UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CLASSIFICAÇÃO DE RANSOMWARE UTILIZANDO MLP, REDUÇÃO DE DIMENSIONALIDADE E BALANCEAMENTO DE CLASSES

GEORGE TASSIANO MELO PEREIRA

DM 19/2023

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil

2023

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GEORGE TASSIANO MELO PEREIRA

CLASSIFICAÇÃO DE RANSOMWARE UTILIZANDO MLP, REDUÇÃO DE DIMENSIONALIDADE E BALANCEAMENTO DE CLASSES

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para obtenção do Grau de Mestre em Engenharia Elétrica na Área de Computação Aplicada.

Orientador: Prof. Dr. Claudomiro de Souza de Sales Júnior

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

P436c Pereira, George Tassiano Melo.
Classificação de ransomware utilizando MLP, redução de dimensionalidade e balanceamento de classes / George Tassiano Melo Pereira. — 2023.
55 f. : il. color.

Orientador(a): Prof. Dr. Claudomiro de Souza de Sales Júnior
Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2023.

1. Aprendizado de Máquina. 2. Segurança de dados. I. Título.

CDD 006.32

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CLASSIFICAÇÃO DE RANSOMWARE UTILIZANDO MLP, REDUÇÃO DE DIMENSIONALIDADE E BALANCEAMENTO DE CLASSES

AUTOR: GEORGE TASSIANO MELO PEREIRA

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 03/07/2023

BANCA EXAMINADORA:

Prof. Dr. Claudomiro de Souza de Sales Júnior
Orientador - PPGEE/UFPA

Profa. Dra. Regiane Silva Kawasaki Francês
Avaliador Externo - FACOMP/UFPA

Profa. Dra. Adriana Rosa Garcez Castro
Avaliador Interno - PPGEE/UFPA

VISTO:

Prof. Dr. Diego Lisboa Cardoso
Coodernador do PPGEE/ITEC/UFPA

Dedico este trabalho a todos que acreditaram no meu potencial.

Agradecimentos

Agradeço a minha família por todo esforço investido na minha educação.

Sou grato ao meu orientador Prof. Dr. Claudomiro Sales por aceitar conduzir este trabalho. Serei eternamente grato pela sua orientação e confiança.

Também agradeço à instituição Universidade Federal do Pará, assim como todos os professores do meu curso pela elevada qualidade do ensino oferecido.

Deixo um agradecimento especial a minha esposa Adriana por estar ao meu lado em todos os momentos da minha vida, e em especial, na realização deste projeto.

“A ciência é muito mais que um corpo de conhecimentos. É uma maneira de pensar.” (Carl Sagan)

Resumo

Ransomware é um tipo de *malware* que impede ou limita o acesso do usuário ao sistema e arquivos até que um resgate seja pago. Combater essa ameaça é difícil devido à sua disseminação rápida e às constantes mudanças nas técnicas de criptografia utilizadas. Algoritmos de aprendizado de máquina, como Redes Neurais Artificiais, têm sido apontados como ferramentas promissoras na classificação de *ransomware*, porque elas podem aprender a identificar padrões e características complexas em grandes quantidades de dados. Isso permite que as redes neurais sejam treinadas com exemplos de amostras de software malicioso, incluindo ransomware, e depois sejam capazes de classificar novos exemplos com alta precisão. Além disso, as redes neurais também são capazes de aprender e se adaptar a mudanças no comportamento do malware, tornando-as ferramentas eficazes para a detecção de novos tipos de ransomware. Neste trabalho, é explorado três tipos de classificação de *ransomware* por RNA dentro de um pipeline composto com redução de dimensionalidade por Kernel PCA e balanceamento de classes com a abordagem de superamostragem aleatória. A MLP (*Multi-layer Perceptron*) alcançou uma média de 98% de acurácia na classificação binária e 85% de acurácia na classificação de família com *goodware*, onde tais valores superam os resultados anteriores e demonstram assim a eficácia da inclusão do balanceamento de classes na melhoria do modelo de detecção de *ransomware*.

Palavras-chave: Ransomware; MLP; Balanceamento de Classes.

Abstract

Ransomware is a type of malware that prevents or limits user access to system and files until a ransom is paid. Combating this threat is difficult due to its rapid spread and constant changes in the encryption techniques used. Machine learning algorithms such as Artificial Neural Networks have been touted as promising tools in classifying ransomware because they can learn to identify complex patterns and features in large amounts of data. This allows neural networks be trained on sample examples of malicious software, including ransomware, and then be able to classify new examples with high accuracy. Furthermore, neural networks are also capable of learning and adapting to changes in malware behavior, making them effective tools for detecting new types of ransomware. In this work, three types of ransomware classification by ANN are explored within a composite pipeline with dimensionality reduction by Kernel PCA and class balancing with the random oversampling approach. The MLP (Multi-layer Perceptron) reached an average of 98% accuracy in the binary classification and 85% accuracy in the goodware family classification, where such values surpass the previous results and thus demonstrate the effectiveness of the inclusion of the class balancing in improving the ransomware detection model.

Key-words: Ransomware; MLP; Class balancing.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

OMS	Organização Mundial de Saúde
ML	Machine Learning
PCA	Análise de Componentes Principais
KPCA	Kernel PCA
IoT	Internet das Coisas
AIDS	Síndrome da Imunodeficiência Adquirida
RSA	Rivest-Shamir-Adleman
MSE	Erro Quadrático Médio
AES	Advanced Encryption Standard
RL	Regressão Logística
SVM	Máquina de vetores de suporte
GP	Gaussian Process
DT	Decision Tree
RF	Random Forest
NB	Naive Bayes
QDA	Quadratic Discriminant Analysis
KNN	K Nearest Neighbor
MLP	Multi-layer Perceptron
XOR	Ou exclusivo
LDA	Análise Discriminante Quadrática
IBL	Instance-based Learning
CPS	Cyber-Physical Systems
SGD	Stochastic Gradient Descent

PSO	Otimização por Enxame de Partícula
ERAND	Ensemble Ransomware Defense
NSGA-II	Non-dominated Sorting Genetic Algorithm II
C&C	Comando e Controle
TPU	Tensor Processing Unit
ROC	Receiver Operating Characteristic
RD	Redução de Dimensionalidade

Sumário

1 Introdução

1.1 Contextualização

O *ransomware* é um tipo de *malware* que tem como objetivo restringir o acesso a sistemas e dados até que um resgate seja pago. Ele pode ser dividido em quatro fases: invasão, comunicação com o servidor de comando e controle do *hacker*, destruição e extorsão, como mostra a Figura 1. Na fase de invasão, os cibercriminosos usam usuários e aplicativos maliciosos, *spam* ou *phishing* para tentar invadir o sistema. Na fase de comunicação, o *malware* se comunica com o servidor de comando e controle do *hacker*. Na fase de destruição, o *malware* pode criptografar os arquivos das vítimas ou bloquear a máquina, impedindo que os usuários acessem seus sistemas. Por fim, na fase de extorsão, o *hacker* exige um pagamento de resgate para liberar os arquivos ou o acesso ao sistema.

Figura 1 – Fases do Ransomware. Fonte: Autor.

A crescente popularidade do *ransomware* é atribuída a vários fatores desde sua primeira aparição em 1989, quando 20.000 disquetes infectados foram distribuídos em uma conferência sobre AIDS pela Organização Mundial de Saúde (OMS), como destacado por (OZ et al., 2021). De acordo com um relatório da empresa de segurança cibernética, *Red Access*, o *ransomware* continuará sendo o inimigo público nº 1 em termos de segurança cibernética em 2023 (WILLIAMS, 2023).

Em 2022, aproximadamente 68% de todas as organizações globais foram vítimas de pelo menos uma infecção de *ransomware* ainda de acordo com (WILLIAMS, 2023). Gangues de *ransomware* estão se tornando cada vez mais sofisticadas e atingindo um número crescente de vítimas, incluindo usuários comuns, organizações empresariais, governamentais, bancos, prefeituras e centros médicos. A pandemia COVID-19 agravou a situação, com ataques de *ransomware* aumentando quase 500% desde o início da pandemia, de acordo com (DALEY, 2021). O *ransomware* tem sido a escolha preferencial dos criminosos cibernéticos há anos e as organizações estão experimentando um aumento na escala e sofisticação desses ataques.

O ransomware gera grandes quantidades de dados de alta dimensão, tornando a classificação sem o uso de inteligência computacional um desafio de tempo e processamento. A redução de dimensionalidade pode ser usada para extrair atributos e reduzir o espaço dimensional, já que

problemas de classificação com um número finito de amostras e alta quantidade de atributos podem causar problemas conhecidos como "maldição da dimensionalidade". Por isso, métodos de redução de dimensionalidade têm sido amplamente aplicados em problemas reais atualmente (CUSACK; MICHEL; KELLER, 2018).

O aprendizado de máquina tem sido eficaz na classificação de ransomware, como mostrado em estudos de autores como (ADAMU; AWAN, 2019), (ABBASI; AL-SAHAF; WELCH, 2020), e (FERNANDO; KOMNINOS; CHEN, 2020). Neste trabalho, redes neurais são utilizadas junto com a técnica de redução de dimensionalidade kernel PCA (KPCA) e balanceamento de classes para encontrar a melhor acurácia na detecção de ransomware, abordando três métodos de classificação utilizados na literatura.

1.2 Estado da Arte

"Aprendizado de máquina é efetivo na detecção de *malwares*, como evidenciado por trabalhos de (FERNANDO; KOMNINOS; CHEN, 2020), (MILOSEVIC; DEHGHANTANHA; CHOO, 2017) (HWANG et al., 2020), (ALENAZI; HINDI; ASSADHAN, 2019), (ADAMU; AWAN, 2019) e (KUMAR; SHARMA, 2023). O *ransomware* é um assunto popular de pesquisa, com vários estudos que se concentram em diferentes aspectos.

O estudo de (OZ et al., 2021) realizou uma ampla pesquisa sobre *ransomware* e soluções de defesa em relação a PCs / estações de trabalho, dispositivos móveis e plataformas *Cyber-Physical Systems* (CPS) e *Internet of Things* (IoT). A pesquisa cobriu 137 estudos de 1990-2020, fornecendo uma visão geral detalhada da evolução do *ransomware*, analisando os principais componentes e apresentando uma taxonomia de notáveis famílias de *ransomware*. Além disso, lista questões pendentes para pesquisas futuras."

O artigo de (ALAM et al., 2018) apresenta uma ferramenta de detecção não supervisionada em duas etapas. A equipe observou os contadores de desempenho de hardware presentes em processadores modernos para identificar atividades de criptografia baseadas em eventos de CPU. Quando o Ransomware Prevention via Performance Counters (RAPPER) identifica alguma atividade suspeita, ele emite um alerta. Este framework utiliza uma combinação de Redes Neurais Artificiais (ANN) e Transformação Rápida de Fourier para ser preciso, rápido e confiável. Não houve fornecimento de exemplos de ransomware durante a fase de treinamento, e o framework identifica anomalias que divergem do comportamento aprendido.

O "R-PackDroid"(MAIORCA et al., 2017) é outra proposta, desta vez para o sistema operacional Android. Os autores usaram informações obtidas a partir de pacotes de API do sistema para diferenciar entre software benigno, ransomware ou malware genérico com o auxílio do classificador Random Forest.

Já o "RansHunt"(HASAN; RAHMAN, 2017) é um framework de detecção baseado na

técnica de Support Vector Machine (SVM). Os autores extraíram mais de 60 mil características de operações dinâmicas e estáticas, como ações de rede, operações de sistema de arquivos, chamadas de API, operações de processo e operações de registro.

Em "Uma Análise de Gráfico de Fluxo de Chamadas de API para Detecção de Ransomware"(CHEN, 2018), a equipe observou o gráfico de fluxo de chamadas de API usadas por um processo suspeito e extraiu 70.000 características para análise. Para a redução da dimensionalidade, eles utilizaram técnicas de correlação e compararam resultados com diferentes técnicas, incluindo SVM, algoritmo Logístico Simples, Random Forest e Naive Bayes.

Os autores (VIGNESWARAN et al., 2018) avaliaram diferentes tipos de redes para classificação e detecção de ransomware. Após a avaliação, eles descobriram que uma rede profunda obteve melhores resultados em comparação com uma rede tradicional superficial. A equipe propôs então uma arquitetura Multi-Layer Perceptron (MLP) para detecção utilizando chamadas de API."

(SHAUKAT; RIBEIRO, 2018) propuseram um sistema de defesa em camadas para proteger contra o ransomware, baseado em uma abordagem híbrida de análise dinâmica e estática. Eles usaram a camada "Strong Trap"para detecção precoce, a camada de aprendizado de máquina para proteção contra invasões zero-day, e a camada de backup de arquivos para proteger os dados do usuário. Eles testaram seu sistema com vários métodos de aprendizado de máquina, incluindo Random Forest, Support Vector Machines, Gradient Tree Boosting e Artificial Neural Networks.

O estudo de (FERNANDO; KOMNINOS; CHEN, 2020) examinou a evolução da detecção de *ransomware* por meio de aprendizado de máquina e técnicas de aprendizado profundo. Avaliaram 19 estudos, incluindo a abordagem algorítmica, o processo de engenharia de recursos e a avaliação de resultados. Este artigo é importante porque explorou novas tendências de ransomware e previsões para sua evolução futura.

(KARIMI; MOATTAR, 2017) se concentraram na detecção de dispositivos móveis Android e IoT. Eles detectaram o ransomware transformando as instruções executáveis em uma imagem em escala de cinza, e depois utilizaram LDA (Alocação Latente de Dirichlet) para reduzir a imagem e métodos de classificação.

O estudo de (BAE; LEE; IM, 2019) propôs um método de detecção de *ransomware* que pode diferenciar arquivos benignos e outros tipos de malware de ameaças de *ransomware*. Seis algoritmos de aprendizado de máquina foram testados, incluindo *Random Forest)* (RF), Regressão Logística (RL), *Naive Bayes* (NB), *Stochastic Gradient Descent* (SGD), *K-Nearest Neighbours* (KNN) e *Support Vector Machine* (SVM).

O estudo de (ABBASI; AL-SAHAF; WELCH, 2020) apresentou um método de seleção de recursos baseado em otimização por enxame de partículas (PSO) para detectar e classificar *ransomware* com dados de análise de comportamento de alta dimensão. O estudo demonstra que

a eficiência da classificação do modelo varia com base na quantidade de recursos selecionados a partir de cada grupo de recursos presente nos dados. A equipe de pesquisa alcançou uma taxa média de acerto de 50% para identificação de diferentes famílias de *ransomware*.

(MOUSSAILEB et al., 2018) detectaram o comportamento de ransomware analisando a ação de travessia de sistema de arquivos. Como a maioria dos ransomware começa a criptografia a partir da raiz do disco rígido, eles conseguem detectá-los usando pastas iscas. Eles usaram vários métodos de aprendizado de máquina, incluindo KNN, árvore de decisão e Random Forest, para classificar o ransomware.

Já o trabalho de (BORAH; BHATTACHARYYA; KALITA, 2021) apresentou a classificação ERAND para defesa contra *ransomware*, usando o NSGA-II para calcular os pesos de cinco classificadores e alcançando precisões elevadas, com resultados acima de 95% para cada família. No entanto, a metodologia utilizada foi obscura e resultou em várias interpretações distintas.

O estudo de (ADAMU; AWAN, 2019) avaliou a previsão de *ransomware* usando algoritmos de aprendizado supervisionado, incluindo *Support Vector Machines* (SVM), *Random Forest* (RF), *Decision Tree* (DT), Naive Bayes (NB), Rede Neural Artificial (ANN), Regressão Logística (RL). O melhor desempenho de classificação binária foi obtido pelo SVM (88,2%) seguido pela RF (84%) e MLP (86%), enquanto a Regressão Logística (65,7%) e o Naive Bayes (52,5%) tiveram desempenho inferior.

O estudo de (SILVA et al., 2020) comparou o desempenho dos algoritmos Naive Bayes, SVM e Árvore de Decisão na classificação de *ransomware* por meio de técnicas estáticas e dinâmicas. A acurácia alcançada foi de 81,65% para Naive Bayes, 70,79% para SVM e 75,66% para Árvore de Decisão.

O estudo de (ALENAZI; HINDI; ASSADHAN, 2019) examinou ajustes finos do classificador Naive Bayes para conjuntos de dados desequilibrados, propondo o *(Fine Tuning Naive Bayes)* FTNB-ID e comparando-o ao Naive Bayes e FTNB original.

O estudo de (CUSACK; MICHEL; KELLER, 2018) propôs um modelo de detecção de *ransomware* baseado em aprendizado de máquina usando dados de tráfego de rede. A solução monitorou a comunicação entre a vítima e o comando e controle (C&C) para impedir a entrega da chave de criptografia e detectar *ransomware*. Foram identificados os oito atributos mais relevantes para detecção de *ransomware*, mas a solução apresentou uma taxa de falsos positivos de 12,5%, o que pode gerar muitos alarmes falsos.

A pesquisa de (SGANDURRA et al., 2016) examina o ransomware de forma automatizada e dinâmica. Eles criaram o EldeRan, baseado na ideia de que o malware exibe ações exclusivas ou significativamente diferentes das ações de software seguro. O sistema monitora um ambiente virtual *sandbox (Cuckoo Sandbox)* e extrai recursos, incluindo chamadas de API do *Windows*, operações de registro, sistema de arquivos, diretórios, extensões de arquivo, descarte

de arquivos e strings executáveis. A seleção de características utiliza o critério de informação mútua para identificar as características mais discriminatórias.

A classificação de executáveis como benignos ou maliciosos é feita pelo classificador de regressão logística regularizado com base nos recursos extraídos. O problema é que este método tem dificuldades para detectar *ransomware* inativo ou que requer interação do usuário para ser ativado, graças à dependência em técnicas de *sandbox*. Além disso, a regressão logística regularizada não é eficiente para lidar com limites de decisão não lineares, tornando difícil para o modelo identificar relações complexas entre recursos.

O trabalho de (KHARRAZ et al., 2015) sugere que o *ransomware* de muitas famílias pode ser detectado monitorando a atividade do sistema de arquivos. Eles conduziram uma análise dinâmica em grande escala de 1359 amostras de *ransomware* de 15 famílias diferentes, coletadas entre 2006 e 2014. Em seguida, a solução UNVEIL foi proposta para análise e detecção de *ransomware*, baseada em padrões de (E/S) de processos de usuário. Infelizmente, o *UNVEIL* só detecta o *ransomware* após alguns arquivos serem criptografados, e pode ser contornado por *ransomware* executado com privilégios do *kernel*.

O estudo de (KUMAR; SHARMA, 2023) aborda a segurança do Android, que tem sido frequentemente alvo de ataques cibernéticos e explorada por usuários não autorizados. autores focam na arquitetura de rede do *ransomware* para Android, analisando detalhadamente e formando um plano para a detecção desses ataques. Eles utilizaram o conjunto de dados CIC-AndMal2017, criado a partir de ataques de *malware* em smartphones Android reais, e analisaram o comportamento de diferentes ataques de *ransomware* para Android. Os resultados deste estudo experimental podem ajudar pesquisadores a desenvolver metodologias de detecção de *ransomware* baseadas em inteligência artificial para a plataforma Android.

Por fim, no trabalho (PEREIRA; JúNIOR, 2022) foi realizado a comparação de 10 técnicas de aprendizado de máquina em conjunto com duas técnicas de redução de dimensionalidade para assim identificar o melhor modelo dentro das abordagens de classificação propostas. Nele está presente os maiores resultados encontrados até o momento em relação a base de dados. Com base nos presentes resultados demonstrados, a MLP foi a principal técnica de classificação em conjunto com o Kernel PCA, e assim servindo como ponto inicial para este trabalho.

1.3 Motivação

A motivação para esta dissertação é impulsionada pela crescente prevalência e sofisticação dos ataques de *ransomware*, que têm causado preocupação em todo o mundo, incluindo empresas, instituições públicas e governos. Nos últimos anos, houve uma série de ataques de *ransomware* de alto perfil que demonstram a necessidade urgente de melhorar as técnicas de detecção e prevenção.

Um exemplo notável é o ataque WannaCry de 2017, que afetou mais de 200.000 computadores em 150 países causando interrupções significativas em hospitais, empresas e infraestruturas críticas (HARKNETT; SMEETS, 2022). Além disso, o ataque NotPetya no mesmo causou danos generalizados, afetando principalmente a Ucrânia, mas também se espalhando globalmente e impactando empresas multinacionais (GREENBERG, 2018).

Outro exemplo de ataque de *ransomware* é o Bad Rabbit, que ocorreu em outubro de 2017 e afetou principalmente a Rússia e a Ucrânia, mas também se espalhou para outros países europeus. O Bad Rabbit se espalhou por meio de sites comprometidos e usou uma técnica de infecção chamada "drive-by download"para infectar os sistemas das vítimas (MAMEDOV; SINITSYN; IVANOV, 2017).Além disso, o ataque Ryuk, que começou em agosto de 2018, teve como alvo várias organizações, incluindo hospitais, empresas e governos locais, causando interrupções significativas (POUDYAL; GUPTA; SEN, 2019).

Esses ataques destacam a importância de desenvolver novas abordagens e metodologias para melhorar a detecção e prevenção de *ransomware*. Com isso, esta dissertação buscará expandir o conhecimento atual sobre detecção de *ransomware*, explorando novas técnicas e abordagens que possam contribuir para aprimorar a eficácia das soluções existentes.

1.4 Justificativa

A literatura apresenta vários desafios para o crescimento do ransomware. Portanto, é necessário desenvolver metodologias que abordem esses desafios para combater o malware. Estudos na literatura usam algoritmos de machine learning para classificar o ransomware, como o trabalho de (ADAMU; AWAN, 2019), (ALENAZI; HINDI; ASSADHAN, 2019), e (SILVA et al., 2020).

Entretanto, há uma lacuna na investigação de soluções que considerem uma variedade de algoritmos e técnicas de redução de dimensionalidade para ganho computacional. Alguns estudos, incluindo o de (SILVA et al., 2020), também não fazem o ajuste de hiperparâmetros dos algoritmos.

O estudo de (FERNANDO; KOMNINOS; CHEN, 2020) aponta que a maioria dos estudos usam *Random Forest, SVM, Decision Tree*, e *Naive Bayes* como algoritmos de classificação. A exploração de técnicas de machine learning para detecção de ransomware é importante porque podem gerar modelos preditivos que aprendem o comportamento do malware e detectam variantes e famílias novas.

Este estudo é relevante para a comunidade científica porque explora a melhor configuração do uso conjunto da MLP, Kernel PCA e o uso do *oversampling*.

1.5 Objetivos

Pesquisadores como (CHEN, 2018), e (GANAPATHI; SHANMUGAPRIYA, 2020) argumentam que a inteligência artificial (IA) pode dar uma contribuição positiva na proteção da sociedade contra a crescente ameaça representada pelo ransomware. Esta dissertação tem como objetivo principal fazer a aplicação de um modelo composto pela MLP, Kernel PCA e *oversampling* classificando *ransomware* para uma análise dinâmica, em que é analisado o comportamento do *malware* em um ambiente já executado.

De maneira mais específica, têm-se os seguintes objetivos:

- Classificar de forma binária todas as classes do *dataset*;
- Classificar as 11 famílias de *ransomwares* incluindo os goodwares;
- Classificar as 11 famílias de *ransomware* excluindo os goodwares;
- Comparar resultados da literatura utilizando classificadores tradicionais de *machine learning*.

1.6 Metodologia

Nesta seção, a metodologia de pesquisa é explicada. A descrição da parte empírica da pesquisa é detalhada no capítulo 3. A presente pesquisa foi constituída em duas partes. A primeira caracteriza-se como um revisão da literatura, com o objetivo de se obter um entendimento sobre o tema e técnicas utilizadas, e a segunda parte é caracterizada pelo estudo de caso, a fim de proceder uma aproximação com o tema de pesquisa, a partir de uma série de procedimentos e experimentos relacionados ao aprendizado de máquina.

A metodologia começa com a preparação do ambiente virtual que serão realizados os experimentos. Logo após é realizada a obtenção das amostras (*ransomwares* e *goodwares*), e em seguida é realizado o pré-processamento dos dados, eliminando dados irrelevantes e duplicados. Seguidamente é feita a validação cruzada dos dados (10% de teste e 90 % de treino). Após, é realizada a aplicação do *RandomOverSampler* e em seguida a redução de dimensionalidade com o Kernel PCA (KPCA). Posteriormente aplica-se a MLP (*Multilayer perceptron*) para classificação. Por fim, a avaliação dos resultados é feita usando acurácia, matriz de confusão, Curva ROC, Curva *Precision-recall* e curva de aprendizado, além de outros gráficos de apoio a análise de eficiência do modelo. Em resumo, a metodologia é definida com base no diagrama da figura 2.

1.7 Organização do Trabalho

Estruturalmente, este trabalho é constituído por:

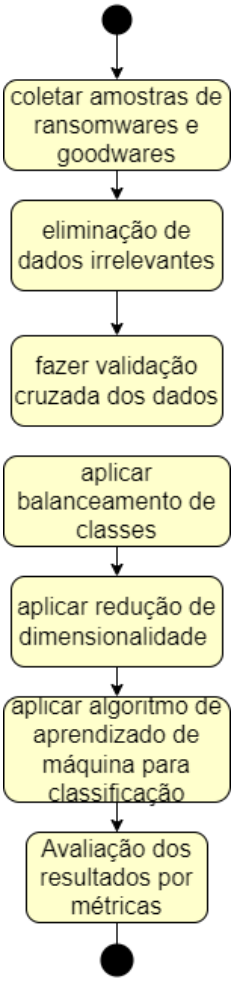

Figura 2 – Diagrama de atividades da metodologia aplicada. Fonte: Autor.

No Capítulo 1 são introduzidas as premissas que norteiam a origem desta dissertação, bem como os objetivos, justificativa e metodologia proposta, além da revisão bibliográfica, descrevendo trabalhos da literatura relacionados às abordagens propostas nesta dissertação, nos quais buscam apresentar o estado da arte neste campo de pesquisa;

No Capítulo 2 apresentam-se a definição do tema principal, os algoritmos de classificação de *machine learning* utilizados nesta dissertação, assim como as técnicas de redução de dimensionalidade, de modo a realizar apresentação do referencial teórico necessário para a compreensão desta dissertação;

No Capítulo 3 busca-se compreender a sequência de atividades desenvolvida, bem como detalhar cada tipo de classificação proposta;

No Capítulo 4 apresentam-se os desempenhos de cada experimento, tal como à análise dos resultados.

No Capítulo 5 apresentam-se às observações finais que resumem as principais descobertas desta pesquisa.

2 Referencial Teórico

2.1 Considerações Iniciais

Este capítulo apresenta conceitos relacionados ao desenvolvimento desta dissertação. Para tal, a seção 2.2 apresenta uma breve discussão sobre a definição de *ransowmare*, tal como sua evolução, enquanto que a seção 2.3 apresenta conceitos básicos relacionados às técnicas de redução de dimensionalidade, e por último a seção 2.4 apresenta os algoritmos de classificação de *machine learning* utilizados para os experimentos.

2.2 Definição de Ransomware

Ransomware é um tipo de malware que criptografa os arquivos de um sistema e exige um pagamento de resgate para descriptografá-los. De acordo com (O'GORMAN; MCDONALD, 2012), "*ransomware* é um tipo de malware que bloqueia o acesso ao computador da vítima ou criptografa arquivos no computador e exige que a vítima pague um resgate para remover a restrição."

Existem dois tipos principais de *ransomware*: *ransomware* de criptografia e *ransomware* de bloqueio de tela. O *ransomware* de criptografia, como o CryptoLocker, criptografa arquivos no computador da vítima e exige um pagamento, geralmente em bitcoin, para fornecer a chave de descriptografia. Se o pagamento não for feito, os arquivos permanecem criptografados e inacessíveis (O'GORMAN; MCDONALD, 2012).

O *ransomware* de bloqueio de tela, como o Locky, bloqueia o acesso ao sistema da vítima exibindo uma mensagem de resgate. Ele não criptografa arquivos, mas impede o acesso ao sistema até que o pagamento seja feito. A vítima não pode usar o computador e é forçada a pagar o resgate ou redefinir o sistema para remover a infecção (O'GORMAN; MCDONALD, 2012).

Scareware é outro tipo de malware que tenta assustar os usuários para que eles paguem por software falso de segurança cibernética. (KOK; AZWEEN; JHANJHI, 2020)

2.2.1 Evolução do Ransomware

A história do *ransomware* começa em 1989, quando o Dr. Joseph Popp distribuiu 20.000 disquetes infectados com o malware conhecido como Aids Trojan ou PC Cyborgpara para colegas pesquisadores que participaram da conferência sobre AIDS da Organização Mundial da Saúde (OMS).

O programa criptografou arquivos depois de ser reiniciado 90 vezes e exigiu um resgate de 378 (USD) para devolvê-los (BATES, 1990). A ideia de sequestrar arquivos e exigir um resgate é antiga, mas o *ransomware* tem sido uma presença constante nas manchetes nos últimos 5 anos.

Em 1996, Young e Yung explicaram as falhas do primeiro malware conhecido como PC Cyborg e apresentaram o conceito de "criptovirologia"(YOUNG; YUNG, 1996). Este conceito propõe o uso de esquemas de criptografia avançados, como criptografia de chave pública, tornando a defesa contra ataques de malware significativamente mais difícil. Eles desenvolveram um malware de prova de conceito para demonstrar esta ideia.

Embora a pesquisa de Young e Yung tenha sido publicada em 1996, o *ransomware* ainda era uma ameaça relativamente silenciosa devido à infraestrutura tecnológica subdesenvolvida e à escassez de conectividade à Internet. No entanto, com o crescimento da Internet e da conectividade, a ideia de extorsão digital foi trazida de volta. Em 2005, o número de usuários conectados à Internet alcançou 1 bilhão e com isso surgiu o GpCode - o primeiro ransomware moderno que usa o algoritmo Rivest-Shamir-Adleman (RSA) para criptografar arquivos do usuário (EMM, 2008).

Em 2008, o *ransomware locker - Randsom.C* se tornou o primeiro conhecido ransomware locker. Ele se espalhava por meio de downloads drive-by e e-mails maliciosos para infectar computadores de destino. Após a infecção, ele bloquearia o desktop e exibiria uma mensagem de resgate falsa, supostamente da Central de Segurança do Windows, exigindo que o usuário ligasse para um número de telefone com tarifa premium para desbloquear o sistema (IMAJI, 2019).

Em 2011-2012 houve o surgimento dos primeiros ransomwares de tela-cheia, como o "*Police Trojan*", que exibia uma mensagem falsa da polícia para extorquir dinheiro dos usuários.

Em 2013 houve o surgimento dos primeiros ransomwares de arquivo, como o "*CryptoLocker*", que criptografava arquivos do usuário e exigia um resgate para desbloqueá-los.

Em 2016 houve o surgimento dos primeiros ransomwares de nuvem, como o "*Petya*", que atacava serviços de nuvem e exigia um resgate para recuperar os dados.

Em 2017 houve o surgimento do primeiro ransomware baseado em inteligência artificial, o "*WannaCry*", que se espalhava rapidamente e exigia um resgate em Bitcoins.

Em 2018-2021 surgiu novos tipos de ransomwares, como o "*Ryuk*"e o "*Maze*", que visam empresas em vez de indivíduos e têm capacidade de exigir resgates maiores.

Durante a pandemia global em 2020, grupos de ransomware mudaram suas táticas, adotando cada vez mais a engenharia social temática COVID-19 para lançar seus ataques. Os cibercriminosos começaram a atacar de duas formas principais: por meio de um falso mapa de calor do COVID-19 que baixava o ransomware ao ser clicado, e através de documentos maliciosos, aparentemente fornecendo informações sobre saúde e segurança relacionadas à

pandemia.

A necessidade de centros de saúde aumentou devido à pandemia, tornando-os vulneráveis a ataques cibernéticos. Como resultado, houve um aumento no número de ataques de ransomware a organizações de saúde, incluindo a surgência de uma nova família de ransomware, chamada Corona, direcionada a hospitais (WUEST, 2020). Esta família criptografava os registros de saúde dos pacientes e exibia uma mensagem de resgate COVID-19, cobrando um pagamento em Bitcoin.

Em resumo, a evolução do ransomware tem sido uma das tendências mais preocupantes na segurança cibernética nos últimos anos. A seguir estão algumas tendências importantes na evolução do ransomware até 2022:

- Cibercrime como serviço: os grupos de cibercriminosos estão oferecendo serviços de ransomware como uma forma de obter lucros;
- Ataques direcionados: os ataques de ransomware estão se tornando mais direcionados, visando empresas específicas e setores críticos, como saúde e infraestrutura;
- Criptografia avançada: os cibercriminosos estão usando criptografia avançada para tornar os dados inacessíveis e exigir resgates maiores;
- Exploração de vulnerabilidades zero-day: os grupos de cibercrime estão explorando vulnerabilidades desconhecidas (zero-day) para aumentar a efetividade dos ataques de ransomware;
- Ransomware em nuvem: os ataques de ransomware estão se expandindo para a nuvem, ameaçando as empresas que armazenam dados e aplicativos na nuvem.

A tabela 1 apresenta um resumo da evolução das principais famílias de *ransomware*, e descreve o ano que foi visto pela primeira vez, o tipo de infecção, a plataforma, o tipo de comunicação com o servidor do *hacker*, bem como o tipo de encriptação, em que pode ser RSA (Algoritmo de Criptografia Assimétrica), AES (Algoritmo de Criptografia Simétrica), MBR (Registro Mestre de Inicialização) e ECC (Criptografia de Curva Elíptica), o tipo de destruição, o uso da extração e trancamento, e o tipo de extorção.

2.3 Técnica de Redução de Dimensionalidade

A mineração de dados é o processo de extração de informações e conhecimento úteis de grandes quantidades de dados. A "maldição da dimensionalidade"é um dos desafios enfrentados na mineração de dados. Isso refere-se à ideia de que quanto mais variáveis (ou dimensões) são adicionadas aos dados, mais difícil se torna para os algoritmos de aprendizado de máquina encontrar padrões significativos nos dados. Isso se deve ao fato de que a quantidade de dados

Tabela 1 – Resumo das principais famílias de Ransomware.

Família	Ano	Infecção	Plataforma	C&C Comm.	Encriptação	Destruição	Extração	Trancar	Extorção
PC CYBORG	1989	Phishing	Windows	Nenhum	Persnonalizado	Copiar			Dinheiro
GPCode	2008	Drive-by-download	Windows	Nenhum	RSA	Copiar			C. pré pago
Archiveus	2008	Drive-by-download	Windows	Nenhum	RSA	Deletar			C.pré pago
Randsom.C	2008	Spam	Windows	Nenhum	Nenhum	Nenhum	X		SMS
Seftad	2008	Spam	Windows	Nenhum	MBR	Copiar		X	C.pré pago
Krotten	2008	Spam	Windows	Nenhum	RSA	Copiar			C.pré pago
Urausy	2009	PhishinX	Windows	Nenhum	Nenhum	Nenhum		X	C.pré pago
Winlock	2010	Spam	Windows	Codificado	Nenhum	Nenhum	X		SMS
Reveton	2012	PhishinX	OS-AXnostic	Nenhum	Nenhum	Nenhum	X	X	C.pré pago
CryptoLocker	2013	PhishinX	Windows	Codificado	RSA	Copiar	X		Bitcoin
CryptoWall	2013	PhishinX	Windows	Codificado	RSA	Copiar			C.pré pago
FakeDefender	2013	App Malicioso	Android	Nenhum	Nenhum	Nenhum		X	C.pré pagos
Lockdroid	2014	App Malicioso	Android	Nenhum	Nenhum	Nenhum		X	C.pré pagos
SimpLocker	2014	App Malicioso	Android	Codificado	AES	Copiar			Bitcoin
TorrentLocker	2014	PhishinX	Windows	Dinâmico	AES+RSA	Deletar			C.pré pago
TrolDesh	2014	PhishinX	Windows	Codificado	AES	Deletar			Bitcoin
CryptoDefense	2014	PhishinX	Windows	Nenhum	RSA	Deletar			Bitcoin
CTBLocker	2015	PhishinX	Windows	Nenhum	ECC	Deletar			Bitcoin
TeslaCrypt	2015	Exploit kits	Windows	Codificado	AES	Deletar			Bitcoin
Fusob	2015	PhishinX	Mobile	Nenhum	Nenhum	Nenhum		X	Giftcards
Chimera	2015	PhishinX	Windows	Codificado	AES	Deletar			Bitcoin
LinuxEncoder	2015	Vulnerability	Linux	Codificado	AES+RSA	Deletar			Bitcoin
Ransom32	2016	Malvertisement	OS-AXnostic	Codificado	AES	Copiar			Bitcoin
Dharma	2016	RDP	Windows	Codificado	AES	Deletar			Bitcoin
Locky	2016	PhishinX	Windows	Dinâmico	AES+RSA	Deletar			Bitcoin
Cerber	2016	PhishinX	Windows	Nenhum	AES	Copiar			Bitcoin
JiXsaw	2016	PhishinX	Windows	Codificado	AES+RSA	Deletar			Bitcoin
KeRanXer	2016	App Malicioso	macOS	Codificado	AES+RSA	Deletar			Bitcoin
Petya	2016	Exploit kits	Windows	Codificado	AES	Copiar		X	Bitcoin
DMALocker	2016	Exploit kits	OS-AXnostic	Codificado	AES	Deletar			Bitcoin
SaXe	2017	Drive-by-download	Windows	Codificado	AES+RSA	Deletar			Bitcoin
BadRabbit	2017	Drive-by-download	Windows	Dinâmico	AES+RSA	Deletar			Bitcoin
WannaCry	2017	Vulnerability	Windows	Codificado	AES+RSA	Deletar			Bitcoin
XoldenEye	2017	Exploit kits	Windows	Codificado	AES	Copiar			Bitcoin
SamSam	2018	RDP	Windows	Codificado	RSA	Deletar			Bitcoin
XandCrab	2018	Drive-by-download	OS-AXnostic	Codificado	AES	Copiar			Bitcoin
Sodikonibi	2019	Exploit kits	Windows	Codificado	AES+ECC	Copiar		X	Bitcoin
Robbinhood	2019	RDP	Windows	Codificado	AES+RSA	Deletar	X		Bitcoin
Maze	2019	Exploit kits	OS-AXnostic	Codificado	AES+RSA	Copiar			Bitcoin
Ryuk	2019	RDP	OS-AXnostic	Codificado	AES+RSA	Copiar			Bitcoin
MeXaCortex	2019	Exploit-kits	OS-AXnostic	Codificado	AES	Copiar			Bitcoin
LockerXaXa	2019	PhishinX	Windows	Nenhum	AES+RSA	Deletar			Bitcoin
Ekans	2019	Vulnerability	ICS	Codificado	AES+RSA	Deletar			Bitcoin
PureLocker	2020	Vulnerability	PC	Dinâmico	AES	Deletar			Proton
Tycoon	2020	Vulnerability	Windows	Dinâmico	AES	Deletar			Proton
CovidLock	2020	App Malicioso	Android	Nenhum	Nenhum	Nenhum		X	Bitcoin
Corona	2020	PhishinX	Windows	Nenhum	AES+RSA	Copiar	X		Bitcoin

Fonte: Adaptado de (OZ et al., 2021)

necessários para treinar um modelo aumenta exponencialmente com o aumento de dimensões (ANOWAR; SADAOUI, 2021).

A redução de dimensionalidade é uma técnica usada para lidar com esse problema, que visa representar dados em uma dimensão mais baixa, preservando ao mesmo tempo o máximo possível de informação relevante. Uma das técnicas de redução de dimensionalidade é o KPCA (Kernel Principal Component Analysis) (MAATEN; POSTMA; HERIK, 2009).

Uma vez que o conjunto de dados têm um grande número de recursos (30.967), foi realizado a aplicação da técnica de redução KPCA com o objetivo de encontrar um subconjunto ideal de recursos. A seguir há uma breve descrição da técnica que foi utilizada.

2.3.1 Kernel PCA

A técnica de Redução de Dimensionalidade de Componentes Principais baseada em Kernel (KPCA) é uma técnica não-linear de análise de dados que visa encontrar uma representação de alta dimensão dos dados em uma dimensionalidade mais baixa, preservando a maior quantidade possível de informação original. Diferentemente da PCA tradicional, a KPCA usa uma transformação baseada em kernel para aplicar o mapeamento não-linear dos dados para uma espaço de alta dimensão, onde é possível encontrar a representação mais significativa em uma dimensionalidade mais baixa.

A KPCA usa o conceito de funções de kernel para medir a similaridade entre pontos de dados. O resultado final da aplicação da técnica é uma série de componentes principais que podem ser interpretadas como novas características lineares dos dados. Estas características são obtidas combinando as características originais dos dados de uma forma não-linear, permitindo que as relações não-lineares entre as características sejam capturadas.

A KPCA é amplamente utilizada em várias aplicações, incluindo análise de imagem, processamento de sinais, *clustering* e classificação. É uma técnica particularmente útil quando os dados apresentam uma estrutura não-linear e é difícil de se representar em uma dimensionalidade mais baixa usando técnicas lineares. Além disso, a KPCA é uma técnica computacionalmente eficiente, sendo muito rápida para processar grandes quantidades de dados (LEVADA, 2020).

As etapas da técnica de redução de dimensionalidade KPCA são descritas a seguir, ainda de acordo com (LEVADA, 2021).

1. Construir a matriz kernel K a partir do dataset de treino X : $K_{i,j} = K\left(\vec{x}_i, \vec{x}_j\right)$
2. Computar a matrix Gram $\tilde{K}$ usando $\tilde{K} = K - 1_n K - K 1_n + 1_n K 1_n$ onde 1_n é a matriz $n \times n$ com todos os elementos igual a $\frac{1}{n}$
3. Usar $K\vec{\alpha}_k = \left(\lambda_k n\right)\vec{\alpha}_k$, onde $\vec{\alpha}_k$ são os autovetores da matriz kernel, para resolver os vetores $\vec{\alpha}_k$ (usando $\tilde{K}$ no lugar do K)
4. Computar o Kernel PCA $y_k\left(\vec{x}\right)$ usando $y_k\left(\vec{x}\right) = \sum_{i=1}^{n} \alpha_{ki} K\left(\vec{x}, \vec{x}_i\right)$ para $k = 1, 2, \cdots, d$

2.4 Algoritmo de Classificação de Machine Learning

A classificação é uma tarefa de aprendizado supervisionado em que o objetivo é prever a categoria ou classe a que uma determinada entrada pertence. Existem vários algoritmos de classificação de aprendizado de máquina, incluindo: Regressão Logística (RL), Árvores de Decisão, *Random Forest*, *Naive Bayes*, SVM (*Support Vector Machine*) KNN (*k-Nearest Neighbors*), e MLP (*Multi-Layer Perceptron*). A seguir há um breve resumo da MLP - algoritmo utilizado para classificação de *ransomware* nessa pesquisa.

2.4.1 Multi-layer Perceptron

O algoritmo *Multi-layer Perceptron (MLP)* é um tipo de rede neural feedforward composta por uma ou mais camadas escondidas de nós conectados por pesos sinápticos (HAYKIN, 2009). Em uma tarefa de classificação, o MLP é treinado com exemplos de treinamento que possuem entradas (características) e saídas (classes). Durante o treinamento, a rede aprende a associar as entradas aos rótulos de saída corretos, e essa associação é baseada nas associações aprendidas pelos pesos sinápticos.

Após o treinamento, a rede MLP pode ser usada para prever a classe de uma entrada não vista antes, baseada nas associações aprendidas durante o treinamento (BISHOP; NASRABADI, 2006). O número de nós na camada de saída depende do número de classes na tarefa de classificação.

O MLP é um algoritmo flexível que pode ser usado em uma ampla variedade de tarefas de classificação, incluindo a classificação de imagens, texto e sinais. A seguir há uma descrição detalhada do funcionamento do MLP:

1. Camada de Entrada: A camada de entrada consiste em nós que representam os atributos ou características de um exemplo de treinamento. Cada nó da camada de entrada representa uma característica e o valor associado é o valor dessa característica para o exemplo em questão;

2. Camada de Saída: A camada de saída consiste em um ou mais nós que representam a previsão ou classificação feita pela rede. O número de nós na camada de saída depende do número de classes na tarefa de classificação ou da dimensão da previsão para tarefas de regressão;

3. Pesos Sinápticos: Cada nó da camada escondida e da camada de saída está conectado a todos os nós da camada anterior por meio de pesos sinápticos. O valor dos pesos é atualizado durante o treinamento da rede para melhorar a precisão das previsões;

4. Treinamento: Durante o treinamento, a rede é alimentada com exemplos de treinamento e as saídas são comparadas com as saídas desejadas. A diferença entre as saídas desejadas e as saídas reais é usada para atualizar os pesos sinápticos da rede, tornando-a mais precisa. O treinamento é repetido várias vezes até que a precisão da rede atinja um nível satisfatório;

As vantagens do Perceptron multicamadas são (LEARN, 2021a):

- Flexibilidade: O MLP é capaz de resolver uma ampla gama de problemas de classificação e pode ser usado em vários tipos de dados, incluindo imagens, texto e sinais;
- Capacidade de Aprendizado: O MLP é capaz de aprender associações complexas entre entradas e saídas e pode ser ajustado para lidar com diferentes graus de complexidade;

- Robustez: O MLP é robusto a ruído nos dados e pode lidar com ausência de dados;
- Capacidade de aprender modelos em tempo real (aprendizado online).

As principais desvantagens do *Multi-layer Perceptron* incluem:

- Complexidade computacional: O treinamento de uma rede MLP pode ser computacionalmente intensivo e exigir muito tempo e recursos para ajustar os pesos da rede;
- Mínimos locais: A MLP pode ser propenso a se ater a mínimos locais durante o treinamento, o que pode resultar em uma menor precisão do modelo;
- Dificuldade com relações não-lineares: O MLP pode ter dificuldade em lidar com relações não-lineares entre entradas e saídas, o que pode afetar a precisão do modelo;
- Overfitting: Se a rede MLP tiver muitas camadas ou muitos neurônios, ela pode se ajustar demais aos dados de treinamento, o que pode resultar em um *overfitting* e uma menor precisão em dados não vistos antes;
- Dependência da escala dos dados: A MLP é sensível à escala dos dados de entrada e pode ser necessário normalizar ou padronizar os dados antes de usá-los como entrada.

2.5 *Random oversampling*

O balanceamento de dados é uma etapa importante no pré-processamento de dados para problemas de classificação com dados desbalanceados. Duas abordagens comuns para lidar com o desbalanceamento de dados são o *oversampling* e o *undersampling*.

O *oversampling* aumenta a quantidade de instâncias da classe minoritária duplicando ou gerando novos exemplos sintéticos. Isso equilibra as classes aumentando a quantidade de dados da classe minoritária e uma das técnicas utilizadas é a *Random oversampling*. O *undersampling* reduz a quantidade de instâncias da classe majoritária removendo aleatoriamente exemplos. Isso equilibra as classes diminuindo a quantidade de dados da classe majoritária.

Random oversampling é uma técnica de reamostragem que foi projetada para equilibrar a distribuição de classe desequilibrada duplicando amostras da classe minoritária para um conjunto de dados de classificação (NAIM; HANNAN; KABIR, 2022). Em *datasets* desbalanceados, uma classe é representada por muito mais exemplos do que as outras classes. Isso pode prejudicar o desempenho dos algoritmos de aprendizado de máquina, pois eles podem ser tendenciosos em relação à classe majoritária.

Esta técnica funciona gerando aleatoriamente novas instâncias da classe minoritária. O número de novas instâncias é escolhido de forma que a classe minoritária tenha o mesmo número de instâncias que a classe majoritária. Essas novas instâncias geradas aleatoriamente são

adicionadas ao conjunto de dados. Este processo é repetido para todas as classes minoritárias até que o conjunto de dados esteja equilibrado. O conjunto de dados equilibrado resultante tem o mesmo número de instâncias para cada classe, eliminando assim o desequilíbrio entre classes.

Este é um método que equilibra o conjunto de dados desequilibrado repetindo amostras da classe minoritária até que a proporção de classe desejada seja obtida (GANGANWAR, 2012). O conjunto de dados equilibrado pode então ser usado para treinar modelos de aprendizado de máquina para melhorar o desempenho preditivo na classe minoritária. Essa técnica é aplicada apenas ao conjunto de treino, mantendo o conjunto de teste original.

3 Abordagem Proposta

3.1 Considerações Iniciais

Neste capítulo é discutido os passos executados no desenvolvimento da dissertação, assim como uma compreensão abrangente da metodologia utilizada, incluindo as ferramentas empregadas, e métricas de avaliação para alcançar os objetivos das classificações propostas.

3.2 Visão Geral da Metodologia

Nesta pesquisa, a base de dados, formada por um conjunto de *features* (API, DROP, REG, FILES_EXT, DIR, STR) com 942 *goodwares* e 585 amostras de ransomwares, é submetida a um pré-processamento para eliminar dados duplicados e irrelevantes. Em seguida, a validação cruzada é realizada para obter os dados de treinamento e teste para o treinamento e validação dos modelos.

A construção do modelo de classificação é feita combinando a técnica de balanceamento de classes *RandomOverSampler*, o Kernel PCA para a realização de redução de dimensionalidade e por fim a MLP como o agente classificador. Com os resultados dos testes, é realizado o ajuste dos hiperparâmetros, armazenando a melhor configuração encontrada para o modelo. Por fim, compara-se os resultados dos modelos construídos e o otimiza-se para determinar o melhor modelo. A ilustração da metodologia é apresentada na Figura 3.

A justificativa pelo uso de uma única técnica de balanceamento de classes em relação a outras se dá por conta que em testes preliminares de diferentes abordagens, tanto de subamostragem quanto de superamostragem, relevaram que a única que tinha reais contribuições na melhoria dos indicadores em análise foi a de *RandomOverSampler*. Enquanto que o uso direto das técnicas de Kernel PCA e a da MLP se dá pelos comprovados resultados do trabalho de (PEREIRA; JúNIOR, 2022) em comparação a outras técnicas de classificação e redução de dimensionalidade.

3.3 Classificações Propostas

A aprendizagem de máquina é o processo de previsão ou inferência baseado no problema ou questão em questão e nos dados disponíveis. Por exemplo, a tarefa de classificação classifica dados em categorias e a tarefa de agrupamento agrupa dados de acordo com sua semelhança.

A aprendizagem de máquina se baseia em padrões dos dados, em vez de ser programada explicitamente (QUINTANILLA, 2021). As seções 3.3.1, 3.3.2 e 3.3.3 apresentam as cinco

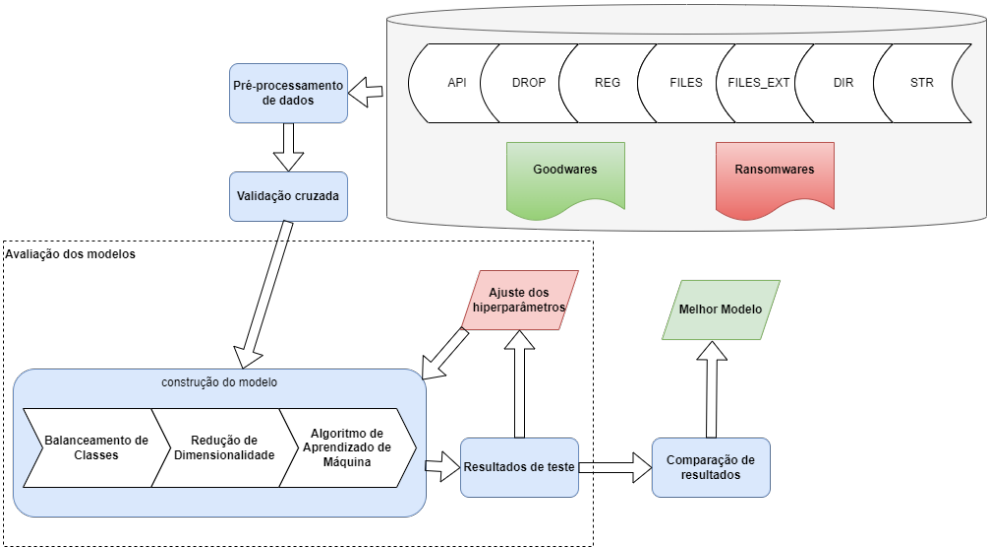

Figura 3 – Ilustração da metodologia aplicada. Fonte: Autor.

tarefas de classificação realizadas como tarefas de aprendizado de máquina que foram treinadas pelos algoritmos classificadores.

3.3.1 Classificação Binária

Classificação binária é um tipo de problema de classificação em aprendizado de máquina, onde o objetivo é prever se um determinado item pertence a uma das duas classes possíveis. Essas duas classes são geralmente rotuladas como "verdadeiro"e "falso", "positivo"e "negativo", ou "1"e "0".

Por exemplo, em uma tarefa de classificação binária de e-mail, o objetivo seria prever se um e-mail é "spam"ou "não-spam". Em uma tarefa de diagnóstico médico, o objetivo seria prever se um paciente tem ou não uma doença. E em uma tarefa de fraude financeira, o objetivo seria prever se uma transação é "fraude"ou "não-fraude".

Neste trabalho, o cenário de classificação binária é a determinação se uma instância é um ransomware ou goodware, representados por 1 ou 0 respectivamente."O esquema desta classificação pode ser visto na Figura 4.

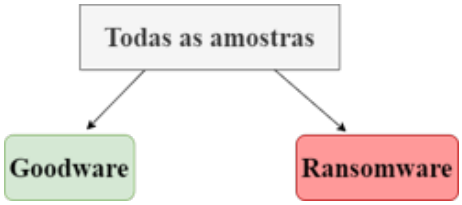

Figura 4 – Esquema da Classificação Binária. Fonte: Autor.

3.3.2 Classificação de Família

Classificação multiclasse é um tipo de problema de classificação em aprendizado de máquina, onde o objetivo é prever a classe de um determinado item em uma lista de mais de duas classes possíveis.

Por exemplo, em uma tarefa de reconhecimento de objetos, o objetivo seria prever a classe de um objeto a partir de uma lista de categorias, como "pessoa", "cachorro", "gato", etc. Em uma tarefa de classificação de sentimentos, o objetivo seria prever se um comentário é "positivo", "negativo"ou "neutro". E em uma tarefa de classificação de frutas, o objetivo seria prever a fruta a partir de uma lista de categorias, como "maçã", "banana", "laranja", etc.

O aprendizado de máquina é frequentemente utilizado para resolver problemas de classificação multiclasse, usando algoritmos como regressão logística, árvores de decisão, redes neurais e outros. Em problemas de classificação multiclasse, também é comum usar a matriz de confusão para visualizar a performance do modelo, mostrando quantos exemplos de cada classe foram corretamente ou incorretamente classificados.

Nessa pesquisa, a classificação multiclasse tem como objetivo identificar a qual das onze famílias de *ransomware* (Tabela 2) uma dada instância pertence. Cada família é representada por um número inteiro de 1 a 11, enquanto a classe dos *goodwares* é representada pelo número 0. A representação visual desta classificação pode ser vista na Figura 5."

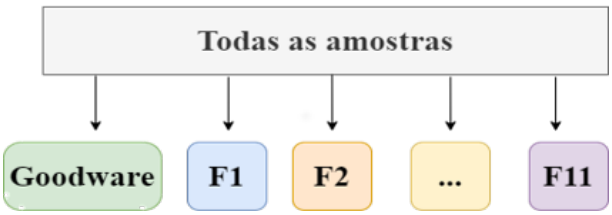

Figura 5 – Esquema da Classificação de Família. Fonte: Autor.

3.3.3 Classificação de Família sem Goodwares

Nesta classificação, apenas as 11 classes de *ransomwares* são consideradas e o objetivo é determinar a qual classe cada instância pertence. Instâncias pertencentes à classe de goodwares não são utilizadas. O esquema desta classificação pode ser visto na Figura 6."

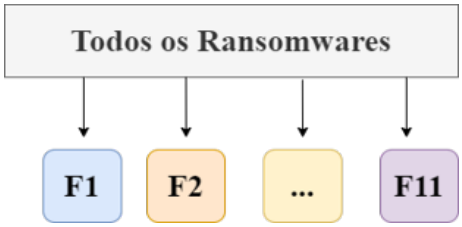

Figura 6 – Esquema da Classificação de Família sem Goodwares. Fonte: Autor.

3.4 Ferramentas Utilizadas

Os algoritmos mencionados neste trabalho foram implementados usando a linguagem Este trabalho utilizou a linguagem Python para implementar seus algoritmos. Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível, com semântica dinâmica. Sua sintaxe clara e fácil de compreender aumenta a legibilidade, minimizando os custos de manutenção. Além disso, a capacidade de usar módulos e pacotes incentiva a modularidade e reutilização de código. O interpretador e a biblioteca padrão estão disponíveis gratuitamente em todas as plataformas principais e podem ser distribuídos livremente (PYTHON, 2021).

Para implementar seus algoritmos, o trabalho também usou a biblioteca *Scikit-Learn*. Essa biblioteca teve início como um projeto do *Google Summer of Code project* em 2007, liderado por David Cournapeau, e sua primeira versão foi lançada em 2010. Atualmente, é mantido como um projeto colaborativo pela comunidade, que recebe doações institucionais e privadas para garantir sua sustentabilidade (LEARN, 2021b). A biblioteca oferece muitos algoritmos de aprendizado supervisionado e não supervisionado, facilitando o desenvolvimento de trabalhos na área de aprendizado de máquina.

As funcionalidades fornecidas pela *Scikit-Learn* incluem:

- Regressão linear, regressão logística, regressão polinomial, regressão ridge, entre outros;
- Classificação, incluindo classificação *K-Nearest Neighbors (KNN)*, Árvore de Decisão, *Naive Bayes*, entre outros;
- Agrupamento, incluindo K-Means, Hierarquical Clustering, entre outros;
- Redução de dimensionalidade, incluindo Análise de Componentes Principais (PCA), técnicas de seleção de características, etc;
- Modelos de mistura, incluindo *Gaussian Mixture Models* (GMM), entre outros;
- Árvores de decisão, incluindo *Random Forest, Gradient Boosting*, etc.
- Métodos de validação cruzada, como *K-fold, leave-one-out*, entre outros;
- Ferramentas para seleção de modelos, incluindo *GridSearchCV* e *RandomizedSearchCV*;
- Seleção de modelo (Curvas de aprendizado, validação cruzada e etc.);
- Seleção de modelo (Curvas de aprendizado, validação cruzada e etc.).

Nesta pesquisa, foi utilizado o *Jupyter Notebook*, um aplicativo *web-based* que permite a estruturação e execução fáceis de partes de código de maneira amigável ainda de acordo com (LEARN, 2021b). Durante a pesquisa, o *Jupyter* foi utilizado para as fases de pré-processamento de dados e treinamento de algoritmos para os modelos de classificação.

Além disso, foi utilizado o *Google Collab* como ambiente de execução, um serviço de armazenamento em nuvem de notebooks para a criação e execução de códigos em *Python* diretamente no navegador, sem a necessidade de instalação de software em uma máquina. O *Google Collab* oferece processamento em Tensor *Processing Unit* (TPU) que pode acelerar o processamento de treinamento de programas e inferência (STARTUPI, 2017).

3.5 Sequência de Atividades

Nesta seção, é descrita detalhadamente as atividades e os passos realizados nas experimentações práticas. Aqui é apresentada uma visão geral das principais etapas que foram seguidas ao longo da dissertação.

3.5.1 Coleta dos Dados

Este trabalho utiliza amostras de ransomware do VirusShare, um site que mantém um banco de dados atualizado de amostras de malware para diversos sistemas operacionais. Essas amostras foram obtidas de (SGANDURRA et al., 2016) que propôs uma solução anti-ransomware chamada Elderan, conforme discutido na revisão de literatura. Elas foram executadas por 30 segundos em um ambiente de análise dinâmica para registrar seu comportamento.

O conjunto de dados, composto por 582 amostras de ransomware de 11 classes diferentes e 942 goodwares, foi analisado em fevereiro de 2016 e contém 30.967 atributos categorizados em sete grupos. De acordo com (ABBASI; AL-SAHAF; WELCH, 2020), os cibercriminosos têm criado novas famílias de ransomware para vítimas específicas, como *Ryuk*, que foi projetado para atingir apenas empresas em 2019.

As amostras de ransomware coletadas são representativas das versões e variantes mais comuns encontradas na natureza, sendo a maioria do tipo cryptoransomware. Para treinamento e teste, foi realizada uma validação cruzada estratificada por divisão aleatória em quatro pacotes. Essa divisão preserva a mesma proporção de cada classe de destino em relação ao conjunto completo. A Tabela 2 apresenta um resumo das instâncias de dados antes e depois do pré-processamento, que será descrito na Seção 3.5.2.

3.5.2 Pré-Processamento dos Dados

Primeiramente, foram removidas instâncias duplicadas com a mesma classe, resultando em 108 remoções. Em seguida, foram excluídas instâncias duplicadas com classes diferentes, sendo 4 remoções em total, resultando em 112 instâncias removidas do dataset. Este processo foi realizado para evitar dados redundantes e possíveis conflitos na classificação, uma vez que não é possível diferenciar qual classe o classificador deve considerar. A Tabela 3 apresenta um resumo das features do conjunto de dados.

Tabela 2 – Resumo das instâncias utilizadas.

Família	**Instâncias do Dataset Original**	**Instâncias após o Pré-Processamento**
Citroni	50	34
CryptLocker	107	100
CryptoWall	46	37
Kollah	25	20
Kovter	64	57
Locker	97	96
Matsnu	59	46
Pgpcoder	4	4
Reverton	90	56
TeslaCrypt	6	6
Trojan-Ransom	34	28
Goodware	942	928
Total	1524	**1412**

Tabela 3 – Resumo do grupo de features do conjunto de dados.

Grupo	**Descrição**	**# Features**
API	Chamadas de API	232
DROP	Extensões de arquivos removidos	346
REG	Operações de Registro Chave	6622
FILES	Operações de arquivos	4141
FILES_EXT	Extensões de arquivos manipulados	935
DIR	Operações em diretório de arquivos	2424
STR	Strings embutidas	16267
Total	-	30967

Os dados de entrada foram codificados como valores binários, onde 0 indica a ausência e 1 indica a presença de uma chamada de sistema. Devido à variabilidade da acurácia de cada técnica de classificação com diferentes quantidades de componentes, foi realizado um ajuste do número de componentes usados aleatoriamente, variando de 1 ao máximo total para todas as técnicas.

3.5.3 Ajustes de Hiperparâmetros

A seleção de modelos para algoritmos de aprendizado de máquina envolve geralmente a otimização de um critério que permite escolher o modelo adequado (CAWLEY; TALBOT, 2010b). Para melhorar o desempenho e obter uma característica de generalização, são necessários ajustes adicionais. Os parâmetros considerados para a MLP e kernel PCA estão descritos na Figura 8 e a faixa dos valores usados nos experimentos está presente na documentação da biblioteca Scikit-learn. Algumas técnicas de ajuste, como validação cruzada e Grid Search com ajustes de hiperparâmetros para reduzir o overfitting, serão abordadas adiante.

- *Cross-Validation*: Para avaliar a capacidade de generalização dos resultados, a validação

cruzada (CV) é amplamente utilizada em processos de aprendizado de máquina (ALLIX, 2015). A validação cruzada minimiza qualquer viés na divisão dos dados e maximiza os resultados estatísticos obtidos nas interações. A Figura 7 apresenta um exemplo de CV com 5-fold, onde o conjunto de dados é dividido em 5 partes iguais, sendo que 4 são usados para treinamento e 1 para testes. Na presente pesquisa, para classificação binária, foi utilizado CV Stratified com 10-fold, enquanto que para as outras classificações foram usados 4-fold devido a existirem apenas 4 instâncias na classe de família (Pgpcoder), conforme apresentado na Tabela 2. O CV Stratified mantém a proporção das classes em todos os folds.

- Hiperparâmetros: Avaliar os melhores parâmetros para um modelo de machine learning é um desafio, uma vez que cada algoritmo de aprendizado de máquina requer diferentes constantes, taxas de aprendizado e outros fatores que podem afetar sua performance, conhecidos como hiperparâmetros (CAWLEY; TALBOT, 2010a). Para aprimorar os resultados, a biblioteca "RandomizedSearchCV"foi implementada na pesquisa, usando validação cruzada de pesquisa aleatória. O metaestimador de busca aleatória é um algoritmo que treina e avalia diferentes versões de um modelo com combinações de hiperparâmetros selecionadas aleatoriamente, para escolher a melhor combinação. Isso resulta em um modelo treinado com hiperparâmetros quase ideais (LEDOUX, 2019).

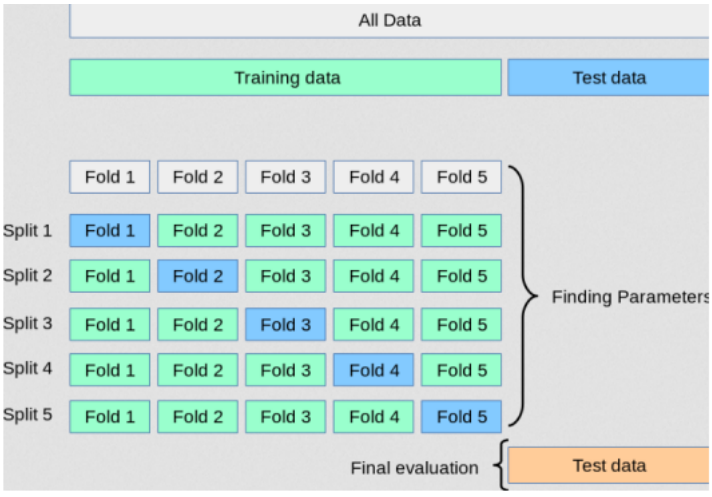

Figura 7 – Exemplo de Cross-validation. Fonte: (PEDREGOSA et al., 2011).

MLP

- Activation [identity, logistic, tanh, relu]
- Hidden layer sizes [(50,50,50), (50,100,50), (100)]
- Solver [sgd, adam]
- Learning_rate [constant, invscaling, adaptive]

KPCA

- N componentes [1-800]

Figura 8 – Parâmetros da MLP e Kernel PCA. Fonte: Autor.

3.5.4 Avaliação dos Resultados

Para avaliar de forma efetiva a metodologia proposta, algumas métricas foram utilizadas, como a seguir:

- Acurácia foi utilizada como a métrica de avaliação em aprendizado de máquina. Conforme destacado por (POWERS; AILAB, 2011), a acurácia é uma métrica comumente utilizada em aprendizado de máquina para medir a capacidade de um modelo de classificação de prever corretamente a classe de um determinado conjunto de dados de teste. Ela é definida como a proporção de previsões corretas feitas pelo modelo sobre o total de previsões realizadas. Em outras palavras, a acurácia mede quanto o modelo está acertando ao prever as classes dos dados de teste. As discussões a respeito dos resultados dessa métrica serão profundamente abordadas na seção 4.3.

  A acurácia pode ser calculada a partir da equação 3.1.

$$Acurácia = \frac{VN + VP}{VP + VN + FP + FN} \tag{3.1}$$

  em que:
  VN = Verdadeiro Negativo
  VP = Verdadeiro Positivo
  FN = Falso Negativo
  FP = Falso Positivo

- A matriz de confusão também foi utilizada para avaliar as classificações multi classe. Ela é uma tabela usada para avaliar o desempenho de um modelo de classificação em aprendizado de máquina. Ela mostra quantos exemplos de cada classe foram corretamente ou incorretamente classificados pelo modelo. A matriz de confusão é geralmente representada como uma tabela quadrada com linhas e colunas correspondentes às classes de destino. As células da tabela contêm a contagem de exemplos que foram classificados corretamente (verdadeiros positivos) ou incorretamente (falsos positivos e falsos negativos) pelo modelo. A matriz de confusão é uma ferramenta útil para compreender o comportamento do modelo em relação a cada classe. Ela permite avaliar a capacidade do modelo de distinguir entre as classes, bem como identificar pontos fortes e fracos do modelo. Por exemplo, se o modelo tem dificuldade em distinguir entre duas classes específicas, isso pode ser indicado por uma alta contagem de falsos positivos e falsos negativos para essas classes na matriz de confusão. A figura 9 mostra como ela pode ser interpretada.

- A curva ROC (Receiver Operating Characteristic) é uma representação gráfica comum usada para avaliar o desempenho de modelos de classificação binária em aprendizado de máquina. A curva ROC mostra a relação entre a taxa de verdadeiros positivos (também

	Verdadeiro Positivo	Verdadeiro Negativo
Falso Positivo	VP	FN
Falso Negativo	FP	VN

Figura 9 – Interpretação da matriz de confusão.
Fonte: Autor.

conhecida como recall) e a taxa de falsos positivos. A taxa de verdadeiros positivos representa a proporção de exemplos positivos que foram corretamente classificados como positivos pelo modelo, enquanto a taxa de falsos positivos representa a proporção de exemplos negativos que foram incorretamente classificados como positivos. A curva ROC é plotada com a taxa de verdadeiros positivos no eixo y e a taxa de falsos positivos no eixo x. Quanto mais próximo a curva ROC estiver da origem (0,0), menos exemplos serão classificados incorretamente como positivos. Quanto mais próximo a curva ROC estiver da esquerda superior, melhor será o desempenho do modelo, pois a taxa de verdadeiros positivos será alta e a taxa de falsos positivos será baixa.

- A curva precision-recall é uma representação gráfica que mostra a relação entre a precisão e o recall de um modelo de classificação binária. É uma ferramenta útil para avaliar o desempenho de um modelo em diferentes pontos de equilíbrio entre precisão e recall. A curva precision-recall é plotada ao variar o limiar de classificação do modelo, ou seja, o valor mínimo que uma previsão deve ter para ser considerada como positiva. Cada ponto na curva representa a precisão e o recall para um dado limiar. A curva precision-recall é geralmente plotada com a precisão no eixo y e o recall no eixo x. A área sob a curva (AUC) é uma medida comum para avaliar a qualidade geral da curva, quantificando o quanto o modelo é capaz de equilibrar precisão e recall. A curva precision-recall é particularmente útil em casos em que o desbalanceamento de classe é uma preocupação, pois permite avaliar o desempenho do modelo em diferentes pontos de equilíbrio entre precisão e recall. Além disso, a curva precision-recall pode ser mais informativa do que a acurácia para modelos com baixa acurácia geral, como por exemplo, quando há muitos exemplos negativos na base de dados.

- A curva de aprendizado é uma representação gráfica que ilustra a progressão do desempenho de um algoritmo de aprendizado de máquina em relação ao aumento do tamanho do conjunto de treinamento. A curva mostra como o desempenho do modelo melhora à medida que mais dados de treinamento são disponibilizados. Geralmente, a curva de aprendizado é plotada com o desempenho do modelo no eixo vertical e o tamanho do conjunto de treinamento no eixo horizontal. Inicialmente, quando o conjunto de treinamento é pequeno, o desempenho do modelo pode ser baixo devido à falta de informações. Conforme mais dados são adicionados, o modelo começa a aprender padrões e tende a melhorar seu desempenho. No entanto, em algum ponto, adicionar mais dados de treinamento pode não

resultar em melhorias significativas, e a curva de aprendizado pode se estabilizar. A curva de aprendizado também pode fornecer informações sobre outros aspectos do processo de treinamento, como a presença de overfitting (quando o modelo se ajusta muito bem aos dados de treinamento, mas não generaliza bem para novos dados) ou underfitting (quando o modelo não é capaz de capturar os padrões dos dados de treinamento). Essa representação gráfica é útil para avaliar o desempenho do modelo, determinar a quantidade adequada de dados de treinamento e identificar possíveis problemas no processo de aprendizado.

4 Experimentos

4.1 Considerações Iniciais

Neste capítulo, descrevemos as implementações dos experimentos de aprendizado supervisionado de máquina, que visam avaliar os objetivos definidos na Seção 1.5. A estrutura do capítulo é a seguinte: a Seção 4.2 fornece detalhes sobre o ambiente de experimentação, a Seção 4.3 apresenta as performance das classificações usando a MLP, bem como a uma análise geral dos resultados obtidos a partir dos experimentos realizados.

4.2 Ambiente de Experimentação

Nesta seção, é fornecida a configuração detalhada para a classificação de ransomware. Os experimentos foram realizados em um computador equipado com um processador Intel Core i9-11900K de 5,30 GHz, 64 GB de memória RAM e o sistema operacional Windows 10 de 64 bits.

Tal ambiente hospedou todas as ferramentas utilizadas no experimentos, sumarizadas nas seguintes atividades:

- Coleta de Dados: Adquirir o conjunto de dados;
- Pré-processamento de Dados: Remoção de informações irrelevantes e eliminação de registros duplicados;
- Validação Cruzada: Divisão do conjunto de dados em treino e teste através da técnica de Validação Cruzada;
- Balanceamento de classes: Uso do RandomOverSampler
- Redução de Dimensionalidade: Utilização da técnicas KPCA;
- Aplicação do Algoritmos de Classificação de Aprendizado de Máquina: Perceptron Multicamadas (MLP);
- Ajuste de hiperparâmetros.

4.3 Classificação das Abordagens propostas

Esta pesquisa propôs realizar três abordagens de classificação, conforme descrito na seção 1.5. O desempenho das abordagens de classificação binária, classificação de família com

goodwares e classificação de família sem goodwares será apresentado nas seções 4.3.1, 4.3.2 e 4.3.3, respectivamente.

4.3.1 Classificação Binária

Acurácia da MLP %	
Trabalho anterior	Trabalho atual
97.73	98.09

Tabela 4 – Tabela dos resultados de acurácia dos modelos na classificação binária

Parâmetros do melhor resultado
Activation [Tanh]
Hidden layer sizes [(50, 50, 50)]
Solver [adam]
Learning rate [Adaptive]
Número de componentes (KPCA) [399]

Tabela 5 – Tabela dos parâmetros otimizados da MLP e KernelPCA para classificação binária

Na classificação binária o melhor modelo com as especificações descritas na Tabela 5 obteve uma acurácia obtida pela validação cruzada foi de 98,09% com o desvio padrão desse resultado foi de 0,02%, o que representa um aumento de 0,36% em comparação ao resultado de (PEREIRA; JúNIOR, 2022), como mostra a Tabela 4, onde não foi utilizada uma técnica de balanceamento da base de dados.

Sobre a perspectiva do tempo de execução para o treinamento do modelo por cada conjunto na validação cruzada foi a média de 21 segundos com um desvio padrão de 1 segundo. Levando em conta a geração de 100 configurações diferentes do pipeline do modelo e o uso de *10-folds*, levaria aproximadamente 21 mil segundos, o que seria equivalente a 6 horas para o término total do programa, considerando sem o uso de paralelismo.

Ao avaliarmos a matriz confusão na Figura 10 da classificação binária, o número falsos negativos foi significativamente maior que o de falsos positivos, o que podemos inferir que o melhor modelo tem a tendência de gerar mais falsos alertas em benefício de maior segurança.

A partir da curva de treinamento na Figura 11, podemos observar que conforme o tamanho do conjunto de treinamento aumenta, a curva de pontuação de treinamento e a curva de pontuação de teste convergem de maneira acentuada até em torno de 900 dados e após há um aumento repentino da distância entre as curvas e o menor aumento da curva de testes. Podemos inferir, portanto, que o melhor custo benefício de termos precisão do modelo em torno de 900 dados.

Quando analisamos a curva de precisão e *recall* na Figura 12 podemos observar o compromisso alcançado pelo modelo na maximização tanto da precisão quanto do *recall*.

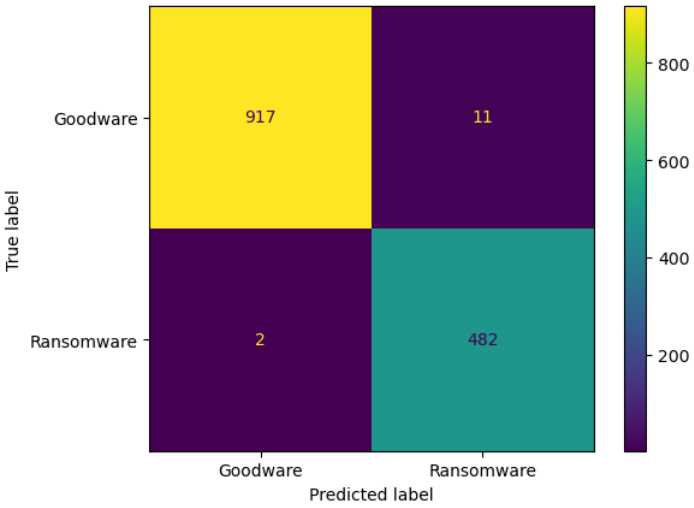

Figura 10 – Gráfico da matriz de confusão da classificação binária.
Fonte: autor.

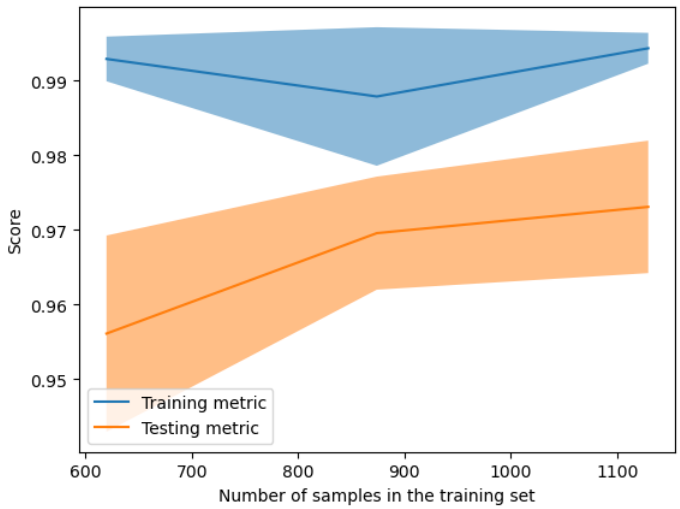

Figura 11 – Gráfico da curva de aprendizado do modelo.
Fonte: autor.

Ao olharmos a curva ROC na Figura 13 podemos observar a grande área abaixo da curva, o que representa que o modelo consegue maximizar a taxa de verdadeiros positivos enquanto minimiza a taxa de falsos positivos.

4.3.2 Classificação de Família

Acurácia da MLP %	
Trabalho anterior	Trabalho atual
84.56	85.19

Tabela 6 – Tabela dos resultados de acurácia dos modelos na classificação de família com *goodware*

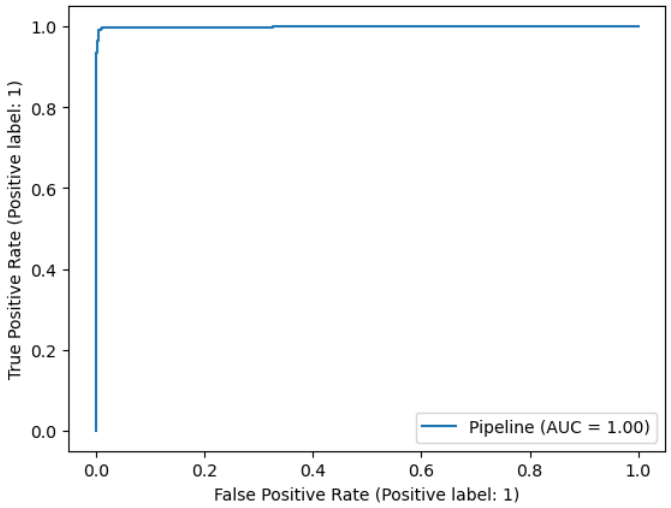

Figura 12 – Gráfico da curva precisão-recall da classificação binária

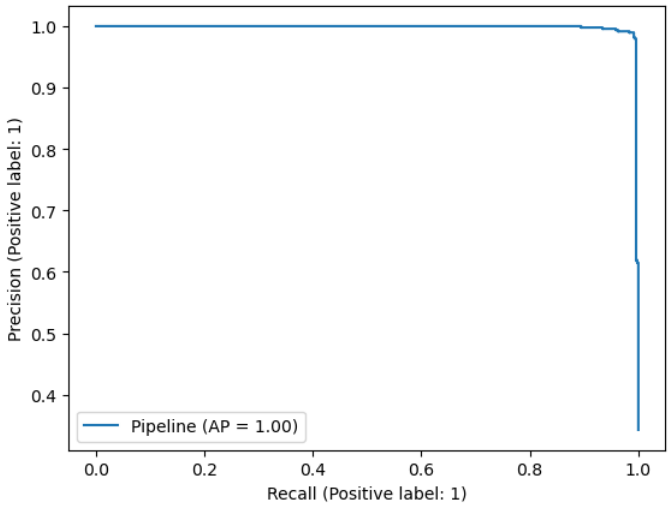

Figura 13 – Gráfico da curva ROC da classificação binária.

Parâmetros do melhor resultado
Activation [Relu]
Hidden layer sizes [(100)]
Solver [adam]
Learning rate [Constant]
Número de componentes (KPCA) [160]

Tabela 7 – Tabela dos parâmetros otimizados da MLP e KernelPCA para classificação de família com *goodware*

O modelo para a classificação de famílias considerando o *goodware* com as especificações na Tabela 7 obteve o resultado de 85,19% de acurácia e desvio padrão em 0,02%, o que representa um aumento em 0,71% em relação ao resultado do trabalho de (PEREIRA; JúNIOR, 2022).

Ao olhar sobre a perspectiva de tempo de execução do treinamento do modelo por cada conjunto na validação cruzada observou-se uma média de 487 segundos com o desvio padrão

de 13 segundos. Foi o mais alto encontrado em todos as análises. Levando em conta as 100 configurações e o uso de *4-folds*, levaria aproximadamente 54 horas para o término do programa sem o uso de paralelismo.

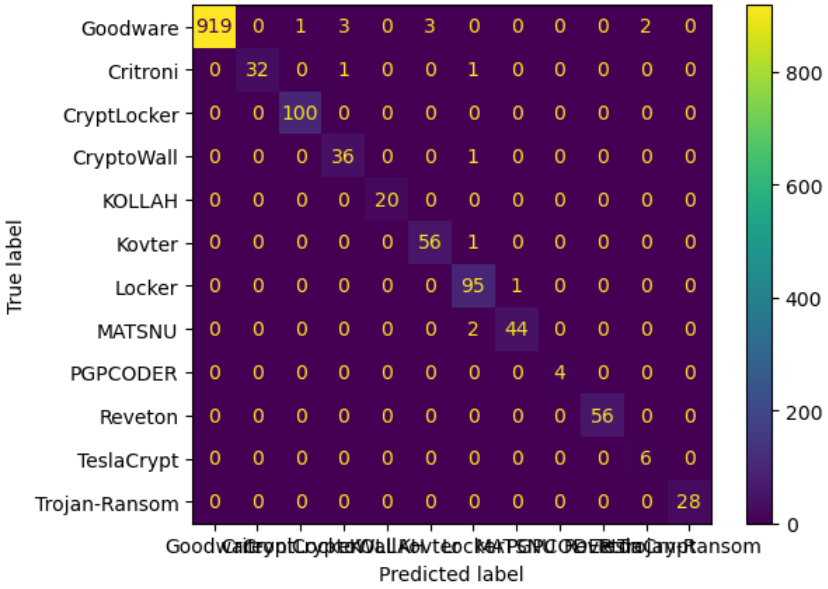

Figura 14 – Gráfico da matriz de confusão da classificação de família dos algoritmos Random Forest e Gaussian Process com as técnicas PCA e KPCA, respectivamente.

Observando a matriz confusão da classificação de família na Figura 14, podemos destacar que nenhuma das instâncias de *ransomwares* foram apontadas como *goodwares*. o que podemos portanto avaliar se esse tipo de classificação não poderia ser usada como critério de detecção de *ransomware* em relação aos *goodwares*, visto que os falsos negativos seriam nulos, e os falsos negativos seriam menores comparado ao da classificação binária, 9 em comparação a 11.

Além disso, em proporção ao número de instancias, a família *TeslaCrypt* teve 2 de suas 8 instâncias identificados incorretamente, o que não aconteceu com a família *PGPCODER* que mesmo possuindo 4 instâncias, foram 100% corretamente identificadas.

Na análise da curva de aprendizado do modelo na Figura 15, podemos observar uma forte velocidade de convergência acontecendo entre 400 e 600 numero de dados, e em seguida uma estabilização quase total do das curvas de treino e teste.

4.3.3 Classificação de Família sem Goodwares

Acurácia da MLP %	
Trabalho anterior	Trabalho atual
66.80	62.80

Tabela 8 – Tabela dos resultados de acurácia dos modelos na classificação de família sem *goodware*

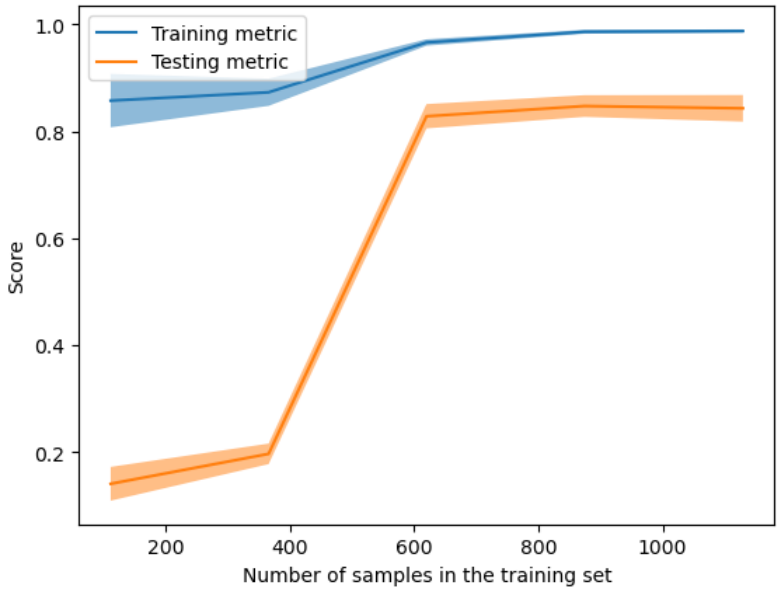

Figura 15 – Gráfico da curva de aprendizado classificação de família com *goodware* do modelo. Fonte: autor.

Parâmetros do melhor resultado
Activation [Relu]
Hidden layer sizes [(100)]
Solver [adam]
Learning rate [Constant]
Número de componentes (KPCA) [606]

Tabela 9 – Tabela dos parâmetros otimizados da MLP e KernelPCA para classificação de família sem *goodware*

O modelo para a classificação de família sem *goodware* com especificação na Tabela 9 obteve uma acurácia de 62,80% e desvio padrão de 0,06%, o que foi 4% menor em comparação com o resultado de (PEREIRA; JúNIOR, 2022). O que podemos inferir que em relação a esse tipo de classificação, a técnica de balanceamento da base de dados, não contribuiu e até reduziu os resultados. Ao analisar a acurácia balanceada foi obtido o valor de 51,97% com desvio padrão de 0,03%, o que denota a baixa confiabilidade do modelo em relação a todas as classes.

Em relação ao tempo de treinamento do modelo por cada conjunto na validação cruzada, foi obtido o menor tempo médio entre todas as análises, de apenas 10 segundos e desvio padrão de 1 segundo. Isso está relacionado ao fato do menor tamanho da base de dados, já que foram retiradas todas as instâncias de *goodwares*, nas quais eram a maioria.

Ao analisar de matriz confusão na Figura 16 podemos destacar uma maior diversidade de erros de classificação no geral, com exceção da classe *PGPCODER* que classificou corretamente para todas suas instâncias.

Quando analisamos a curva de aprendizado na Figura 17, podemos observar uma convergência mais uniforme entre treino e teste, e ainda com a perspectiva que o aumento do

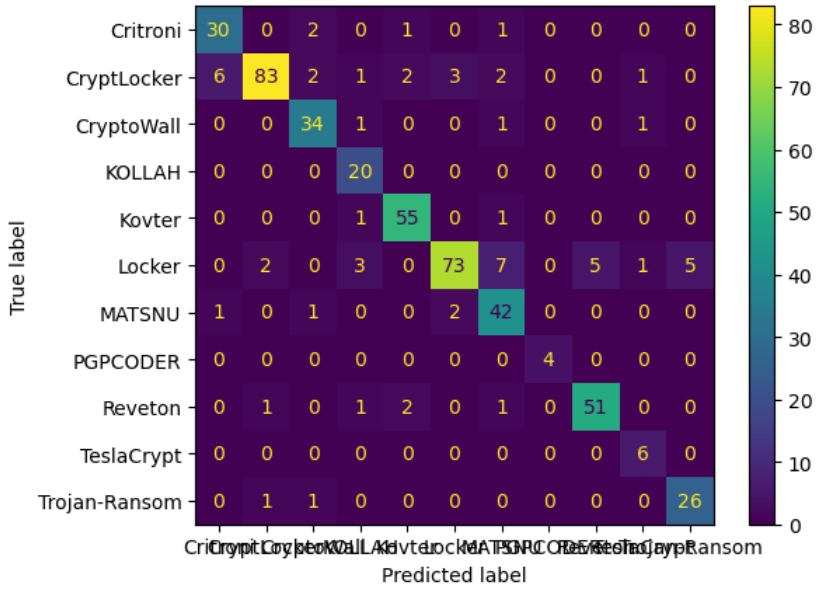

Figura 16 – Gráfico da matriz de confusão da classificação de família sem *goodwares* do modelo. Fonte: autor.

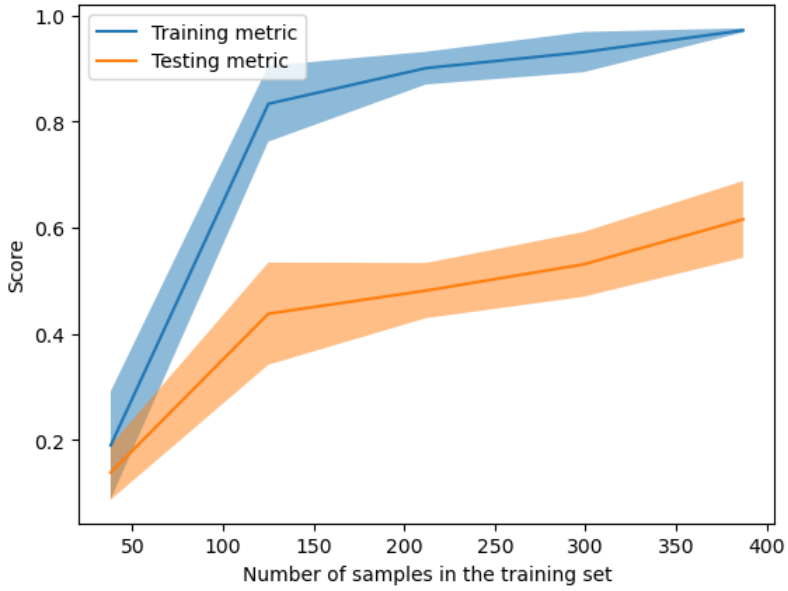

Figura 17 – Gráfico da curva de aprendizado do modelo.

maior numero de dados poderia continuar contribuindo para a melhoria do teste. Isso reforça a deficiência da base de dados em não possuir um maior número de instâncias em cada classe para que se possa alcançar melhores resultados nessa classificação.

Quando analisamos as melhores configurações encontradas sobre cada tipo de análise, podemos destacar que o uso do adam performou melhor em comparação com o *SGD* e o *LBFGS*. As classificações que envolveram família tiveram melhores resultados usando a configuração de 100 neurônios na camada escondida enquanto a classificação binária usou 3 camadas escondidas com 50 neurônios cada. A taxa de aprendizado do tipo constante foi melhor utilizado nos cenários de família em comparação a binária que usou o modelo adaptativo. sobre a redução

de dimensionalidade, a análise por família não necessariamente exigiu um maior número de componentes, pois na classificação de família sem *goodware*, foi a que utilizou maior numero, enquanto a com *goodware* foi a menor entre todas as análises.

5 Conclusões

O assunto *ransomware* representa um desafio significativo para os pesquisadores e especialistas em segurança de Tecnologia da Informação. Com rápida disseminação, e desenvolvimento de sofisticadas técnicas de criptografia, estudos relacionados ao combate de ransomware requerem que pesquisadores apresentem constantes abordagens para acompanhar técnicas evolutivas.

Neste contexto, técnicas de *machine learning*, baseadas em modelos preditivos tem sido investigadas na literatura como uma ferramenta promissora para a detecção de códigos maliciosos. Em geral, algoritmos de aprendizado de máquina podem ser categorizados como preditivos (ou aprendizado supervisionado) ou por descoberta de padrões (aprendizado não supervisionado). A aprendizagem supervisionada consiste em usar dados de treinamento rotulados para ajustar um modelo matemático, e em seguida, aplicar este modelo treinado para prever a ocorrência em dados novos, enquanto algoritmos de aprendizagem não supervisionados identificam padrões ocultos em dados não rotulados.

Este trabalho auxilia nesse esforço, apresentando uma abordagem dinâmica e classificando *ransomware* considerando três abordagens. O presente trabalho teve como objetivo explorar o uso de uma pipeline composta de uma rede neural artificial, o KPCA para redução de dimensionalidade e o RandomOverSampler como técnica de balanceamento de classes. O dataset usado foi disponibilizado por (SGANDURRA et al., 2016) era composto por 582 amostras de *ransomware* pertencentes a 11 classes de família e 942 amostras de *goodwares*.

Inicialmente, e a fim de proporcionar uma contextualização sobre o tema, são introduzidas as premissas que norteiam a origem desta dissertação e para uma melhor compreensão sobre os elementos que compõem os experimentos realizados, é apresentado um referencial teórico com principais tópicos importantes para a compreensão desta dissertação, bem como uma visão geral das principais características dos algoritmos utilizados.

Em seguida, apresentou-se um conjunto de trabalhos correlatos, com o objetivo de posicionar a pesquisa em relação ao tema. Embora a revisão da literatura tenha reportado trabalhos com resultados muito promissores, a presente pesquisa buscou abordar diferentes abordagens e o emprego das técnicas de machine learning, redução de dimensionalidade e balanceamento de classes. Como objetivo geral, a pesquisa buscou fazer a classificação de *ransomware* e obter melhores resultados que os anteriormente encontrados.

A pesquisa utilizou, como base, ferramentas na linguagem Python, a qual possui uma vasta quantidade de bibliotecas para manipulação de dados. Na classificação binária, foi utilizado 10-fold no cross-validation, e nas demais classificações foram adotados 4-fold. Além do mais, foi implementado no scikit-learn a biblioteca *"RandomizedSearchCV"* (validação cruzada de pesquisa aleatória), o qual permite treinar e avaliar uma série de modelos tirando sorteios

aleatórios de um conjunto predeterminado de distribuições de hiperparâmetros. O algoritmo escolhe a versão mais bem-sucedida do modelo visto depois de treinar N diferentes versões do modelo com diferentes combinações, como explicado na seção 3.5.3.

Ao fim dos experimentos, coleta dos dados e análise observa-se que os modelos gerados obtiveram melhores resultados para as abordagens de classificação binária e de família com goodware e não houve melhoria sobre a classificação de família sem goodware em comparação ao trabalho de (PEREIRA; JúNIOR, 2022). A partir dos resultados, podemos mostrar a maior contribuição sobre o assunto por aumentar cada vez mais a segurança de sistemas, e principalmente podemos notar o potencial do uso da classificação de família com goodware como também uma outra maneira de identificação de ransomwares em sistemas.

Como trabalhos futuros, à medida que bases de *ransomware* maiores e confiáveis estejam disponíveis, planeja-se a ampliação da pesquisa fazendo novas avaliações com os melhores modelos encontrados, e também realizar a construção de um novo dataset que atenda melhor o equilíbrio do número de instâncias de cada classe e assim melhor favorecendo as análises de família.

Referências

ABBASI, M. S.; AL-SAHAF, H.; WELCH, I. Particle swarm optimization: A wrapper-based feature selection method for ransomware detection and classification. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 2020. v. 12104 LNCS, p. 181–196. ISBN 9783030437213. ISSN 16113349. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-43722-0_12>. Citado 3 vezes nas páginas 16, 17 e 35.

ADAMU, U.; AWAN, I. Ransomware prediction using supervised learning algorithms. In: *Proceedings - 2019 International Conference on Future Internet of Things and Cloud, FiCloud 2019*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. p. 57–63. ISBN 9781728128887. Citado 3 vezes nas páginas 16, 18 e 20.

ALAM, M. et al. Rapper: Ransomware prevention via performance counters. arXiv, 2018. Disponível em: <https://arxiv.org/abs/1802.03909>. Citado na página 16.

ALENAZI, F.; HINDI, K.; ASSADHAN, B. Fine-tuning naïve bayes for imbalanced datasets. *International Conference on Data Science*, CDATA'19, 7 2019. Citado 3 vezes nas páginas 16, 18 e 20.

ALLIX, K. Challenges and outlook in machine learning-based malware detection for android. 10 2015. Disponível em: <https://orbilu.uni.lu/handle/10993/24900>. Citado na página 37.

ANOWAR, F.; SADAOUI, S. Incremental learning framework for real-world fraud detection environment. *Computational Intelligence*, v. 37, p. 635– 656, 01 2021. Citado na página 26.

BAE, S. I.; LEE, G. B.; IM, E. G. Ransomware detection using machine learning algorithms. In: . John Wiley and Sons Ltd, 2019. v. 32, p. e5422. ISSN 15320634. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.5422https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5422https://onlinelibrary.wiley.com/doi/10.1002/cpe.5422>. Citado na página 17.

BATES, J. Trojan horse: Aids information introductory diskette version 2.0. Virus Bulletin, 1990. Citado na página 24.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4. Citado na página 28.

BORAH, P.; BHATTACHARYYA, D. K.; KALITA, J. K. Cost effective method for ransomware detection: An ensemble approach. In: . [S.l.]: Springer Science and Business Media Deutschland GmbH, 2021. v. 12582 LNCS, p. 203–219. ISBN 9783030656201. ISSN 16113349. Citado na página 18.

CAWLEY, G.; TALBOT, N. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, v. 11, p. 2079–2107, 07 2010. Citado na página 37.

CAWLEY, G. C.; TALBOT, N. L. C. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, v. 11, p. 2079–2107, 2010. Citado na página 36.

CHEN, L. Deep transfer learning for static malware classification. 2018. Disponível em: <https://arxiv.org/abs/1812.07606>. Citado 2 vezes nas páginas 17 e 21.

CUSACK, G.; MICHEL, O.; KELLER, E. Machine learning-based detection of ransomware using sdn. In: *SDN-NFVSec 2018 - Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization, Co-located with CODASPY 2018*. [S.l.]: Association for Computing Machinery, Inc, 2018. v. 2018-January, p. 1–6. ISBN 9781450356350. Citado 2 vezes nas páginas 16 e 18.

DALEY, B. The increase in ransomware attacks during the covid-19 pandemic may lead to a new internet. 2021. Disponível em: <https://theconversation.com/the-increase-in-ransomware-attacks-during-the-covid-19-pandemic-may-lead-to-a-new-internet-162490>. Citado na página 15.

EMM, D. Cracking the code: The history of gpcode. *Computer Fraud Security*, v. 2008, p. 15–17, 09 2008. Citado na página 24.

FERNANDO, D. W.; KOMNINOS, N.; CHEN, T. A study on the evolution of ransomware detection using machine learning and deep learning techniques. *IoT*, MDPI AG, v. 1, p. 551–604, 12 2020. Citado 3 vezes nas páginas 16, 17 e 20.

GANAPATHI, P.; SHANMUGAPRIYA, D. Handbook of research on machine and deep learning applications for cyber security. igi global. 2020. Disponível em: <http://doi:10.4018/978-1-5225-9611-0>. Citado na página 21.

GANGANWAR, V. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, v. 2, n. 4, p. 42–47, 2012. Citado na página 30.

GREENBERG, A. The untold story of notpetya, the most devastating cyberattack in history. *Wired, August*, v. 22, 2018. Citado na página 20.

HARKNETT, R. J.; SMEETS, M. Cyber campaigns and strategic outcomes. *Journal of Strategic Studies*, Taylor & Francis, v. 45, n. 4, p. 534–567, 2022. Citado na página 20.

HASAN, M.; RAHMAN, M. M. Ranshunt: A support vector machines based ransomware analysis framework with integrated feature set. p. 1–7, 12 2017. Citado na página 16.

HAYKIN, S. *Neural networks and learning machines, 3/E*. [S.l.]: Pearson Education India, 2009. Citado na página 28.

HWANG, J. et al. Two-stage ransomware detection using dynamic analysis and machine learning techniques. *Wireless Personal Communications*, Springer, v. 112, p. 2597–2609, 6 2020. ISSN 1572834X. Citado na página 16.

IMAJI, A. *Ransomware Attacks: Critical Analysis, Threats, and Prevention methods*. Tese (Doutorado), 03 2019. Citado na página 24.

KARIMI, A.; MOATTAR, M. H. Android ransomware detection using reduced opcode sequence and image similarity. *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, p. 229–234, 2017. Citado na página 17.

KHARRAZ, A. et al. Cutting the gordian knot: A look under the hood of ransomware attacks. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.]: Springer Verlag, 2015. v. 9148, p. 3–24. ISBN 9783319205496. ISSN 16113349. Citado na página 19.

KOK, S.; AZWEEN, A.; JHANJHI, N. Evaluation metric for crypto-ransomware detection using machine learning. *Journal of Information Security and Applications*, v. 55, p. 102646, 2020. ISSN 2214-2126. Disponível em: <https://www.sciencedirect.com/science/article/pii/S221421262030805X>. Citado na página 23.

KUMAR, A.; SHARMA, I. Understanding the behaviour of android ransomware attacks with real smartphones dataset. In: *2023 International Conference for Advancement in Technology (ICONAT)*. [S.l.: s.n.], 2023. p. 1–5. Citado 2 vezes nas páginas 16 e 19.

LEARN scikit. 1.17. neural network models (supervised). 2021. Disponível em: <https://scikit-learn.org/stable/modules/neural_networks_supervised.html>. Citado na página 28.

LEARN scikit. About us. 2021. Disponível em: <https://scikit-learn.org/stable/about.html>. Citado na página 34.

LEDOUX, J. Tune hyperparameters with randomized search. 2019. Disponível em: <https://jamesrledoux.com/code/randomized_parameter_search>. Citado na página 37.

LEVADA, A. Lecture notes in dimensionality reduction for unsupervised metric learning. *Universidade Federal de São Carlos – Departamento de Computação, Notas da aula de Reconhecimento de Padrões*, 5 2020. Citado na página 27.

LEVADA, A. Análise de componentes principais (pca) e análise discriminante linear (lda). *Universidade Federal de São Carlos – Departamento de Computação, Notas da aula de Reconhecimento de Padrões*, 5 2021. Citado na página 27.

MAATEN, L. V. D.; POSTMA, E.; HERIK, J. Dimensionality reduction: A comparative review. In: . [S.l.: s.n.], 2009. Citado na página 26.

MAIORCA, D. et al. R-packdroid: Api package-based characterization and detection of mobile ransomware. Association for Computing Machinery, New York, NY, USA, p. 1718–1723, 2017. Disponível em: <https://doi.org/10.1145/3019612.3019793>. Citado na página 16.

MAMEDOV, O.; SINITSYN, F.; IVANOV, A. Bad rabbit ransomware. *Retrieved May*, v. 1, p. 2021, 2017. Citado na página 20.

MILOSEVIC, N.; DEHGHANTANHA, A.; CHOO, K. K. R. Machine learning aided android malware classification. *Computers and Electrical Engineering*, Elsevier Ltd, v. 61, p. 266–274, 7 2017. ISSN 00457906. Citado na página 16.

MOUSSAILEB, R. et al. Ransomware's early mitigation mechanisms. Association for Computing Machinery, New York, NY, USA, 2018. Disponível em: <https://doi.org/10.1145/3230833.3234691>. Citado na página 18.

NAIM, F. A.; HANNAN, U. H.; KABIR, M. H. Effective rate of minority class over-sampling for maximizing the imbalanced dataset model performance. In: SPRINGER. *Proceedings of Data Analytics and Management: ICDAM 2021, Volume 2*. [S.l.], 2022. p. 9–20. Citado na página 29.

O'GORMAN, G.; MCDONALD, G. *Ransomware: A growing menace*. [S.l.]: Symantec Corporation Arizona, AZ, USA, 2012. Citado na página 23.

OZ, H. et al. A survey on ransomware: Evolution, taxonomy, and defense solutions. 2021. Disponível em: <https://doi.org/10.1145/nnnnnnn.nnnnnnn>. Citado 3 vezes nas páginas 15, 16 e 26.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Disponível em: <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>. Citado 2 vezes nas páginas 9 e 37.

PEREIRA, G. T. M.; JúNIOR, C. de Souza de S. Ransomware classification by machine learning and dimensionality reduction. *Journal of Engineering Research*, Atena Editora, v. 2, p. 2–9, 11 2022. Citado 6 vezes nas páginas 19, 31, 42, 44, 46 e 50.

POUDYAL, S.; GUPTA, K. D.; SEN, S. Pefile analysis: a static approach to ransomware analysis. *Int J Forens Comput Sci*, v. 1, p. 34–39, 2019. Citado na página 20.

POWERS, D.; AILAB. Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *J. Mach. Learn. Technol*, v. 2, p. 2229–3981, 01 2011. Citado na página 38.

PYTHON. What is python? executive summary. 2021. Disponível em: <https://www.python.org/doc/essays/blurb/>. Citado na página 34.

QUINTANILLA, L. Tarefas de aprendizado de máquina no ml.net. 2021. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/machine-learning/resources/tasks>. Citado na página 31.

SGANDURRA, D. et al. Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. 9 2016. Disponível em: <http://arxiv.org/abs/1609.03020>. Citado 3 vezes nas páginas 18, 35 e 49.

SHAUKAT, S. K.; RIBEIRO, V. J. Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, p. 356–363, 2018. Citado na página 17.

SILVA, A. et al. Aplicação de algoritmos de aprendizado de máquina na unificação das técnicas de análise estática e dinâmica na classificação de ransomware. *PESQUISA EDUCAÇÃO A DISTÂNCIA*, 03 2020. Citado 2 vezes nas páginas 18 e 20.

STARTUPI. Google lança nova unidade de processamento focada em machine learning. 2017. Disponível em: <https://startupi.com.br/2017/05/google-lanca-nova-unidade-de-processamento-focada-em-machine-learning/>. Citado na página 35.

VIGNESWARAN, R. et al. Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, p. 1–6, 2018. Citado na página 17.

WILLIAMS, S. Ransomware to remain public enemy 1 in 2023 - report. 2023. Disponível em: <https://securitybrief.co.nz/story/ransomware-to-remain-public-enemy-1-in-2023-report>. Citado na página 15.

WUEST, C. Digital coronavirus: Yet another ransomware combined with infostealer. *Acronis Blog*, 2020. Disponível em: <https://www.acronis.com/en-us/blog/posts/digital-coronavirus-yet-another-ransomware-combined-infostealer>. Citado na página 25.

YOUNG, A.; YUNG, M. Cryptovirology: extortion-based security threats and countermeasures. *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, p. 129–140, 1996. Citado na página 24.