Estudo de Métodos de Detecção de Ransomware Utilizando Inteligência Artificial

Carlos Eduardo Chagas Romar[1], **Rodrigo Cardoso Silva**[1]

[1]Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie
01.302-907 – São Paulo – SP – Brasil

caduromar@gmail.com, rodrigoc.silva@mackenzie.br

Abstract. *The present article investigates the contributions of research on detection of ransomware malware using artificial intelligence. The main motivations for this study are the destructive nature of ransomware, the difficulty of reversing a ransomware infection, and how important it is to detect it before it infects a system. Machine learning is coming to the forefront of combating ransomware, so we've tried to identify weaknesses in machine learning approaches and how they can be strengthened. The threat posed by ransomware is exceptionally high, with new variants and families continually being found on the internet and the "dark web". Recovering from ransomware infections is difficult given the nature of the encryption they use. Exploring machine learning and deep learning approaches when it comes to detecting ransomware is of great interest because machine learning and deep learning can detect "zero-day" threats. These techniques can generate predictive models that can learn ransomware behavior and use that knowledge to detect variants and families that have yet to be seen. In this research, we review prominent research studies showing machine or deep learning approaches to detecting ransomware to see if these techniques are viable.*

Resumo. *O presente artigo investiga as contribuições da pesquisa sobre a detecção de ransomware malware usando inteligência artificial. As principais motivações para isso estudo são a natureza destrutiva do ransomware, a dificuldade de reverter uma infecção por ransomware, e como é importante detectá-lo antes de infectar um sistema. O aprendizado de máquina está chegando na vanguarda do combate ao ransomware, então tentamos identificar pontos fracos na máquina abordagens de aprendizagem e como elas podem ser fortalecidas. A ameaça representada pelo ransomware é excepcionalmente alta, com novas variantes e famílias continuamente sendo encontradas na internet e na dark web. Recuperar-se de infecções por ransomware é difícil, dada a natureza da criptografia usada por eles. A exploração de abordagens de aprendizado de máquina e aprendizado profundo quando se trata de detectar ransomware é de grande interesse porque o aprendizado de máquina e o aprendizado profundo podem detectar ameaças de zero-day. Essas técnicas podem gerar modelos preditivos que podem aprender o comportamento de ransomware e usar esse conhecimento para detectar variantes e famílias que ainda não foram vistas. Nesta pesquisa, revisamos estudos de pesquisa proeminentes que mostram abordagens de aprendizado de máquina ou profundo para detectar ransomware com o intuito de analisar se essas técnicas são viáveis.*

1. Introdução

No mundo digital, onde toda informação é armazenada digitalmente, as informações podem ser acessadas a qualquer momento, podem ser acessadas via internet e facilmente, tudo é feito sem problemas com um clique, sem esforço e de forma eficiente. A digitalização ajudou a diminuir a criminalidade se aplicado no todo, fazendo as coisas facilmente e têm diminuir o trabalho de documentação. Mas ainda cria um problema de segurança para informações pessoais e confidenciais de um Individual, além de muitos roubos ou ataques cibernéticos.

Os principais tipos de malware são vírus, cavalo de troia (*trojan*), *ransomware*, *backdoor*, *worm*, *bot*, *spyware* e *rootkit* [CERT.br 2020]. Dentre eles o *ransomware* oferece uma ameaça em particular devido ao crescimento em sua popularidade. Isso se deve ao fato que os *hackers* só precisam encontrar uma pequena vulnerabilidade em um dispositivo para bloqueá-lo. Se esse dispositivo estiver conectado a uma rede, os *hackers* podem encontrar ainda mais dispositivos para infectar [Adkisson 2018].

O nome *ransomware* vem da palavra em inglês *ransom*, que significa resgate pago por algo ou alguém que foi sequestrado, nesse caso, os dados das vítimas, que ilustra perfeitamente a forma de ação do malware.

As táticas e técnicas de *ransomware* continuaram a evoluir, o que demonstra a crescente sofisticação tecnológica dos atores e uma crescente ameaça de ransomware para organizações em todo o mundo [FBI 2021]. Enquanto os ataques de *ransomware* contra corporações, governos e infraestruturas críticas as entradas estão crescendo rapidamente, os ataques contra consumidores individuais estão diminuindo. O primeiro ransomware da história surgiu em 1989. Era chamado de Trojan da AIDS, mas parece rudimentar hoje em dia. Ele se espalhou através de disquetes e envolveu o envio de 189 dólares para uma caixa postal no Panamá para pagar o resgate. Após isso houveram diversos ataques ao redor do mundo, mas o ataque WannaCry em 2017 foi o pior ataque antes visto. O Wannacry atacou muitos hospitais, empresas, organizações do governo e pelo menos 150 universidades, tendo um total de mais de 200.000 vítimas. Bloqueou todos os computadores e exigiu resgate [Mohurle and Patil 2017].

Não só os ataques se tornaram mais frequentes, como cada vez mais tornaram-se mais graves. Novas versões do malware aparecem com frequência sendo capacitadas para evitar métodos antivírus e de detecção de intrusão. Neste texto iremos apresentar soluções de detecção de ransomware utilizando inteligência artificial e verificar se são uma alternativa viável às já existentes.

2. Ransomware

Ransomware é um tipo de malware projetado para impedir ou reduzir o acesso que um usuário tem a seu dispositivo, sistema operacional ou arquivos. *Ransomware* é normalmente encontrado nas formas de *locker ransomware* e *crypto-ransomware*. *Locker ransomware* exibe uma tela de bloqueio que impede que a vítima acesse seus computadores, muitas vezes fingindo ser a aplicação da lei exigindo dinheiro pago em troca do acesso ao computador. O *crypto-ransomware* criptografa arquivos-chave no computador do usuário do sistema, usando esquemas de criptografia complexos e taxas de demanda, geralmente na forma de criptomoeda para descriptografar os arquivos da vítima. Em sua história,

o *ransomware* tornou-se mais proeminente, avançado, e destrutivo. A ascensão do ransomware é atribuída a muitos fatores diferentes desde que apareceu pela primeira vez em 1989. O surgimento do *ransomware* como serviço também aumentou a disponibilidade do *ransomware* para criminosos em potencial menos talentosos tecnicamente. Oferta CryptoLocker, CryptoWall e Locky este tipo de serviço com a variante CryptoWall, gerando mais de 320 milhões de dólares em receita durante sua vida útil [Groot 2022].

Relatórios do ano de 2021 indicam danos totais e lucros de *ransomware* chegando a US$ 49.2 milhões [FBI 2021]. Com o malware em constante evolução e novas versões de famílias de malware se comportando de maneira diferente de seus antecessores, as abordagens de detecção tradicionais terão mais dificuldade em detectá-los [Mirza Baig and Lindskog 2012].

2.1. O Ataque Ransomware

A estrutura de um ataque de *ransomware* segue a metodologia apresentada na Figura 1. Existem diversos vetores de infecção que normalmente realizam infecções de ransomware. Em primeiro lugar, o vetor mais proeminente sendo e-mails maliciosos, conhecidos como *Phishing*, a carga útil é entregue como um anexo de e-mail enviados por spam usando *botnets* e outros *hosts* comprometidos [Zimba 2017]. Os *exploit kits* são outro método proeminente de infecção. *Exploit kits* são pacotes de *software* que examinam um sistema em busca de vulnerabilidades com a intenção de infectá-lo com *software* malicioso [CyberPedia 2018]. Outro método proeminente de infecção são os *downloads drive-by*, nos quais as vítimas são atraídas para sites maliciosos que executam códigos maliciosos [Damien Warren Fernando and Chen 2020].

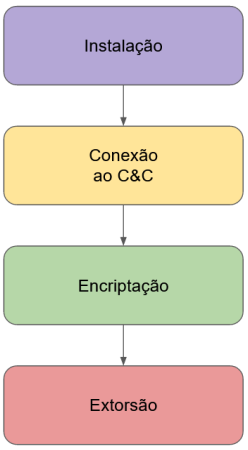

Figure 1. Metodologia do ataque ransomware

2.1.1. Instalação

A instalação ocorre depois que a carga foi despejada no sistema. Um método proeminente de instalação é o *download dropper* [Liska and Gallo 2017]. Essa abordagem usa um arquivo inicial que envolve o uso de um pequeno pedaço de código para evitar a detecção e chegar ao centro de comando e controle (C&C). Os autores de *ransomware* tentam dividir

a execução em diferentes scripts e processos para evitar a detecção baseada em assinatura realizada por Antivírus [Liska and Gallo 2017]. Quando uma organização é alvo de um ataque, o *ransomware* se espalha pela rede, determinando os locais de compartilhamento de arquivos e infectando-os para maximizar os danos e aumentar o possível resgate. Os executáveis não são executados até que o maior número de máquinas tenham sido infectadas.

2.1.2. Servidor de Comando e Controle

Depois que o ransomware é instalado em um sistema, ele chega a um centro de C&C em busca de instruções [Liska and Gallo 2017]. Os centros C&C respondem com um número variável de solicitações que dão as instruções ao ransomware sobre como proceder com a execução. Algumas variantes de ransomware relatam quantidades significativas de informações do sistema, o que pode dar aos invasores uma ideia de que tipo de sistema que atacaram e se vale a pena ir além de apenas um ataque de ransomware. o ransomware entra em contato com o centro de C&C para obter as chaves de criptografia após a instalação para garantir que as chaves sejam mantidas segredo [Sophos 2020]. É quase impossível descriptografar arquivos sem as chaves de descriptografia [Sophos 2020]. Centros de comando e controle se diferem de família para família de ransomware; alguns utilizam HTTP normal (Hypertext Transfer Protocol); alguns usam serviços complexos baseados em Tor para se conectar [Liska and Gallo 2017].

2.1.3. Encryption and Extortion

O ransomware moderno usa criptografia assimétrica, então o *ransomware* vem com uma chave pública RSA (Rivest–Shamir–Adleman), usada pelo *ransomware* para estabelecer um canal seguro para seu servidor de comando e controle [Tailor and Patel 2017]. Criptografia de chave pública significa que o conteúdo enviado entre o servidor e o cliente (sistema infectado) serão criptografados de forma que terceiros terão muita dificuldade em descriptografar. O fator chave neste processo é que a chave pública só pode descriptografar mensagens que foram criptografadas pela chave privada correspondente. Esta chave privada é mantida no servidor, que só os atacantes têm acesso, impossibilitando a recuperação da vítima. Variantes diferentes de *ransomware* criptografam arquivos de maneiras diferentes. Alguns utilizam métodos de criptografia simétrica, outros métodos de criptografia assimétrica. Métodos de criptografia simétricos geram uma chave localmente e encriptam arquivos usando essa chave, cuja principal vantagem é que utiliza poucos recursos do sistema, reduzindo as chances de serem detectados. As chaves assimétricas usaram uma chave pública que pode criptografar, mas o processo de descriptografia requer a chave privada correspondente, que é armazenada apenas no servidor de comando e controle.

3. Detecção de Ransomware

A detecção de *ransomware*, usando inteligência artificial, segue um padrão muito específico. Deve haver uma seleção de recursos, seja por meio de um recurso personalizado métodos de seleção ou algoritmos predeterminados; assim que isso for concluído e o conjunto de recursos ideal for encontrado, os dados, organizados por esses recursos, serão

inseridos no algoritmo de aprendizado profundo ou aprendizado de máquina de sua escolha. O algoritmo será treinado e depois testado. Algoritmos de aprendizagem permitem que o computador aprenda por si só. Os algoritmos de aprendizado operam de diferentes maneiras, sendo as principais as supervisionadas e as não supervisionadas. Algoritmos de aprendizado supervisionado exigirão um conjunto de treinamento, no caso de amostras de dados de *software* benigno e *ransomware*, para que o algoritmo possa aprender a identificar padrões que distinguem os dois um do outro. Os métodos de aprendizado não supervisionados são alimentados com conjuntos de dados que não são rotulados e tentarão encontrar padrões que possam construir modelos, para distinguir os tipos de dados [Damien Warren Fernando and Chen 2020].

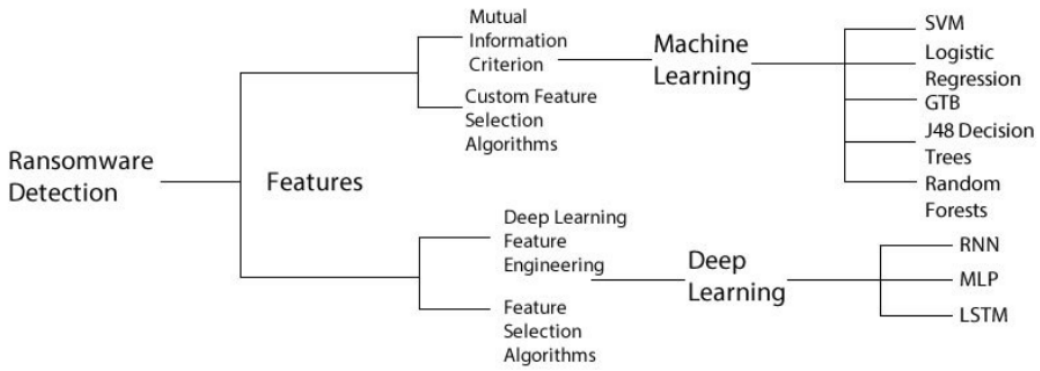

Figure 2. Taxonomia da detecção de ransomware
[Damien Warren Fernando and Chen 2020]

3.1. Dificuldades na implementação de Inteligência Artificial na detecção de ransomware

Vários desafios dificultam a detecção de ransomware. Em primeiro lugar, a ideia de usar métodos baseados em heurísticas parecem altamente arriscadas devido à velocidade com que o ransomware evolui. Abordagens que não utilizam da inteligência artificial não parecem apropriadas devido a uma variedade de *malware* que exibe a capacidade para evoluir e mudar, impossibilitando seu conhecimento até mesmo de algumas soluções baseadas em aprendizado de máquina [Damien Warren Fernando and Chen 2020].

A infecção por *ransomware* deve ser detectada precocemente porque, uma vez que os arquivos são criptografados, se torna quase impossível descriptografá-los sem pagar por uma chave de descriptografia. Caso optem por não pagar, as alternativas são recuperar o sistema com um *backup*, ou contar com o fato de que os desenvolvedores do ransomware cometeram algum erro ou utilizaram algum método que torna chaves acessíveis.

Como o *ransomware* é propagado por vários métodos, isso significa que a detecção precoce tem que levar em consideração os diferentes métodos de propagação que o ransomware provavelmente usará.

A previsão da evolução do Ransomware também será um desafio que se mostrará crítico quando detectar *ransomware* no futuro. É impossível prever exatamente quais novas técnicas serão implementadas no futuro.

A incorporação de IA (Inteligência Artificial) na defesa contra ataques *ransomware* fornece o maior desafio de todos. Com invasores usando técnicas semelhantes baseadas em IA para aqueles empregados por sistemas de defesa baseados em IA, a configuração de medidas defensivas terá que levar em conta vetores de ataque adaptáveis [Damien Warren Fernando and Chen 2020].

4. Estudos de Destecção de Ransomware com IA

4.1. EldeRan

O sistema EldeRan foi desenvolvido por um grupo de estudantes ingleses a partir da premissa observacional que *ransomware* executa ações que se diferenciam de *softwares* benignos. O sistema monitora ações realizadas pelo programa, como chamadas de API (*Application Programming Interface*) do Windows, operações de chave de registro, operações do sistema de arquivos, operações de diretório, o conjunto de operações feitas por extensão de arquivo, arquivos descartados e as *strings* do executável. Feita a coleta, os dados passam por um classificador de Regressão Logística Regularizado (*Regularised Logistic Regression classifier*) que retorna se o programa analisado é um *ransomware* ou um *goodwear*, caso o programa seja benigno.

O componente de seleção de características deste sistema usa o critério de informação mútua, que permite que as características mais discriminantes sejam obtidas. Os recursos usados são binários, portanto, é a presença ou ausência de um recurso que é utilizado como valor. O critério de informação mútua dá ao usuário a capacidade de quantificar a quantidade de discriminação que cada recurso adiciona ao classificador, dando ao sistema uma medida de quão dependente ou independente os recursos são para saber se um arquivo é *ransomware* ou benigno. Este conjunto de recursos é reduzido de um inicial lista de recursos de 30.967. De acordo com o critério de informação mútua, as características mais significativas no as 400 funcionalidades finais foram relacionadas a operações de chave de registro, com 48,25% das funcionalidades nas 100 finais sendo operações de chave de registro. A próxima categoria mais relevante são os recursos de estatísticas da API, que tornam até 24% das características finais. Os 24% restantes dos recursos somam menos de 10% individualmente. Os recursos adicionais consistem em diretórios percorridos, arquivos abertos, excluídos e modificados entre outros diretórios e atividades relacionadas a arquivos, que não são especificadas no trabalho de pesquisa [Daniele Sgandurra and Lupu 2016].

Os recursos são alimentados para o classificador de regressão logística regularizado para classificar os executáveis como sejam benignos ou maliciosos. A regressão logística é conhecida por ser eficaz na classificação quando há múltiplas variáveis a serem consideradas; no entanto, como o classificador usa 400 recursos, o modelo é muito vulnerável a problemas de *over-fitting*. O *over-fitting* foi reduzido usando uma função de regularização que atribui uma função de penalidade de custo a cada recurso que evitará o ajuste excessivo. A justificação do uso da regressão logística regularizada é que a regressão logística é mais fácil de treinar e adicionar novos samples em oposição a um método, como SVM, como Naïve Bayes, assumiram independência entre os recursos, mas a suposição feita ao tentar detectar *ransomware* é que há uma forte dependência entre as características. Devido ao alto volume de recursos no conjunto de dados, o algoritmo escolhido para classificação foi o algoritmo de Regressão Logística. O método visa modelar

a probabilidade log-posterior das diferentes classes dados os dados por meio de funções lineares dependendo das características [Daniele Sgandurra and Lupu 2016].

Os experimentos realizaram um comparativo com o antivírus VirusTotal. Os experimentos utilizaram um dataset de 942 softwares benignos e 582 amostras de *ransomware*. Nos testes, o sistema alcançou uma taxa de detecção de 96.34%, com uma taxa de falso positivo de 1.12%. O antivírus VirusTotal obteve uma taxa de detecção 96.89% e uma taxa de falso positivo de 0.66%. A fase final do experimento foi feita com categorias de *ransomware* desconhecidas, fora do dataset utilizado, simulando ameaças zero-day. Nesse teste o sistema teve uma taxa de detecção de 93.3%. [Daniele Sgandurra and Lupu 2016].

4.1.1. Pontos Positivos

O sistema alcançou taxas relativamente altas (96.34%) nas categorias de *ransomware* em que foi treinado e uma taxa de 93.3% em detecção de *ransomware* de categorias desconhecidas. Com sua análise de recursos estáticos e dinâmicas, o sistema consegue atingir taxas de detecção capazes de competir com sistemas de antivírus disponíveis no mercado [Damien Warren Fernando and Chen 2020].

4.1.2. Pontos Negativos

O principal problema do EldeRan, constatado pelos próprios criadores do sistema, é a dificuldade de detectar *ransomware* que permanecem dormentes por um período muito grande de tempo ou que dependem do input do usuário para sua ativação devido à limitações do ambiente de teste. Além disso, o sistema de regressão utilizado não é adequado para limites de decisão não lineares, o que significa que o modelo pode achar difícil encontrar relacionamentos complexos entre recursos [Damien Warren Fernando and Chen 2020].

4.2. RansomWall

O RansomWall utiliza um recurso de camadas para detectar ataques *ransomware* em tempo real. O sistema é montado em 5 camadas, sendo a primeira a camada de análise estática, que analisa o executável em um contexto estático, strings, por exemplo. A segunda camada é uma camada de captura que utiliza arquivos e diretórios falsos. Esses arquivos e diretórios são utilizados para que o *ransomware* ataque-os antes de outros arquivos do usuário, a análise do *ransomware* mostra que uma grande proporção deles usa uma abordagem de pesquisa em profundidade ao procurar arquivos para criptografar. A terceira camada é o mecanismo de análise dinâmica que fornece dados comportamentais para o executável. As duas camadas finais são a camada de backup e as camadas de aprendizado de máquina que lidam com o backup de arquivos se uma infecção por *ransomware* é detectada. A camada de aprendizado de máquina classifica as amostras com base no modelo, que é treinado offline.

A camada de aprendizado de máquina é composta por regressão logística, *support vector machines*, ANNs (Artificial Neural Networks), “random forests” e “gradient

tree boosting" . O aprendizado de máquina A camada é baseado em "Aprendizado Supervisionado Sequencial com Janela Deslizante de Média Móvel". A saída é benigno ou *ransomware*, portanto, é por isso que os classificadores são usados. O treinamento é feito *offline*, a execução do sistema ocorre em tempo real em que as camadas estática, dinâmica e trap envia dados para um coletor de recursos, que converte os dados no conjunto de recursos. Se um processo for marcado como suspeito, os valores dos recursos dos dados são enviados para a camada de aprendizado de máquina, que então processo usando a seleção de algoritmos para determinar se a amostra é benigna ou ransomware. O algoritmo exato para decidir quais recursos usar dentre os recursos fornecidos pelas três camadas não são especificados[Shaukat and Ribeiro 2018].

O Gradient Tree Boosting funciona em um sistema do tipo gradiente descendente no qual constrói modelos progressivamente. Primeiramente é construído um modelo simples, após o qual será calculado o erro residual para o modelo, e então um novo modelo será construído para tentar corrigir os erros do primeiro modelo. O algoritmo reconstrói continuamente o modelo para reduzir o erro no modelo anterior até que a previsão do modelo esteja em um nível aceitável. A função gradiente descendente tentará reduzir o gradiente para fechar a lacuna entre os valores reais e os valores previstos [Shaukat and Ribeiro 2018].

O sistema utilizou um conjunto de 574 amostras entre 12 categorias de *ransomware* criptográficos, e 442 *softwares* benignos em ambientes reais de usuários [Shaukat and Ribeiro 2018]. O resultado dos testes realizados é extremamente promissor, com uma taxa de detecção de 98.52% e uma taxa de falsos positivos de apenas 0.0056% [Damien Warren Fernando and Chen 2020].

4.2.1. Pontos Positivos

O RansomWall, por ser multi-camadas, possui uma grande vantagem. Suas camadas estáticas, dinâmicas e *honey pot* dão à esse sistema uma grande vantagem contra os demais. Além disso possui uma camada extra de proteção, a camada de backup realiza o backup dos arquivos de forma preventiva assim que um *ransomware* é detectado. Sua taxa de detecção é excelente, em 98.52% e uma quantidade muito baixa de falsos positivos, porém ainda existem certas desvantagens que devem ser levadas em consideração.

4.2.2. Pontos Negativos

Em termos de limitações, o sistema RansomWall usa um conjunto de dados de 574 ransomware amostras e 442 arquivos benignos. Por causa de como os arquivos benignos podem ser variados, levanta a questão como para saber se o sistema tem treinamento suficiente sobre o comportamento normal. Outra desvantagens do sistema é que, por ser *offline*, não analisa nenhum comportamento de rede, que pode ser uma das maneiras mais rápidas de se detectar um ataque.

4.3. NetConverse

NetConverse é uma análise de algoritmos de aprendizado de máquina em um conjunto de dados do Windows tráfego de rede *ransomware*. A pesquisa leva em consideração

o desenvolvimento de variantes de *ransomware* que estão evoluindo de forma a evitar a detecção de aprendizado de máquina. O conjunto de dados é composto por tráfego de rede baseado em conversação. Essa abordagem reconhece a análise dinâmica, as técnicas têm limitações e novas variantes de *ransomware* podem ser redesenhadas na tentativa de diminuir a taxa de detecção por algoritmos de aprendizado de máquina.

A seleção de recursos é feita usando o TShark, que gera dados estatísticos e calculados junto com extração de recursos estáticos. TShark é uma extensão do analisador de rede, Wireshark. Cada rede O arquivo PCAP (*Packet Capture*) é mesclado em um conjunto de dados, com base nos recursos extraídos do arquivo PCAP. Os recursos consistem no protocolo usado, na origem e no endereço de destino dos pacotes, e duração das ligações. Esta pesquisa tem uma abordagem diferente da maioria dos outros estudos que foram analisados nesta pesquisa. Eles contam com um programa pronto, o TShark, para fazer seu recurso extração em vez de usar um algoritmo de seleção de recursos [Damien Warren Fernando and Chen 2020].

Os dados retirados do TShark são executados por meio de vários algoritmos de aprendizado de máquina: bayesiano redes, MLP (*Multilayer Perceptron*), J48, KNN (*K-Nearest Neighbour*), *Random Forests* e LMT (*Logistic Model Tree*). A maior precisão alcançada foi pelo algoritmo J48, que alcançou uma taxa de precisão de 97,1%. Todos os experimentos foram realizados no WEKA com validação cruzada de 10 vezes abordagem usando todos os dez recursos extraídos. O algoritmo J48 atinge a maior taxa de precisão com uma taxa de falsos positivos muito baixa. Todos os dados usados nos experimentos são extraídos de máquinas virtuais executadas na estação de trabalho VMWare com todos os classificadores não sendo ajustados. O conjunto de dados é composto por 264 arquivos benignos e 210 exemplos de arquivos *ransomware* [Omar M. K. Alhawi 2018].

4.3.1. Pontos Positivos

O principal ponto positivo deste estudo é a sua taxa de detecção. Seu teste com diferentes algoritmos permite a escolha do algoritmo mais adequado para realizar a detecção. Além disso, sua ampla abordagem de recursos de rede permite que o sistema possa rodar em praticamente qualquer dispositivo sem grandes exigências do hardware.

4.3.2. Pontos Negativos

Embora atingir uma taxa de detecção de 97,1% seja impressionante, o conjunto de dados usado continha apenas 210 ransomware amostras e 264 arquivos benignos. O treinamento extremamente limitado que os modelos receberam em comportamento benigno provavelmente fará com que o modelo fique confuso quando for implantado no mundo real a menos que a expansão e o ajuste sejam realizados. A falta de detalhamento também reforça a afirmação dos autores que esta pesquisa atua como uma linha de base que outros pesquisadores podem construir, porque os algoritmos não receberam nenhum ajuste para permitir que sejam criados especificamente para ransomware.

4.4. API Sequence-Based Detection

Essa pesquisa propõe uma forma de detecção com uma abordagem diferente da maioria dos outros modelos propostos, que é a diferenciação de *ransomware* de outros *malwares*. Ela utiliza sequências de execuções de API que são convertidas em sequências de "n-grams". Essas sequências de "n-grams" são usadas para detectar os *ransomware* e diferenciá-los de outros *malwares* e *softwares* benignos. Cada elemento da entrada pode ser representado como "1" se um "*n-gram*" aparecer na sequência de "*n-gram*" ou como "0" se o "*n-gram*" não estiver presente na sequência [Damien Warren Fernando and Chen 2020].

Os recursos utilizados nesta pesquisa são chamadas de API do Windows. As sequências de chamadas de API de cada executável são reunidas e depois convertidas em "n-gramas". As sequências de "*n-grams*" são usadas para diferenciar entre malware, *ransomware* e software benigno. A extração das chamadas da API do Windows é feita pela ferramenta Intel Pin. Os valores de CF-NCF (Frequência fora da classe) são calculados para cada amostra, e esses valores atuam como os pesos de cada elemento [Seong Il Bae and Im 2019].

O CF-NCF atua como um indicador para modelos de classificação, baseados no *Term-Frequency-Inverse Document Frequency*. Esta técnica é utilizada para enfatizar as características de cada classe, dando assim maior capacidade de diferenciar entre *malware*, *ransomware* e arquivos benignos. CF-NCF calcula pesos em um elemento em uma classe, para fornecer uma maior precisão em experimentos de classificação [Seong Il Bae and Im 2019].

4.4.1. Pontos Positivos

Este estudo foca em distinguir *ransomware* de outros tipos de *malware*. O ideal em uma detecção é parar o ataque antes que ele ocorra e se o sistema consegue detectar o tipo de *malware*, ele consegue tomar medidas específicas para aquele tipo de ataque. Os resultados obtidos são bem promissores, principalmente levando em conta o foco na diferenciação do *ransomware* de outros malwares.

4.4.2. Pontos Negativos

Uma desvantagem que essa pesquisa apresenta é a falta de consideração na evolução dos ransomware no futuro. Outro problema é que o dataset apresenta uma quantidade baixa de *softwares* benignos, com apenas 300 amostras, comparado à 1900 amostras de *malwares*.

5. Conclusão

Em termos das direções que a detecção de *ransomware* está tomando, há evidências convincentes de que *machine learning* e *deep lerning* desempenharão um papel fundamental no futuro da detecção de *ransomware*[Damien Warren Fernando and Chen 2020]. A julgar pelos estudos pesquisados, a precisão e as taxas de detecção fornecidas por essas abordagens são muito difíceis de contestar. A forma como a inteligência artificial é usada para a detecção precisa se adaptar à medida que o *ransomware* evolui, tendo de ser treinadas

para aprender e antecipar novas tendências de *ransomware* que aparecerão no futuro. A evolução do *ransomware* pode tentar evitar ou enganar as técnicas de *machine learning* com aprendizagem adversária. É essencial que os modelos de *machine learning* e *deep learning* sejam ajustados e modificados para levar em consideração cepas polimórficas e híbridas de *ransomware*.

Seria útil usar recursos de rede, dados estáticos e comportamentais para ter uma visão abrangente do método de detecção ao invés de depender apenas de recursos de rede, estáticos ou comportamentais separadamente. Seria aconselhável ter sistemas com vários componentes de Machine Learning ou Deep Learning, cada um lidando com um aspecto do processo de detecção. Por exemplo, um algoritmo lidaria com dados comportamentais, outro lidaria com o hardware e um terceiro algoritmo lidaria com a rede de dados.

Uma lacuna na maioria dessas técnicas de pesquisa é o fato da detecção parecer depender da execução do *ransomware*, esses métodos ainda levam à imposição de algum dano ao sistema ao confiar em padrões comportamentais, como frequências de leitura/gravação e padrões de acesso a arquivos [Damien Warren Fernando and Chen 2020]. A maioria dos estudos revisados não abordam diretamente a detecção precoce, e os estudos que abordam não fornecem uma explicação do porque o período de tempo de detecção escolhido é ideal para *ransomware*. É importante detectar o *ransomware* antes que eles criptografem arquivos; portanto, os pesquisadores precisam declarar por quanto tempo suas amostras de teste e treinamento levaram em média, para iniciar a criptografia de arquivos. O aspecto referente ao tempo precisa ser analisado em profundidade para saber quando um *ransomware* precisa ser interrompido para preservar arquivos.

Foram revisadas pesquisas que usam inteligência artificial para a detecção de *ransomware*. Em geral, as abordagens revisadas apresentam altas taxas de detecção acima dos 90%. Esses modelos são todos treinados em uma mistura de recursos de rede, comportamentais ou estáticos. Enquanto a maioria é sistemas conceituais, como RansomWall e o sistema EldeRan, alguns foram testados por meio de implantação. Os resultados alcançados nos dão confiança de que os modelos de machine e deep learning podem ser implantados para detectar *ransomware*. No entanto, sua capacidade de resistir ao teste do tempo e evoluir com a rápida evolução do *ransomware* é discutível.

No geral, acredito que o *ransomware* continuará a evoluir e se tornará mais desafiador para os sistemas de detecção inteligente. A evolução do *ransomware* e o avanço dos mecanismos de evasão de detecção seguem o padrão geral da evolução do *malware*. É importante reconhecer que a detecção baseada em assinatura já foi o método de detecção mais comumente usado e como ele se tornou obsoleto ao longo do tempo. Os criadores de sistemas de detecção inteligentes que usam sistemas de inteligência artificial terão que criar maneiras de que esses sistemas possam resistir ao teste do tempo e não serem deixados para trás, como abordagens heurísticas; estudos que avaliamos neste artigo abordam a adaptação à mudança, mas acredito que seja um aspecto de detecção que não é pesquisado o suficiente.

References

Adkisson, T. (2018). Why ransomware is the most popular tool for cyberattacks. Newsy.

CERT.br (2020). Cartilha de segurança para interne. In *Códigos Maliciosos*.

CyberPedia (2018). What is an exploit kit? CyberPedia.

Damien Warren Fernando, N. K. and Chen, T. (2020). A study on the evolution of ransomware detection using machine learning and deep learning techniques.

Daniele Sgandurra, Luis Muñoz-González, R. M. and Lupu, E. C. (2016). Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. Department of Computing, Imperial College London.

FBI (2021). Internet crime report 2021.

Groot, J. D. (2022). A history of ransomware attacks. In *The Biggest and Worst Ransomware Attacks of All Time*. Digital Guardian.

Liska, A. and Gallo, T. (2017). Ransomware. In *Defending Against Digital Extortion*. O'Reilly Media.

Mirza Baig, Pavol Zavarsky, R. R. and Lindskog, D. (2012). The study of evasion of packed pe from static detection. World Congress on Internet Security.

Mohurle, S. and Patil, M. (2017). A brief study of wannacry threat: Ransomware attack 2017. International Journal of Advanced Research in Computer Science.

Omar M. K. Alhawi, James Baldwin, A. D. (2018). Leveraging machine learning techniques for windows ransomware network traffic detection. University of Salford.

Seong Il Bae, G. B. L. and Im, E. G. (2019). Ransomware detection using machine learning algorithms. Hanyang University.

Shaukat, S. K. and Ribeiro, V. J. (2018). Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. Indian Institute of Technology.

Sophos (2020). Ransomware. In *How an attack works*. Sophos.

Tailor, J. P. and Patel, A. D. (2017). A comprehensive survey. In *Ransomware Attacks Prevention, Monitoring and Damage Control*. International Journal of Research and Scientific Innovation.

Zimba, A. (2017). Malware-free intrusion. In *A Novel Approach to Ransomware Infection Vectors*. International Journal of Computer Science and Information Security.