Classificação de dados baseada em redes complexas para detecção de binários empacotados

Ricardo Barbosa Lima Filho

Universidade Federal de Uberlândia
Faculdade de Computação
Bacharelado em Sistemas de Informação

Monte Carmelo - MG
2022

Ricardo Barbosa Lima Filho

Classificação de dados baseada em redes complexas para detecção de binários empacotados

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Prof. Dr. Murillo Guimarães Carneiro

Monte Carmelo - MG
2022

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4144 - http://www.portal.facom.ufu.br/ facom@ufu.br

ATA DE DEFESA - GRADUAÇÃO

Curso de Graduação em:	Sistemas de Informação - campus Monte Carmelo				
Defesa de:	FACOM31804 - Trabalho de Conclusão de Curso 2				
Data:	19/08/2022	Hora de início:	10h00	Hora de encerramento:	11h20
Matrícula do Discente:	31911BSI023				
Nome do Discente:	Ricardo Barbosa Lima Filho				
Título do Trabalho:	Classificação de dados baseada em redes complexas para detecção de binários empacotados				
A carga horária curricular foi cumprida integralmente?	(X) Sim ( ) Não				

Reuniu-se de forma remota na Plataforma Microsoft Teams, link https://teams.microsoft.com/l/meetup-join/19%3amIwYs8UvQ0ZeAcZlZEejrO2mhC3mrHKdnliwaZ-7tUc1%40thread.tacv2/1660655653072?context=%7b%22Tid%22%3a%22cd5e6d23-cb99-4189-88ab-1a9021a0c451%22%2c%22Oid%22%3a%22256d2afa-0920-498f-9861-bf3a34ad9d01%22%7d, a Banca Examinadora, designada pelo Colegiado do Curso de Graduação em Sistemas de Informação - campus Monte Carmelo, assim composta: Professores: Dr. Kil Jin Brandini Park (FEELT/UFU); Dr. Rodrigo Sanches Miani (FACOM/UFU); Dr. Murillo Guimarães Carneiro (FACOM/UFU), orientador do candidato.

Iniciando os trabalhos, o presidente da mesa, Dr. Murillo Guimarães Carneiro, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao discente a palavra, para a exposição do seu trabalho. A duração da apresentação do discente e o tempo de arguição e resposta foram conforme as normas do curso.

A seguir, o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

(X) Aprovado com Nota [100]

OU

( ) Aprovado sem nota.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Murillo Guimarães Carneiro**, **Professor(a) do Magistério Superior**, em 19/08/2022, às 11:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 19/08/2022, às 15:56, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Kil Jin Brandini Park**, **Professor(a) do Magistério Superior**, em 20/08/2022, às 15:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3849446** e o código CRC **9E90494F**.

Dedico este trabalho aos meus pais, por nunca terem medido esforços para me proporcionar ensino de qualidade, carinho e amparo durante toda esta trajetória, e ao meu orientador Dr. Murillo Carneiro, que conduziu o trabalho sempre prestando o auxílio necessário.

Agradecimentos

Aos meus pais, Valéria e Ricardo, que me inspiraram a sempre esforçar e nunca desistir, para atingir meus sonhos.

As minhas irmãs, Laís e Paula, que são minhas companheiras e meu porto seguro, sempre me ajudando em momentos difíceis.

Ao meu professor e orientador Dr. Murillo Carneiro, que me guiou e auxiliou durante este trabalho, com dedicação e excelência.

A todos os professores, que me ensinaram e despertaram meu interesse em conhecer e buscar novos conhecimentos.

Aos meus colegas e amigos, em que pude compartilhar importantes momentos da minha vida em que sempre lembrarei com gratidão.

Aos envolvidos, professores e colegas de outros cursos da UFU, em que tive o prazer de poder trabalhar junto, ensinando e aprendendo para o progresso coletivo.

A todos que contribuiram, diretamente e indiretamente no desenvolvimento deste trabalho.

"Quem pensa pouco, erra muito."
(Leonardo Da Vinci)

Resumo

A área de segurança da informação está constantemente sendo testada com novas vulnerabilidades e desafios para corrigi-lás, um exemplo são os binários empacotados que são executáveis maliciosos ofuscados no processo de compactação de um arquivo. No entanto, sistemas de antivírus encontram dificuldades para detectar o empacotado como perigoso, devido a técnica de ocultamento presente no arquivo.

A classificação de dados que é um ramo do aprendizado de máquina apresenta diversas abordagens ao analisar os atributos físicos dos dados e não investigam padrões de formação, podendo limitar o desempenho. Redes complexas em aprendizado de máquina são estruturas que apresentam padrões de conexões não triviais, nem completamente regular e nem completamente aleatório. Mediante a sua versatilidade, o seu uso para classificação de dados tem se tornado cada vez mais relevante, por resultados bem sucedidos ao analisar suas medidas e propriedades para classificação de uma instância. A confeção de uma rede interliga dados por meio de suas características em comum, com isso é possível interpretar padrões estruturais e topológicos.

Diante deste cenário, este trabalho propõe a construção de uma rede baseado em k-vizinhos mais próximos e o aprendizado de padrões da rede via conformidade padrão através de seis medidas de rede selecionadas na literatura: assortatividade, coeficiente de agrupamento, grau médio, intermedialidade, menor caminho médio e proximidade. Os resultados apontam uma acurácia média superior a noventa por cento, além de um melhor resultado nos sete empacotadores analisados em relação a diferentes classificadores da literatura, validando a rede gerada e auxiliando a detecção de arquivos empacotados.

Palavras-chave: Redes complexas, conformidade padrão, classificação de dados, segurança da informação, binários empacotados.

Abstract

The information security field is constantly being tested with new vulnerabilities and challenges to fix them, an example is packed binaries that are malicious executables obfuscated in the process of compressing a file. However, antivirus systems find it difficult to detect the package as dangerous, due to the hiding technique present in the file.

Data classification, which is a branch of machine learning, takes several approaches when analyzing physical data, which can limit performance. Complex networks in machine learning are structures that do not follow a regular or random pattern. Due to its versatility, its use for data classification has become increasingly relevant, due to successful results when analyzing its measures and properties for labeling an instance. The construction of a network interconnects data, through their common characteristics, with this it is possible to interpret structural and topological patterns.

In view of this scenario, this work proposes the construction of a network based on k-nearest neighbors and the learning of network patterns via pattern compliance through six network measures selected in the literature: assortativity, clustering coefficient, average degree, betweenness, shortest mean path and closeness. The results point to an average accuracy of more than ninety percent, in addition to a better result in the seven packagers analyzed in relation to different classifiers in the literature, validating the generated network and helping the detection of packed files.

Keywords: complex networks, high level rating, data classification, information security, packed executables.

Lista de ilustrações

Lista de tabelas

Lista de siglas

AM	Aprendizado de Máquina
R	Assortatividade
CART	Classification And Regression Trees
CC	Coeficiente de agrupamento
ED	Distância euclidiana
FSG	Fast Small Good
K	Grau médio
IA	Inteligência Artificial
B	Intermedialidade
kNN	K-vizinhos mais próximos
MDP	Markov Decision Process
MPRESS	Matcode Compressor
MC	Menor caminho médio
MLP	Multilayer Perceptron
NB	Naive Bayes
P	Proximidade
RF	Random Forest
SVM	Support Vector Machine
TI	Tecnologia da Informação
UPX	Ultimate Packer for eXecutables

Sumário

Capítulo 1

Introdução

Malware é um termo derivado da fusão de "malicioso" e "software". É usado para definir qualquer tipo de aplicativo cujo objetivo é causar danos a usuários ou sistemas. Este aplicativo pode ser utilizado para roubo de dados, interceptação de informações, bombardeio de publicidade, danos ao sistema físico ou danos aos dados (IDIKA; MATHUR, 2007).

Aplicações maliciosas têm evoluído de forma a cada vez mais permanecerem ocultas no sistema alvo. Além disso, essas aplicações utilizam de técnicas de ofuscação e/ou anti-análise a fim de evitar a descoberta de seu comportamento durante a execução (BOTACIN et al., 2019). Um problema bastante interessante que explora a utilização de técnicas de ofuscação são os arquivos binários empacotados, que devido ao arquivo ser um empacotado e sua informação real estar "ofuscada", sistemas convencionais de antivírus acabam não detectando o arquivo como malicioso (KINDREDSEC, 2020).

Existem algumas maneiras para analisar e estudar o comportamento de aplicações maliciosas, a fim de criar medidas de proteção contra o mesmo. Dentre as maneiras adotadas para estudo estão a engenharia reversa e o aprendizado de máquina, que visam analisar ou categorizar tais aplicações, entretando a engenharia reversa pode não ser uma boa técnica para analisar arquivos empacotados, devido a ocultação dos dados presentes no arquivo (KINDREDSEC, 2020).

O BinStat (PARK; RUIZ; MONTES, 2011) é um software capaz de classificar um arquivo executável empacotado através da análise bloco a bloco dos arquivos, utilizando o classificador Decision Tree (Árvore de decisão). Já em (ASSIS et al., 2019) é proposto a adoção de mais classificadores com heurísticas diferentes visando aprimorar o desempenho preditivo, para isso foram realizados diversos testes com alguns dos classificadores mais conhecidos da literatura e os resultados apresentam uma melhora significativa, principalmente no mapeamento de arquivos desempacotados.

A classificação de dados baseada em redes complexas via conformidade padrão (SILVA; ZHAO, 2012) é um tema recente na literatura, mas que tem mostrado ser uma abordagem bastante promissora para diferentes problemas, devido a análise ser realizada considerando

padrões topológicos de formação da rede. Nesta técnica, a classificação ocorre ao verificar em qual classe da rede a inserção de um novo item apresenta menor variação em suas medidas (CARNEIRO, 2016).

Visto que não há um estudo detalhado sobre um classificador de alto nível para a análise de arquivos binários empacotados, o projeto tem por foco, portanto, preparar um ambiente de aprendizado de máquina para classificação dos dados via conformidade padrão.

1.1 Problema

Proteger sistemas contra possíveis ameaças se torna um desafio à medida que não sabemos sobre o ataque, uma prova disso são os binários empacotados. Embora estes executáveis possam ser adotados com intenção legítima, os desenvolvedores de malware utilizam essas ferramentas para esconderem executáveis maliciosos dentro de um arquivo compactado, com isso, uma potencial ameaça pode não ser detectada por um sistemas de antivírus. Portanto, é necessário um ambiente capaz de classificar os dados, para contribuir com as medidas de proteção do mesmo.

1.2 Hipótese

Utilizar um classificador de alto nível via conformidade padrão para classificação de binários empacotados permite soluções com melhor desempenho preditivo para o problema.

1.3 Objetivos

O objetivo geral deste trabalho é o desenvolvimento de modelos de aprendizado de máquina baseados em redes para análise e detecção de executáveis empacotados.

Os objetivos específicos deste trabalho são:

- ❑ Desenvolver estratégias de aprendizado em redes complexas para o problema abordado.
- ❑ Avaliar as estratégias desenvolvidas em comparação com outros modelos de trabalhos relacionados.

1.4 Resultados esperados

Espera-se contribuir para área de segurança cibernética, através de um sistema de aprendizado de máquina baseado em redes complexas, a fim de classificar os dados, en-

contrando padrões nas aplicações e finalmente promovendo positivamente em busca de medidas de prevenções do mesmo.

1.5 Organização da Monografia

O texto pós capítulo 1, está organizado da seguinte maneira:

- **Capítulo 2 (Fundamentação Teórica):** Apresenta todos os conceitos importantes utilizados para o desenvolvimento e bom entendimento do trabalho.
- **Capítulo 3 (Materias e Método):** São apresentados as bases de treino e teste, bem como os métodos empregados para confecção do ambiente de aprendizado.
- **Capítulo 4 (Resultados Experimentais):** São apresentados os resultados obtidos pela rede, bem como uma comparação com o resultado de diferentes classificadores e por fim, uma análise detalhada do desempanho de cada medida de rede.
- **Capítulo 5 (Conclusão):** é apresentado a conclusão do projeto, em vista do que foi programado e do que foi de fato alcançado, além dos trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Nesse capítulo serão apresentados os conceitos utilizados no trabalho de classificação, algoritmos, estratégias e conceitos necessários para entender aspectos fundamentais do aprendizado de máquina e do problema de detecção de binários empacotados.

2.1 Aprendizado de Máquina

A Inteligência Artificial (IA) é uma vasta área do ramo de Tecnologia da Informação (TI), que abrange sub-áreas como o Aprendizado de Máquina (AM), pois essencialmente a capacidade de aprender é um comportamento de suma importância para um comportamento inteligente (BATISTA, 2003). Um ambiente de AM, não somente ajuda na solução de problemas complexos, mas também pode melhorar o nosso próprio raciocínio lógico (MONARD et al., 1997).

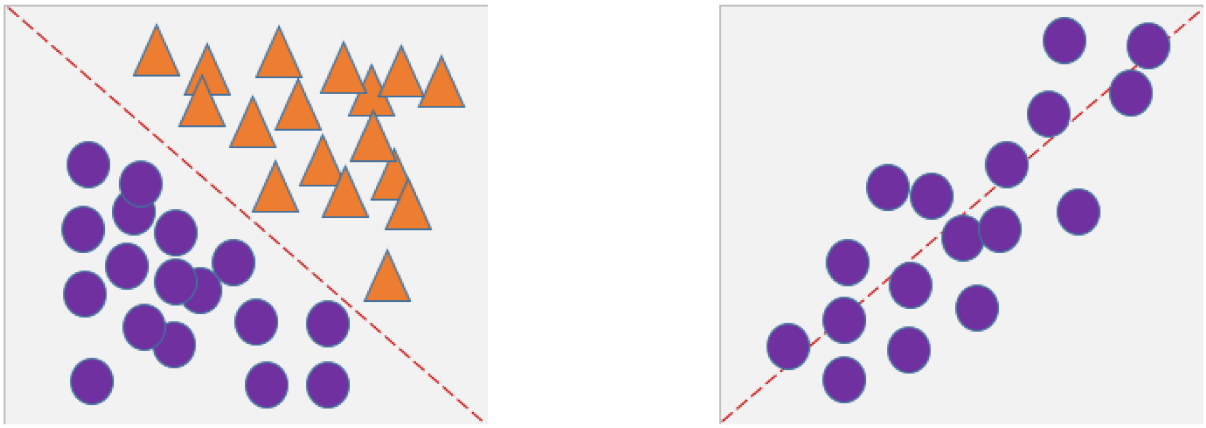
A AM é uma área de pesquisa voltada para técnicas computacionais que buscam aprender determinados padrões e compartamento advindo de obsevações ou amostras, também denominadas experiências, e a partir da experiência coletada durante o aprendizado, busca melhorar o desempenho (CARNEIRO, 2016). Existem três tipos de paradigmas clássicos de aprendizado de máquina: supervisionado, não supervisionado e por reforço.

2.1.1 Aprendizado Supervisionado

Quando o ambiente tenta prever uma variável específica (alvo) a partir do conjunto de características das outras variáveis descritivas (atributos) da instância, dizemos que o aprendizado é supervisionado. Neste cenário há duas categorias: regressão e classificação. Na regressão as variáveis são mapeadas para alguma função contínua, ou seja, dada uma base de dados, são encontradas retas que representam a proximidade das instâncias, com isso é possível categorizar novos dados em relação a sua proximidade em cada função. Na classificação, os dados são previstos em uma saída discreta, em que os mesmos são

agrupados de acordo com um atributo alvo, logo novas instancias são previstas de acordo com a combinação dos outros atributos (CERQUEIRA, 2010).

A Figura 1 ilustra as duas principais ferramentas do aprendizado supervisionado. Ao lado esquerdo temos a classificação linear, que busca um linha reta (função linear) capaz de separar os dados em dois conjuntos. Já a regressão linear, representada ao lado direito, busca uma linha reta que passa próximo a um conjunto de dados.



(a) Classificação. (b) Regressão.

Figura 1 – As duas tarefas principais do aprendizado supervisionado: classificação (esquerda) e regressão (direita).

Fonte: Adaptada (EDELL, 2015).

2.1.2 Aprendizado Não Supervisionado

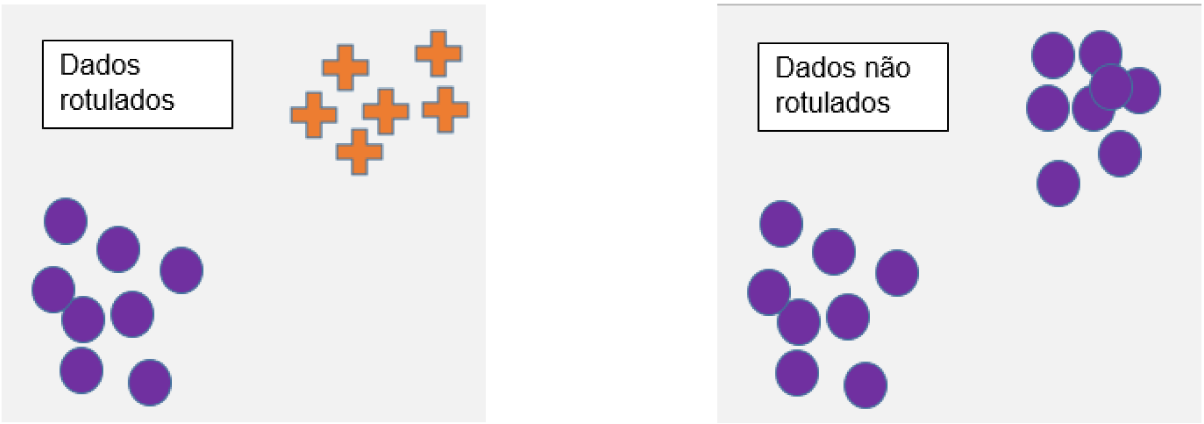
No aprendizado não supervisionado, o algoritmo aprende em dados de teste que não foram rotulados, desta forma esta metodologia identifica padrões/semelhanças nos dados e grava as tendências com maior ocorrência (BRUNIALTI et al., 2015).

Alguns exemplos de situações que o aprendizado não supervisionado é viável seriam: sistemas de recomendações, detecção de anomalias, registro de compras (associação de produtos e perfil do consumidor), entre outras. Dentre as técnicas utilizadas em ambientes com tais características temos dois grandes grupos: Associação e Clusterização.

- **Associação:** permite a descoberta de regras e dependências e a identificação de conjuntos de itens que costumam aparecer juntos. Os algoritmos buscam encontrar relações entre itens, examinar eventos simultâneos, e assim poder entender novos modelos para melhores resultados (PELEGRIN et al., 2012).

- **Clusterização (ou agrupamento):** é uma técnica que permite agrupar conjuntos de dados denominados clusters, com base em similaridade ou métricas de distância. Dessa forma os objetos pertencentes a um mesmo cluster possuem similaridades entre si e, ao mesmo tempo, os objetos pertencentes a clusters diferentes apresentem alta dissimilaridade (PELEGRIN et al., 2012).

A Figura 2 apresenta a diferença mais relevante entre o aprendizado supervisionado e não supervisionado. Na figura da direita, temos a representação do aprendizado supervisionado, evidenciando a presença de rótulos (classes) no dados. Já a figura da esquerda representa o aprendizado não supervisionado e dados não rotulados.



(a) Aprendizado Supervisionado. (b) Aprendizado Não Supervisionado.

Figura 2 – Principal diferença entre o aprendizado supervisionado e o não supervisionado. A direita representa o aprendizado supervisionado com dados rotulados, e a esquerda representa o aprendizado não supervisionado com dados não rotulados.

Fonte: Adaptada (WAUKE, 2020).

2.1.3 Aprendizado por reforço

O Aprendizado por Reforço, conhecido como modelo de aprendizado semi-supervisionado em Machine Learning (AM), é uma técnica para permitir que um agente tome ações e interaja com um ambiente, a fim de maximizar as recompensas totais (ACADEMY, 2022).

Nesta abordagem a ideia é que o agente busque realizar uma determinada tarefa, inicialmente por tentativa e erro. Posteriormente, os resultados de cada tentativa, bem-sucedida ou não, são usados para treinar o agente por meio de um sistema de recompensa/penalidade e determinar se a ação realizada foi eficaz. Portanto, o agente aprende com os erros e acertos e, eventualmente, espera-se que ele execute tarefas com proficiência (ACADEMY, 2022).

Geralmente este tipo de aprendizado é recomendado em situações em que o agente conhece as regras, mas não sabe qual a melhor sequência de ações, pois as mesma são iterativamente aprendidas, igualmente em um jogo de tabuleiro.

A Figura 3 exemplifica o modelo matemático conhecido como MDP. A sequência inicia-se no instante (t), o agente seleciona uma ação, muda para um novo estado, recebe uma recompensa e então o ciclo é reiniciado para o próximo instante $(t + 1)$.

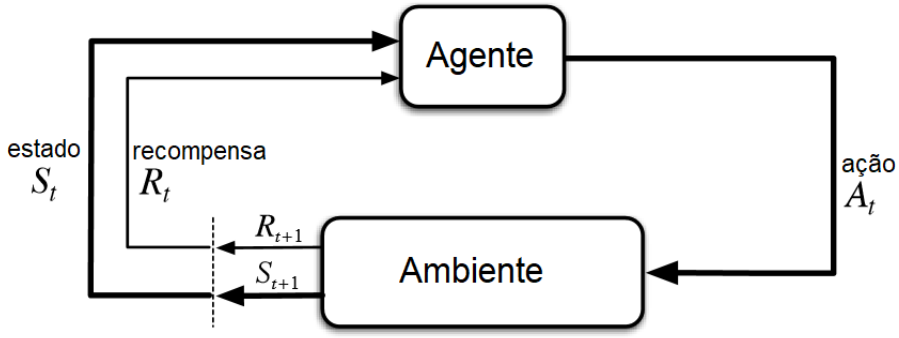


Figura 3 – Fluxo de atividades realizados no aprendizado por reforço, denominado MDP. Fonte: (NEVES, 2020).

2.2 Classificação de dados

A etapa de classificação tem por característica mapear conjuntos de dados não rotulados em classes pré-definidas, para isso, os modelos de classificação são construídos a partir de dados de treino que possuem informação quanto às classes que pertencem. Por fim, os modelos são aplicados sobre uma base de teste onde nenhuma informação a respeito das classes é disposta, e então os modelos classificam de acordo com os rótulos previamente conhecidos. Por exemplo, em uma base de dados sobre pacientes com uma determinada doença, pode-se classificar estes pacientes em grupos que tem ou não tem a enfermidade, com isso um novo paciente pode ser classificado em um dos grupos, de acordo com suas características (CASTANHEIRA, 2008).

Uma das tarefas mais comuns, a Classificação, visa identificar a qual classe um determinado registro pertence. Nesta tarefa, o modelo analisa o conjunto de registros fornecidos, com cada registro já contendo a indicação à qual classe pertence, a fim de "aprender" como classificar um novo registro (aprendizado supervisionado) (CAMILO; SILVA, 2009).

Dividir dados em conjunto de teste e conjunto de treinamento é uma parte importante da avaliação de modelos de mineração de dados. Geralmente, quando é dividido um conjunto de dados em um conjunto de treinamento e um conjunto de teste, a maioria dos dados é usada para treinamento e uma pequena parte dos dados é usada para teste. Com os modelos já processados, o conjunto de treinamento é utilizado e finalmente o modelo pode realizar previsões no conjunto de testes (PETERMANN, 2006).

2.2.1 K-Vizinhos mais próximos (KNN)

O algoritmo KNN é uma dos mais simples em termos de complexidade e implementação, além de ser um dos mais conhecidos pela literatura. Este classificador visa categorizar um novo o novo objeto em relação a sua proximidade com os demais objetos contidos na base de treinamento, atribuindo a classe dominante entre seus k vizinhos mais próximos,

visto que k é a quantidade de vizinhos a serem considerados pelo experimento (ARAúJO, 2018).

A escolha do valor de k para um valor ímpar evita eventuais empates para problemas binários, pois quando ocorre quantidades iguais de classes (k sendo par), duas estratégias podem ser adotadas: a classe da instância da próxima, ou a escolha de um rótulo de classe aleatório.

A Figura 4 exemplifica situações em que os valores escolhidos para k foram respectivamente três e seis. Quando o valor de k é igual a três, a classe para a instância em questão será a Casse B (roxo), por outro lado, quando o k é igual a seis, então o novo objeto será classificado como Classe A (verde), visto que a maioria das classes de seus vizinhos é da Classe A.

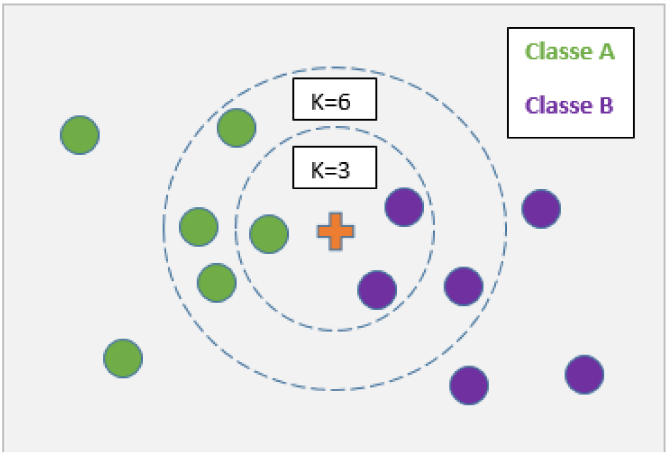


Figura 4 – Vizinhos mais próximos do ponto central, para valores de $k = 3$ e $k = 6$.

Fonte: Adaptada (CHOUINARD, 2022).

2.2.2 Validação e desempenho preditivo

Para este trabalho foi adotado como medida avaliativa para validação dos resultados obtidos a acurácia, que pode ser definida como a proximidade de um resultado experimental com o seu valor de referência real. Assim, ela determina o grau de exatidão. Em outras palavras, é uma medida que quantifica o nível percentual de acerto.

A acurácia é baseada na contagem de registros previstos corretamente, quantificando este valor em porcentagem. A Equação 1 representa a fórmula da acurácia, em que o Total representa um conjunto total de dados analisados.

$$AC = \frac{VerdadeirosPositivos + VerdadeirosNegativos}{Total} \tag{1}$$

2.3 Redes Complexas

O termo redes complexas pode ser explicado pela segmentação do termo, em que redes seriam interconexões entre objetos relacionados e complexas seria devido a suas características topológicas não serem triviais, nem totalmente aleatórias e nem totalmente regulares (CARNEIRO, 2016).Visto que para obtenção de resultados na rede é feito cálculos e análise sobre sua topologia, dizemos então que redes complexas é ligada a cálculos estatísticos.

Uma rede é um grafo no qual há um conjunto de vértices (ou nós) um conjunto de arestas (ou arcos) que conectam esses vértices. As arestas estabelecem algum tipo de relação entre dois vértices, além de conter ou não pesos de acordo com o problema modelado. Tais conexões com valores (pesos) podem definir a capacidade ou intensidade, em que o tráfego flui pela rede (CARNEIRO, 2016) Além disso, o grafo pode ser direcionado ou não. Em um grafo direcionado (dígrafo), cada aresta tem um sentido (direção) que conecta um vértice origem à um vértice destino (VERA, 2011).

A literatura contempla diversas medidas de rede, que representam características estruturais diferentes da rede, utilizadas em várias aplicações (NEWMAN, 2003; COSTA et al., 2007; RUBINOV; SPORNS, 2010). Para este trabalho foram selecionadas seis medidas, levando em conta o problema tanto na classificação de empacotados e arquivos originais:

- **Assortatividade:** calcula a tendência de conexão entre vértices, verificando a relação ao grau de cada um (NEWMAN, 2003). O coeficiente de assortatividade r representa o coeficiente de correlação de Pearson de grau entre pares de nós conectados, com isso os valores giram em torno de $[-1, 1]$, em que valores positivos indicam uma correlação entre nós de grau semelhante, enquanto valores negativos indicam relações entre nós de grau diferente (CARNEIRO, 2016).

  A assortatividade é dada pela Equação 2, em que L é o número de conexões na rede e i_u, k_u seriam os graus dos vértices i e k, que juntos, compõe uma aresta u.

$$r = \frac{L^{-1}\sum\limits_{u} i_u k_u - \left[L^{-1}\sum\limits_{u} \frac{1}{2}\left(i_u + k_u\right)\right]^2}{L^{-1}\sum\limits_{u} \frac{1}{2}\left(i_u^2 + k_u^2\right) - \left[L^{-1}\sum\limits_{u} \frac{1}{2}\left(i_u + k_u\right)\right]^2} \tag{2}$$

- **Coeficiente de agrupamento:** mede o grau com que os nós de um grafo tendem a agrupar-se (CARNEIRO, 2016). Em outras palavras, dado um determinado nó, o coeficiente de agrupamento estima o quão perto os seus vizinhos estão, e consequentemente a probabilidade de se interligarem.

  Assumindo que o vértice i tem k vizinhos, existem no máximo $k_i(k_i - 1)$ possibilidades de arestas entre eles, se houver um vizinho u de i está conectado a qualquer outro vizinho s de i (CARNEIRO, 2016). Logo, o coeficiente de agrupamento é dado pela equação 3.

$$CC_i = \frac{|e_{us}|}{k_i(k_i - 1)} \tag{3}$$

  O coeficiente de agrupamento médio de um rede é dado pela equção 4. Onde $CC_i \in [0, 1]$.

$$CC = \frac{1}{V}\sum_{i=0}^{V} CC_i \tag{4}$$

- **Grau médio:** é uma medida simples, que quantifica a quantidade de conexões de um vértice da rede, representada pela equação 5.

$$k_i = \sum_{j=0}^{n} a_{ij} \tag{5}$$

  Dado todos os vértices de uma rede, o grau médio define a quantidade média de vértices adjacentes, representado pela equação 6.

$$k = \frac{1}{n}\sum_{i=0}^{n} k_i \tag{6}$$

  A Figura 5 mostra o grau de cada vértice de um grafo com quatro nós. Utilizando a equação 6, logo temos que o grau médio da rede é de 1.5.

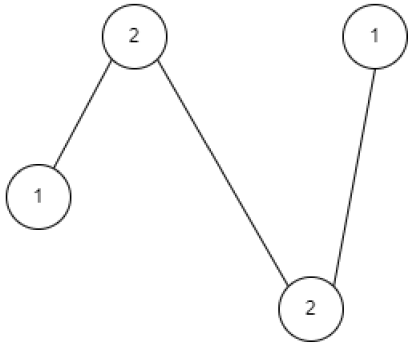


Figura 5 – Grau dos nós de uma pequena rede gerada com apenas quatro vértices.

Fonte: Autoria própria.

- **Menor caminho médio:** seja um par de nós (i, j), o menor caminho possível dentre todos os possíveis é chamado de menor caminho médio ou distância geodésica. Essa medida representa a maneira mais eficiente de percorer a rede, minimizando o tempo e custo para torca de informações (BORGWARDT; KRIEGEL, 2005; VERA, 2011).

  Considerando um conjunto de vértices V pertencente a um grafo G, o menor caminho médio entre um ponto i e j onde $i, j \in V$ é dado pela equação 7.

$$MC = \frac{1}{n(n-1)} \sum_{i \neq j} D_{i,j} \tag{7}$$

- **Proximidade:** mede o quão próximo um vértice está dos demais nós da rede, em outras palavras, está medida expressa o inverso da distância geodésica média de um ponto em relação aos demais (CARNEIRO, 2016). O cálculo dessa medida é dado pela equação 8.

$$P = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{n-1}{\sum_{j=1}^{n} d(i,j)} \right) \tag{8}$$

- **Intermedialidade:** A intermedialidade, também conhecida como betweenness é uma medida de centralidade de vértices. Tecnicamente, quanto maior o valor de Intermedialidade de um vértice, maior é o tráfego de informações e acesso do mesmo (BILZã, 2015). O cálculo dessa medida é dado pela equação 9.

$$b_i = \sum_{u,v \in V-i} n_{uv}^{i} \tag{9}$$

2.4 Binários empacotados

O empacotamento é um meio de distribuir um executável em um estado compactado ou ofuscado, tornando mais difícil detectar, analisar estaticamente ou realizar uma engenharia reversa. No contexto do malware, uma vez que a carga útil maliciosa é compactada ou ofuscada, os serviços de segurança, como os antivírus que realizam análise estática automatizada podem ter problemas para sinalizar o binário como malicioso, tornando uma grande vantagem para os desenvolvedores de malware (KINDREDSEC, 2020).

Para o empacotamento de arquivos, temos diversas arquiteturas, porém a mais utilizada é a arquitetura de empacotamento "stub-payload", representado pela Figura 6. Nesta arquitetura, é criado um novo executável que contém dois componentes principais: o conteúdo compactado original e um pequeno pedaço de código responsável por descompactar o arquivo original e executá-lo. Esse pequeno trecho de código costuma ser chamado de stub (KINDREDSEC, 2020).

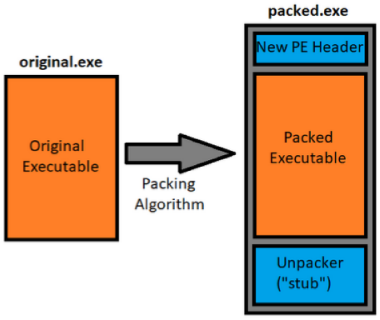


Figura 6 – Transformação de um executável original em um arquivo empacotado, evidenciando as diferenças entre as arquiteturas de cada arquivo.

Fonte: (KINDREDSEC, 2020).

Algumas informações podem ser verificadas para identificar a compactação de um arquivo em específico:

- **Nomes de seção não padrão:** em um executável tradicional, você geralmente terá as mesmas seções todas as vezes (.text, .data, .rsrc, etc). No entanto, muitos compactadores definem suas próprias seções personalizadas, o que indica que o executável não é padrão e que talvez pode estar empacotado (CALHOUN; COLES, 2008).

- **Seções com um tamanho bruto pequeno, mas um tamanho virtual grande:** Quando há um arquivo com um tamanho bruto pequeno, isso indica que o executável real não contém nenhum dado bruto naquela seção específica. No entanto, quando o executável é carregado na memória, o tamanho bruto não é mais relevante e, em vez disso, o tamanho virtual de cada seção específica é alocado na memória (CALHOUN;

COLES, 2008). Se uma seção está sendo alocada em uma grande quantidade de espaço virtual, mas não contém dados brutos reais, isso pode ser um indicativo na qual o código descompactado pode eventualmente ser gravado, o que geralmente é feito por algoritmos de descompactação.

- **Seções com entropia muito alta:** A palavra entropia se refere à variância e "aleatoriedade" de um dado, logo um texto em uma linguagem específica, ou um código em assembly, uma vez que as mesmas seguem padrões determinísticos e por isso possuem uma baixa entropia, por outro lado, quando um arquivo é criptografado ou compactado, os padrões não se tornam tão "previsíveis" e então temos um arquivo com uma alta entropia (OYA, 2016).

Ao apresentar algumas maneiras de identificar um arquivo binário empacotado é possível notar a dificuldade para uma análise manual, ou até mesmo para um antivírus comum devido a técnicas de ofuscação. Portanto é viável o desenvolvimento de um ambiente de alto nível capaz de classificar os dados a nível computacional.

2.5 Trabalhos Relacionados

Existem diferentes trabalhos que estão relacionados com a classificação de empacotados (ASSIS et al., 2019; PARK; RUIZ; MONTES, 2011), apresentando resultados significativos para identificação do mesmo, porém todas as abordagens utilizadas analisam somente atributos físicos da base de dados. Nesse contexto temos a classificação por redes complexas, que na literatura temos a classificação voltada a conformidade padrão (CARNEIRO; GAMA; RIBEIRO, 2021; CARNEIRO et al., 2014), que visa classificar um novo objeto na classe em que apresentar maior conformidade com os padrões da rede, e a classificação por caracterização de importância (CARNEIRO; ZHAO, 2018), em que os objetos não rotulados são classificados na classe em que ele recebe maior importância. Esses trabalhos relacionados serão descritos a seguir.

2.5.1 BinStat

BinStat é uma aplicação desenvolvida por Kil Jin Brandini Park, Rodrigo Ruiz, Antônio Montes, cujo o objetivo é analisar e classificar executáveis empacotados (PARK; RUIZ; MONTES, 2011).

Sua metodologia se baseia na classificação dos valores por Árvore de Decisão para validação, com isso, cada executável é analisado pela segmentação do arquivo em blocos, e os mesmos são medidos por treze cálculos estatíticos. Por fim, os resultados são expostos na árvore de decisão, retornando a classe que cada bloco pertence (PARK; RUIZ; MONTES, 2011).

A aplicação é subdividida em quatro módulos: Módulo estatístico, Módulo de decisão, Módulo de classificação e Parser do Resultado.

Na fase de treinamento, os arquivos executáveis empacotados e não empacotados serão fornecidos como entrada para o módulo de estatística, que por sua vez é fornecido para o módulo de tomada de decisão, onde a árvore de decisão será treinada. Após construir a árvore, a mesma é passada para o módulo de classificação, e então o módulo de classificação poderá receber solicitações de classificação de arquivos executáveis com status de embalagem desconhecido.

Na fase de produção, o módulo de decisão não é ativado. Dessa forma, os cálculos a respeito do executável gerados pelo módulo estatístico são repassados ao módulo de classificação, que alimenta o parser de resultado, encarregado de formatar a saída para processamentos futuros (PARK; RUIZ; MONTES, 2011).

Em (ASSIS et al., 2019) são realizados testes em diferentes classificadores clássicos, presentes na literatura e comparado o desempenho entre cada classificador na detecção do empacotados. Os resultados apontam que classificadores que apresentam algum tipo de correlação linear, são mais indicados para mapear os arquivos originais, por outro lado, os classificadores não lineares apresentam resultados melhores para os empacotados. Logo, os resultados obtidos apontam uma melhoria no trabalho de (PARK; RUIZ; MONTES, 2011), ao considerar mais de um classificador, visto que há abordagens mais recomendadas para cada situção do problema.

2.5.2 Classificação de alto nível via conformidade padrão

Inicialmente proposta em (SILVA; ZHAO, 2012), consiste na união de classificadores de baixo e alto nível, passando a considerar não somente atributos físicos de cada instância, mas também os padrões de formação, capturados pelas medidas de rede (CARNEIRO, 2016). Portanto, a proposta atrelada as medidas de rede, são utilizadas para um melhor desempenho preditivo dos classificadores de baixo nível.

Tal técnica pode ser dividida em dois passos majoritários, que seriam treinamento e classificação. A etapa de treinamento consiste na criação da rede a partir dos dados de entrada na forma de vetor de atributos, que será utilizada para calcular as medidas de rede para cada classe presente na base de dados, além de serem utilizadas como padrão de conformidade para a inserção de qualquer novo atributo na rede (CARNEIRO, 2016). Na etapa de classificação, para cada novo elemento inserido na rede, será recalculadas todas as medidas para cada classe e analisado o impacto gerado no componente (classe) da rede. Portanto a classe que apresentar a menor variação tornará o rótulo da instância por apresentar maior conformidade com os padrões previamente calculados.

Em (CARNEIRO; GAMA; RIBEIRO, 2021) é realizado um estudo detalhado sobre o desempenho preditivo de oito medidas de rede selecionadas da literatura, com resultados validados via conformidade padrão. Os resultados indicam que medidas como o menor ca-

minho médio e assortatividade, além de apresentarem bons resultados preditivos, também são mais robustos a variações estruturais da rede.

Já em (CARNEIRO et al., 2014) temos um trabalho que consiste em uma abordagem denominada classificação de alto nível em K-associados ótimo, combinando com a classificação via conformidade padrão. A técnica proposta, alem de combinar termos de baixo e alto nível, classifica os dados não apenas por características físicas (atributos), mas também verificando padrões de formação através da conformidade com o padrão da rede.

2.5.3 Classificação via caracterização de importância

Essa técnica foi proposta em (CUPERTINO; ZHAO; CARNEIRO, 2015) e ela utiliza o processo de passeio aleatório para classificar os dados por facilidade de acesso. Em tese, quanto maior a facilidade de acesso de um vértice, mais conexões o mesmo possui e consequentemente ele é mais importância (CARNEIRO, 2016). Por meio de três definições básicas esse processo pode ser explicado:

1. Espaço de estados: dado um passeio aleatório em um grafo G e uma sequência de estados $E[e_1, ..., e_n]$, a probabilidade de ir de um estado a outro é independente dos estados anteriores (GALLAGER, 2011).

2. Probabilidade de transição: é a probabilidade da mudança estado de um caminhante aleatório (CARNEIRO, 2016).

3. Probabilidades limitantes: em um determinado momento após um caminhante aleatório realizar um número infinito de transições, o passeio atinge um estado estacionário (CARNEIRO, 2016).

Em (CARNEIRO; ZHAO, 2018) foi proposta uma abordagem de classificação por caracterização de importância, em que uma nova instância não rotulada é classificada pelo conceito de importância caracterizado pela medida PageRank do Google das redes de dados subjacentes. Além da abordagem, foi apresentada uma nova medida de rede denominada eficiência diferencial espaço-estrutural, para combinar as características físicas e topológicas dos dados de entrada.

Capítulo 3

Materias e Métodos

Nesse capítulo serão apresentadas todas a fases do projeto, como uma descrição detalhada sobre a base de dados empregada, o pré-processamento do mesmo, os métodos e o ambiente de desenvolvimento adotado e por fim a definição do algoritmo para avalição da medidas de rede, via conformidade padrão.

3.1 Base de Dados

Neste trabalho foram utilizadas as bases geradas em (PARK; RUIZ; MONTES, 2011), através de um módulo estatístico que segmenta a entrada em blocos de 1024 bytes, com isso, para cada bloco serão calculados o valor do seu histograma de frequência, que será utilizado como entrada como entrada por treze cálculos estatísticos.

Ao todo foram utilizadas nove bases de dados, sendo um conjunto de dados destinados para treino e os demais para testes, em que sete conjuntos de teste representam empacotadores distintos e um conjunto representa arquivos que não foram empacotados.

Na Tabela 1a é possível perceber que há apenas seis ferramentas de empacotemento binário, já na Tabela 1b temos um empacotador a mais que seria o Themida, portanto na fase de treinamento do ambiente não haverá intâncias destinadas ao empacotador Themida.

Tabela 1 – Distribuição de blocos, atributos e classes na base de treinamento e teste

(a) Treino

Origem	**Conjunto de Treino**	**Atributos**	**Classes**
Original	55035	13	0
UPX	23251	13	1
FSG	27852	13	1
Mew 11	24203	13	1
MPRESS	26026	13	1
XComp	22767	13	1
PECompact	24814	13	1
Somatório	183502	13	2

(b) Teste

Origem	**Conjunto de Teste**	**Atributos**	**Classes**
Original	1390	13	0
UPX	656	13	1
FSG	650	13	1
Mew 11	576	13	1
MPRESS	663	13	1
XComp	655	13	1
PECompact	711	13	1
Themida	26285	13	1
Somatório	31586	13	2

Sobre os atributos, são treze representando cada cálculo estatístico e um destinado aos rótulos (1 para empacotado e 0 para não empacotado):

- **Índice de Simpson:** É um índice útil para medir a diversidade de elementos. Sua equação é dada em 10.

$$S_{b_i} = \frac{\sum_{f=0}^{n} k_f(k_f - 1)}{N(N-1)} \tag{10}$$

Fonte: (MELO, 2008)

- **Distância de Camberra:** A distância de camberra é utilizada para medir a distância entre pares de pontos em um espaço vetorial. Sua equação é dada em 11.

$$CA_{b_i} = \sum_{f=0}^{n} \frac{|X_f - X_{f+1}|}{|X_f| + |X_{f+1}|} \tag{11}$$

Fonte: (LANCE; WILLIAMS, 1966)

- **Distância de Ordem de Minkowski:** É uma métrica em um espaço vetorial normado, a qual pode ser considerada como uma generalização de ambas as distâncias euclidiana e Manhattan. Sua equação é dada em 12.

$$M_{b_i} = \sqrt[\lambda]{\sum_{f=0}^{n} |X_f - X_{f+1}|^{\lambda}} \tag{12}$$

Fonte: (TABISH; SHAFIQ; FAROOQ, 2009)

- **Distância de Manhattan:** A distância de manhattan é a soma dos comprimentos da projeção da linha que combina eixos e coordenadas. Sua equação é dada em 13.

$$MH_{b_i} = \sum_{f=0}^{n} |X_f - X_{f+1}| \tag{13}$$

Fonte: (GOMES, 2022)

- **Distância de Chebyshev:** A distância de chebyshev é definida pelo espaço vetorial em que a distância entre dois vetores é a maior de suas diferenças ao longo de qualquer dimensão de coordenada. Sua equação é dada em 14.

$$CH_{b_i} = max_f \left| X_f - X_{f+1} \right| \tag{14}$$

Fonte: (BRAUN; KOZAKEVICIUS, 2014)

- **Distância de Bray Curtis:** O índice de Bray-Curtis pode ser expresso como uma proporção de similaridade ou dissimilaridade (distância) na abundância das espécies. Em qualquer um dos casos seus valores vão de um máximo de um ao mínimo de zero. Sua equação é dada em 15.

$$BC_{b_i} = \frac{\sum_{f=0}^{n} \left| X_f - X_{f+1} \right|}{\sum_{f=0}^{n} \left(X_f + X_{f+1} \right)} \tag{15}$$

Fonte: (THOMAS; JOY, 2006)

- **Separação Angular:** O termo separação (ou distância) angular é tecnicamente sinônimo de ângulo, mas tem o objetivo de sugerir a distância linear entre esses objetos. Sua equação é dada em 16.

$$AS_{b_i} = \frac{\sum_{f=0} X_f \cdot X_{f+1}}{\left(\sum_{f=0}^{n} X_f^2 \cdot \sum_{f=0}^{n} X_{f+1}^2 \right)^{\frac{1}{2}}} \tag{16}$$

Fonte: (THOMAS; JOY, 2006)

- **Coeficiente de Correlação:** A correlação procura entender como uma variável se comporta em um cenário onde outra está variando, visando identificar se existe alguma relação entre a variabilidade de ambas, quantificando essa relação. Sua equação é dada em 17.

$$CC_{b_i} = \frac{\sum_{f=0}^{n} \left(X_f - \bar{X}_{b_i} \right) \cdot \left(X_{f+1} - \bar{X}_{b_i} \right)}{\left(\sum_{f=0}^{n} \left(X_f - \bar{X}_{b_i} \right)^2 \cdot \sum_{f=0}^{n} \left(X_{f+1} - \bar{X}_{b_i} \right)^2 \right)^{\frac{1}{2}}} \tag{17}$$

Fonte: (THOMAS; JOY, 2006)

- **Entropia:** Usada para medir o grau de desordem de um sistema. Quanto maior for a variação de entropia de um sistema, maior será sua desordem. Sua equação é dada em 18.

$$E(R) = -\sum_{v \in \Delta_n} t(r_v) \log_2 t(r_v) \tag{18}$$

Fonte: (THOMAS; JOY, 2006)

- **Divergência de Kullback – Leibler:** Também chamada de entropia relativa, é uma medida não-simétrica da diferença entre duas distribuições de probabilidade. Sua equação é dada em 19.

$$KL_{b_i}\left(X_f \parallel X_{f+1}\right) = \sum_{f=0}^{n} X_f \log \frac{X_f}{X_{f+1}} \tag{19}$$

Fonte: (KULLBACK; LEIBLER, 1951)

- **Divergência de Jensen-Shannon:** É um método para medir a similaridade entre duas distribuições de probabilidade. Sua equação é dada em 20.

$$JSD_{b_i}\left(X_f \parallel X_{f+1}\right) = \frac{1}{2} D\left(X_f \parallel \left(\frac{1}{2}\left(X_f + X_{f+1}\right)\right)\right) + \frac{1}{2} D\left(X_{f+1} \parallel \left(\frac{1}{2}\left(X_f + X_{f+1}\right)\right)\right) \tag{20}$$

Fonte: (ENDRES; SCHINDELIN, 2003)

- **Divergência de Itakura – Saito:** É uma medida da diferença entre um espectro original $P(\omega)$ e uma aproximação $\hat{P}(\omega)$ desse espectro. Embora não seja uma medida perceptiva, pretende-se refletir o contraste perceptual. Sua equação é dada em 21.

$$BF_{b_i}\left(X_f, X_{f+1}\right) = \sum_{f=0}^{n} \left(\frac{X_f}{X_{f+1}} - \log\left(\frac{X_f}{X_{f+1}}\right) - 1\right) \tag{21}$$

Fonte: (ITAKURA, 1968)

- **Variação Total:** A função para para variação total é dada pela equação em 22.

$$\delta_{b_i}\left(X_f, X_{f+1}\right) = \frac{1}{2} \sum_{f} \left|X_f - X_{f+1}\right| \tag{22}$$

Fonte: (THOMAS; JOY, 2006)

Sobre as bases, temos um somatório de sete empacotadores distintos que foram coletados pelos trabalhos de (PARK; RUIZ; MONTES, 2011) e que foram testados isoladamente na fase de teste. Dentre os compactadores temos todos apresentados na Tabela 2.

Tabela 2 – Todas as sete ferramentas de empacotamento, seguidas de suas versões e sistemas operacionais em que são destinadas.

Ferramenta	**Versão**	**Sistema Operacinal**
Fast Small Good (FSG)	freeware, versão 2.0	Windows
MEW 11	freeware, versão 1.2	Windows
Matcode Compressor (MPRESS)	freeware, versão 2.18	Windows
PECompact	shareware, versão 3.00.2	Windows
Themida	shareware, versão 2.2.7.0	Windows
Ultimate Packer for eXecutables (UPX)	freeware, versão 3.05	Linux
XComp97	freeware, versão 0.97	Windows

Fonte: (ASSIS et al., 2019) Adaptada.

3.2 Pré-processamento dos dados

Para a base de treino e teste, foram estudadas alternativas de normalização, dentre elas temos a normalização pela norma euclidiana (norma 2). Técnicas de normalização de dados geralmente possuem o mesmo objetivo, que seria em transformar os dados em uma mesma ordem de grandeza.

Dado um vetor $u = (u_1, ..., u_n)$, pertencente ao conjunto numérico $\mathbb{R}^n$. A norma euclidiana, do vetor u é dada pela equação 23.

$$\| u \|_2 = \sqrt{\sum_{i=0}^{n} |u_i|^2} \tag{23}$$

Não foram necessários mais ajustes nas bases de dados, apenas tranformar os dados em uma mesma ordem de grandeza.

3.3 Métodos empregados e ambiente experimental

A construção do algoritimo que utiliza medidas de redes complexas, foi implementado na linguagem Python.

Para criar um ambiente baseado em redes complexas, foi preciso analisar bibliotecas e métodos que proporcionem a criação da rede, cálculo das medidas de desempenho e validação dos resultados obtidos.

Primeiramente, foi necessário analisar meios para construir o grafo. Dentre as analisadas, a escolhida foi o método de formação de rede kNN, além da distância euclidiana com medida de dissimilaridade.

A geração de um grafo kNN $A[i, j]$, em que i é o vértice, e j seria outro vértice diretamente ligado. A rede é gerada a partir da entrada de dados na forma de vetor de atributos sem o rótulo das intâncias, um valor k-vizinhos a serem considerados e uma medida de proximidade utilizada na geração da árvore (Distância euclidiana).

Para realização dos cálculos das medidas de rede temos como entrada de dado um grafo, correspondente a uma classe da rede. As medidas de assortatividade, coeficiente de agrupamento e grau médio, não demandam muito processamento da máquina, além de serem as medidas inicialmente propostas na classificação de alto nível em (SILVA; ZHAO, 2012). As demais medidas de rede como menor caminho médio, intermedialidade e proximidade, demandam mais processamento a medida que a densidade da rede é maior, no entanto essas medidas expressam características valiosas da rede, reforçando a necessidade da utilização para análise.

Por último, é importante destacar que foram utilizadas um total de seis medidas de rede e para validação do resultados obtidos por cada medida, foi utilizado a acurácia média em relação a instâncias previstas acertivamente.

3.4 Algoritmo

Para análise do resultados é necessário a construção de ambiente de aprendizado, começando pela construção do grafo da base de treino em forma de vetor de atributos. A composição da rede foi realizada pelo método kNN e utilizando a distância euclidiana como medida de proximidade, comentado na seção 3.3.

Com a rede gerada, as medidas são aplicadas para cada classe da rede, para que todo novo dado seja definido antes da inserção. Na etapa de testes um novo objeto é inserido na rede e as medidas são recalculadas, logo o rótulo que a nova instância receberá é da classe que apresentar maior conformidade padrão com a medida de rede em questão. A seguir são apresentados os passos principais do algoritmo.

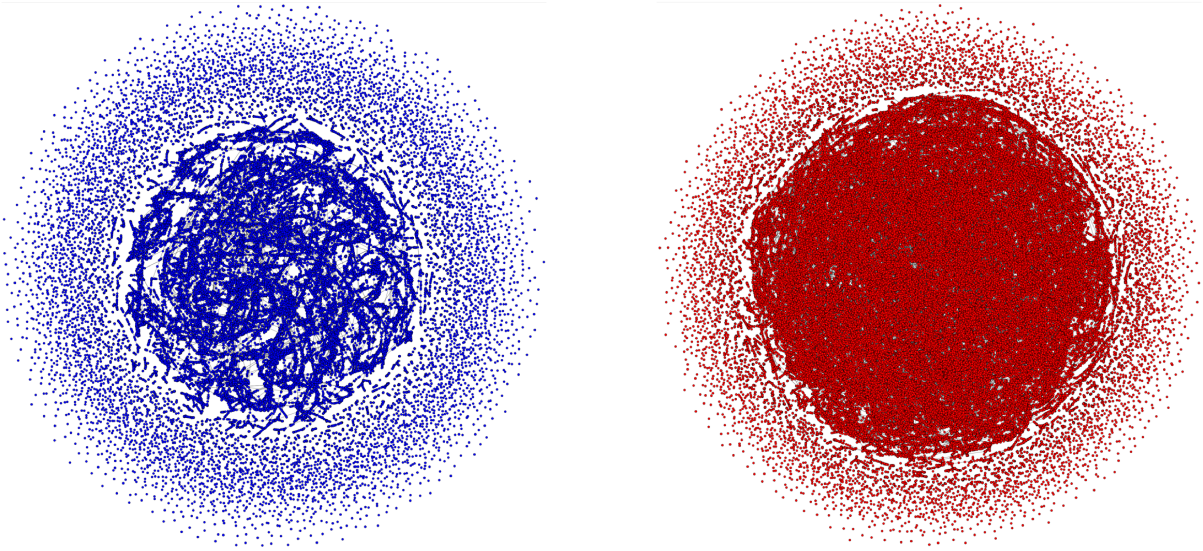
1. Normalização das bases de treino e teste;
2. Construção do grafo kNN da base de treino;
3. Cálculo das seis medidas de rede para cada classe de dados;
4. Aplicação do método de classificação em redes para cada uma das oito bases de testes;
    - 4.1 Quando um novo objeto é conectado ao grafo, todas as medidas de rede são recalculadas;
    - 4.2 A classificação do objeto é dada pela classe que obteve maior conformidade padrão.
5. Finalmente, é aplicada a acurácia para quantificar em porcentagem os dados previstos com acerto.

O Algoritmo 1 apresenta em detalhes os passos para a classificação de alto nível via conformidade padrão.

Algorithm 1 Comformidade padrão

Entrada: X_{teste}
Saída: Dados classificados
// Etapa de treinamento
Construção da rede utilizando X_{treino}
Cálculo das medidas de rede para cada classe
// Etapa de teste
Seleção temporária de conexões para y
recalcular as medidas de rede para cada classe
Classificação de y por conformidade padrão

A Figura 7 exemplifica as redes geradas para arquivos desempacotados 7a e arquivos empacotados 7b, considerando três vizinhos mais próximos como parâmetro de formação da rede-kNN. Visualmente é possível notar que a rede gerada para os empacotados é mais densa, além dos vértices estarem mais próximos dos demais.



(a) Arquivos desempacotados. (b) Arquivos empacotados.

Figura 7 – Exemplo da rede gerada considerando $k = 3$: Arquivos desempacotados (esquerda) e Arquivos empacotados (direita).

Capítulo 4

Resultados experimentais

Nesse capítulo será mostrado a condução dos experimentos e os resultados obtidos pela classificação de alto nível em comparativo com diferentes classificadores, nas bases geradas em (PARK; RUIZ; MONTES, 2011).

4.1 Condução dos experimentos realizados

Para os testes realizados, a escolha dos valores de k adotados na formação da rede foram de um a cinco, além da Distância euclidiana (ED) como medida de dissimilaridade.

Ao realizar teste inicias foram detectados resultados melhores para valores de $k = 1$, além de um rápido tempo de execução, sendo assim, foi conduzido em paralelo outro teste com seis medidas de rede, mas com apenas um valor para k (um). Neste teste também foi adotado a mesma medida de dissimilaridade.

Na etapa de análise de resultados foi realizado um comparativo em que o desempenho da rede será avaliado em relação a diferentes classificadores já aplicados para o problema proposto em (ASSIS et al., 2019). A ideia desta análise seria para validar a hipótese do trabalho apresentada na Seção 1.2 e comprovar a eficiência do classificador de alto nível para o problema proposto.

Por fim, na etapa final de discussão dos resultados, foi realizado uma análise mais detalhada em relação ao desempenho de cada medida de rede e sensibilidade dos parâmetros da rede. Esta etapa é importante, pois é através de uma visão geral do trabalho feito que será possível traçar os trabalhos futuros e melhorias para o ambiente de aprendizado desenvolvido.

4.2 Resultados da classificação de alto nível

Como comentado anteriormente, o primeiro experimento foi realizado com apenas três medidas de rede (Assortatividade (R), Coeficiente de agrupamento (CC), Grau médio (K)), e com valores entre $[1, 5]$ em k, juntamente de três baterias de testes. A escolha de apenas algumas medidas em relação ao valor de k, que se mostra necessitar de muito processamento a medida que o valor de vizinhos são maiores, aumentando em larga escala o tempo de resolução para o problema proposto para as medidas de Proximidade (P), Intermedialidade (B) e Menor caminho médio (MC), dificultando a resolução devido a densidade da rede.

A Tabela 3 apresenta os resultados obtidos e os respectivos valores para o parâmetro k. Os resultados estão relacionados às maiores acurácias encontradas dado as variações do parâmetro de construção da rede (k). É possível identificar que os melhores resultados são em relação aos arquivos empacotados, visto que o classificador de alto nível apresentou dificuldades em mapear os arquivos originais (desempacotados), principalmente para as medidas de rede R e K.

Para os programas de empacotamento é possível inferir que a medida de rede Grau médio apresenta os melhores resultados preditivos, em que para todos os sete compactadores os valores de acurácia foi superior a 98%. Enquanto que para arquivos originais/descompactados temos que o Coeficiente de agrupamente obteve o melhor resultado com um valor de 82.8%, sendo um resultado razoável em comparação com as medidas Assortatividade e Grau Médio, que por sua vez apresentaram resultados insatisfatórios no melhor cenário.

Uma outra análise realizada foi em relação à sensibilidade das medidas para a mudança no valor do parâmetro k, em que para os empacotados o coeficiente de agrupamento demonstra suscetível ao incremento de vizinhos, apesentando resultados melhores, equanto as medidas de Assortatividade e Grau Médio decaem bastante. Para os arquivos originais os resultados obtidos pelas medidas de Assortatividade e Grau Médio apresentaram melhorias para valores k maiores, mas não de grande valia para o problema, entretanto o Grau Médio demostrou uma grande queda nos resultados. Logo é possível afirmar que os melhores resultados tanto para arquivos desempacotados e empacotados, giram em torno do valor um para o parâmentro k.

Tabela 3 – Resultados preditivos para cada medida de rede adotada, pela métrica euclidiana de formação da rede. Entre parênteses temos o valor de K usado para construir a rede. Os arquivos empacotados estão separados para cada ferramenta de empacotamento selecionada para a análise.

Medidas	**Desempacotados**	**Ferramentas de empacotamento binário**						
	Original	***UPX***	***FSG***	***Mew 11****	***MPRESS***	***XComp***	***PECompact***	***Themida****
Assortatividade	61.2 (k=3)	97.5 (k=1)	96.9 (k=1)	99.3 (k=1)	95.7 (k=1)	97.1 (k=1)	97.2 (k=1)	99.6 (k=1)
Coef. de Agrup.	**82.8** (k=1)	90.2 (k=5)	92.1 (k=5)	92.0 (k=3)	90.7 (k=5)	88.5 (k=5)	90.4 (k=5)	92.8 (k=5)
Grau Médio	61.3 (k=2)	**98.4** (k=1)	**98.3** (k=1)	**99.8** (k=1)	**98.4** (k=1)	**98.3** (k=1)	**98.2** (k=1)	**99.8** (k=1)

4.3 Análise com outras medidas de rede

Seguindo para o segundo experimento, agora temos todas as seis medidas comentadas na Seção 2.3, que seriam Assortatividade, Coeficiente de Agrupamento, Grau médio, Intermedialidade, Menor Caminho Médio e Proximidade, todas para o parâmetro k igual a um, novamente com a mesma medida de dissimilaridade comentada na Seção 4.1.

A Tabela 4, apresenta os melhores resultados obtidos para cada uma das seis medidas de rede. Ao analisar os resultados, ainda temos que a medida de rede com o melhor desempenho para detecção dos empacotados é o Grau Médio, e com o advento de mais três novas medidas temos uma melhoria no melhor resultado para arquivos desempacotados de 4.9%, para a medida de Proximidade com o melhor resultado de 89.7%. Também é possível notar que as medidas de Coeficiente de Agrupamento e Proximidade que possuem resultados melhores para os desempacotados, decaem consideravelmente tratando-se dos empacotados, e analogamente podemos inferir a mesma análise para as demais medidas que possuem um bom desempenho ao analisar as ferramentas, mas com um melhor resultado bem abaixo para os arquivos originais.

Tabela 4 – Resultados preditivos para cada medida de rede adotada, considerando as redes formadas para $k = 1$. Os arquivos empacotados estão separados para cada ferramenta de empacotamento selecionada para a análise.

Medidas	**Desempacotados**	**Ferramentas de empacotamento binário**						
	Original	***UPX***	***FSG***	***Mew 11****	***MPRESS***	***XComp***	***PECompact***	***Themida****
Assortatividade	54.9	97.5	96.9	99.3	95.7	97.0	97.1	99.6
Coef. de Agrup.	82.8	81.5	84.1	87.3	79.0	77.5	81.8	88.9
Grau Médio	48.0	**98.4**	**98.3**	**99.8**	**98.4**	**98.3**	**98.2**	**99.8**
Intermedialidade	56.9	88.4	90.1	92.1	90.3	85.3	89.1	91.9
Menor Caminho	52.4	93.2	93.6	94.9	93.6	90.2	93.3	94.5
Proximidade	**89.7**	79.2	82.1	86.9	77.6	77.0	79.8	88.6

4.4 Comparativo entre diferentes classificadores

Como parâmetro de validação para os melhores resultados conquistados em ambos testes, é feito uma comparação com alguns dos mais conhecidos classificadores da área: Classification And Regression Trees (CART), Random Forest (RF), K-vizinhos mais próximos (kNN), Naive Bayes (NB), Multilayer Perceptron (MLP), Support Vector Machine (SVM) e Árvore de decisão (C5.0).

Como podemos observar na Tabela 5, os resultados obtidos no trabalho pela classificação de alto nível, mostrou ter o melhor desempenho em todos os sete compactadores escolhidos para o estudo extraídos em (ASSIS et al., 2019). Vale destacar os resultados obtidos para os empacotadores *Themida e *Mew 11, por apresentarem os melhores resultados dentre os demais (99.8%). Para arquivos não empacotados a Rede Complexa obteve um resultado satisfatório de oitenta e nove ponto sete por cento (89.7%), logo atrás do NB com uma acurácia de noventa ponto oito por cento (90.8%).

Também é possível perceber é que os classificadores não lineares tiveram uma certa dificuldade em mapear arquivos desempacotados, por outro lado estes classificadores apresentaram bons resultados ao categorizar os empacotados.

Esse fenômeno pode ser explicado pela estrutura das classes da base de dados, em que para uma base de dados bidimensional, um classificador linear tenta encontrar uma função linear que produza uma reta capaz de separar as amostras em suas classes. Por outro lado, um classificador não-linear busca encontrar uma função não-linear para classificar as amostras e, desse modo, consegue produzir ajustes mais complexos que representam, com maior precisão, a estrutura das classes das amostras contidas na base (SOUZA, 2018). Levando em conta esta explicação, é possível determinar que para o problema dos empacotados, a adoção de um classificador linear para classificar os arquivos empacotados e utilizar um classificador não linear para os arquivos originais, seria uma boa opcção.

Outro ponto interessante é que todos os resultados do classificador kNN foram superados pela classificação de alto nível, revelando que esta abordagem pode entregar um desempenho melhor para o problema proposto.

Tabela 5 – Resultado preditivo dos arquivos originais e empacotados em comparação com diferentes classificadores. Os arquivos empacotados estão separados de acordo com a ferramenta de empacotamento.

Classificador	**Desempacotados**	**Ferramentas de empacotamento binário**						
	Original	***UPX***	***FSG***	***Mew 11****	***MPRESS***	***XComp***	***PECompact***	***Themida****
CART	78.3	94.7	94.9	98.1	93.7	96.8	94.0	99.7
RF	82.9	95.7	97.8	98.6	97.3	95.3	95.2	**99.8**
kNN	81.6	96.6	97.8	98.4	96.5	96.3	96.2	99.7
NB	**90.8**	78.4	92.9	90.5	74.1	76.8	78.3	99.3
MLP	79.0	91.9	95.1	99.3	90.3	96.3	92.7	99.7
SVM	83.5	95.7	97.8	99.3	95.8	97.1	95.8	**99.8**
C5.0 (Binstat)	80.2	94.3	96.7	97.9	96.3	95.7	95.3	99.5
Redes Complexas	89.7	**98.4**	**98.4**	**99.8**	**98.4**	**98.3**	**98.2**	**99.8**

4.5 Discussão dos resultados

Os resultados apresentados em ambos experimentos foram de suma importância para o problema em questão, pois no primeiro teste ao considerar o incremento de vizinhos na classificação de alto nível, foi possível perceber a sensibilidade das medidas em que os resultados decaíram consideravelmente, logo o segundo teste foi realizado para avaliar mais medidas de rede, mas com apenas o valor um para o parâmetro k. No geral os resultados apresentados na Tabela 6 foram ótimos, com uma alta taxa de acerto para os arquivos empacotados, porém com uma certa dificuldade para mapear os originais. Também é notável que algumas medidas são mais apropriadas para categorizar os empacotados e as demais são melhores para mapear os originais, em que a medida de Grau médio obteve o melhor desempenho para os empacotados e a medida de Proximidade teve o melhor desempenho em catalogar os desempacotados.

Finalmente será discutido sobre as demais medidas de rede que não apareceram na Tabela 6. A Assortatividade apresentou bons resultados para os arquivos empacotados, superando até mesmo alguns classificadores como CART, NB, MLP e C5.0. O menor caminho medio apresentou resultados inferiores aos da Assortatividade, mas ainda sim razoáveis para os empacotados. Já a Intermedialidade não apresentou resultados pertinentes, além de ser a medida que mais demandou processamento, portanto para o problema em questão, esta medida de rede não é recomendada. A última medida é o coeficiente de agrupamento que apresentou uma facilidade maior para mapear os arquivos originais, com resultados melhores que os classificadores CART, kNN, MLP e C5.0.

Tabela 6 – Ranking das medidas de rede mais efetivas para cada ferramenta de empacotamento.

Origem	**Melhor medida**	**Melhor resultado**
Original	Proximidade	89.7
UPX	Grau Médio	98.4
FSG	Grau Médio	98.3
Mew 11	Grau Médio	99.8
MPRESS	Grau Médio	98.4
XComp	Grau Médio	98.3
PECompact	Grau Médio	98.2
Themida	Grau Médio	99.8

Capítulo 5

Conclusão

O trabalho teve como objetivos principais a confecção de um ambiente de aprendizado de alto nível para detecção de binários empacotados, além de avaliar os resultados frente a outros classificadores renomados presentes na literatura. Os objetivos foram propostos pois não havia um estudo detalhado sobre um classificador de alto nível, via conformidade padrão para a análise de arquivos binários empacotados, visto que esta abordagem considera a estrutura de formação dos dados de entrada e também ao considerar diferentes medidas de rede, é possível obter bons resultados tanto para os arquivos desempacotados e empacotados. Desse modo os objetivos específicos foram atingidos, provando a eficiência da classificação de alto nível para o problema, além da validação da base de dados utilizada e confeccionada em (PARK; RUIZ; MONTES, 2011), gerada através da segmentação do arquivo em blocos e estes expostos a treze cálculos estatísticos, bem escolhidos frente ao cenário em questão.

No geral o algoritmo apresentou resultados satisfatórios para todos os programas de empacotamento binário, superando a margem de noventa e oito por cento (98%) de acurácia para cada empacotador analisado, com destaque para os empacotadores Themida e Mew, que foram melhores mapeados pelo ambiente, atingindo os resultados mais próximos ao cem por cento. Em contrapartida, os blocos originais, ou não empacotados, apresentaram alguns bons resultados, mas bem abaixo dos obtidos anteriormente, o que mostra uma certa dificuldade não somente da classificação de alto nível, mas dos demais classificadores analisados na Seção 4.4.

Alguns fatores podem ser levantados quanto à facilidade de mapeamento dos empacotados e a queda na margem de acerto dos não empacotados. O primeiro fator talvez seja pela distribuição de blocos analisados em empacotados e desempacotados, em que temos uma distribuição menor para blocos originais e que pode implicar na predição menos assertiva do mesmo. A segunda razão pode ser pela própria estrutura dos arquivos, em que alguns empacotadores podem ter arquiteturas similares, diferenciando apenas no processo de empacotamento. Como comentado na Seção 2.4, a arquitetura Stub-payload é apenas umas das arquiteturas mais utilizadas e que possívelmente tornem alguns resultados

iguais de blocos referentes a empacotadores diferentes, explicando o fato do Themida ter o melhor desempenho mesmo sem nenhum bloco analisado na fase de treino. O mesmo pode não acontecer para arquivos originais dificultando a categorização do mesmo.

Por fim, comentando sobre a ferramenta de detecção de empacotados Binstat, é válido afirmar que a arquitetura composta de um módulo estatístico que transforma as informações bloco a bloco de um arquivo em dados para análise computacinal até um resultado da categoria do arquivo via árvore de decisão C5.0, é funcional e auxilia na detecção de um empacotado, com uma alta taxa de acerto. Entretanto, este trabalho provê evidências de que uma abordagem de alto nível via conformidade padrão proporciona resultados significativos para o problema, devido a sua flexibilidade e poder de caracterização das medidas de rede. Logo espera-se que a investigação ampla de tais modelos contribua tanto em outras aplicações da literatura quanto na área de segurança da informação.

Em trabalhos futuros planeja-se utilizar uma gama maior de medidas de rede e métodos de construção de rede presentes na literatura, além da utilização da validação cruzada para a lapidação dos resultados preditivos. Também pretende-se utilizar a classificação de alto nível frente a mais desafios da segurança da informação como a detecção da esteganografia em arquivos maliciosos.

Referências

ACADEMY, D. S. **Deep Learning Book**. [S.l.]: Online, 2022. Citado na página 19.

ARAúJO, J. P. de. **Análise da Taxa de Convergência da Regra de Classificação dos k-Vizinhos Mais Próximos**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, Outubro 2018. Citado na página 21.

ASSIS, C. R. O. et al. A comparative analysis of classifiers in the recognition of packed executables. In: **2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)**. [S.l.: s.n.], 2019. p. 1356–1360. Citado 6 vezes nas páginas 14, 26, 27, 33, 36 e 39.

BATISTA, G. E. de A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - USP, São Carlos, SP, 2003. Citado na página 17.

BILZã, M. d. A. **Rotulação de indivíduos representativos no aprendizado semissupervisionado baseado em redes: caracterização, realce, ganho e filosofia**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - USP, São Carlos, 2015. Citado na página 24.

BORGWARDT, K. M.; KRIEGEL, H.-P. Shortest-path kernels on graphs. In: IEEE. **Fifth IEEE international conference on data mining (ICDM'05)**. [S.l.], 2005. p. 8–pp. Citado na página 24.

BOTACIN, M. et al. Introdução à engenharia reversa de aplicações maliciosas em ambientes linux. In: _____. [S.l.: s.n.], 2019. Citado na página 14.

BRAUN, E.; KOZAKEVICIUS, A. Revisitando conjuntos e distâncias para encontrar pontos vizinhos. **XX EREMAT-Encontro Regional de Estudantes de Matemática da Região Sul fundação Universidade Federal do Pampa**, p. 13–16, 2014. Citado na página 31.

BRUNIALTI, L. et al. Aprendizado de máquina em sistemas de recomendação baseados em conteúdo textual: Uma revisão sistemática. In: **Anais do XI Simpósio Brasileiro de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2015. p. 203–210. Citado na página 18.

CALHOUN, W. C.; COLES, D. Predicting the types of file fragments. **Digital Investigation**, v. 5, p. S14–S20, 2008. ISSN 1742-2876. The Proceedings of the Eighth Annual DFRWS Conference. Citado 2 vezes nas páginas 25 e 26.

CAMILO, C.; SILVA, J. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. 2009. [Online; accessed 10-janeiro-2020]. Citado na página 20.

CARNEIRO, M. G. **Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - USP, São Carlos, 2016. Citado 7 vezes nas páginas 15, 17, 22, 23, 24, 27 e 28.

CARNEIRO, M. G.; GAMA, B. C.; RIBEIRO, O. S. Complex network measures for data classification. In: **2021 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2021. p. 1–8. Citado 2 vezes nas páginas 26 e 27.

CARNEIRO, M. G. et al. Network-based data classification: combining k-associated optimal graphs and high-level prediction. **Journal of the Brazilian Computer Society**, SpringerOpen, v. 20, n. 1, p. 1–14, 2014. Citado 2 vezes nas páginas 26 e 28.

CARNEIRO, M. G.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 29, n. 8, p. 3361–3373, 2018. Disponível em: <doi:10.1109/TNNLS.2017.2726082>. Citado 2 vezes nas páginas 26 e 28.

CASTANHEIRA, L. G. **Aplicação de técnicas de mineração de dados em problemas de classificação de padrões**. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Setembro 2008. Citado na página 20.

CERQUEIRA, P. H. R. **Um estudo sobre reconhecimento de padrões: um aprendizado supervisionado com classificador bayesiano**. Dissertação (Mestrado) — Escola Superior de Agricultura Luiz de Queiroz - USP, Janeiro 2010. Citado na página 18.

CHOUINARD, J. C. **k-Nearest Neighbors (KNN) in Python**. 2022. Disponível em: <https://www.jcchouinard.com/k-nearest-neighbors/>. Citado na página 21.

COSTA, L. d. F. et al. Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citado na página 22.

CUPERTINO, T. H.; ZHAO, L.; CARNEIRO, M. G. Network-based supervised data classification by using an heuristic of ease of access. **Neurocomputing**, Elsevier, v. 149, p. 86–92, 2015. Citado na página 28.

EDELL, L. **IS MACHINE LEARNING THE NEW EPM BLACK?** 2015. Disponível em: <https://scorecardstreet.wordpress.com/2015/12/09/is-machine-learning-the-new-epm-black/>. Citado na página 18.

ENDRES, D.; SCHINDELIN, J. A new metric for probability distributions. **IEEE Transactions on Information Theory**, v. 49, n. 7, p. 1858–1860, 2003. Citado na página 32.

GALLAGER, R. G. Discrete stochastic processes. **OpenCourseWare: Massachusetts Institute of Technology**, 2011. Citado na página 28.

GOMES, T. T. **Previsão de cargas multinodais com o uso de Rede Neural ARTMAP Manhattan**. Tese (Doutorado) — Faculdade de Engenharia de Ilha Solteira – Unesp, Maio 2022. Citado na página 30.

IDIKA, N.; MATHUR, A. P. A survey of malware detection techniques. **Purdue University**, Citeseer, v. 48, n. 2, p. 32–46, 2007. Citado na página 14.

ITAKURA, F. Analysis synthesis telephony based on the maximum likelihood method. **Reports of the 6 Int. Cong. Acoust.**, 1968. Citado na página 32.

KINDREDSEC. The basics of packed malware: Manually unpacking upx executables. 2020. [Online; accessed 10-janeiro-2020]. Citado 2 vezes nas páginas 14 e 25.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **The annals of mathematical statistics**, JSTOR, v. 22, n. 1, p. 79–86, 1951. Citado na página 32.

LANCE, G. N.; WILLIAMS, W. T. Computer Programs for Hierarchical Polythetic Classification ("Similarity Analyses"). **The Computer Journal**, v. 9, n. 1, p. 60–64, 1966. Citado na página 30.

MELO, A. S. O que ganhamos 'confundindo' riqueza de espécies e equabilidade em um índice de diversidade? **BIOTA NEOTROPICA**, v. 3, n. 8, 2008. Citado na página 30.

MONARD, M. C. et al. **Uma introdução ao aprendizado simbólico de máquina por exemplos**. [S.l.]: ICMSC-USP, 1997. Citado na página 17.

NEVES, E. C. **Aprendizado por Reforço**. 2020. Disponível em: <https://medium.com/turing-talks/aprendizado-por-reforço-1-introduç~ao-7382ebb641ab>. Citado na página 20.

NEWMAN, M. E. The structure and function of complex networks. **SIAM review**, SIAM, v. 45, n. 2, p. 167–256, 2003. Citado na página 22.

OYA, J. K. M. **Utilização de árvores de decisão para aprimorar a classificação de fragmentos**. Dissertação (Mestrado) — Departamento de Engenharia Elétrica - UNB, Dezembro 2016. Citado na página 26.

PARK, K. J. B.; RUIZ, R.; MONTES, A. Binstat: Tool for recognition of packed executables. **The International Journal of Forensic Computer Science**, v. 6, n. 1, p. 44–58, 2011. Citado 7 vezes nas páginas 14, 26, 27, 29, 32, 36 e 41.

PELEGRIN, D. C. et al. A shell de data mining orion: Classificação, clusterização e associação. **Congresso Sul Brasileiro de Computação (SULCOMP)**, v. 1, 2012. Citado na página 18.

PETERMANN, R. J. **Modelo de mineração de dados para classificação de clientes em telecomunicações**. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica, 2006. Citado na página 20.

RUBINOV, M.; SPORNS, O. Complex network measures of brain connectivity: uses and interpretations. **Neuroimage**, Elsevier, v. 52, n. 3, p. 1059–1069, 2010. Citado na página 22.

SILVA, T. C.; ZHAO, L. Network-based high level data classification. **IEEE Transactions on Neural Networks and Learning Systems**, v. 23, n. 6, p. 954–970, 2012. Citado 3 vezes nas páginas 14, 27 e 34.

SOUZA, N. A. de. **Aumentando o poder preditivo de classificadores lineares através de particionamento por classe**. Dissertação (Mestrado) — Centro de Ciências em Gestão e Tecnologia – CCGT - UFUSCar, Janeiro 2018. Citado na página 39.

TABISH, S. M.; SHAFIQ, M. Z.; FAROOQ, M. Malware detection using statistical analysis of byte-level file content. In: **Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics**. New York, NY, USA: Association for Computing Machinery, 2009. (CSI-KDD '09), p. 23–31. Citado na página 30.

THOMAS, M.; JOY, A. T. **Elements of information theory**. [S.l.]: Wiley-Interscience, 2006. Citado 2 vezes nas páginas 31 e 32.

VERA, A. M. **Propriedades de redes complexas de telecomunicações**. Dissertação (Mestrado) — Escola de Engenharia de São Carlos, Agosto 2011. Citado 2 vezes nas páginas 22 e 24.

WAUKE, J. **Um guia para iniciantes sobre como as máquinas aprendem (Machine Learning)**. 2020. Disponível em: <https://jobu.com.br/2020/10/24/um-guia-para-iniciantes-sobre-como-as-maquinas-aprendem-machine-learning/>. Citado na página 19.