine* - Máquina de vetores de suporte.

disponíveis. Além disso, não costuma haver a descrição dos parâmetros de configuração empregados durante a classificação. A suposição é que os parâmetros usados no estado-da-arte estejam relacionados às configurações padrões do Weka. Admite-se que não há como comparar obras científico-acadêmicas baseado em suposições. Em acréscimo, a comparação com classificadores padrões do Weka não iria agregar qualquer relevância aos classificadores autorais propostos.

Um dos objetivos específicos diz respeito a demonstrar a eficiência do nosso antivírus, dotado de Aprendizado Extremo, em comparação ao Aprendizado Profundo (*Deep Learning*). Além do intervalo de duração de treinamento, em ordem de segundos, o nosso antivírus é capaz de apresentar acurácia estatisticamente superior a *Deep Learning* de última geração. O antivírus proposto tem a sua acurácia comparada com a *Deep Learning* de SANTOS, *et al.* 2019. Tal obra do estado-da-arte não visa a detecção de *malwares* e, sim, o reconhecimento óptico de caracteres. Logo, a técnica de SANTOS, *et al.* 2019 sofre uma adaptação em sua camada de neurônios de entrada. Ao invés de processamento digital de imagem, os atributos de entrada dizem respeito à extração de características dos *malwares*.

Nos últimos anos, redes neurais convolucionais profundas têm sido cada vez mais usadas em várias tarefas como; visão computacional, reconhecimento de objetos, e aplicativos de diagnósticos através sinais ou imagens biomédicas. Uma desvantagem das redes profundas é o longo tempo de treinamento. Podem custar dias para ajustar os pesos com métodos iterativos baseados gradientes descendentes [22]. Como agravante, as redes profundas apresentam baixa capacidade de paralelismo porque as camadas convolucionais são sequenciais. Logo, uma camada só pode ser executada após a camada imediatamente anterior ter concluído o seu trabalho. Isso pode ser um obstáculo em aplicativos que precisam de treinamento frequente como os antivírus .

A *Deep Learning* de SANTOS, *et al.* 2019 apresenta uma única camada convolucional e não há retropropagação de dados. Portanto, a referida técnica tem grande capacidade de paralelismo desde que o supercomputador tenha recurso computacional (memória) suficiente. A camada convolucional emprega 30 mil filtros simultaneamente de modo que o tempo de treinamento é compatível com aplicativos que precisam de treinamento frequentemente.

Ao invés de filtros convolucionais aleatórios, a *Deep Learning* de SANTOS, *et al.* 2019 desenvolve filtros PCA visando a extração de componentes principais das regiões de interesse referentes ao vetor de entrada de dados. Cada filtro é convertido em uma matriz de Toeplitz denotada por $W_{detect} \in \mathbb{R}^{L^2xJ^2}$, com $L = (J + W - 1)$ para um mapa de características de tamanho LxL, onde $We\ J$ se referem ao tamanho dos filtros e o número de pixels, respectivamente [22].

A adaptação de uma técnica de Aprendizado Profundo em função do reconhecimento de padrão de *malwares* tem como objetivo desmistificar um senso comum equivocado. Devido aos excelentes resultados obtidos por técnicas de *Deep Learning* foi criado um senso comum de que o Aprendizado Profundo é capaz de prover as melhores acurácias em qualquer tipo de aplicação. Entretanto, deve-se levar em consideração que as redes neurais profundas, especificamente, as redes convolucionais são baseados na convolução linear de filtros. Embora exerça um papel fundamental em aplicações computacionais, a convolução de filtros está limitada a aplicações onde haja a formação de gradiente de fluxo de vetores, o que não ocorre em todas as aplicações existentes.

Considere, por exemplo, as imagens biomédicas oriundas de mamógrafos. As imagens são repletas de ruídos que atrapalham o reconhecimento da lesão mamária [15]. Logo, a convolução de filtros é fundamental no sentido de eliminar os ruídos e, portanto, descartar pequenas irregularidades no achado correspondente ao câncer em potencial. Técnicas convolucionais como, por exemplo, filtros gaussianos são essenciais na redução de ruídos em imagens biomédicas [8].

Como contra-exemplo, considere o repositório ilustrado na Tabela 2. As características são completamente desconexas entre si apesar de em uma codificação binária, pertencerem a uma mesma região de vizinhança. Um aplicativo suspeito de tentar capturar informações de rede não tem qualquer correlação com o fato dele acessar a galeria de imagens ou o e-mail da vítima. Isso posto, ao se aplicar a convolução linear de filtros no repositório, ilustrado na Tabela 2, o acesso à galeria, contendo o valor 0, seria tratado como ruído. A explicação é que sua região de vizinhança possui valores positivos. Em síntese, o aplicativo suspeito seria acusado de acessar a galeria de imagens da vítima mesmo que a extração de características tenha auditado o inverso.

Dadas as limitações do Aprendizado Profundo, a formação de gradiente de fluxo de vetores, como nas imagens biomédicas, é essencial para o sucesso de tais técnicas. De forma oposta, repositórios de aprendizado estatístico, formado por atributos não-correlatos, são inadequados a redes neurais profundas, especificamente, redes convolucionais. Aprendizado Profundo sofre desvantagem quando aplicado ao reconhecimento de padrão de *malwares*. Em síntese, não há rede convolucional concebida para a detecção de malware – o que se usa são adaptações alternativas.

Tabela 2. Ilustrativo de repositório de aprendizado estatístico visando o reconhecimento de *malwares*.

Atributos		
capturar informações de rede	acessar a galeria de imagens	acessar o e-mail da vítima
1	0	1

III. REDES NEURAIS ELM

Redes neurais são modelos de inteligência computacional frequentemente utilizados para resolver problemas de classificação, tendo como principal característica o poder de generalização diante de dados não apresentados previamente à rede.

Em grande parte das redes neurais, como a MLP (*Multilayer Perceptron* - Perceptron com Múltiplas Camadas) [25], é necessário um conhecimento sobre os parâmetros da rede para obter máximo desempenho na solução do problema. Uma preocupação comum nesse tipo de rede é evitar se ater a mínimos locais [9], sendo necessário adicionar métodos de controle da rede para sair dessas regiões. Outra característica comum nesse tipo de rede é a grande quantidade de tempo de treinamento necessário para tornar a rede apta a realizar classificações corretamente.

A rede ELM (*Extreme Learning Machine* – Máquina de Aprendizado Extremo) tem como principal característica a velocidade de treinamento e predição de dados comparada a outros classificadores. A rede ELM é uma rede de camada escondida única, não interativa. O processo de aprendizagem da rede ELM é baseado na inversa generalizada de Moore-Penrose (pseudo-inversa), onde são calculados os pesos entre a camada escondida e a camada de saída [10].

A aprendizagem da rede ELM é realizada em lote, onde todos os dados são apresentados à rede antes do ajuste dos pesos referentes às ligações sinápticas entre os neurônios da camada escondida e de saída. A aprendizagem da ELM contém uma única iteração. Em síntese, não há retropropagação de dados, consequentemente a ELM não apresenta problemas com *overfitting* (treinamento em excesso).

Por conter uma única iteração, o treinamento da ELM é mais rápido do que as abordagens convencionais. Além disso, não é necessário determinar o máximo número de iterações, uma vez que o algoritmo não é iterativo. Em acréscimo, por não se basear no método de gradiente descendente, a rede não sofre o problema de mínimo local nem é necessária a definição de um parâmetro de taxa de aprendizagem.

Matematicamente, na rede ELM os atributos de entrada x_{ik} correspondem ao conjunto $\{x_{it} \in R;\ i \in N^*,\ i = 1, \dots, n;\ t \in N^*,\ t = 1, .. v\}$. Logo, há n características extraídas da aplicação e v vetores de dados de treinamento. A camada escondida h_j, constituída por m neurônios, é representada pelo conjunto $\{h_j \in R;\ j \in N^*,\ j = 1, \dots, m\}$.

O processo de treinamento da ELM é rápido por ser composto por poucas etapas. Inicialmente, os pesos de entrada w_{ji} e $biasb_{jt}$ são definidos de maneira aleatória. Dada uma função de ativação $f: \mathbb{R} \to \mathbb{R}$, o processo de aprendizagem é dividido em três passos:

1. Atribuição aleatória de pesos w_{ji}, correspondente aos pesos entre a camada de entrada e a camada escondida, e *bias* b_{jk}.
2. Calcular a matriz H, que corresponde à saída dos neurônios da camada escondida.
3. Calcular a 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 s.

Assim como na MLP, a saída dos neurônios da camada escondida, correspondente à matriz H, é calculada através do *kernel* K, entradas e pesos da camada escondida, conforme mostra a Equação (1).

$$H_{jt} = \begin{bmatrix} K(11) & \cdots & K(1N) \\ \vdots & \ddots & \vdots \\ K(V1) & \cdots & K(VN) \end{bmatrix} \quad (1)$$

Diferente das redes com retropropagação, na rede ELM não é necessário definir critério de parada para treinamento nem criar mecanismos para que a rede não perca a capacidade de generalização. O motivo é que a rede ELM apresenta uma única iteração. Desse modo, não é necessária a separação de conjunto de dados em treinamento, validação e teste. Basta a divisão em conjuntos de treinamento e teste, permitindo um maior número de amostras para esses dois conjuntos em comparação às redes neurais baseadas em retropropagação.

Uma vez treinada a rede, os padrões de teste são apresentados juntamente com a saída desejada. A rede não sofrerá mais ajustes e apenas calculará o resultado obtido para cada conjunto de teste apresentado. Ao comparar os dados esperados com os obtidos é avaliado o grau de precisão da rede ELM.

IV. MELMS (ELMS MORFOLÓGICAS)

O antivírus criado é apresentado a redes neurais ELM visando o reconhecimento de padrão de *malwares*. Ao invés de *kernels* convencionais, o trabalho cria *kernels* autorais para as ELMs.

No presente artigo, são criadas as mELMs (ELMs morfológicas), que consistem em 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 à qualquer fronteira de decisão.

Morfologia Matemática é uma teoria completa de processamento não-linear amplamente utilizado 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 [23].

A morfologia é baseada nas transformações de formas preservando as relações de inclusão dos objetos. Há duas operações morfológicas fundamentais: Erosão e Dilatação [23]. A Morfologia Matemática pode ser considerada uma teoria construtiva porque todas as operações são construídas tendo como base Erosões e Dilatações. Matematicamente, Erosão e Dilatação são formalizadas de acordo com a Equação (2) e a Equação (3), respectivamente:

$$\varepsilon_g(f)(u) = \bigcap_{v \in S} f(v) \vee \bar{g}(u - v) \quad (2)$$

$$\delta_g(f)(u) = \bigcup_{v \in S} f(v) \wedge g(u - v), \quad (3)$$

onde $f: S \to [0,1]$ e $g: S \to [0,1]$ são imagens normalizadas em forma de matriz nomeada de formato S, onde $S \in \mathbb{N}^2$. f diz respeito à imagem original.

A Morfologia Matemática tem o processo básico de deslocamento do elemento estruturante g sobre a imagem original. O elemento estruturante g é uma pequena matriz bidimensional cujos valores de seus coeficientes determinam o objetivo a ser alcançado durante o processamento [8]. As operações morfológicas de Erosão e Dilatação visam maximizar e minimizar a região do(s) objetos(s) sobreposto(s) pela matriz bidimensional g, respectivamente [23].

O pixel é definido através do par cartesiano $(u, f(u))$, onde u é a posição associada ao valor $f(u)$. v é a matriz de *f(u),* abrangida por g. Os operadores $\cup$ e $\vee$ estão associados à operação de máximo, enquanto $\cap$ e $\bigwedge$ estão associados à operação de mínimo. g é o elemento estruturante tanto para Erosão quanto para Dilatação [23]. $\bar{g}$ é a negação de g.

Na Equação (2) inicialmente ocorre a negação do elemento estruturante $\bar{g}$. Logo, acontece a operação de máximo $\vee$ denotada por $f(v) \vee \bar{g}(u - v)$, onde $f(v)$ diz respeito à matriz da imagem original f abrangida (casada) por $\bar{g}$. $f(v)$ é nomeada tecnicamente de região ativa da imagem. Por fim, o valor $\varepsilon_g(f)(u)$, na posição u, da imagem erodida [8] recebe o mínimo valor entre os máximos, através do operador $\cap$. $\varepsilon_g(f)(u)$ obtém o valor 0 associado ao preto absoluto. A Erosão sobrepõe $\bar{g}$ à imagem original f. O objetivo é que as áreas similares ao $\bar{g}$ se expandam. Ao se associar o 1's ao branco absoluto e 0's ao preto absoluto, a Erosão aumenta as áreas mais escuras e elimina as regiões com maior intensidade [23].

A Equação (3) exibe a atuação da operação morfológica de Dilatação. Por precedência matemática, ocorre a operação de mínimo $\bigwedge$ denotada por $f(v) \wedge g(u - v)$, onde $f(v)$ diz respeito à matriz da imagem original f abrangida (casada) por g. Logo, o valor $\delta_g(f)(u)$, na posição u, da imagem dilatada recebe o máximo valor entre os mínimos, através do operador $\cup$. A Dilatação sobrepõe o elemento estruturante g à imagem original f. O objetivo é que as áreas similares ao g se expandam. Ao se associar o 1's ao branco absoluto e 0's ao preto absoluto, a dilatação aumenta as áreas com tonalidade mais intensas e elimina as regiões escuras [23].

O trabalho proposto cria as mELMs (*Morphological Extreme Learning Machines*). Como estudo de caso, as mELMs são aplicadas na distinção entre executáveis benignos e *malwares*. As mELMs são inspiradas na Morfologia Matemática tendo como base os operadores não-lineares de Erosão e Dilatação. Dada a Equação (2) referente ao operador de imagem erosão, o *kernel* ELM de Erosão pode ser definido de acordo com a Equação (4), onde $\{\mathrm{i} \in N^*,\ \mathrm{i} = 1, \ldots, n;\ j \in N^*,\ j = 1, \ldots, m,;\ t \in N^*,\ t = 1, .. v\}$. Logo, há n neurônios na camada de entrada (sem o *bias*), m neurônios na camada escondida e v vetores de dados de treinamento.

$$K_\varepsilon(t, i) = \bigcap_{i=1}^{n} (x_{it} \vee \bar{w}_{ji}) + b_{jt} \quad (4)$$

A Figura 1 exibe a atuação do *kernel* morfológico de Erosão, descrito na Equação (4). Ao traçar uma analogia entre a operação de processamento de imagem e o *kernel* mELM, o branco absoluto está associado aos 1's, enquanto o preto absoluto ao 0's. A região ativa da imagem $f(v)$ corresponde aos atributos de entrada $x_{1t,}\, x_{2t},\, \ldots, x_{nt}$. O elemento estruturante negado $\bar{g}$ diz respeito aos pesos $\bar{w}_{ji}$ das ligações sinápticas entre as camadas de entrada e escondida.

No exemplo ilustrado na Figura 1 por precedência matemática, inicialmente ocorre a negação dos pesos referentes às ligações sinápticas entre a camada de entrada e escondida $\bar{w}_{ji}$. Suponha o conjunto de pesos $w_{1i} = \{0, 1, 0, 1\}$, então $\bar{w}_{ji} = \{1, 0, 1, 0\}$ de acordo com a lógica difusa ou lógica *fuzzy*: $\bar{A} = 1 - A$. Em sequência, acontecem as operações de máximo $\vee$ denotada por $x_{it} \vee \bar{w}_{ji}$. Logo, as resultantes das operações máximas são somadas ao *bias* b_{jt}. Por fim, $H_{1t} = K_\varepsilon(ti)$ recebe o mínimo valor entre os máximos, através do operador $\cap$. A matriz H foi detalhada na Equação (1).

$$K_\delta(t, i) = \bigcup_{i=1}^{n} (x_{it} \wedge w_{ji}) + b_{jt} \quad (5)$$

Similarmente ao *kernel* Erosão, a Equação (6) define o *kernel* Dilatação inspirado na Equação (3) e referente ao operador morfológico de Dilatação.

$$K_\delta(t, i) = \bigcup_{i=1}^{n} (x_{it} \wedge w_{ji}) + b_{jt} \quad (6)$$

A Figura 2 exibe a atuação do *kernel* morfológico Dilatação, descrito na Equação (6). Suponha o conjunto de pesos $w_{1i} = \{0, 1, 0, 1\}$. Por precedência matemática, acontecem as operações de mínimo $\wedge$ denotadas por $x_{it} \wedge w_{ji}$. Logo, as resultantes das operações mínimas são somadas ao *bias* b_{jt}. Por fim, $H_{1t} = K_\delta(ti)$ recebe o máximo valor entre os mínimos, através do operador $\cup$. A matriz H foi detalhada na equação (1).

Nos exemplos mostrados na Figura 1 e na Figura 2, os pesos e atributos entradas são binários. Em aplicações reais, no entanto, os pesos são fracionários. Então, em nossas mELMs, as operações de *máximo* e *mínimo* são implementadas através de desvios condicionais (*if else*) ao invés de operações lógicas AND e OR.

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 Figura 3 (a). Seguindo o mesmo raciocínio, *kernels*

[8] Imagem erodida: imagem digital após sofrer a operação morfológica de erosão [23].

sigmoide, RBF (*Radial Basis Function* – Função de Base Radial) e senoide são capazes de resolver problemas separáveis por função sigmoidal, radial e senoidal, vistos na Figura 3 (b), na Figura 3 (c) e na Figura 3 (d), respectivamente.

Então, 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*, no entanto, pode ser necessária, caso contrário a rede neural composta, por um *kernel* desajustado, por gerar resultados não satisfatórios.

Como contra-exemplo, observe o emprego do *kernel* Linear aplicado a distribuições sigmoide e senoide apresentados na Figura 4 (a) e na Figura 4 (b), respectivamente. As precisões das classificações expostas na Figura 4 (a) e na Figura 4 (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 Figura 5 (a), a Figura 5 (b), a Figura 5 (c) e a Figura 5 (d) exibem a atuação do mELM kernel de erosão nas distribuições linear, sigmoide, radial e senoide, com as respectivas precisões de 100%, 93,07%, 98,18% e 99,50%. A Figura 6 (a), a Figura 6 (b), a Figura 6 (c) e a Figura 6 (d) exibem a atuação do mELM *kernel* dilatação nas distribuições linear, sigmoidal, radial e senoidal, com as respectivas precisões 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 nossas mELMs são capazes de naturalmente detectar e modelar as regiões *n*-dimensionais referentes às distintas classes por empregar a Morfologia Matemática, que lida naturalmente com a detecção das formas dos corpos presentes nas imagens [23].

V. Metodologia Proposta

Dada as limitações dos antivírus comerciais, o trabalho proposto visa criar um antivírus, dotado de inteligência artificial, capaz de diferenciar aplicativos *malwares* de benignos de forma preventiva. A Figura 7 exibe o diagrama da metodologia proposta em diagrama de blocos.

Inicialmente, são extraídas as características do conjunto de executáveis oriundos da base [19]. Logo, as redes neurais artificiais utilizam, como atributos de entrada, essas características extraídas das executáveis os quais são classificados entre benignos e *malwares*. Os resultados da classificação estão descritos nas seções VI e VII. Quanto aos materiais, todos os experimentos foram realizados em um supercomputador em nuvem dotado de 250 GB de memória RAM, 8 processadores e 300 GB de armazenamento em massa.

De modo a não haver comparações injustas, o antivírus de LIMA, *et al* (2019) e a *Deep Learning* de SANTOS, *et al,* 2019 são treinados e testados no mesmo supercomputador empregado pelo antivírus autoral. Enfatiza-se que a aquisição de um supercomputador foi devido à réplica e à comparação com os trabalhos do estado-da-arte. O antivírus autoral exige baixa capacidade de processamento e de armazenamento. Enfatiza-se que o antivírus autoral poderia ser usado em qualquer computador de mesa convencional.

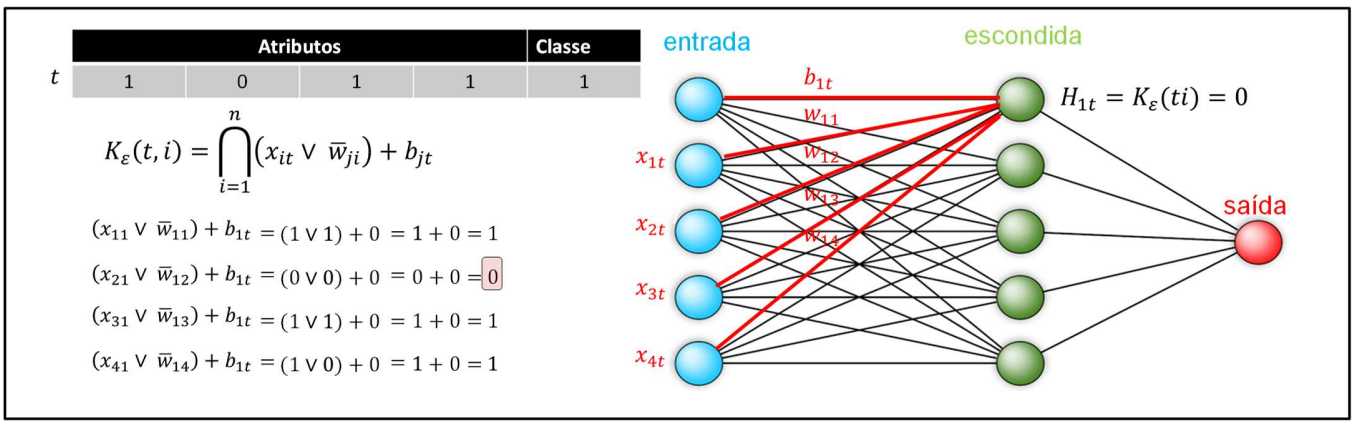

Figura 1. Atuação ilustrada do *kernel* autoral de Erosão.

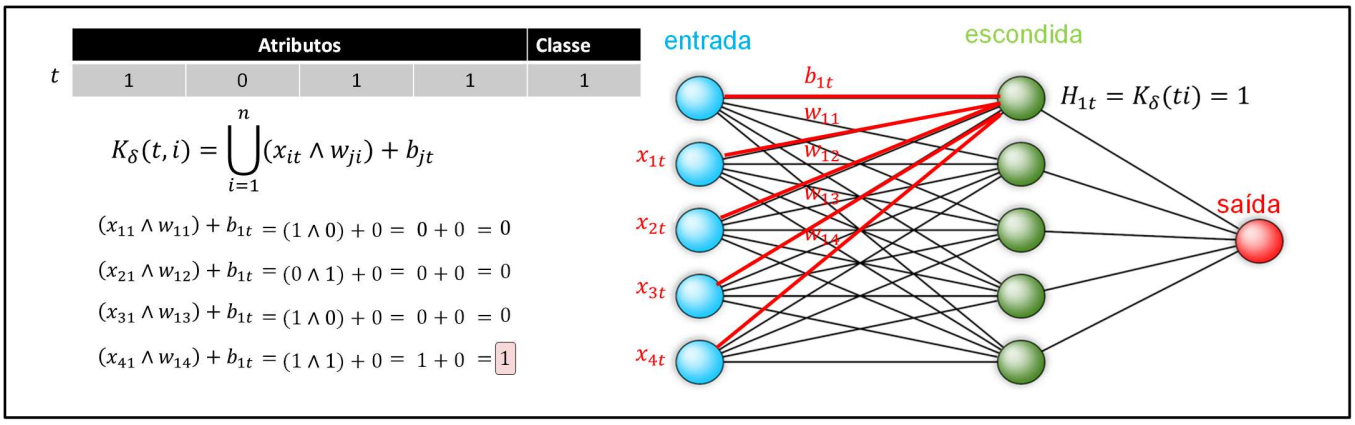

Figura 2 Atuação ilustrada do kernel autoral de Dilatação.

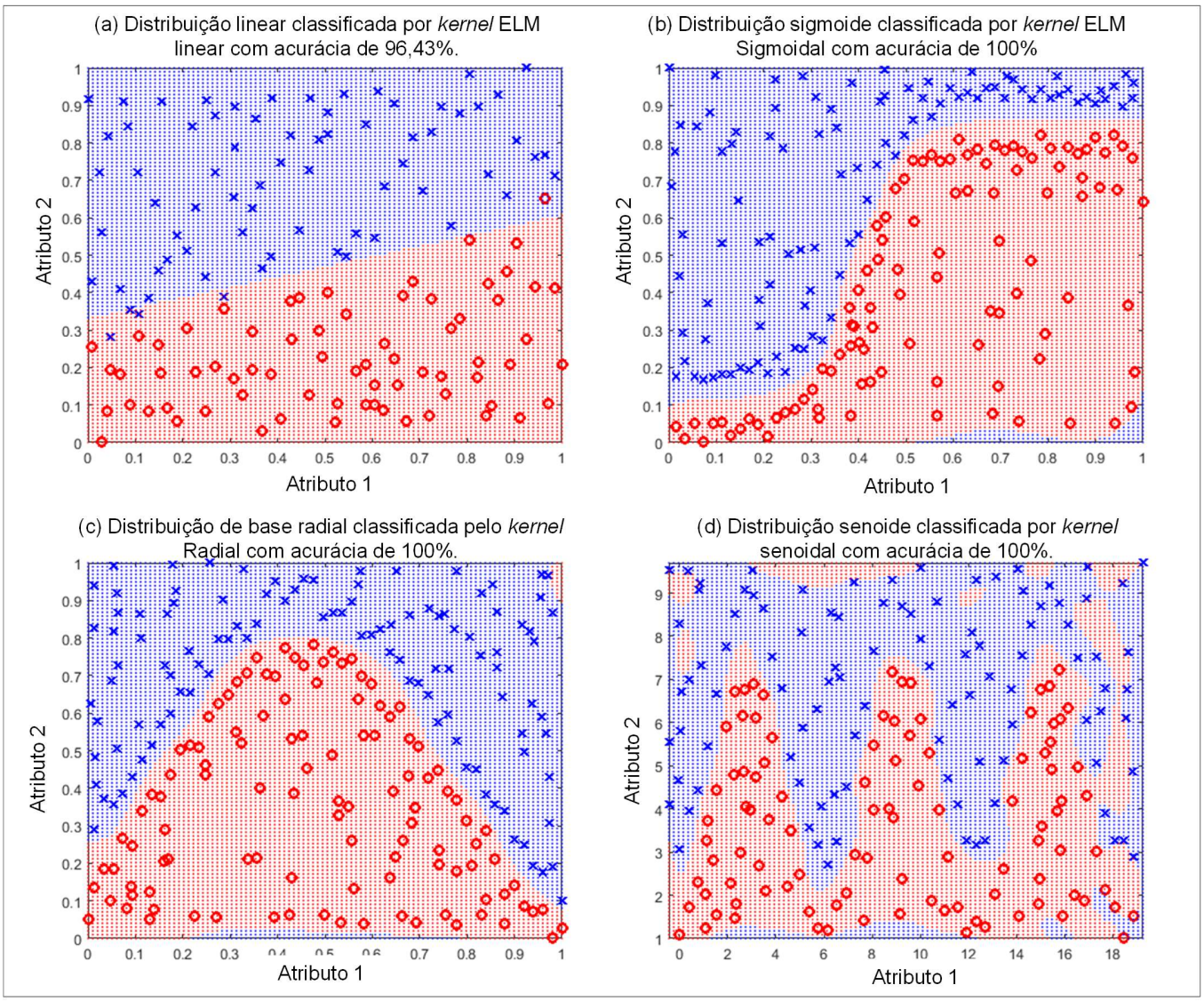

Figura 3. Atuações bem-sucedidas dos *kernels* compatíveis com os conjuntos de dados.

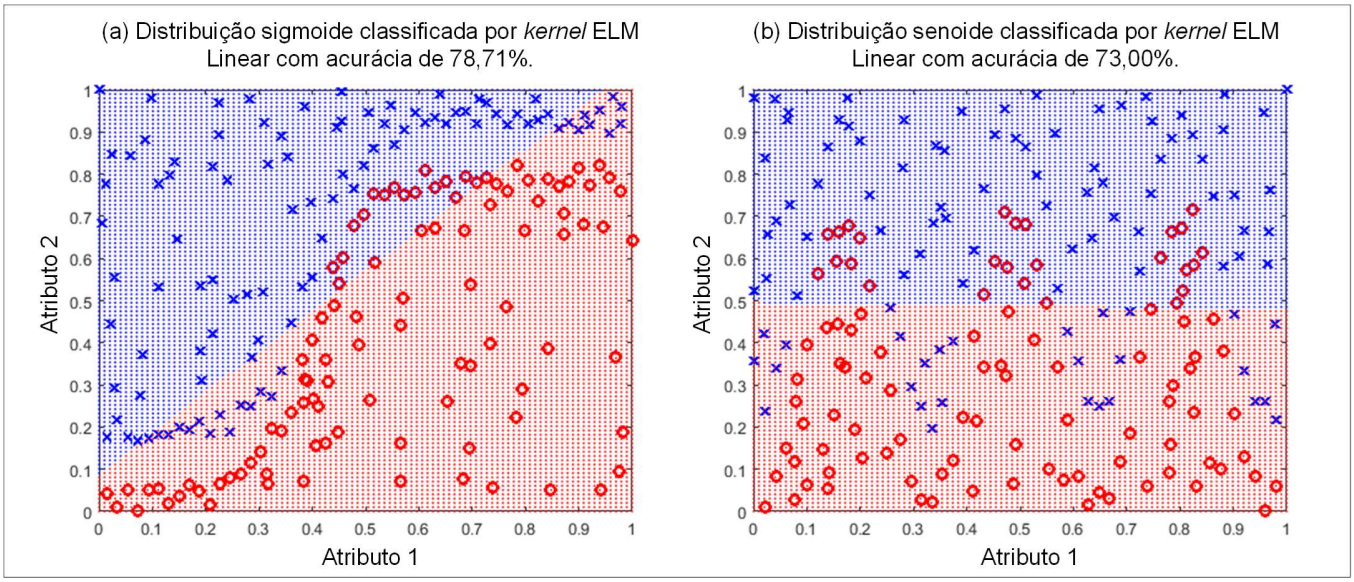

Figura 4. Atuações malsucedidas do *kernel* Linear em conjuntos de dados não-linearmente separáveis.

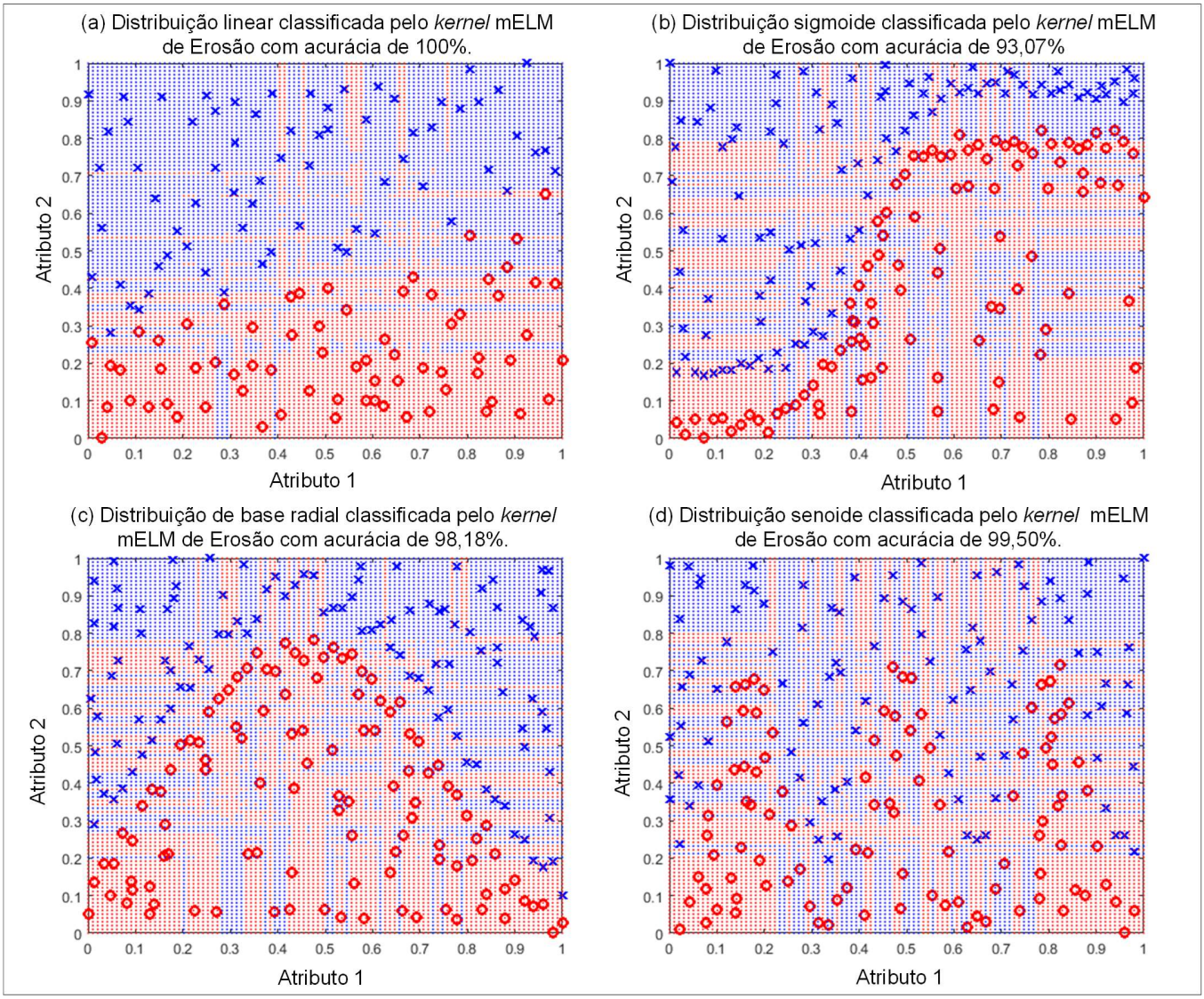

Figura 5. Atuações bem-sucedidas do mELM *kernel* Erosão em diversos conjuntos de dados.

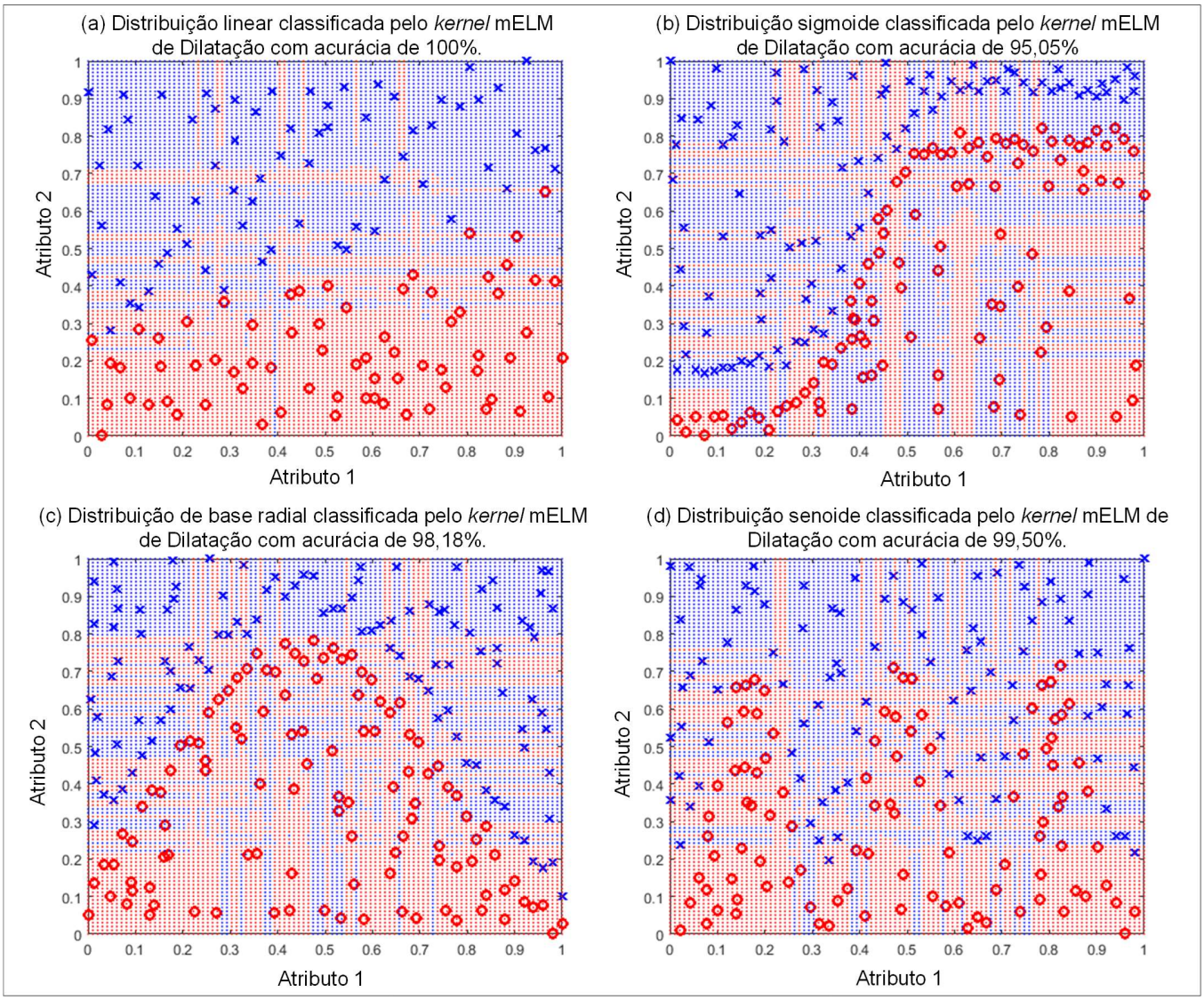

Figura 6. Atuações bem-sucedidas do mELM *kernel* Dilatação em diversos conjuntos de dados.

A. Extração de Características dos Executáveis

Quanto às características, o executável passa por um processo de *disassembling* visando a Engenharia Reversa do arquivo binário. Logo, as características, referentes ao executável, podem ser estudadas e, portanto, é possível investigar a intenção maliciosa do aplicativo suspeito. Então, a extração de características possibilita a definição de uma sequência de ações suspeitas. A descrição completa da extração de características empregada está na obra de LIMA, *et al* (2019). No total, são extraídas 630 características, de cada executável. De forma sintética, o nosso antivírus avalia os seguintes grupos de características.

- ✓ Características relacionadas à criptografia dados. Tal estratégia é típica de *ransomwares,* os quais sequestram os dados da vítima através da criptografia. Para descriptografar os dados, o invasor pede ao usuário um montante monetário para que possa ter de volta todos os seus dados;
- ✓ Características relacionadas à coleta de medidas dos elementos gráficos em tela, utilizado por hackers em conjunto com técnicas maliciosas de *ransomware*;
- ✓ Características relacionadas a *spywares* como *keyloggers* (captura de informações do teclado visando o furto de senhas e *logins*) e *screenloggers* (filmagem da tela da vítima). O antivírus criado visa monitorar se o arquivo suspeito tenta monitorar a atividade da Internet do usuário e informações particulares. Além disso, é investigado se o arquivo auditado tenta coletar senhas bancárias *on-line* e outras informações confidenciais e enviar os dados para seu criador;
- ✓ Características relacionadas a indícios de que o computador tenha sofrido fragmentação no disco rígido de modo a deteriorar o desempenho do computador;
- ✓ Características relacionadas ao Sistema Operacional. A forense digital averigua se são criados perfis e auditadas

informações internas (eg.: *drivers*) do Sistema Operacional Windows;

- ✓ Características relacionadas à inicialização do Windows. Audita-se caso o arquivo suspeito tenta modificar configurações de *boot*. Também é auditado se houve instalação de um *bootkit* (arquivos maliciosos com a finalidade de alterar e infectar o MBR[9] da partição) por meio de modificações no disco rígido;
- ✓ Características relacionadas ao Registro (Regedit) do Windows. Cabe ressaltar que a vítima pode não estar livre da infecção de um *malware* mesmo após a sua detecção e eliminação. A persistência das malfeitorias, mesmo após a exclusão do *malware*, ocorre devido à inserção de entradas (chaves) maliciosas no *Regedit*. Logo, quando o sistema operacional é inicializado, o *cyber*-ataque recomeça devido à chave mal-intencionada invocar a vulnerabilidade explorada pelo malware (ex.: redirecionar a página inicial do Internet Explorer);
- ✓ Características relacionadas à Antiforense Digital, incluindo técnicas de remoção, ocultação e subversão de evidências com o objetivo de reduzir as consequências dos resultados de análises forense. Isto é, o antivírus criado investiga se o arquivo suspeito tenta suspender a sua própria execução até que um determinado intervalo de tempo limite tenha decorrido. Tal estratégia típica dos *malwares* que ficam inativos até o término da quarentena dos antivírus comerciais;
- ✓ Características relacionadas à análise de GUI [10] do programa suspeito. O antivírus criado audita se o arquivo suspeito tenta detectar formas através de visão computacional e processamento digital de imagem;
- ✓ Características relacionadas à perícia ilícita da memória principal (RAM) do sistema local. O antivírus criado investiga se o aplicativo suspeito tenta reservar, confirmar ou alterar o estado de uma região de páginas no espaço de endereço virtual de um processo;
- ✓ Características relacionadas ao tráfego de rede. Averígua-se se o arquivo testado tenta consultar servidores DNS e criar uma sessão FTP ou HTTP em tempo de execução;
- ✓ Características pertencentes a *sniffers*. O antivírus investiga se o aplicativo suspeito tenta ler dados dos pacotes de rede, leitura esta que é feita a partir de requisições prévias do sistema local;
- ✓ Características típicas de *backdoors,* quando a vítima passa a receber comandos (ordens) remotos. Em uma aplicação convencional, um *socket* [11] é criado no servidor e aguarda (*listen*) uma conexão com o(s) usuário(s). De forma inversa, os *malwares* podem criar sockets, no sistema local, aguardando (*listen*) que um computador mal intencionado remoto requisite uma conexão e, portanto, possa receber as informações íntimas (senhas numéricas, imagens) da vítima;
- ✓ Características relacionadas a programas aplicativos utilitários. O antivírus criado verifica se o arquivo suspeito tenta alterar as configurações dos programas utilitários do Sistema Operacional Windows (ex.: reproduzir vídeos/áudios pelo Windows Media Player).

É importante ressaltar que cada uma destas características individualmente não necessariamente representa um comportamento de risco e o objetivo do antivírus proposto é exatamente diferenciar aqueles programas que as possuem legitimamente daqueles que as possuem para fins maliciosos. Então, a detecção de malwares deve ocorrer através do cruzamento de informações e, consequentemente, a ponderação de todos os comportamentos auditados. O antivírus proposto pondera, estatisticamente, as características auditadas através do uso de *Data Science* e máquinas de aprendizado extremo.

B. Classificadores

Quanto ao reconhecimento de padrão de *malwares*, 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 trabalho proposto utiliza onze redes ELMs cada uma dotada de um tipo de *kernel* distinto. No estado da arte, sete desses *kernels* são descritos por HUANG, *et al,* (2012), são eles; Linear, Polinomial, Transformada de Wavelets, Sigmoidal, Senoide, *Hard* Limite e Função de Base Triangular [10]. Além disso, são empregados quatro outros *kernels* autorais. Os quatro *kernels* são; Dilatação, Erosão, *fuzzy*-dilatação e *fuzzy*-erosão. Os *kernels* autorais Dilatação e Erosão foram detalhados na seção IV.

As fmELMs (*fuzzy*-morfológica ELMs) têm obtido sucesso no tratamento de imagens biomédicas, especificamente, na detecção e classificação de câncer de mama [3][4]. A solução autoral diz respeito à criação de aproximações dos operadores morfológicos clássicos de forma que seus tempos de execução e andamentos se mantenham uniformes independente dos valores dos dados de entrada. Logo, os desvios condicionais são substituídos por operações aritméticas que apresentam tempo de execução uniforme além de computacionalmente menos onerosas do que os desvios condicionais. As fmELMs são capazes de apresentar acurácias médias superiores a 90% [3][4].

Quanto aos *kernels* Polinomial e Wavelets, é necessário determinar os parâmetros (C, γ). Os parâmetros C e γ variam exponencialmente em sequencias crescentes, matematicamente de acordo com a função 2^n, onde $n = \{-24, 10, 0, 10, 25\}$. No *kernel* linear, há a investigação

[9] MBR: *Master Boot Record* – Tabela de Partição Mestre. MBR corresponde ao setor de inicialização da partição do disco rígido.

[10] GUI: *Graphical User Interface* – Interface Gráfica do Usuário.

[11] *Socket*: função visando o estabelecimento de comunicação em rede entre cliente e servidor. Disponível em: https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-socket. Acesso em março de 2020.

apenas do parâmetro de custo C, não cabendo a exploração do parâmetro do *kernel* γ [10].

Os *kernels* Sigmoidal, Senoidal, *Hard* Limite, Função de Base Triangular, Dilatação, Erosão, *fuzzy*-Dilatação e *fuzzy*-Erosão empregam arquiteturas dotadas de camada escondida. Então, há a investigação quanto à quantidade de neurônios da camada escondida desses *kernels*. A hipótese é verificar se arquiteturas que exijam um maior volume de cálculos, como por exemplo, aumentar a quantidade de neurônios na camada escondida, são capazes de gerar taxas de acertos superiores em comparação com arquiteturas que exijam uma menor quantidade de cálculos. Há a avaliação de duas arquiteturas, elas empregam 100 e 500 neurônios em suas respectivas camadas escondidas. Tais arquiteturas possuem lastro de excelentes acurácias na aplicação de redes ELM na área de Engenharia Biomédica [5].

São investigados 10 *folds,* referentes à validação cruzada do método *k-fold,* para cada arquitetura. O objetivo é que os resultados alcançados não sejam influenciados pelos conjuntos destinados ao treinamento e teste. Para isso, o total de aplicativos é dividido em dez partes. Na primeira execução, a primeiro parte é destinada ao conjunto de teste, enquanto as demais são reservadas ao treinamento. Essa alternância ocorre por dez execuções até que todas as dez partes tenham sido aplicadas à fase de teste. Como dito anteriormente, na rede ELM não há retropropagação de dados. Logo, o objetivo do método de validação cruzada *k-fold* não é estabelecer um critério de parada para evitar o *overfitting* (treinamento em excesso), mas sim verificar se o classificador sofre mudanças abruptas em suas acurácias a depender dos conjuntos destinados ao treinamento e teste. Também há o cuidado metodológico de selecionar equitativamente, de forma randômica, exemplares benignos e *malwares* para cada *fold*. O objetivo é que classificadores tendenciosos, em relação a uma determinada classe, não tenham suas taxas de acerto favorecidas.

Os *kernels* Polinomial, *Wavelets* e Linear não possuem camada escondida [10]. Os cálculos são baseados na transformação dos dados de entrada e podem trabalhar de maneira aproximadamente similar à dos *kernels* contendo arquiteturas dotadas de camadas escondidas [10]. No caso dos *kernels* Polinomial, *Wavelets* e Linear, há 10 execuções distintas referentes à validação cruzada do método *k-fold*, onde $k = 10$.

Por outro lado, nos *kernels* Sigmoidal, Senoidal, *Hard* Limite, Função de Base Triangular, Dilatação, Erosão, *fuzzy*-Dilatação e *fuzzy*-Erosão são geradas 300 execuções. São investigados 30 diferentes conjuntos de pesos iniciais referentes às ligações sinápticas entre os neurônios. A semente do gerador aleatório varia entre 1 e 30 de maneira incremental. Ademais, para cada conjunto de pesos sinápticos são investigados 10 *folds* referentes ao método *k-fold*, onde $k = 10$.

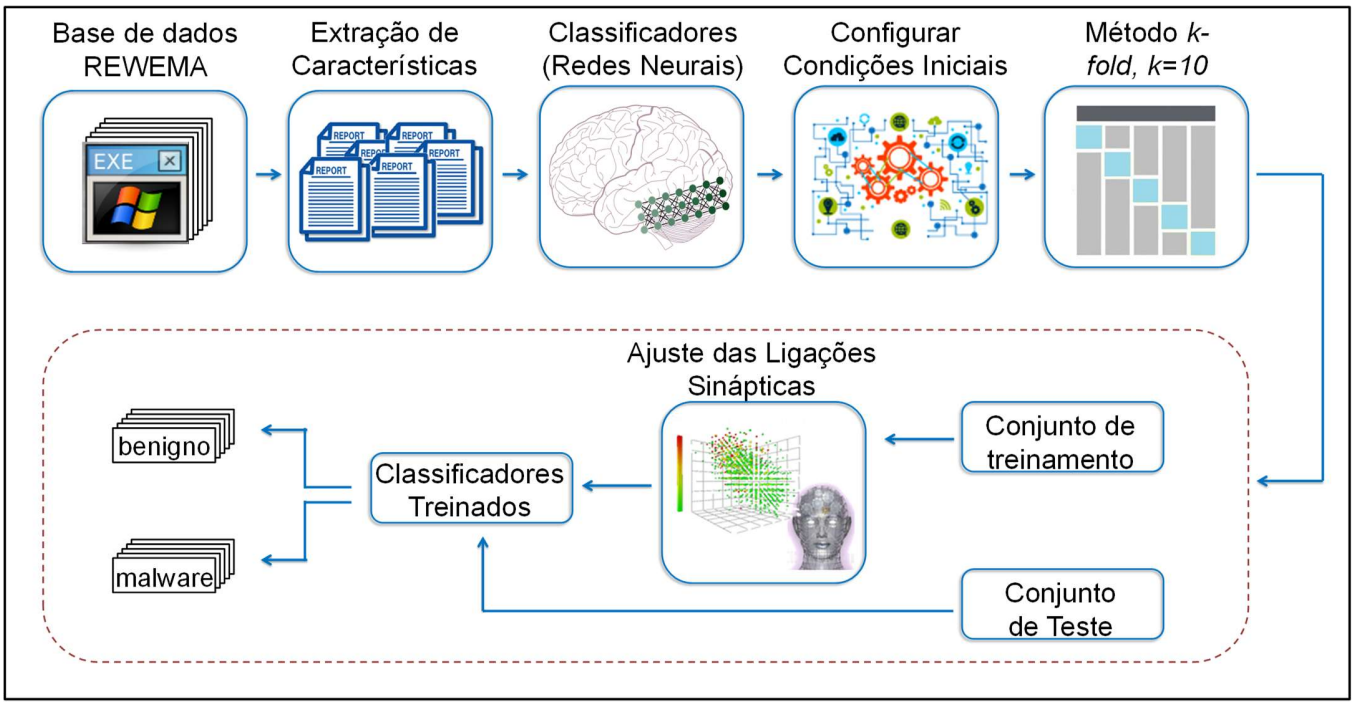

Figura 7. Diagrama da metodologia proposta.

VI. Resultados das Redes ELMs

Esta seção mostra os resultados utilizando a rede neural ELM, com suas configurações descritas na seção V.B. A extração de características dos executáveis está relatada na seção V.A. A classificação agrupa as imagens em duas classes: benigna e *malware*. Quanto à base de dados, foram usadas 3136 executáveis malignos e outros 3136 executáveis benignos disponíveis em (REWEMA, 2019).

A Tabela 3 detalha os resultados obtidos pelas redes ELMs com os *kernels* Polinomial e Wavelets. Cada linha da Tabela 4 contém 10 execuções referentes à validação cruzada do método *k-fold*, onde $k = 10$. Em relação à precisão na fase de teste, o máximo desempenho médio foi de 97.83% na distinção entre casos benignos e *malware* através do *kernel* Polinomial dotado dos parâmetros $(C, \gamma) = (2^{10}, 2^{-24})$. Na Tabela 3, na Tabela 4 e na Tabela 5, há apenas as descrições do melhor e pior caso, nessa ordem, para cada *kernel* ELM.

A Tabela 4 exibe os resultados alcançados pela rede ELM com kernel Linear. Há a investigação apenas do parâmetro de custo C, não cabendo a exploração do parâmetro do kernel γ em um *kernel* Linear [10]. Cada linha da Tabela 4 contém 10 execuções referentes à validação cruzada do método *k-fold*, onde $k = 10$. A máxima e mínima precisão foi de 97,19% e 94,45%, respectivamente. Logo, a investigação do parâmetro de custo C, é capaz de maximizar a precisão quanto à identificação de *malwares*.

Tabela 3. Resultados das redes ELMs (melhor e pior casos). Os parâmetros (C, γ) variam de acordo com o conjunto $\{2^{-24}, 2^{-10}, 2^{0}, 2^{10}, 2^{25}\}$.

Kernel	(C, γ)	*Acerto treino (%)*	*Acerto teste (%)*	*Tempo treino (seg.)*	*Tempo teste (seg.)*
Polinomial	$(2^{10}, 2^{-24})$	97,84 ± 0,09	**97,83 ± 0,75**	7,03 ± 0,08	0,56 ± 0,02
	$(2^{-10}, 2^{-24})$	50,01 ± 0,00	49,92 ± 0,00	7,94 ± 0,23	0,61 ± 0,02
Wavelets	$(2^{10}, 2^{0})$	**100,00 ± 0,00**	71,97 ± 2,58	12,63 ± 0,38	0,54 ± 0,02
	$(2^{10}, 2^{25})$	55,68 ± 0,22	54,58 ± 1,61	6,33 ± 0,06	0,50 ± 0,01

Tabela 4. Resultados das redes ELMs dotada de kernel Linear (melhor e pior casos). Os parâmetros C variam de acordo com o conjunto $\{2^{-24}, 2^{-10}, 2^{0}, 2^{10}, 2^{25}\}$.

Kernel	C	*Acerto treino (%)*	*Acerto teste (%)*	*Tempo treino (seg.)*	*Tempo teste (seg.)*
Linear	2^{0}	**97,58 ± 0,27**	**97,19 ± 0,82**	4,02 ± 0,11	0,20 ± 0,02
	2^{-10}	94,54 ± 0,19	94,45 ± 1,15	3,77 ± 0,06	0,19 ± 0,01

Tabela 5. Resultados das redes ELMs (melhor e pior casos). A quantidade de neurônios na camada escondida varia de acordo com o conjunto $\{100, 500\}$.

Kernel	Neurônios	*Acerto treino (%)*	*Acerto teste (%)*	*Tempo treino (seg.)*	*Tempo teste (seg.)*
Sigmoidal	500	79,75 ± 0,27	77,34 ± 1,72	1,15 ± 0,16	0,02 ± 0,01
	100	75,25 ± 0,56	74,89 ± 1,57	0,30 ± 0,06	0,01 ± 0,01
Senoidal	500	72,90 ± 0,61	66,08 ± 1,81	1,19 ± 0,11	0,03 ± 0,01
	100	63,36 ± 0,61	61,15 ± 1,85	0,30 ± 0,05	0,01 ± 0,00
Hard limite	100	50,01 ± 0,00	49,92 ± 0,00	0,28 ± 0,03	0,00 ± 0,00
	500	50,01 ± 0,00	49,92 ± 0,00	1,36 ± 0,22	0,02 ± 0,01
Base triangular	500	50,07 ± 0,04	50,05 ± 0,15	0,85 ± 0,09	0,02 ± 0,00
	100	50,02 ± 0,03	50,02 ± 0,11	0,28 ± 0,07	0,01 ± 0,01
Fuzzy-Dilatação	500	96,57 ± 0,46	95,49 ± 1,15	1,17 ± 0,08	0,02 ± 0,00
	100	95,46 ± 0,53	95,26 ± 1,02	0,37 ± 0,10	0,01 ± 0,01
Fuzzy-Erosão	500	96,57 ± 0,46	95,42 ± 1,15	1,21 ± 0,21	0,02 ± 0,01
	100	95,28 ± 0,55	95,03 ± 1,05	0,37 ± 0,03	0,01 ± 0,00
Dilatação	500	**99,92 ± 0,01**	**99,80 ± 0,14**	9,32 ± 1,31	0,85 ± 0,13
	100	99,56 ± 0,16	99,44 ± 0,31	2,21 ± 0,29	0,18 ± 0,06
Erosão	500	96,16 ± 3,02	95,95 ± 3,26	10,96 ± 0,70	1,05 ± 0,13
	100	84,72 ± 7,76	84,56 ± 7,93	2,19 ± 0,37	0,19 ± 0,01

A Tabela 5 detalha os resultados obtidos pelas redes ELMs com os kernels Sigmoidal, Senoidal, Hard Limite, Função de Base Triangular, *fuzzy*-Dilatação, *fuzzy*-Erosão, Dilatação e Erosão. Cada linha da Tabela 5 contém 300 execuções distintas. São investigados 30 diferentes conjuntos de pesos iniciais referentes às ligações sinápticas entre os neurônios. A semente do gerador aleatório varia entre 1 e 30 de maneira incremental. Então, para cada conjunto de pesos sinápticos são investigados 10 *folds* referentes ao método k-*fold*, onde $k = 10$, totalizando as 300 execuções mencionadas. Em relação à precisão, o máximo desempenho médio foi de 99,80% com desvio padrão de 0.14 através do mELM *kernel* Dilatação dotado de 500 neurônios na sua camada escondida. Logo, o nosso *kernel* Dilatação apresenta a excelente qualidade de não sofrer mudanças abruptas em função das condições iniciais.

Dentre as melhores configurações de cada *kernel*, a melhor e a pior acurácia média, resultante do treinamento, foram 100,00% e 50,01 % através do *kernel* Wavelets e do *kernel Hard* Limite, respectivamente. Dentre as melhores configurações da fase de teste, a melhor e a pior precisão média foram 99,80% e 49,92% através do *kernel* autoral Dilatação e do *kernel Hard* Limite, respectivamente. Dadas as diferenças de resultado obtidas, pode-se concluir que a escolha de um *kernel* adequado é essencial para maximizar a precisão quanto à identificação de *malwares*.

Os *kernels* do estado-da-arte não foram capazes de gerar acurácias superiores aos *kernels* morfológicos autorais mesmo quando configurados com o quíntuplo de neurônios na camada escondida. Cabe atentar que o *kernel* autoral Dilatação obteve as melhores acurácias médias, em relação ao estado-da-arte, independente da arquitetura explorada. O trabalho proposto estima que os *kernels* morfológicos sejam capazes de se adequar a qualquer fronteira decisão. Por se inspirar na Morfologia Matemática, as mELMs são capazes de modelar qualquer forma presente nas fronteiras de decisões das redes neural.

Uma questão fundamental, em redes neurais artificiais, diz respeito à adequação das condições iniciais em relação à natureza do problema. Em redes ELMs, denotam-se condições iniciais como (*i*) as configurações iniciais dos pesos sinápticos e (*ii*) o conjunto de amostras reservadas ao aprendizado (*k-fold*). Em síntese, um *kernel* pode obter resultados satisfatórios com um conjunto de parâmetros iniciais. Tal fato não implica que esse mesmo *kernel* terá sucesso quando for inicializado possuindo um outro conjunto de condições iniciais.

O *kernel* autoral Dilatação apresenta a melhor precisão acompanhada de uma dispersão ajustada. O *kernel* Dilatação possui um desvio padrão de 0,14 na fase de teste dotado de 500 neurônios em sua camada escondida. Conclui-se que o *kernel* Dilatação apresenta a excelente qualidade de não sofrer mudanças abruptas em função das condições iniciais. Cabe ressaltar que quanto maior for a quantidade de configurações investigadas, maior o nível de entropia tende e, consequentemente, haverá uma menor homogeneidade entre as amostras. Mesmo com a adversidade de possuir 30 vezes mais amostras do que os *kernels* Polinomial, Wavelets e Linear, o *kernel* autoral Dilatação apresenta um desvio padrão menor do que tais classificadores. A dispersão do *kernel* Dilatação será exibida visualmente no gráfico *boxplot* apresentado na Figura 8 na seção a seguir.

VII. Resultados em Relação ao Estado-da-Arte

Nesta seção, o antivírus proposto é comparado com o antivírus de LIMA, *et al* (2019). Na etapa de classificação, o nosso antivírus é dotado do *kernel* mELM Dilatação e contém 500 neurônios em sua camada escondida. Enquanto o antivírus de LIMA, *et al* (2019) emprega redes neurais baseadas em retropropagação de dados.

O antivírus autoral também é comparado a *Deep Learning* de SANTOS, *et al,* 2019. Tal obra do estado-da-arte não visa a detecção de *malwares* e, sim, o reconhecimento óptico de caracteres. Logo, a técnica de SANTOS, *et al,* 2019 sofre uma adaptação em sua camada de neurônios de entrada. Ao invés de processamento digital de imagem, os atributos de entrada dizem respeito à extração de características dos *malwares*. A *Deep Learning* de SANTOS, *et al.* 2019 apresenta uma única camada convolucional e não há retropropagação de dados. A camada convolucional emprega todos seus 30 mil filtros simultaneamente de modo que o tempo de treinamento é compatível com aplicativos que precisam de treinamento frequentemente.

A Tabela 6 descreve os resultados obtidos pelo antivírus proposto e pelo estado-da-arte. Cada linha da Tabela 6 contém 300 execuções. São investigados 30 diferentes conjuntos de pesos iniciais referentes às ligações sinápticas entre os neurônios. A semente do gerador aleatório varia entre 1 e 30 de maneira incremental. Então, para cada conjunto de pesos sinápticos são investigados 10 *folds* referentes ao método *k-fold*, onde $k = 10$. O antivírus autoral apresenta a maior acurácia média mesmo quando comparado ao melhor cenário obtido pela obra de LIMA, *et al* (2019).

No tocante a obra de LIMA, *et al* (2019), a mínima taxa de acerto foi de 48,87% para a rede que emprega o Rprop (Retropropagação Resiliente). Essa abordagem possui uma arquitetura com duas camadas escondidas com 100 neurônios cada uma. Enquanto, a máxima precisão foi de 98,13% para a rede inspirada em retropropagação de Gradiente Escalonado. A arquitetura dessa rede contém uma camada escondida com 100 neurônios. Quanto à rede *Deep Learning* de SANTOS, *et al* (2019), a acurácia média foi de 98,95% com desvio padrão de 2,72%.

A Figura 8 e a Figura 9 são representações gráficas dos resultados descritos na Tabela 6. A Figura 8 (a) e a Figura 8 (b) apresentam os *boxplots* referentes às precisões ótimas durante a fase de treinamento e de teste, respectivamente. A Figura 8 (a) apresenta os *boxplots*, da etapa de treinamento, tanto do antivírus autoral quanto do estado-da-arte. A melhor acurácia média, resultante do treinamento, foi de 100.00% através da *Deep Learning* de SANTOS, *et al.* (2019). O antivírus de LIMA, *et al.* (2019) obteve precisão média de 48,82% e 98,82%, no seu pior e melhor cenário,

respectivamente. O antivírus autoral obteve um desempenho médio de 99,92% com desvio padrão de 0,01%. Logo, o antivírus criado apresenta a vantajosa característica de não sofrer mudanças abruptas em função das condições iniciais (ligações sinápticas e *k-fold*).

Tabela 6. Comparação entre o antivírus proposto e o estado-da-arte.

Técnica	***Acerto treino (%)***	***Acerto teste (%)***	***Tempo treino (seg.)***	***Tempo teste (seg.)***
Antivírus autoral	**99,92 ± 0,01**	**99,80 ± 0,14**	9,32 ± 1,31	0,85 ±0,13
LIMA, *et al* (2019), pior configuração	48,82 ± 7,03	48,87 ± 6,83	1,15 ± 0,12	0,06 ± 0,01
LIMA, *et al* (2019), melhor configuração	98,82 ± 0,95	98,13 ± 0,68	37,14 ± 10,90	0,04 ± 0,01
Deep Learning de SANTOS, *et al* (2019)	100,00 ± 0,01	98,95 ± 2,72	887,49 ± 102.03	2,44 ± 0,30

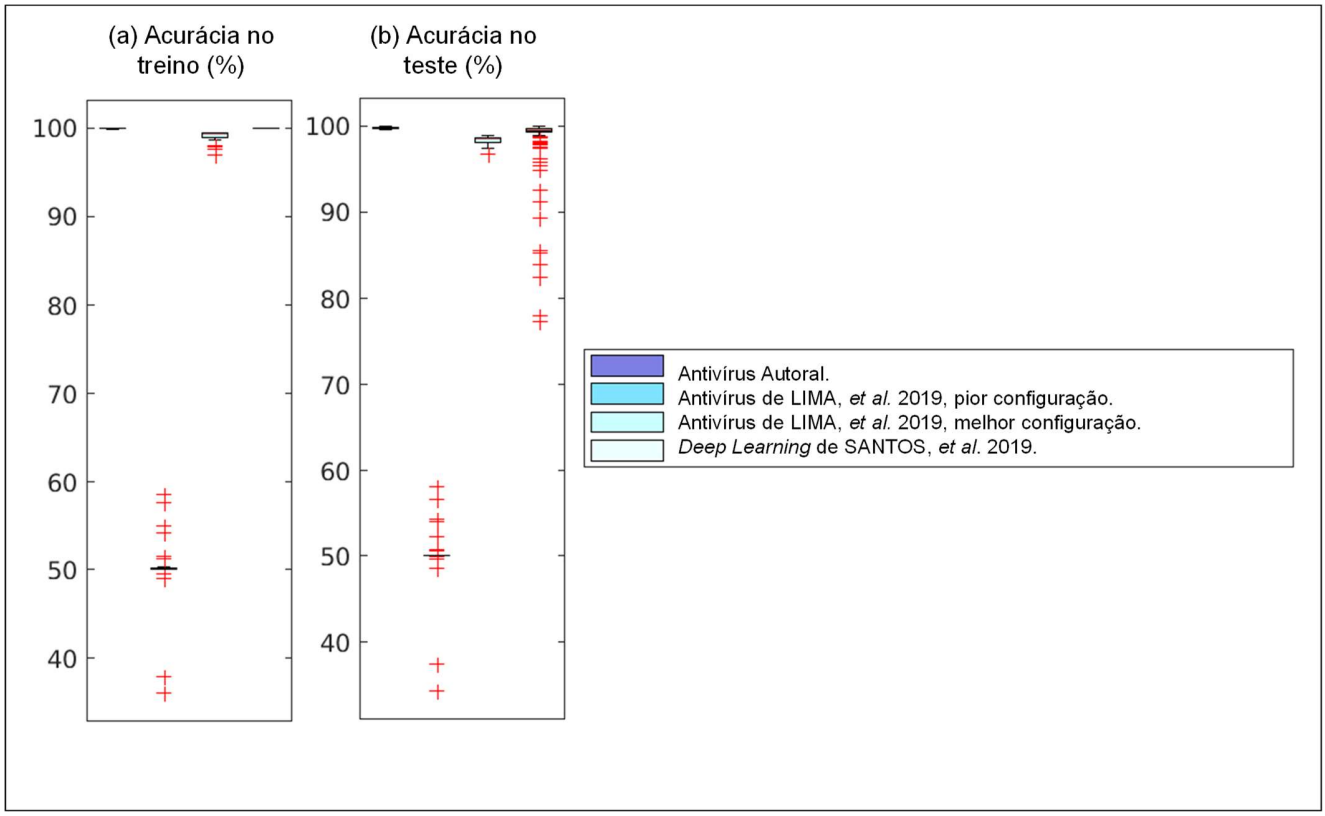

Figura 8. (a) *Boxplot* referente às acurácias de treinamento. (b) *Boxplot* referente às acurácias de teste.

A Figura 8 (b) exibe os *boxplots* referentes às melhores precisões na fase de teste. A melhor precisão média, resultante do teste, foi de 99,80% através do antivírus autoral. A *Deep Learning* de SANTOS, *et al* (2019) alcançou um desempenho médio de 98,95%. O antivírus de LIMA, *et al* (2019) obteve precisão média de 48,87% e 98,13%, no seu pior e melhor cenário, respectivamente. Logo, corrobora-se que redes neurais baseadas em retropropagação podem sofrer variações abruptas, em suas acurácias, a depender dos seus parâmetros de configurações. Então, foi salutar a decisão de LIMA, *et al.* (2019) em explorar distintas funções de aprendizado, gradientes e arquiteturas de modo a otimizar a precisão de suas redes neurais baseadas em retropropagação de dados.

A Figura 9 (a) e a Figura 9 (b) apresentam os *boxplots* referentes aos tempos gastos durante a fase de treinamento e de teste, respectivamente. Em relação ao tempo de treinamento, a *Deep Learning* de SANTOS, *et al* (2019) é mais lenta em comparação as demais visto que são empregados 30.000 filtros convolucionais em seu processamento de dados [22]. Por outro lado, o antivírus autoral consome apenas 9,32 segundos para concluir, em média, o seu treinamento. Apesar de o aprendizado ser baseado em retropropagação, a obra de LIMA, *et al.* (2019) conclui seu treinamento na ordem de segundos. Em relação ao tempo consumido durante a fase de teste, as aplicações consumiram tempos bastante próximos sem grandes discrepâncias.

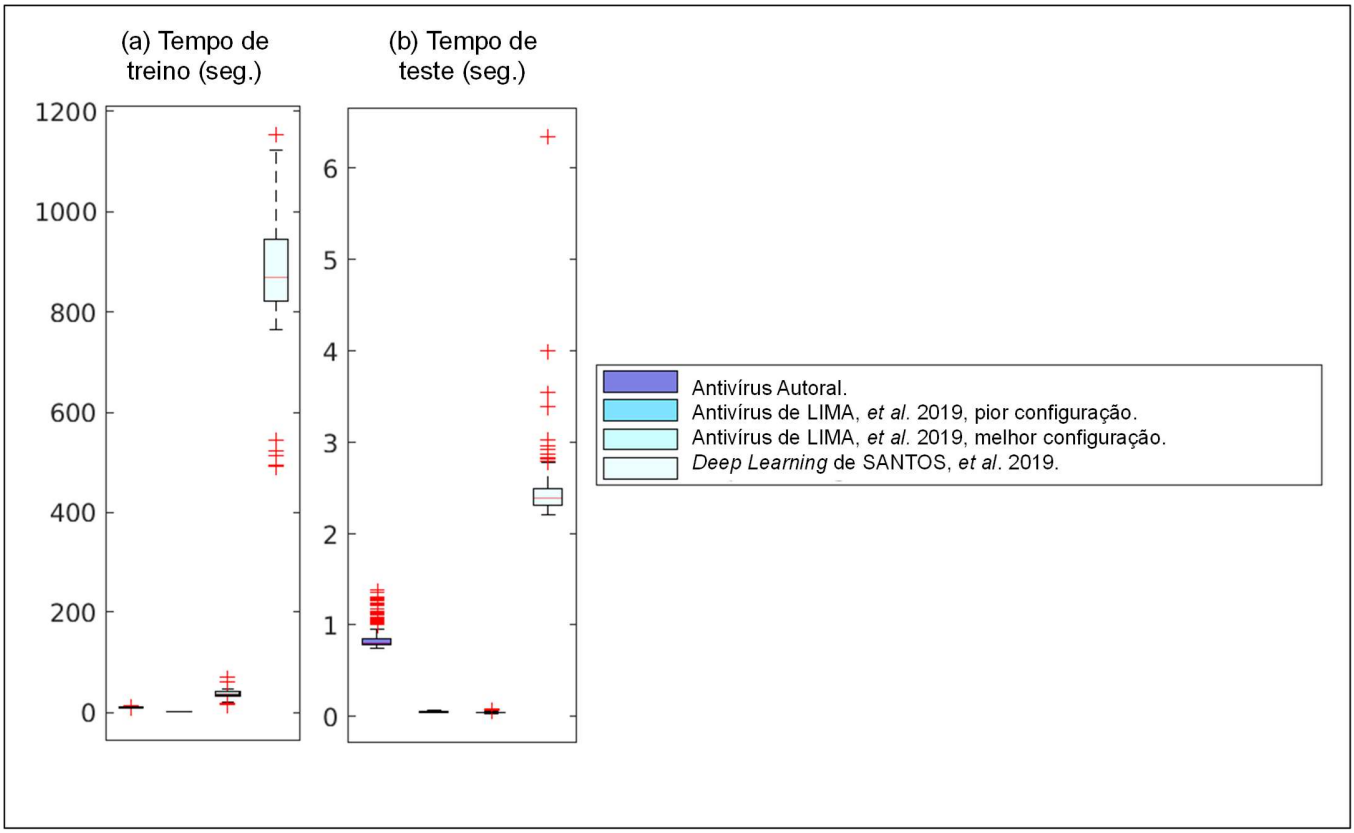

Figura 9. (a) Boxplot referente aos tempos gastos durante o treinamento. (b) Boxplot referente aos tempos consumidos durante a fase de teste.

A Tabela 7 exibe as matrizes de confusão das técnicas apresentadas na Tabela 6 em termos percentuais. A matriz de confusão é importante para a verificação da qualidade de um aprendizado supervisionado. Na Tabela 7, B. e M. são abreviaturas de Benigno e *Malware*. As classes desejadas estão dispostas no rótulo vertical enquanto as classes obtidas estão no rótulo horizontal [12].

Na Tabela 7, por exemplo, o antivírus proposto, na fase de teste, classificou em média, de maneira equivocada, 0,35% casos como benignos quando se tratava de casos malignos (falso negativo). Seguindo o mesmo raciocínio, houve a classificação média de 0,04% casos ditos como *malwares* quando se tratava de aplicações benignas (falso positivo). Na matriz de confusão, a diagonal principal é ocupada por casos nos quais a classe obtida coincide com a classe desejada. Então, um bom classificador deve ter uma diagonal principal ocupada por valores altos enquanto as demais posições devem possuir valores baixos. Na Tabela 7, as diagonais principais estão em negrito.

Na perícia forense digital, um falso positivo implicaria em um aplicativo benigno impedido de ser executado pelo sistema. Um falso negativo, no entanto, pode implicar em um *malware* que não foi detectado. Vale salientar que os *malwares* podem gerar malefícios irreversíveis e irrecuperáveis para toda a rede mundial de computadores. Isso posto, um falso negativo pode implicar na perda da dignidade, das finanças e da saúde mental da vítima. Enfatiza-se que o antivírus autoral apresenta o menor percentual médio de falsos negativos com apenas 0,35%.

Ainda quanto à Tabela 7, sensibilidade e especificidade dizem respeito à capacidade do antivírus identificar os aplicativos *malwares* e benignos, respectivamente. O trabalho proposto apresenta a matriz de confusão em termos percentuais de modo a facilitar a interpretação da sensibilidade e especificidade. Em síntese, a sensibilidade e a especificidade estão apresentadas na própria matriz de confusão, descrita na Tabela 7. Por exemplo, o antivírus proposto alcança, em média, 99,65% tanto em relação à sensibilidade quanto a verdadeiros positivos. Seguindo o mesmo raciocínio, o antivírus autoral obtém, em média, 99,96% tanto para especificidade quanto para verdadeiros negativos.

A Tabela 8 exibe os testes de hipótese paramétrico *t-students* e não-paramétrico *Wilcoxon* entre o antivírus criado e o estado-da-arte. É possível concluir que o antivírus autoral é estatisticamente distinto em comparação a todas as demais amostras. A explicação é que, tanto no teste paramétrico *t-students* quanto no teste não-paramétrico *Wilcoxon*, a hipótese nula foi rejeitada. Logo, as qualidades dos antivírus propostos são estatisticamente distintas.

Na Figura 10, o antivírus autoral demonstrou uma grande vantagem quando comparado ao estado-da-arte. O nosso antivírus foi capaz de alcançar a melhor precisão média com

[12] Matriz de confusão. Disponível em: https://www.mathworks.com/help/stats/confusionmat.html. Acesso em maio de 2020.

99,80% de acurácia acompanhada de um tempo de treinamento correspondente a 9,32 segundos. Nesse caso, a proporção entre precisão média e tempo de treinamento obtida pelo nosso antivírus é 4 e 96 vezes superior ao antivírus de LIMA, *et al.* (2019) e a *Deep Learning* de SANTOS, *et al.* (2018), respectivamente. Ainda quanto à Figura 10, o pior cenário do antivírus de LIMA, *et al.* (2019) foi desconsiderado. Não faz sentido adotar um antivírus com taxas de precisão em torno de 50% porque é quase um resultado aleatório.

A relação média entre a acurácia percentual e tempo de treinamento em ordem inversa é empregada em Engenharia Biomédica [14]. Admite-se que o estabelecimento de tal relação assume papel importante na área de Segurança da Informação visto que são lançados 8 (oito) novos *malwares* por segundo [11]. Então, paradoxalmente um antivírus recém-lançado já pode estar obsoleto e necessitar de um novo treinamento mediante uma vulnerabilidade recém-descoberta. Em síntese, o tempo de aprendizado de um antivírus não deve ser discrepante em comparação à taxa de criação de novos *malwares* mundialmente.

Tabela 7. Matrizes de confusão das técnicas apresentadas na Tabela 6 (%).

'Técnica		***Treino***		***Teste***	
		M.	B.	M.	B.
Antivírus autoral	M.	**99,83 ± 0,03**	0,17 ± 0,03	**99,65 ± 0,26**	0,35 ± 0,26
	B.	0,00 ± 0,00	**100,00 ± 0,00**	0,04 ± 0,11	**99,96 ± 0,11**
Antivírus de LIMA, *et al* (2019), melhor configuração	M.	**99,50 ± 0,13**	0,50 ± 0,13	**99,22 ± 0,31**	0,78 ± 0,31
	B.	1,41 ±1,33	**98,59 ± 1,33**	2,62 ± 1,20	**97,38 ± 1,20**
Deep Learning de SANTOS, *et al* (2019)	M.	**100,00 ± 0,00**	0,00 ± 0,00	**98,73 ± 3,17**	1,27 ± 3,17
	B.	0,00 ± 0,00	**100,00 ± 0,00**	0,80 ± 2,23	**99,20 ± 2,23**

Tabela 8. Teste de hipótese *t-students* e *Wilcoxon* entre o antivírus criado e o estado-da-arte.

Comparação	***t-students (teste paramétrico)***		***Wilcoxon (teste não-paramétrico)***	
	Hipótese	***Valor p***	***Hipótese***	***Valor p***
Antivírus autoral *vs* LIMA, *et al* (2019), pior configuração	1	5.20745e-321	1	2.82789e-103
Antivírus autoral *vs* LIMA, *et al* (2019), melhor configuração	1	1.51156e-150	1	9.68508e-101
Antivírus autoral *vs* *Deep Learning* de SANTOS, *et al* (2019)	1	1.26316e-07	1	1.81344e-36

VIII. CONCLUSÃO

A meta do trabalho proposto foi suprir as limitações dos antivírus comerciais os quais, mesmo com faturamento bilionário, apresentam baixa efetividade e vêm sendo criticados por institutos de pesquisa há mais de uma década.

A forma de atuação dos antivírus comerciais se baseia em assinaturas, quando o executável suspeito é comparado a uma lista negra confeccionada a partir de denúncias prévias (e isso requer que já tenha havido vítimas). Admite-se que listas negras são efetivamente nulas mediante a atual taxa mundial de criação de *malwares* que é de 8 (oito) novos aplicativos maliciosos por segundo [11]. Conclui-se que *malwares* tem capacidade de ludibriar os antivírus e demais mecanismos de *cyber*-vigilância [13][20].

Nesse trabalho, nós propomos um antivírus, dotado de inteligência artificial, capaz de identificar *malwares* por meio de modelos baseados em redes neurais de treinamento rápido e de alta acurácia. O nosso antivírus é dotado de máquinas morfológicas de aprendizado extremo (mELMs). As nossas mELMs são inspiradas na teoria de processamento de imagem da Morfologia Matemática. Nossa mELM, através do *kernel* de Dilatação, alcança o melhor resultado comparado às redes neurais clássicas ELMs dotadas de distintos *kernels*. O *kernel* autoral de Dilatação consegue distinguir aplicativos *malwares* dos benignos em 99,80% dos casos acompanhado de um tempo de treinamento de 9,32 segundos.

A explicação do sucesso da nossa máquina morfológica de aprendizado 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 empregadas pelas redes neurais clássicas. O mapeamento da fronteira de decisão, realizado pelos nossos *kernels* morfológicos, emprega os próprios valores das amostras reservadas ao treinamento. Nossas máquinas morfológicas interpretam a fronteira de decisão de uma rede neural como imagem *n*-dimensional, onde *n* diz respeito à quantidade de características extraídas, contendo distintos corpos capazes de serem delineados através do uso da Morfologia Matemática. Logo, nossas máquinas morfológicas lidam naturalmente com o delineamento e modelagem das regiões mapeadas às distintas classes pertencentes a qualquer repositório de aprendizado estatístico.

Em nossa metodologia, as características extraídas dos executáveis serviram como atributos de entrada das redes neurais artificiais empregadas como classificadores. Logo, as redes neurais se tornaram capazes de reconhecer o padrão de comportamentos previamente classificados como suspeitos em tempo real. Portanto, através de nosso antivírus, é possível detectar o comportamento maliciosos do executável suspeito, de maneira preventiva, antes mesmo dele ser executado pelo usuário. Quanto à base de dados, o trabalho proposto empregou a REWEMA dotada de 6272 executáveis, igualitariamente, agrupados em arquivos benignos e *malwares*. Como a base REWEMA é disponibilizada livremente, há a total possibilidade de a metodologia proposta ser replicada, por terceiros.

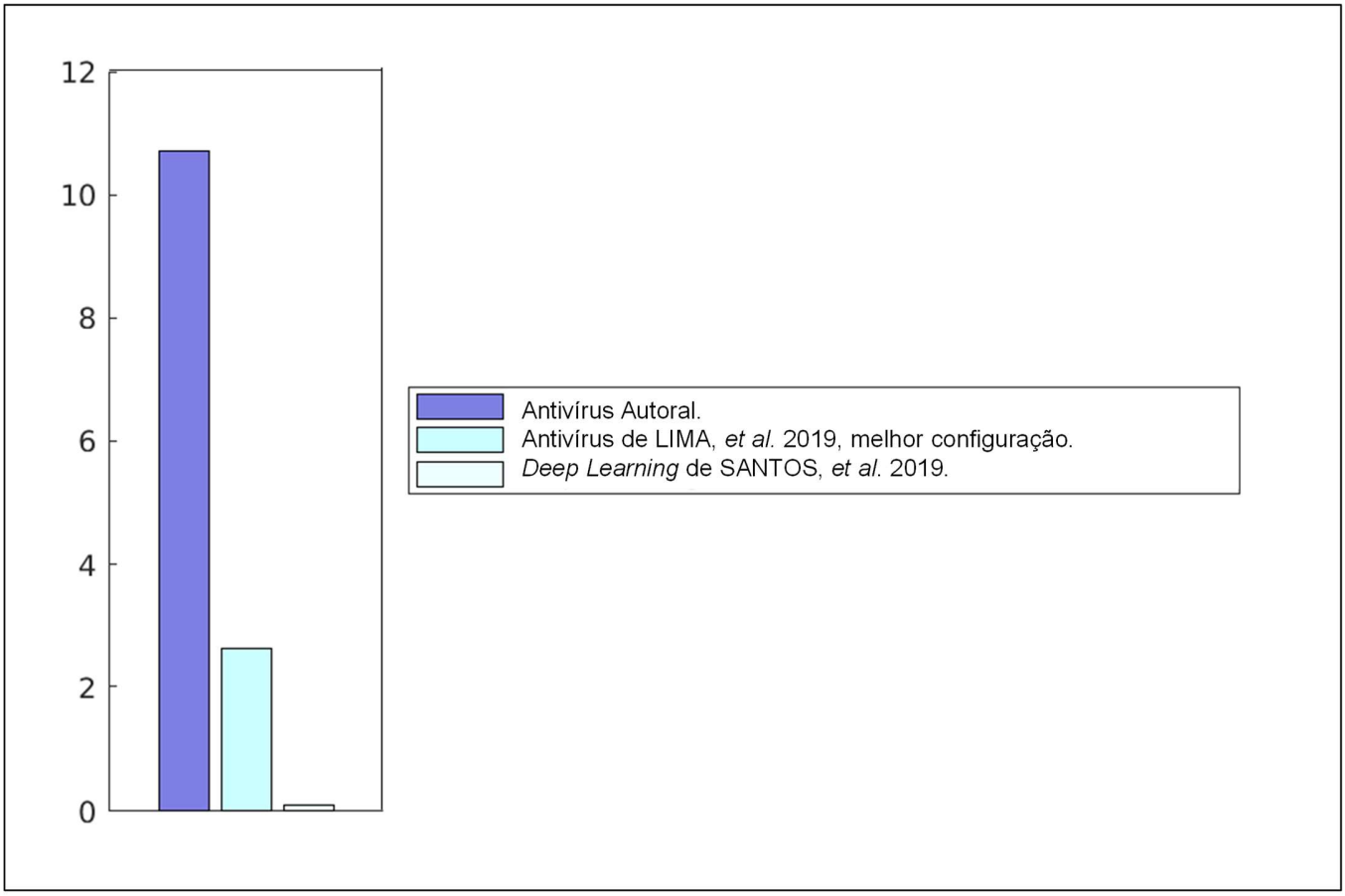

Figura 10. Proporção entre acurácia média e tempo de treinamento obtida pelo antivírus autoral e o estado-da-arte.

REFERÊNCIAS

[1] ALAZAB, M. Profiling and Classifying the Behavior of Malicious Codes. The Journal of Systems and Software 100 (2015) 91–102., 2015.

[2] AMOR, N. B.; BENFERHAT, S.; AND ELOUEDI, Z. Naive bayes vs decision trees in intrusion detection systems. In Proceedings of the 2004 ACM symposium on Applied computing, pages 420–424., 2004.

[3] AZEVEDO, W. W. et al. Fuzzy Morphological Extreme Learning Machines to detect and classify masses in mammograms. In: 2015 IEEE International Conference on Fuzzy Systems (FUZZIEEE), 2015, Istanbul. , 2015 a.

[4] 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), 2015, Killarney., 2015 b.

[5] AZEVEDO, WASHINGTON W. ; SANTANA, M. A. ; DA SILVA-FILHO, ABEL G. ; LIMA, S. M. L. ; SANTOS, W. P. . 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, 2020, v. , p. 1-300.

[6] BAI, J.; WANG, J.; ZOU, G. A Malware Detection Scheme Based on Mining Format Information. The Scientific World Journal. (2014), Article ID 260905, 11 pages., 2014.
[7] DING, Y. et al. Control Flow-Based Opcode Behavior Analysis for Malware Detection. Computers & Security 44 (2014) 65-74., 2014.
[8] GONZALES, R. C.; WOODS, R. E. Digital Image Processing. Segunda Edição. Editora Prentice Hall, 2002.
[9] HUANG, G. B. et al. Classification ability of single hidden layer feedforward neural networks., The IEEE Transactions on Neural Networks and Learning Systems. 2000;11(3):799-801. doi: 10.1109/72.846750, 2000.
[10] HUANG, G. B. et al. Extreme Learning Machine for Regression and MultiClass Classification. IEEE Transactions on Systems, Man, and Cybernetics. 42(2), (2012) 513-519., 2012.
[11] INTEL. McAfee Labs: Threat Report. Disponível em: https://www.mcafee.com/ca/resources/reports/rp-quarterly-threats-mar-2018.pdf. Acesso em março de 2020., 2018.
[12] KUMAR, A.; KUPPUSAMY, K. S.; AGHILA, G. A Learning Model to Detect Maliciousness of Portable Executable using Integrated Feature Set. Journal of King Saud University – Computer and Information Sciences., 2017.
[13] LIMA, S. M. L. et al. Antivírus dotado de Rede Neural Artificial visando Detectar *Malwares* Preventivamente. iSys - Brazilian Journal of Information Systems, [S.l.], v. 11, n. 4, p. 31-62, 2019.
[14] LIMA, S. M. L.; SILVA-FILHO, A. G.; DOS SANTOS, W. P. 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.
[15] 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. 134, (2016), 11-29., 2016.
[16] 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. 1ed.New York: Nova Science, 2020, v. 1, p. 27-64.
[17] LIMA, S. M. L. Limitation of COTS Antiviruses: Issues, Controversies, and Problems of COTS Antiviruses. In: Maria Manuela Cruz-Cunha,Nuno Ricardo Mateus-Coelho. (Org.). Handbook of Research on Cyber Crime and Information Privacy. 1ed.: , 2020, v. 1, p. 396-413.
[18] MICROSOFT. Trustworthy Computing | 2013 Microsoft Computing Safety Index (MCSI) Worldwide Results Summary, 2013.
[19] REWEMA. REWEMA (Retrieval of 32-bit Windows Architecture Executables Applied to Malware Analysis). Disponível em: https://github.com/rewema. Acesso em março de 2020., 2019.
[20] SANS. SANS Institute InfoSec Reading Room. Out with The Old, In with The New: Replacing Traditional Antivirus, Disponível em: https://www.sans.org/reading-room/whitepapers/analyst/old-new-replacing-traditional-antivirus-37377. Acesso em junho de 2017, 2018.
[21] SANTOS, I. et al. Opcode Sequences as Representation of Executables for Data-Mining-based Unknown Malware Detection. Information Sciences 231 (2013) 64–82., 2013.
[22] SANTOS, M. M.; SILVA FILHO, A. G.; SANTOS, W. P. Deep convolutional extreme learning machines: Filters combination and error model validation. NEUROCOMPUTING, v. 329, p. 359-369, 2019.
[23] SANTOS, W. P. Mathematical Morphology In Digital Document Analysis and Processing, New York: Nova Science. Chapter 8, (2011) 159-192., 2011.
[24] TOFLER, A. The Third Wave., Editora: Bantam Books. ISBN 0-553-24698-4, 1981.
[25] XIANG, C.; DING, S. Q.; LEE, T. H. Geometrical interpretation and architecture selection of MLP. The IEEE Transactions on Neural Networks and Learning Systems. 2005 Jan;16(1):84-96., 2005.