Uso do TF-IDF na Comparação de Dados para Detecção de Ransomware

Augusto Parisot [1], **Lucila M. S. Bento** [2], **Raphael C. S. Machado** [1]

[1] Instituto de Computação – Universidade Federal Fluminense (UFF)
Av. Gal. Milton Tavares de Souza, s/n, São Domingos, Niterói, RJ

[2]Instituto de Matemática e Estatística – Universidade do Estado do Rio de Janeiro (UERJ)
Rua São Francisco Xavier, 524 - 6o andar, Maracanã, Rio de Janeiro, RJ

aparisot@id.uff.br, lucila.bento@ime.uerj.br, raphaelmachado@ic.uff.br

Abstract. *Ransomware attacks represent one of the most significant cyber threats faced by users and organizations worldwide. This paper employs the TF-IDF technique, widely used in natural language processing, to analyze data from dynamic analysis reports generated by the Cuckoo Sandbox. We compared various types of data to determine which are most effective in detecting this threat. In our evaluation, we explored preprocessing methods alongside classic machine learning algorithms. The results indicate that Random Forest and SVM, when processing String data with StandardScaler, achieved accuracies of up to 98%, proving to be the most effective approaches.*

Resumo. *Os ataques de ransomware representam uma das maiores ameaças cibernéticas enfrentadas por usuários e organizações em todo o mundo. Este artigo emprega a técnica TF-IDF, amplamente usada em processamento de linguagem natural, para processar dados de relatórios de análise dinâmica gerados pelo Cuckoo Sandbox. Comparamos diferentes tipos de dados, a fim de revelar quais podem ser usados com maior eficácia na detecção dessa ameaça. Para a avaliação, investigamos métodos de pré-processamento junto com algoritmos de aprendizado de máquina clássicos. Os resultados indicam que Random Forest e SVM, ao processarem dados de String com StandardScaler, alcançaram acurácia de até 98%, destacando-se como as abordagens mais eficazes.*

1. Introdução

Os sistemas operacionais Windows, amplamente utilizados em contextos que variam desde ambientes corporativos até pessoais, são frequentemente alvos de ações maliciosas, especialmente ataques de ransomware. Esses ataques não só criptografam dados críticos mas também ameaçam expô-los — uma prática conhecida como dupla extorsão [Kaspersky 2021]—, demandando resgates financeiros significativos e evoluindo constantemente para desafiar as estratégias de detecção tradicionais. Adicionalmente, o ransomware também apresenta impacto na sociedade, por meio da interrupção de atividades essenciais, como serviços públicos, saúde ou segurança.

Dados recentes apontam para um aumento preocupante na frequência e na sofisticação dos ataques de ransomware direcionados a sistemas Windows, causando perdas financeiras e operacionais significativas para as organizações afetadas. Em 2023,

tais ataques custaram às organizações uma média de US$ 4,45 milhões em danos, com projeções apontando para um aumento até o final de 2024 [IBMSecurity 2023a]. Além disso, estima-se que os danos globais causados por ransomwares superarão US$ 265 bilhões até 2031, com expectativa de um novo ataque ocorrendo a cada 2 segundos [Razaulla et al. 2023].

Em resposta a este cenário desafiador, a comunidade científica e o mercado intensificaram seus esforços para desenvolver métodos eficazes de detecção de malware, focando principalmente nas técnicas de análise dinâmica [Mohanta and Saldanha 2020]. Esse tipo de análise é realizada em ambientes controlados como o Cuckoo Sandbox [Guarnieri et al. 2012], permitindo observação a execução do ransomware e a coleta de dados característicos ao seu comportamento e fundamentais para identificar e mitigar ataques em tempo real, como chamadas de API, tráfego de rede, informações sobre strings presentes no código do malware e/ou capturadas durante sua execução e assinaturas conhecidas de comportamentos maliciosos.

Este estudo utiliza a técnica de Frequência de Termo-Inverso da Frequência do Documento (TF-IDF, do inglês *Term Frequency–Inverse Document Frequency*), reconhecida na literatura por sua eficácia na análise textual [Prachi. et al. 2023, Cen et al. 2024], para avaliar dados obtidos em relatórios de análise dinâmica. Aplicamos o TF-IDF para analisar e comparar diferentes seções de relatórios produzidos pelo Cuckoo Sandbox, identificando quais delas são mais eficazes na detecção de atividades maliciosas de ransomware em sistemas Windows. Avaliamos famílias de ransomware que estiveram envolvidas em um grande número de incidentes cibernéticos nos últimos anos [IBMSecurity 2023b, IBMSecurity 2024, Horowitz 2023, Team 2023]. Além disso, utilizamos métodos de pré-processamento, como StandardScaler e PCA, e testamos seis algoritmos de aprendizado de máquina (ML, do inglês *Machine Learning*) amplamente adotados para detecção de malware [Singh and Singh 2021, Chang et al. 2022, Begovic et al. 2023, Cen et al. 2024], para avaliar a eficácia de cada seção analisada.

As principais contribuições deste artigo incluem:

- ✓ Demonstração da aplicação eficaz do TF-IDF na análise de dados de análise dinâmica.
- ✓ Desenvolvimento e disponibilização de conjuntos de dados contendo informações derivadas de chamadas de API, atividades de rede, assinatura e informações sobre string.
- ✓ Comparação detalhada das seções dos relatórios do Cuckoo Sandbox, identificando as mais eficazes para a detecção de ransomware.
- ✓ Avaliação de seis algoritmos de aprendizado de máquina, proporcionando insights sobre a melhor abordagem para a detecção de ransomware.
- ✓ Disponibilização das ferramentas desenvolvidas.

O artigo está organizado da seguinte forma: a Seção 2 explora trabalhos relacionados; a Seção 3 detalha a metodologia utilizada, incluindo a configuração do sandbox e o processamento de dados; a Seção 4 apresenta os resultados dos experimentos, destacando o desempenho comparativo das seções dos relatórios na detecção de ransomware; e, finalmente, a Seção 5 discute as implicações dos resultados, identifica limitações e propõe direções futuras para a pesquisa.

2. Trabalhos Relacionados

As campanhas de ransomware têm se mostrado altamente lucrativas para os cibercriminosos, impulsionando uma evolução contínua e proliferação de variantes maliciosas, principalmente para a plataforma Windows [Kaspersrky 2021]. Para enfrentar esse desafio crescente, o mercado e a comunidade acadêmica constantemente trabalham na proposta de sistemas de análise inteligentes que exigem mínima intervenção humana, muitas vezes empregando uma combinação de técnicas como o TF-IDF para a detecção de ransomware [Begovic et al. 2023, Cen et al. 2024].

Diversas metodologias têm sido exploradas para extrair e transformar características de malwares em conjuntos de dados utilizáveis. Como exemplo pode ser citado o trabalho de Al et al. [Al-rimy et al. 2019], no qual os autores transformaram chamadas de API em documentos, aplicando TF-IDF para classificar os ransomwares de maneira precoce. Outro trabalho interessante é o de Zhang et al. [Zhang et al. 2019], onde sequências de N-gramas de códigos de operação de malwares foram analisadas usando TF-IDF, ilustrando como os algoritmos de ML podem ser empregados subsequentemente na classificação. Similarmente, Dinh et al. [Dinh et al. 2019] extraíram dados dos relatórios do Cuckoo Sandbox para construir conjuntos de dados personalizados a partir de seções específicas dos relatórios.

Chen et al. [Chen et al. 2019] propuseram um sistema automatizado que utiliza TF-IDF junto com a Análise Discriminante Linear de Fisher e Árvores Extremamente Aleatorizadas para extrair padrões de malware e facilitar a detecção precoce. Essa abordagem é utilizada para analisar amostras de malware recém-descobertas dentro de ambientes controlados para gerar relatórios de análise dinâmica, extraindo e classificando automaticamente características discriminativas indicativas de atividades maliciosas.

Qin et al. [Qin et al. 2021], foi introduzido um novo método para extração de características usando TF-IDF em sequências de chamadas de API capturadas pelo Cuckoo Sandbox. O método proposto combina características a nível de documento e categoria. Seus resultados mostram um desempenho superior em modelos de ML, como regressão logística e classificação de vetor de suporte, comparado ao uso isolado do TF-IDF.

Zhang et al. [Zhang et al. 2023] propuseram um sistema que coleta sequências de chamadas de API na fase pré-cifragem do ransomware, transformando-as em vetores de características usando TF-IDF e treinando modelos de ML para detecção precoce de ransomware. Eles também desenvolveram uma Ontologia de Contramedidas de Defesa contra Ransomware, que permite deduzir automaticamente contramedidas de defesa.

Dabas et al. [Dabas et al. 2023] apresentaram um método de detecção de malware, focando na extração de informações de chamadas de API em três formas: uso, frequência e sequência de chamadas de API. Esses conjuntos de características foram enriquecidos com TF-IDF e usados para treinamento e teste de modelos de ML. Seus experimentos com vários algoritmos de ML revelam que a precisão de detecção usando o conjunto de características integradas da API supera conjuntos de características isoladas para todos os algoritmos testados.

Kim et al. [Kim and Kim 2024] enfrentaram o desafio de padronizar tamanhos variáveis de sequências de chamadas de API. Os autores utilizaram a técnica TF-IDF no lugar de recorrer ao comum *padding* com zeros, que muitas vezes é impraticável para

grandes discrepâncias de dados. O método proposto transforma os dados em um tamanho fixo, elevando os valores das sequências de chamadas de API benignas e maliciosas que são significativas, enquanto diminui os valores das que são irrelevantes. Essa técnica de pré-processamento mostrou bons resultados, alcançando maior precisão e menores taxas de falsos positivos quando comparada com outras abordagens tradicionais na literatura.

Neste estudo, expandimos os fundamentos estabelecidos para empregar a técnica de TF-IDF não apenas na classificação de chamadas de API, mas também na integração de dados de assinatura, comunicação de rede e strings extraídas de amostras de malware. Esse enfoque multidimensional visa aprimorar nossa capacidade de detecção, permitindo identificar quais tipos de dados são mais eficazes para detectar ataques de ransomware. Buscamos determinar qual combinação de ferramental técnico e tipo de dados oferece o melhor desempenho em termos de precisão e eficiência na detecção.

3. Métodos e Ferramentas

A Figura 2 mostra as duas etapas empregadas, a saber, extração de características e construção do modelo. A primeira etapa começa com a coleta das amostras de ransomware e sua submissão à máquina do Cuckoo Sandbox, que realiza a geração dos relatórios JSON com os dados de comportamento do código malicioso. Os dados presentes nesses relatórios correspondem ao conjunto de dados considerado no presente estudo. Os dados extraídos dos relatórios são processados utilizando a técnica de TF-IDF para selecionar características significativas para a detecção de ransomware. Após esta seleção, os dados são categorizados como benignos ou maliciosos, formando o conjunto de dados que é dividido em dados de treino e de teste. Utilizamos os dados de treino para ajustar seis modelos de ML, reconhecidos na literatura por sua eficácia na classificação de malware [Faceli et al. 2021, Maniriho et al. 2024a, Benmalek 2024]. Após o treinamento, a eficácia dos modelos é então avaliada através de testes com o conjunto de teste para verificar sua precisão na classificação dos ransomwares.

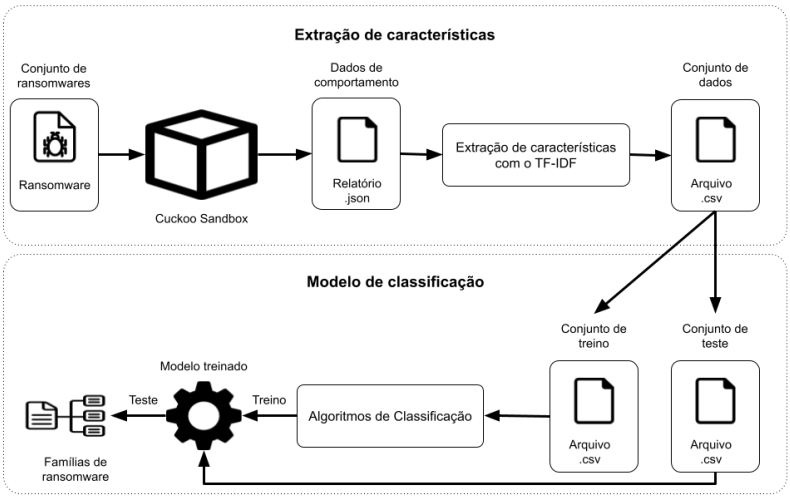


Figura 1. Visão geral da abordagem adotada

As amostras de ransomware analisadas neste estudo – pertencentes as famílias REvil, Ryuk, LockBit, Conti, Clop, Egregor, NetWalker e MountLocker – foram selecionadas devido ao seu impacto significativo em ataques cibernéticos recen-

tes [IBMSecurity 2023b, IBMSecurity 2024, Horowitz 2023, Team 2023]. Obtivemos essas amostras dos repositórios VirusTotal, VirusShare, Malware Bazaar e Hybrid-Analysis, usando scripts Python para recuperar as amostras com base em seus hashes SHA256. Para formação do conjunto de amostras benignas, foram utilizados 100 programas de uso comum, dentre eles aplicações nativas do Windows, como bloco de notas, visualizador de imagens e calculadora e outros aplicativos de uso diário, como editores de texto, VNC, WinRAR e WinZIP. A quantidade de amostras maliciosas coletadas é apresentada na Tabela 1.

Tabela 1. Quantidade de amostras por família

Ryuk	Revil	NetWalker	MountLocker	LockBit	Egregor	Conti	Clop	Total
52	629	78	17	49	45	104	15	989

Para replicar as condições do mundo real, ocultamos a natureza virtual do ambiente de análise por meio do uso das ferramentas Paranoid Fish e INETSIM, para ocultar rastros de virtualização e simular serviços de internet, respectivamente. Também populamos o ambiente de análise com histórico de uso, arquivos de usuários e programas comuns usados no dia a dia, como editores de texto, navegadores, reprodutores de vídeo e compactadores. A máquina *host* executava Ubuntu 20.04.4 LTS (64 bits) em um CPU Intel Core i7-8550U com 16GB de RAM DDR4, enquanto a VM executava Windows 7 Ultimate SP1 x64 no VirtualBox versão 6.1.32, alocado com 4 núcleos e 4GB de RAM. Para obter os dados relacionados ao comportamento dos malwares, todas as ações realizadas pelos ransomwares no Cuckoo Sandbox foram monitoradas durante 600 segundos, seguindo a recomendação de [Black et al. 2020].

3.1. TF-IDF

O acrônimo TF-IDF representa *Term Frequency [Luhn 1958] - Inverse Document Frequency [Jones 1972]*, uma técnica comum em Processamento de Linguagem Natural para identificar palavras importantes dentro de um documento. Esta técnica é baseada na teoria de modelagem de linguagem e tem como premissa que palavras que aparecem frequentemente em muitos documentos não são boas discriminadoras, e, portanto, devem ter um peso reduzido em tarefas de classificação ou busca [Luhn 1958, Jones 1972, Zhang et al. 2019].

O cálculo do TF-IDF é realizado em duas etapas, descritas pelas Equações 1 e 2.

$$TF(i,j) = \frac{n(i,j)}{\sum_k n(k,j)} \tag{1}$$

onde $TF(i,j)$ é a frequência do termo i no documento j, $n(i,j)$ é o número de ocorrências do termo i no documento j, e $\sum_k n(k,j)$ é o número total de termos no documento j.

$$IDF(i) = \log\left(\frac{|D|}{|\{j : i \in j\}|}\right), \tag{2}$$

onde $|D|$ é o número total de documentos no corpus e $|\{j : i \in j\}|$ é o número de documentos que contêm o termo i. Essa medida ajuda a determinar se um termo é comum ou raro entre todos os documentos.

$$TF \cdot IDF(i,j) = TF(i,j) \times IDF(i), \tag{3}$$

A Equação 3 combina as duas medidas anteriores para calcular o peso de cada termo no documento j, ponderando a frequência do termo pela sua raridade no conjunto de termos. O valor TF-IDF alto indica uma alta importância do termo no documento específico em relação ao conjunto de todos os termos.

O uso do TF-IDF na detecção de ataques é vantajoso por sua simplicidade de ser calculado e eficácia em diferenciar termos úteis para análise. No entanto, uma limitação que não pode deixar de ser mencionada é que ele não capta o contexto ou a semântica dos termos, tratando cada palavra de forma isolada. Além disso, esta abordagem pode levar a um modelo esparso e de alta dimensionalidade, especialmente com um vocabulário extenso como é o caso dos relatórios do Cuckoo Sandbox, tornando-se custoso em termos de memória e processamento [Vajjala et al. 2020]. Embora essas limitações estejam presentes, a técnica pôde ser aplicada sem obstáculos no escopo do presente estudo.

3.2. Análise de comportamento dos ransomwares

A análise comportamental dos ransomwares, especialmente utilizando os dados extraídos de relatórios JSON gerados pelo Cuckoo Sandbox, revela *insights* cruciais sobre a natureza dessas ameaças. No entanto, além dos dados importantes para identificação de ações maliciosas listados anteriormente, os relatórios JSON contêm ruídos, como colchetes e legendas, que não contribuem para a análise de segurança e devem ser filtrados durante a extração de recursos, o que pode ser feito trivialmente pelo TF-IDF.

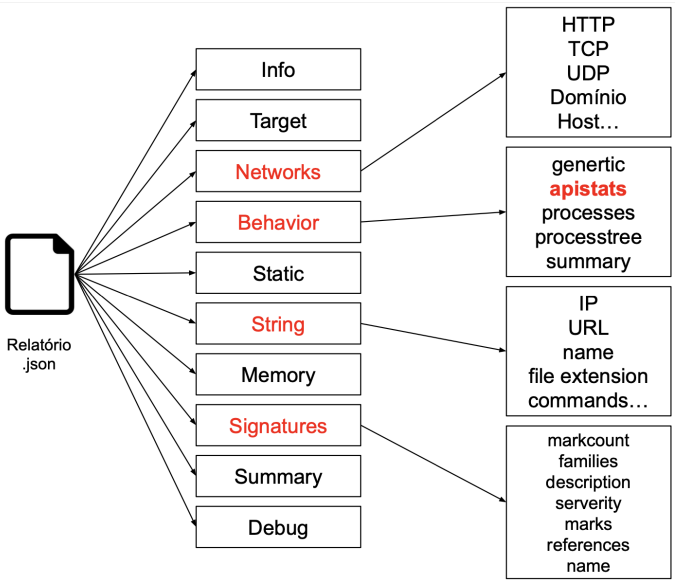


Figura 2. Seções típicas dos relatórios JSON do Cuckoo Sandbox

A Figura 2 ilustra as seções típicas encontradas nos relatórios JSON do Cuckoo Sandbox, das quais extraímos quatro principais categorias de características para análise:

Network (Rede): Os ransomwares frequentemente se comunicam via Internet. Com essa seção é possível obter informações sobre o monitoramento e análise dos protocolos de rede utilizados, focando em protocolos comuns como HTTP, TCP, UDP, ICMP e SMTP.

Behavior (Comportamento): Sumariza as ações executadas pelo malware, com foco na subseção ”apistats”que documenta as chamadas de API realizadas, proporcionando uma visão quantitativa da frequência de cada chamada.

Signatures (Assinaturas): Essa seção lista as assinaturas que foram acionadas durante a análise, refletindo comportamentos típicos de malwares conhecidos.

String: Aqui são listadas todas as strings extraídas do arquivo sob análise, sendo úteis para identificar URLs suspeitas, nomes e caminhos de arquivos, strings presentes no código do ransomware e extraídas da memória, entre outras informações.

O valor do TF-IDF de cada termo é diretamente proporcional à sua frequência no relatório e inversamente proporcional à frequência de sua ocorrência em todos os relatórios, ajudando a destacar termos que são unicamente relevantes para certas famílias de ransomware. Iniciamos a extração dos dados transformando os valores associados às chaves de cada relatório em texto, que foi processado para calcular os valores de TF e IDF, e posteriormente armazenados em um *Dataframe* do Pandas. Este processo inclui a remoção de palavras comuns e a normalização do texto antes da aplicação do TF-IDF. As características mais relevantes foram então selecionados usando as técnicas StandardScaler [Freeman and Chio 2018] para padronização e Análise de Componentes Principais (PCA) [Wold et al. 1987] para redução de dimensionalidade, a fim de otimizar a detecção dos ransomwares.

Após a extração e o pré-processamento dos dados, procedemos com a aplicação dos algoritmos de ML para classificar as amostras como benignas ou maliciosas. Os algoritmos utilizados foram escolhidos por serem bem estabelecidos na literatura. Os algoritmos utilizados são Árvore de Decisão (DT) [Freeman and Chio 2018], Floresta Aleatória (RF) [Freeman and Chio 2018], K-Vizinhos Mais Próximos (KNN) [Freeman and Chio 2018], Naive Bayes (NB) [Freeman and Chio 2018], Máquinas de Vetores de Suporte (SVM) [Freeman and Chio 2018] e Multilayer Perceptron (MLP) [Vang-Mata 2020]. Esses algoritmos foram escolhidos por sua eficácia comprovada na detecção de padrões complexos em dados de alta dimensionalidade, típicos em análises voltadas a deteção de malwares [Maniriho et al. 2024b, Maniriho et al. 2024a, Benmalek 2024]. A implementação foi realizada usando a biblioteca SciKit-Learn[1], que oferece ferramentas robustas e eficientes para construção e validação de modelos de aprendizado de máquina.

Os scripts utilizados para seleção e download das amostras, processamento e classificação estão disponíveis em [*endereço ocultado devido a identificação dos autores*]. Os dados preparados e as características extraídas estão disponíveis em [*endereço ocultado devido a identificação dos autores*]para o acesso e a replicação do presente estudo.

3.3. Critérios de avaliação

A avaliação dos classificadores foi realizada utilizando o conjunto de dados construído, conforme descrito na Seção 3.2. As métricas Precision, Recall, F1-Score e Accuracy, comumente utilizadas em análises de classificação binária, foram empregadas para mensurar o desempenho dos modelos em identificar amostras como benignas ou maliciosas.

[1]`https://scikit-learn.org/stable/`

Precision: Esta métrica mede a exatidão das previsões positivas feitas pelo modelo, sendo calculada como a razão entre os verdadeiros positivos (VP) e a soma dos verdadeiros positivos e falsos positivos (FP):

$$Precision = \frac{VP}{VP + FP}$$

Recall: Mede a capacidade do modelo de identificar corretamente todas as instâncias positivas reais, calculada como a razão entre os verdadeiros positivos e a soma dos verdadeiros positivos e falsos negativos (FN):

$$Recall = \frac{VP}{VP + FN}$$

F1-Score: Fornece uma média harmônica entre Precision e Recall, refletindo um equilíbrio entre essas métricas. É particularmente útil em situações onde é importante ter um balanço entre Precision e Recall:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Accuracy (Acurácia): Mede a proporção total de previsões corretas (incluindo verdadeiros positivos e verdadeiros negativos) em relação ao total de casos examinados. É uma métrica geral de desempenho que indica a eficácia global do modelo em todas as classificações:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

Essas métricas são fundamentais para avaliar a eficácia dos modelos de detecção de ransomware. Priorizamos uma alta Accuracy para refletir efetivamente a performance global do modelo. Além disso, um alto Recall é essencial para assegurar que todas as instâncias maliciosas sejam detectadas, minimizando a ocorrência de falsos negativos, enquanto uma alta Precision reduz o número de falsos positivos, crucial para a aplicabilidade prática do modelo em cenários reais.

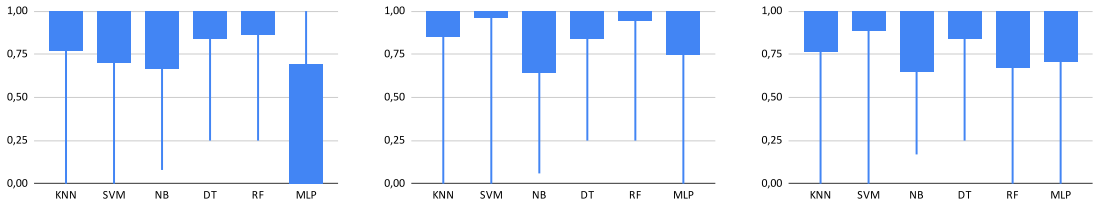
4. Resultados e Discussão

Esta seção apresenta os resultados dos experimentos focados em classificar amostras como benignas ou maliciosas, explorando qual tipo de dado extraído pelo Cuckoo Sandbox fornece melhores *insights* para detecção. Os experimentos foram conduzidos usando Python versão 3.9 na máquina *host* descrita na Seção 3. Realizamos a classificação binária, onde cada família de ransomware foi avaliada individualmente. Os experimentos consideraram duas divisões de dados: 70:30 (onde 70% dos dados foram usados para treinamento e 30% para teste) e 50:50 (onde 50% dos dados foram usados para treinamento e 50% para teste). As avaliações foram realizadas em três cenários: sem otimização (Normal), com aplicação do *StandardScaler* para normalização dos dados e PCA com $n = 100$ para redução de dimensionalidade.

No que se refere aos dados, cada ransomware é representado no conjunto de dados sem otimização por um vetor com diferentes quantidades de características, que dependem da seção que está sendo analisada: 332.480 características de Chamadas de API, 413

características de Assinatura, 9.676 características de Rede e 140.433 características de String.

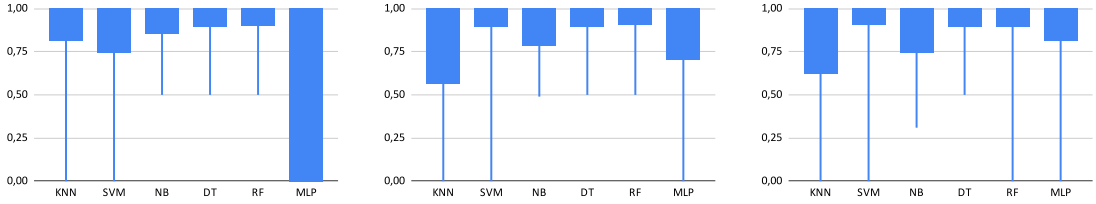
Os gráficos das Figuras 3 e 4 mostram a distribuição das métricas Precision e Recall para os diferentes algoritmos e métodos de pré-processamento, considerando a divisão dos dados em 70:30.



(a) Normal (b) StandardScaler (c) PCA

Figura 3. Distribuição da métrica Precision para os algoritmos e métodos de pré-processamento (70:30)

A métrica Precision indicam que os algoritmos DT, RF e SVM possuem grande potencial para a detecção de ransomware por apresentarem uma baixa taxa de falsos positivos. Em contraste, os algoritmos KNN e MLP mostram alta variabilidade, especialmente quando não são aplicados métodos de pré-processamento, indicando que a normalização dos dados com o StandardScaler e a redução de dimensionalidade com o PCA possuem um papel importante na precisão desses algoritmos e, consequentemente, na diminuição de falsos positivos. Situação similar também foi observada para a métrica Precision ao considerar a divisão dos dados em 50:50.



(a) Normal (b) StandardScaler (c) PCA

Figura 4. Distribuição da métrica Recall para os algoritmos e métodos de pré-processamento (70:30)

A distribuição de Recall apresentada na Figura 4 também aponta o potencial dos algoritmos DT, RF e SVM, já que apresentam baixas taxas de falsos negativos. Na divisão dos dados em 50:50, podemos observar algo semelhante.

Os mapas de calor nas Figuras 5 e 6 ilustram o desempenho dos algoritmos, mostrando as variações de Precision (Pr), Recall (Re) e F1-Score (F1) por tipo de informação e método de pré-processamento, onde as variações no desempenho são destacadas.

Os resultados para a divisão de dados 70:30, mostrados na Figura 5, destacam o algoritmo RF como tendo o melhor desempenho médio geral, mantendo altas métricas de Precision, Recall e F1-Score para os diferentes tipos de dados e pré-processamentos, especialmente com dados de Assinatura e String. Os dados de String se destacam ao

Algoritmo	Normal			StandardScaler			PCA			Normal			StandardScaler			PCA		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
	Assinatura									Rede								
KNN	0,85	0,83	0,84	0,95	0,90	0,91	0,95	0,90	0,92	0,85	0,90	0,85	0,91	0,81	0,83	0,86	0,83	0,83
SVM	0,88	0,88	0,82	0,86	0,79	0,81	0,94	0,85	0,88	0,91	0,91	0,89	0,87	0,92	0,87	0,91	0,93	0,89
NB	0,76	0,93	0,80	0,69	0,91	0,72	0,77	0,81	0,74	0,67	0,85	0,69	0,66	0,85	0,68	0,70	0,70	0,64
DT	0,93	0,85	0,88	0,93	0,85	0,88	0,93	0,85	0,88	0,86	0,88	0,85	0,86	0,88	0,85	0,86	0,88	0,85
RF	0,94	0,92	0,91	0,94	0,93	0,91	0,87	0,93	0,89	0,87	0,90	0,84	0,88	0,92	0,88	0,68	0,83	0,73
MLP	0,43	0,49	0,49	0,74	0,80	0,77	0,92	0,93	0,91	0,39	0,39	0,32	0,89	0,90	0,88	0,86	0,88	0,86
	String									API								
KNN	0,94	0,85	0,88	0,78	0,91	0,81	0,84	0,92	0,85	0,69	0,75	0,71	0,73	0,43	0,50	0,69	0,48	0,55
SVM	0,74	0,69	0,71	0,98	0,97	0,97	0,94	0,97	0,95	0,65	0,76	0,68	0,87	0,71	0,77	0,84	0,77	0,79
NB	0,94	0,97	0,95	0,94	0,91	0,92	0,76	0,97	0,81	0,82	0,81	0,80	0,75	0,75	0,71	0,85	0,84	0,83
DT	0,83	0,97	0,88	0,83	0,97	0,88	0,83	0,97	0,88	0,88	0,85	0,85	0,88	0,85	0,85	0,88	0,85	0,85
RF	0,89	0,96	0,91	0,91	0,98	0,93	0,89	0,98	0,93	0,90	0,84	0,86	0,90	0,84	0,86	0,80	0,77	0,77
MLP	0,53	0,51	0,49	0,76	0,99	0,82	0,67	0,81	0,73	0,25	0,37	0,30	0,82	0,67	0,67	0,71	0,76	0,73

Figura 5. Resultados por tipo de informação (70:30)

serem processados com StandardScaler, com os quais o SVM alcançou valores elevados para as métricas Precision (0,98) e F1-Score (0,95), superando sua performance tanto sem otimização quanto com PCA. Esse fato pode ser atribuído à capacidade do StandardScaler de normalizar características, reduzindo o viés e melhorando a capacidade do SVM de distinguir entre classes em dados textuais complexos, que são particularmente sensíveis a variações de escala, como é o caso dos dados de String. De modo similar, o StandardScaler possibilitou que o MLP alcançasse o maior valor de Recall (0,99) com os dados de String.

Por outro lado, a análise de Chamadas de API processadas com StandardScaler e PCA tiveram desempenho variável e, muitas vezes com melhorias pouco significativas se comparado com os dados sem otimização, exceto para o algoritmo MLP que apresentou aumentos significativos em todas as métricas comparadas aos conjuntos de dados sem otimização. Esse resultado sugere que a normalização e a redução de dimensionalidade podem contribuir para atenuar o ruído e destacar padrões mais relevantes nas Chamadas de API percebidas pelo MLP. Também é interessante notar que para os dados de Assinatura e Rede processados com StandardScaler, o NB obteve valores baixos de Precision (0,69 e 0,66, respectivamente), mas melhores valores de Recall (0,91 e 0,85, respectivamente), o que poderia indicar um possível desbalanceamento das classes. Contudo, considerando que medidas foram tomadas para evitar o desbalanceamento e que esse fenômeno não foi observado de maneira consistente em todos os algoritmos, esses resultados do NB foram considerados atípicos.

Adicionalmente, os dados de String frequentemente exibiram desempenhos comparáveis ou superiores aos de Assinatura, Rede e Chamadas de API para a maioria dos algoritmos. Isso pode ser atribuído ao fato de o Cuckoo Sandbox fornecer, nesta seção, uma riqueza de detalhes sobre strings codificadas no ransomware e comandos capturados durante a análise dinâmica.

Os resultados para a divisão 50:50 mostrados na Figura 6 reafirmam que em termos de desempenho médio geral o RF se destaque em todas as métricas e tipos de dados,

Algoritmo	Normal			StandardScaler			PCA			Normal			StandardScaler			PCA		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
	Assinatura									**Rede**								
KNN	0,96	0,90	0,93	0,94	0,86	0,89	0,94	0,85	0,89	0,87	0,89	0,87	0,85	0,76	0,79	0,84	0,80	0,81
SVM	1,00	0,85	0,90	0,98	0,86	0,91	0,95	0,91	0,92	0,86	0,91	0,89	0,87	0,95	0,90	0,94	0,92	0,92
NB	0,83	0,94	0,83	0,77	0,93	0,79	0,80	0,76	0,75	0,71	0,88	0,72	0,71	0,88	0,72	0,75	0,75	0,69
DT	0,89	0,85	0,86	0,89	0,85	0,86	0,89	0,85	0,86	0,85	0,93	0,87	0,85	0,93	0,87	0,85	0,93	0,87
RF	0,90	0,91	0,89	0,91	0,98	0,92	0,89	0,91	0,89	0,88	0,91	0,88	0,92	0,92	0,91	0,83	0,87	0,84
MLP	0,52	0,69	0,57	0,90	0,87	0,87	0,93	0,89	0,91	0,38	0,44	0,39	0,86	0,90	0,87	0,85	0,86	0,85
	String									**API**								
KNN	0,96	0,88	0,91	0,76	0,93	0,80	0,83	0,92	0,84	0,86	0,74	0,75	0,85	0,42	0,53	0,80	0,47	0,57
SVM	0,67	0,70	0,66	0,98	0,93	0,95	0,94	0,95	0,94	0,60	0,79	0,67	0,99	0,69	0,79	0,95	0,76	0,81
NB	0,95	0,98	0,95	0,96	0,93	0,94	0,79	0,98	0,85	0,84	0,81	0,81	0,78	0,81	0,76	0,79	0,87	0,81
DT	0,85	0,97	0,89	0,85	0,97	0,89	0,85	0,97	0,89	0,85	0,88	0,85	0,85	0,88	0,85	0,85	0,88	0,85
RF	0,95	0,95	0,95	0,96	0,96	0,96	0,96	0,97	0,96	0,91	0,79	0,83	0,91	0,85	0,88	0,82	0,80	0,83
MLP	0,42	0,60	0,46	0,73	0,99	0,80	0,73	0,91	0,79	0,34	0,38	0,29	0,70	0,83	0,70	0,76	0,73	0,73

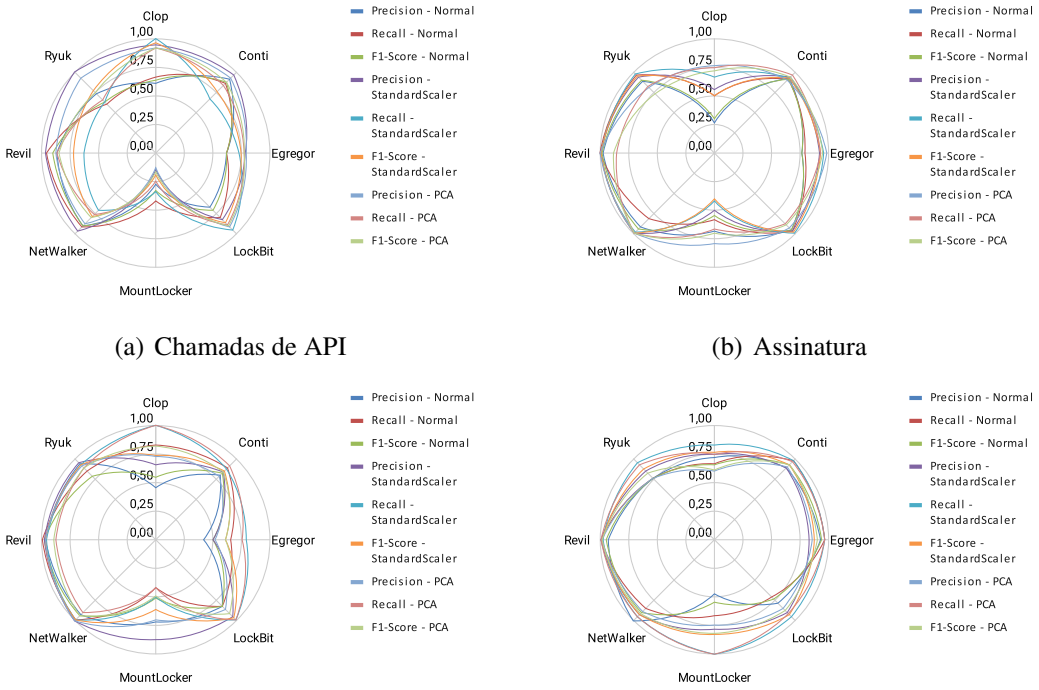
Figura 6. Resultados por tipo de informação (50:50)

especialmente nas categorias de Assinatura e String. Este algoritmo tende a performar bem em cenários de classificação complexos devido à sua capacidade de lidar com *overfitting* e sua eficácia em trabalhar conjuntos de dados desbalanceados. O SVM também apresenta um bom desempenho geral em Precision, especialmente com pré-processamento com StandardScaler, onde atinge resultados quase perfeitos. Isso sugere que o SVM, conhecida por sua eficácia em espaços de alta dimensão, consegue extrair padrões significativos das informações de String, Assinatura e Chamadas de API.

Em relação aos tipos de informação, os dados de String continuam sendo os mais eficazes, alinhado com o observado para a divisão 70:30. Na divisão 50:50, também foi possível observar resultados notáveis para os dados de Assinatura, os quais são derivados diretamente de padrões conhecidos de malware, evidenciando sua utilidade prática na identificação de atividades maliciosas já amplamente conhecida.

A análise por família de ransomware indicam que certas famílias, como NetWalker e Revil, foram consistentemente bem detectadas em vários tipos de dados e métodos de pré-processamento. Este fato sugere que as características dessas famílias são distintivas e bem capturadas pelos métodos empregados, facilitando sua identificação. Em contrapartida, famílias como MountLocker e Clop apresentaram variações significativas de desempenho, possivelmente devido à menor representatividade no conjunto de dados ou à natureza sofisticada de seus comportamentos, que pode não ser totalmente capturada pelos métodos utilizados. A Figura 7 ilustra como diferentes famílias de ransomware respondem a cada tipo de dado e método de pré-processamento.

Note que o NetWalker apresenta o melhor desempenho global para os dados de Assinatura e Rede, com resultados quase perfeitos quando adotados os métodos de pré-processamento. O Egregor apresenta ótimos resultados para os dados de String e bons resultados para os dados de Assinatura, em contraste com os resultados mais baixos para os dados de Rede e Chamadas de API. Ainda sobre as Chamadas de API, esses dados foram os que possibilitaram os melhores resultados para o Clop, possivelmente por terem sido capazes de capturar minimamente suas ações típicas de tentativa de desabilitação



(a) Chamadas de API (b) Assinatura

(c) Rede (d) String

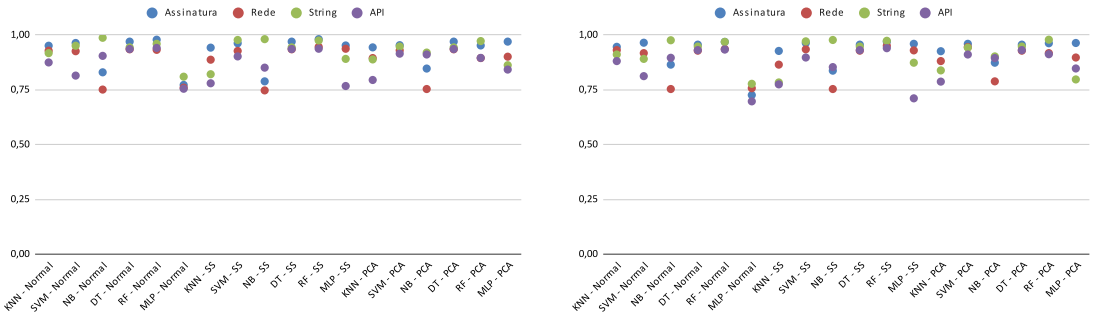
Figura 7. Resultados por família de ransomware (70:30)

de softwares de segurança, serviços de backup e outras defesas que poderiam impedir a criptografia. Por outro lado, os dados de Chamadas de API foram os que apresentaram a maior inconsistência nas métricas avaliadas para o Ryuk.

A divisão do conjunto de dados 50:50 também foi analisada e podemos destacar que as famílias NetWalker e Revil continuaram apresentando resultados robustos e consistentemente altos, indicando que as características dessas famílias são capturadas efetivamente pelos modelos, independentemente da mudança na divisão dos dados. Além disso, os dados de Assinatura e String frequentemente resultaram nos melhores desempenhos, comparado a Rede e Chamadas de API, em ambas as divisões, o que pode ter ocorrido devido a natureza mais direta e menos ambígua das informações capturadas nessas categorias. Também é importante destacar que o Clop mostrou uma melhoria significativa em todas as métricas e tipos de dados da divisão 70:30 para a 50:50, especialmente com o uso dos métodos de pré-processamento. No que se refere a classificação correta de amostras maliciosas, os destaques para a divisão 50:50 foram o Egregor e o MountLocker que, usando os dados de String, alcançaram Recall 1,00 para os dados sem otimização e com StandardScaler para o Egregor e com StandardScaler e PCA para o MountLocker.

A acurácia fornece uma medida abrangente do desempenho do modelo ao indicar a proporção de previsões corretas — tanto verdadeiros positivos quanto verdadeiros negativos — em relação ao total de casos testados e os resultados observados nos experimentos são mostrados na Figura 8.

Para a divisão do conjunto de dados em 70:30 (Figura 8(a)), os algoritmos RF e DT mostraram alto desempenho em todos os tipos de dados e métodos de pré-processamento,



(a) 70:30 (b) 50:50

Figura 8. Acurácia por tipo de informação

especialmente para os dados sem otimização, indicando que estes são robustos independentemente da transformação dos dados. Como esperado, o StandardScaler melhorou a acurácia de algoritmos que são sensíveis à variação na escala dos dados, como pode ser observado para o MLP em quase todos os tipos de informação. Já o PCA mostrou resultados mistos, ajudando na redução de ruído em alguns casos, mas também podendo ocasionar a perda de informações relevantes para a classificação em outros.

Confirmando o observado anteriormente, os dados de String se destacaram com o SVM e StandardScaler, alcançando a acurácia mais alta (0,98), o que reforça que o ajuste fino da escala dos dados pode extrair características mais discriminativas desta informação textual. Ao considerar especificamente os dados de Assinatura, a maior acurácia (0,98 nos dados sem otimização e StandardScaler) foi alcançado com o RF. Para os dados de Rede, o RF com o StandardScaler também foi o que apresentou a melhor acurácia (0,95). Embora as Chamadas de API não tenham alcançado as maiores acurácias comparativamente aos outros tipos de dados, ainda mostrou bons resultados com RF, DT e SVM (0,94, 0,93 e 0,92, respectivamente).

Para a divisão do conjunto de dados em 50:50 (Figura 8(b)), o RF e o SVM continuaram exibindo um ótimo desempenho em quase todos os tipos de informação e métodos de pré-processamento, especialmente o RF que mostra uma acurácia superior aos demais algoritmos na maioria das configurações, não sendo surpreendente que ele continue mostrando os melhores resultados para os dados de Assinatura e String, especialmente com StandardScaler e PCA. Também é relevante mencionar que o NB alcançou uma acurácia de 0,98 para os dados de String, equiparando-se ao RF e ao SVM, embora não tenha atingido valores tão elevados de acurácia para os dados de Assinatura, Rede e Chamadas de API.

Os resultados mostram que os dados de String quando classificados com RF e SVM são particularmente eficazes para a detecção de ransomware. Especificamente, na divisão dos dados em 50:50, o RF atingiu uma acurácia de 0,98 com dados pré-processados usando StandardScaler e PCA, enquanto o SVM alcançou a mesma acurácia tanto com dados sem otimização quanto com aqueles processados com StandardScaler. Isso sugere que a utilização de dados textuais detalhados presentes na seção String dos relatórios JSON do Cuckoo Sandbox é altamente recomendável para detectar com alta precisão os ataques de ransomware.

5. Conclusões e Trabalhos Futuros

Este estudo explorou a eficácia dos métodos de pré-processamento e algoritmos de ML na detecção de ransomware, com um enfoque especial no uso da técnica TF-IDF para processar dados extraídos de relatórios de análise dinâmica do Cuckoo Sandbox. A análise comparativa entre os diferentes tipos de dados — Assinatura, String, Rede e Chamadas de API — revelou suas respectivas eficácias em contextos de detecção de ransomware. Os dados de String tratados com StandardScaler e analisados usando RF e SVM emergiram como as combinações mais eficazes, sugerindo uma direção clara para o uso de dados textuais detalhados em conjunto com métodos de ajuste de escala como uma estratégia efetiva para identificar ransomware. Além disso, a capacidade do PCA de reduzir ruído e destacar características importantes foi confirmada, embora com resultados variando de acordo com o tipo de dados e o algoritmo aplicado.

O cenário de evolução constante das ameaças cibernéticas, considerando o ransomware em particular, corresponde a um desafio para desenvolver um conjunto de dados completo e atualizado para treinamento e avaliação de modelos, de modo que trabalhos futuros poderiam enriquecer o conjunto de dados disponibilizado, integrando outras amostras de ransomware. O uso do INETSIM, embora restrinja as comunicações do ransomware, pode ter distorcido a precisão do conjunto de dados e afetado o desempenho dos modelos na classificação dos dados de Rede, indicando o papel crítico da integridade e equilíbrio do conjunto de dados.

Pesquisas futuras poderiam ser explorado o impacto de outras técnicas de pré-processamento de dados e sua combinação com algoritmos de ML não abordados neste estudo, como as técnicas de *Deep Learning*, que necessitaria de testes maisabrangentes para refinamento de seus parâmetros e mais tempo de pesquisa. Uma área promissora inclui a aplicação de métodos de aprendizado profundo, que podem ser capazes de identificar padrões complexos em grandes conjuntos de dados de ransomware que técnicas mais tradicionais podem não detectar. Além disso, além de expandir o conjunto de dados com novas variantes de ransomware, seria interessante considerar dados obtidos de outras ferramentas além do Cuckoo Sandbox, para validar a robustez do uso de TF-IDF associado a algoritmos de ML em diferentes ambientes e cenários. A implementação de modelos híbridos que combinem diferentes tipos de dados também pode ser explorada para identificar se produzem sistemas de detecção de ransomware mais robustos e confiáveis.

Pesquisas futuras também poderiam refinar os parâmetros do MLP para aumentar a precisão da classificação, dado que os resultados encontrados no presente estudo apresentam grande margem para melhoria. O desenvolvimento de um sistema de detecção de ransomware, testando classificadores em tempo real, é um próximo passo importante. Além disso, como o Windows 7 já está obsoleto há anos, novos experimentos em ambiente Windows 10 e 11 serão realizados para comparar os resultados, visto que alguns ransomwares chamam APIs diferentes dependendo do sistema operacional, havendo APIs adicionais e atualizadas disponíveis no Windows 10 e Windows 11, e sendo essas duas versões aquelas que concentram o maior número de usuários atualmente [Statcounter 2024].

Referências

Al-rimy, B. A. S., Maarof, M. A., and Shaid, S. Z. M. (2019). Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Generation Computer Systems*, 101:476–491.

Begovic, K., Al-Ali, A., and Malluhi, Q. (2023). Cryptographic ransomware encryption detection: Survey. *Computers & Security*, 132:103349.

Benmalek, M. (2024). Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges. *Internet of Things and Cyber-Physical Systems*, 4:186–202.

Black, P., Sohail, A., Gondal, I., Kamruzzaman, J., Vamplew, P., and Watters, P. (2020). Api based discrimination of ransomware and benign cryptographic programs. In *International Conference on Neural Information Processing*, pages 177–188. Springer.

Cen, M., Jiang, F., Qin, X., Jiang, Q., and Doss, R. (2024). Ransomware early detection: A survey. *Computer Networks*, 239:110138.

Chang, K., Zhao, N., and Kou, L. (2022). A survey on malware detection based on api calls. In *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*, pages 464–471.

Chen, Q., Islam, S. R., Haswell, H., and Bridges, R. A. (2019). Automated ransomware behavior analysis: Pattern extraction and early detection. In *International Conference on Science of Cyber Security*, pages 199–214. Springer.

Dabas, N., Ahlawat, P., and Sharma, P. (2023). An effective malware detection method using hybrid feature selection and machine learning algorithms. *Arabian Journal for Science and Engineering*, 48(8):9749 – 9767.

Dinh, P. V., Shone, N., Dung, P. H., Shi, Q., Hung, N. V., and Ngoc, T. N. (2019). Behaviour-aware malware classification: Dynamic feature selection. In *2019 11th International Conference on Knowledge and Systems Engineering*, pages 1–5. IEEE.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. F. d. (2021). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC.

Freeman, D. and Chio, C. (2018). *Machine Learning and Security: Protecting Systems with Data and Algorithms*. O'Reilly Media.

Guarnieri, C., Tanasi, A., Bremer, J., and Schloesser, M. (2012). The cuckoo sandbox. *Accessed: Dec*, 16:2018.

Horowitz, M. (2023). Check point 2023 security report.

IBMSecurity (2023a). Cost of a data breach report 2023.

IBMSecurity (2023b). X-force threat intelligence index 2023.

IBMSecurity (2024). X-force threat intelligence index 2024.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Kaspersky (2021). Ransomware double extortion and beyond: Revil, clop, and conti.

Kaspersrky (2021). Ataques de ransomware direcionados crescem 700%.

Kim, M. and Kim, H. (2024). A dynamic analysis data preprocessing technique for malicious code detection with tf-idf and sliding windows. *Electronics*, 13(5).

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Maniriho, P., Mahmood, A. N., and Chowdhury, M. J. M. (2024a). A systematic literature review on windows malware detection: Techniques, research issues, and future directions. *Journal of Systems and Software*, 209:111921.

Maniriho, P., Mahmood, A. N., and Chowdhury, M. J. M. (2024b). A systematic literature review on windows malware detection: Techniques, research issues, and future directions. *Journal of Systems and Software*, 209:111921.

Mohanta, A. and Saldanha, A. (2020). *Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware*. Springer.

Prachi., Dabas, N., and Sharma, P. (2023). Malanalyser: An effective and efficient windows malware detection method based on api call sequences. *Expert Systems with Applications*, 230:120756.

Qin, B., Zhang, J., and Chen, H. (2021). Malware detection based on tf-(idf&icf) method. *Journal of Physics: Conference Series*, 2024(1):012030.

Razaulla, S., Fachkha, C., Markarian, C., Gawanmeh, A., Mansoor, W., Fung, B. C. M., and Assi, C. (2023). The age of ransomware: A survey on the evolution, taxonomy, and research directions. *IEEE Access*, 11:40698–40723.

Singh, J. and Singh, J. (2021). A survey on machine learning-based malware detection in executable files. *Journal of Systems Architecture*, 112:101861.

Statcounter (2024). Desktop windows version market share worldwide: May 2023 - may 2024.

Team, T. I. (2023). 2023 state of ransomware.

Vajjala, S., Majumder, B., Gupta, A., and Surana, H. (2020). *Practical Natural Language Processing: A Comp. Guide to Building Real-world NLP Systems*. O'Reilly Media.

Vang-Mata, R. (2020). *Multilayer Perceptrons: Theory and Applications*. Computer Science, Technology and Applications Series. Nova Science Publishers.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., and Sangaiah, A. K. (2019). Classification of ransomware families with machine learning based on n-gram of opcodes. *Future Generation Computer Systems*, 90:211–221.

Zhang, S., Du, T., Shi, P., Su, X., and Han, Y. (2023). Early detection and defense countermeasure inference of ransomware based on api sequence. *International Journal of Advanced Computer Science and Applications*, 14(10):632 – 641.