

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BRUNA DE CAMARGO RUBIO

**CLASSIFICAÇÃO E OTIMIZAÇÃO DE CARACTERÍSTICAS PARA
DETECÇÃO DE ANOMALIAS EM REDES DE COMPUTADORES**

BAURU
2016

BRUNA DE CAMARGO RUBIO

**CLASSIFICAÇÃO E OTIMIZAÇÃO DE CARACTERÍSTICAS PARA
DETECÇÃO DE ANOMALIAS EM REDES DE COMPUTADORES**

Trabalho de Conclusão de Curso do Curso
de Bacharelado em Ciência da Computação
da Universidade Estadual Paulista “Júlio
de Mesquita Filho”, Faculdade de Ciências,
Campus Bauru.

Orientador: Prof. Dr. Kelton Augusto Pontara
da Costa

BAURU
2016

Rubio, Bruna de Camargo.

Classificação e otimização de características para detecção de anomalias em redes de computadores / Bruna de Camargo Rubio, 2016

81 p. : il.

Orientador: Kelton Augusto Pontara da Costa

Monografia (graduação)–Universidade Estadual Paulista. Faculdade de Ciências, Bauru, 2016

1. Segurança de redes de computadores. 2. Detecção de anomalia. 3. Inteligência artificial. 4. Aprendizado de máquina. 5. Classificadores de padrões. 6. Floresta de caminhos ótimos. 7. Metaheurística. 8. Otimização por enxame de partículas. I. Universidade Estadual Paulista. Faculdade de Ciências. II. Título.

Bruna de Camargo Rubio

Classificação e Otimização de Características para Detecção de Anomalias em Redes de Computadores

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Kelton Augusto Pontara da Costa
Orientador

Prof^a Dr^a Simone das Graças Domingues Prado
Prof^a. Disciplina

Prof. Dr. João Paulo Papa
Universidade Estadual Paulista “Júlio de Mesquita Filho”
Faculdade de Ciências
Departamento de Computação

Bauru, 24 de Novembro de 2016.

*Dedico este trabalho à Vida que, assim
como a Ciência, é construída aos poucos
por todos que se interessam por ela.*

Agradecimentos

Primeiramente agradeço aos meus pais, Ricardo e Simone, que me criaram da melhor forma possível com muito amor e carinho; que sempre me apoiaram, e me ensinaram à ir além, e que este ainda é um pequeno passo do que vêm pela frente.

À minha família, meus avós Alcides, Iracema, Celestino e Maria, meus tios e em especial aos meus primos: Amanda, Marcelo, Erick e Joyce que são muito mais que primos, são os irmãos que não tive e me fizeram perceber que pra família, não importa a distância, todos estão sempre no coração e sem os quais eu não sei como seria a vida.

Agradeço também ao meu lindo, Gustavo de Rosa, que apareceu na minha vida aos poucos, e foi ganhando um espaço cada vez maior em meu coração, e que eu amo mais do que as palavras podem descrever. Que este tempo que passamos juntos, seja apenas o começo de algo muito maior que conquistaremos, sempre juntos. Que todo o amor e dedicação que tem por mim, eu possa retribuir a cada dia.

Ao novo integrante da família, Prince, o gostosolden, que já em tão pouco tempo me mostrou uma nova forma de amor, pura e completa, e que alegra todos os meus dias.

Às minhas amigas, Sofia, a qual conheci no meu primeiro dia na cidade sanduíche, e que não importa o tempo 15, 20 anos sempre estaremos juntas; Nathalie que apesar das diferenças tivemos uma sintonia, que espero levar além da universidade.

Ao meu orientador, Kelton, que me ajudou e me aguentou durante o desenvolvimento deste trabalho, para que no fim, fosse um trabalho digno para encerrar este ciclo.

Aos meus professores, da faculdade e da escola, que me ensinaram, direta e indiretamente, e me passaram o conhecimento necessário para chegar à este momento.

À UNESP Bauru, a qual me mostrou um novo mundo, com alegrias, responsabilidades e amigos que passaram em minha vida e que de certa forma não vou me esquecer. A qual tenho o orgulho de poder dizer: Eu sou da UNESP!

*"One day you'll leave this world behind
So live a life you will remember"*
The Nights - Avicii

Resumo

A evolução das redes de computadores, em especial da internet, traz à tona um sério problema enfrentado atualmente: a Segurança de Redes de Computadores. Tal preocupação deve-se ao crescente número de pessoas interessadas em obter informações não autorizadas. Uma das áreas mais pesquisadas dentro da segurança de redes é a detecção de intrusão, especificamente, a detecção de anomalia. Um dos desafios desta área é a grande variedade e complexidade das anomalias, o que leva pesquisadores e empresas à investirem em abordagens inovadoras, tal como, a inteligência artificial. Portanto, este trabalho propõe a utilização de técnicas de aprendizado de máquina com o intuito de criar uma abordagem eficaz para a detecção de anomalias. A utilização de um classificador de padrões baseado em Florestas de Caminhos Ótimos (*Optimum-Path Forest* - OPF) aliada à meta-heurística da Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) busca uma solução que minimize o número de características, sem a perda significativa da acurácia de detecção. A proposta mostrou-se comprovada através dos resultados obtidos, que apresentaram melhores taxas de classificação com um menor número de características.

Palavras Chaves: Segurança de Redes de Computadores; Detecção de Anomalia; Inteligência Artificial; Aprendizado de Máquina; Classificadores de Padrões; Floresta de Caminhos Ótimos; Meta-heurística; Otimização por enxame de Partículas.

Abstract

The development of computer networks, particularly the internet, brought up a serious problem nowadays: the Computer Network Security. This issue is due to a growing number of people interested in obtaining unauthorized information. One of the most researched areas within the network security is intrusion detection, specifically, anomaly detection. The main challenge in this area is the wide variety and complexity of anomalies, leading researchers and companies to invest in innovative approaches, such as artificial intelligence. Therefore, this work proposes to use machine learning techniques in order to create an effective approach to anomaly detection. A pattern recognition methodology based on Optimum-Path Forest (OPF) allied with a meta-heuristic Particle Swarm Optimization (PSO) provides a solution that minimizes the number of features, without significant loss of detection accuracy. The proposal is confirmed by the results obtained, which showed best classification rates with a smaller number of features.

Key Words: Computer Network Security; Anomaly detection; Artificial Intelligence; Machine Learning; Pattern Recognition; Optimum Path Forest; Meta-heuristic; Particle Swarm Optimization.

Listas de ilustrações

Figura 1 – Ataques mundiais à redes de computadores em Setembro de 2016	15
Figura 2 – Rede de Computadores.	18
Figura 3 – Concepção original da Ethernet.	20
Figura 4 – Camadas Modelo OSI.	21
Figura 5 – Camadas Modelo OSI x TCP/IP.	22
Figura 6 – Protocolos e Camadas TCP/IP em comparação ao Modelo OSI.	23
Figura 7 – Arquitetura LAN sem fio.	28
Figura 8 – Segurança em Rede de Computadores.	30
Figura 9 – Criptografia com chave simétrica.	31
Figura 10 – Criptografia com chave pública.	32
Figura 11 – <i>Firewall</i>	34
Figura 12 – Tipo de <i>Firewalls</i>	35
Figura 13 – Classificação dos SDIs.	36
Figura 14 – Arquitetura da metodologia de Detecção de Anomalia.	38
Figura 15 – Classificação das técnicas de detecção.	39
Figura 16 – Exemplo <i>tcpdump</i>	42
Figura 17 – Classificação binária linear.	43
Figura 18 – Ajuste de treinamento de uma classificação.	44
Figura 19 – Funcionamento OPF.	47
Figura 20 – Espaço de busca com bando de pássaros (representação PSO).	50
Figura 21 – Sistematização PSO.	51
Figura 22 – Fluxograma das etapas de desenvolvimento do trabalho.	53
Figura 23 – Representação gráfica do problema proposto.	54
Figura 24 – Fluxograma de criação da Base de Dados.	55
Figura 25 – Manipulação da base no <i>Wireshark</i>	57
Figura 26 – Amostra da Base de Dados criada.	59
Figura 27 – Amostra da Base de Dados criada no formato entendido pelo OPF.	60
Figura 28 – Gráfico da acurácia de classificação pura x iteração em 10% da base de dados. .	64
Figura 29 – Gráfico da acurácia de classificação pura x iteração em 50% da base de dados. .	64
Figura 30 – Gráfico da acurácia de classificação pura x iteração em 100% da base de dados. .	65
Figura 31 – Gráfico da acurácia x iteração do processo de otimização.	66
Figura 32 – Gráfico da acurácia x quantidade de características selecionadas durante o processo de otimização.	66
Figura 33 – Gráfico de comparação de acurácia da classificação com a otimização.	67
Figura 34 – Janela principal da interface.	69
Figura 35 – Caminho para as pastas de instalação das Bibliotecas.	69

Figura 36 – Resultados da operação Classificação e Opção de salvar resultados.	70
Figura 37 – Informações adicionais para o PSO.	70
Figura 38 – Resultados da operação Classificação + Otimização e melhores características. .	71
Figura 39 – Janelas auxiliares: Ajuda e Informações	71

Listas de tabelas

Tabela 1 – Principais tipos de mensagens ICMP.	27
Tabela 2 – Aplicações e protocolos utilizados.	29
Tabela 3 – Comandos <i>tcpdump</i>	41
Tabela 4 – Comparação das técnicas de classificação de padrões na KDDCup.	54
Tabela 5 – Parâmetros do comando <i>tcpdump</i>	56
Tabela 6 – Assinatura de intrusões.	58
Tabela 7 – Formato base de dados	60
Tabela 8 – Programação de execução do OPF na base original.	62
Tabela 9 – Média dos resultados da classificação pura da base de dados.	63
Tabela 10 – Média dos resultados da classificação pura pela porcentagem da base de dados.	65
Tabela 11 – Média dos resultados da otimização.	65
Tabela 12 – Comparação dos resultados da classificação com a otimização.	67
Tabela 13 – Codificação dos Tipos de Anomalias.	79
Tabela 14 – Codificação dos Tipos IGMP.	80
Tabela 15 – Codificação das mensagens DHCPv6.	80
Tabela 16 – Codificação dos Tipos de Pacotes.	80
Tabela 17 – Codificação dos Protocolos.	81

Sumário

1	INTRODUÇÃO	14
1.1	Problema	15
1.2	Justificativa	16
1.3	Objetivos	16
1.3.1	Objetivo Geral	16
1.3.2	Objetivos Específicos	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Redes de Computadores	18
2.1.1	História das Redes de Computadores	19
2.1.2	Protocolos de Redes	21
2.2	Segurança de Redes	29
2.2.1	Ferramentas de Segurança	31
2.3	Anomalias	36
2.3.1	Detecção de Anomalias	37
2.3.2	Ferramentas de Detecção	40
2.4	Classificação de Padrões	42
2.4.1	Floresta de Caminhos Ótimos	46
2.5	Otimização	48
2.5.1	Otimização por Enxame de Partículas	50
3	DESENVOLVIMENTO	52
3.1	Método de Pesquisa	53
3.2	Criação da Base de Dados	55
3.2.1	Captação	55
3.2.2	Rotulação	57
3.3	Classificação	59
3.4	Otimização	60
3.5	Experimentos	61
3.6	Resultados	63
3.7	Interface Gráfica	68
4	CONCLUSÃO	73
	REFERÊNCIAS	74

APÊNDICES

78

APÊNDICE A – CODIFICAÇÃO NO PROCESSO DE ROTULAÇÃO 79

1 Introdução

A crescente difusão das redes de computadores, em especial a *internet*, altera constantemente os atuais parâmetros de segurança através de ataques de intrusões, requerendo assim, sistemas de detecção e prevenção mais aprimorados. Atualmente, os Sistemas de Detecção de Intrusão (SDI) tornam-se desatualizados rapidamente devido à demanda e à complexidade dos ataques que surgem a todo momento.

A interconectividade difundida pelas redes de computadores, especificamente pela *internet*, requer novos métodos de segurança além dos já conhecidos como, por exemplo, *firewalls*, criptografia, controle de acesso, sistemas de detecção e prevenção de intrusão, dentre outras.

Os Sistemas de Detecção e Prevenção de Intrusão utilizam diversas metodologias, tais como, baseada em assinatura, baseada em anomalia ou um sistema híbrido (GARCÍA-TEODORO et al., 2009). Neste trabalho, será aplicada a abordagem baseada em anomalia, a qual parte do princípio que o comportamento anormal é raro e diferente do comportamento definido como normal (WU; BANZHAF, 2010).

Diversas técnicas estão sendo utilizadas a fim de melhorar tais sistemas de detecção, dentre elas, a Inteligência Artificial (IA), que prevê a criação de sistemas capazes de tomar decisões por si só. De acordo com Winston e Pendergast (1984, p. 1), o principal objetivo da inteligência artificial é fazer máquinas inteligentes, propósito o qual sobrepõe o entendimento da própria inteligência e do desenvolvimento da utilidade das máquinas.

A utilização de sistemas inteligentes na detecção de intrusão tem por objetivo monitorar redes de computadores, com o intuito de aprender e detectar novos ataques através de um treinamento, automatizando o processo e tornando a IA uma eficiente ferramenta na detecção de anomalias.

Existem muitas abordagens para a utilização de IA na detecção de anomalias, como mineração de dados (LI; LEE, 2003), sistemas imunológicos artificiais (GUANGMIN, 2008), algoritmos genéticos (SELVAKANI; RAJESH, 2007), *Ball Vector Machine* e *Extreme Learning Machine* (CAI; PAN; CHENG, 2012), Redes Neurais Artificiais com Perceptron Multicamadas (HAYKIN, 1998) e Máquina de Vetores de Suporte (*Support Vector Machine* - SVM) (CORTES; VAPNIK, 1995), dentre outras.

Este trabalho propõe uma abordagem baseada na classificação e otimização das características dos pacotes capturados na rede. A etapa de classificação é responsável por aprender o que é anomalia e assim, ser capaz de classificar novas amostras em duas possíveis classes: anomalia ou não anomalia, e a outra etapa é responsável pela otimização da função

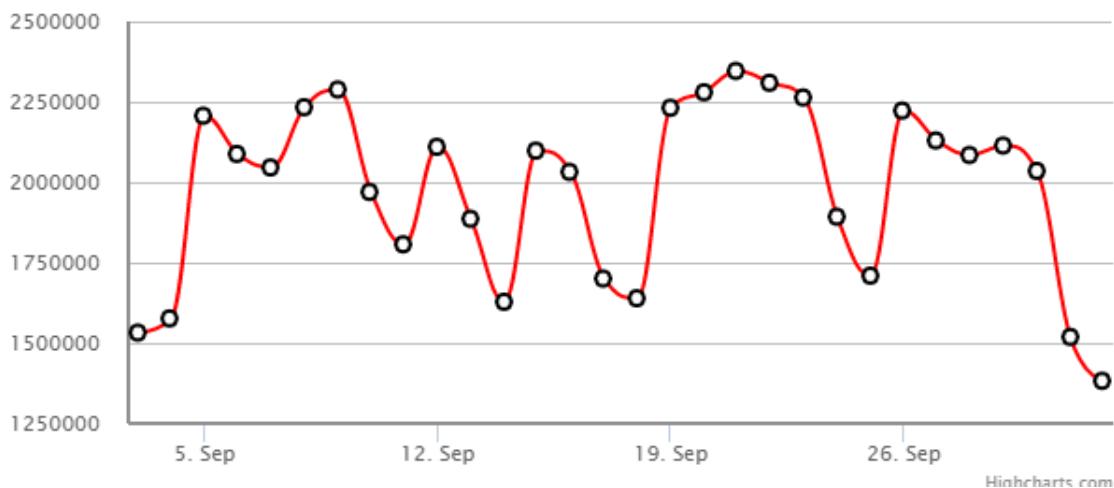
que representa tais características, com o intuito de encontrar o melhor valor que aumenta a acurácia na detecção.

1.1 Problema

Atualmente, a intercomunicação garantida pela internet é intrínseca à sociedade. Contudo, quanto maior a utilização da rede, maior o número de dados que trafegam por ela. Estes dados podem ser de diversas naturezas, tais como, informações pessoais, transações bancárias, negociações internacionais e também, informações de defesa nacional.

O aumento exponencial dos dados que trafegam na rede dificulta a análise de detecção de anomalias, ou seja, de possíveis ataques. Nota-se um crescimento na criação de pacotes maliciosos que buscam recuperar tais informações, a fim de usá-las indevidamente. Como exemplo, segundo informações do CERT.br¹, o ano de 2014 teve um aumento de 197% de incidentes de segurança em redes conectadas à *internet* em relação ao ano de 2013. Para isso, são criadas as mais diversas formas de invasões, bem como, *spams*, *malwares*, conteúdos maliciosos, vírus, *pishing*, *trojans*, *worms*, dentre outras (SECURELIST, 2013). A Figura 1 mostra o número de ataques à rede de computadores no mundo.

Figura 1: Ataques mundiais à redes de computadores em Setembro de 2016.



Fonte: SecureList²

No entanto, tais ferramentas definidas como sistema de Detecção de Intrusões de Rede (SDIR), não são suficientes para detectar e bloquear todas as intrusões, devido à alta complexidade e volatilidade destes pacotes, ocasionando falhas de reconhecimento e, consequentemente, diminuição da sensibilidade da detecção. Portanto, a problemática é buscar

¹ CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. Notícia disponível em: <http://www.cgi.br/noticia/releases/cert-br-registra-aumento-de-ataques-de-negacao-de-servico-em-2014>. Com data de acesso em 23 de Março de 2016.

² <https://securelist.com/statistics>. Com data de acesso em 03 de Outubro de 2016.

uma abordagem para a detecção de intrusões, minimizando a porcentagem de falso-positivos ou falso-negativos, acarretando em um aumento da acurácia da detecção.

1.2 Justificativa

Devido à sua melhor efetividade, a detecção de anomalias é amplamente utilizada em aplicações como: detecções de fraudes em cartões de crédito, detecção de intrusão em redes de computadores, pesquisas militares, dentre outras (CHANDOLA; BANERJEE; KUMAR, 2009).

Uma abordagem simplificada da detecção da anomalias é procurar por amostras que fogem ao comportamento definido como normal e posteriormente classificá-las como anômalas. Entretanto, existem fatores que dificultam essa detecção, tais como, a definição do que é o comportamento normal e a similaridade dos pacotes maliciosos aos normais, dificultando assim, a conceitualização computacional do que é uma anomalia.

Como visto anteriormente, o emprego de técnicas de inteligência artificial, tais como: redes neurais artificiais, reconhecimento de padrões e aprendizado de máquina, têm efetividade na área. A capacidade da Inteligência Artificial em aprender informações ocasiona um aumento na possibilidade de identificar anomalias desconhecidas.

Contudo, a dificuldade na utilização de técnicas inteligentes é a definição de quais são os melhores parâmetros, principalmente pela grande quantidade de informações (características) que os pacotes fornecem. Algumas dessas características são desnecessárias para a determinação do que é uma anomalia, podendo assim, serem descartadas durante a análise.

Motivado pela necessidade de métodos mais eficientes de detecção e prevenção de intrusões, este trabalho propõe um estudo de possíveis melhorias à área de segurança em redes de computadores. Visando aliar a utilização de técnicas atuais, como a Inteligência Artificial, propõe-se a classificação e otimização das características dos pacotes que trafegam em uma rede de computadores.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem por objetivo aplicar os conhecimentos obtidos durante o curso, com enfoque na área de segurança em redes de computadores e propor a utilização de técnicas inteligentes para verificar os resultados obtidos na detecção.

1.3.2 Objetivos Específicos

São enfrentados constantemente problemas de ataques virtuais, os quais ocasionam a perda ou roubo de informações pessoais. Logo, é necessário uma busca de métodos mais efetivos na área de Segurança em Redes de Computadores.

Portanto, este trabalho visa:

- a) conhecer novas técnicas que podem otimizar a Detecção de Intrusão em uma rede de computadores;
- b) criar uma nova base de dados para aplicação e validação do trabalho, além de disponibilizar uma nova fonte de dados para pesquisas posteriores;
- c) introduzir a otimização de um classificador no contexto de detecção de anomalias em redes de computadores;
- d) aprender e utilizar técnicas de classificação em problemas atuais e relevantes, tal como, Florestas de Caminhos Ótimos;
- e) aprender e utilizar técnicas meta-heurísticas para a otimização, tal como, a Otimização por Enxame de Partículas, a qual tem sido muito utilizada em pesquisas atuais;
- f) avaliar a eficácia das técnicas inteligentes estudadas.

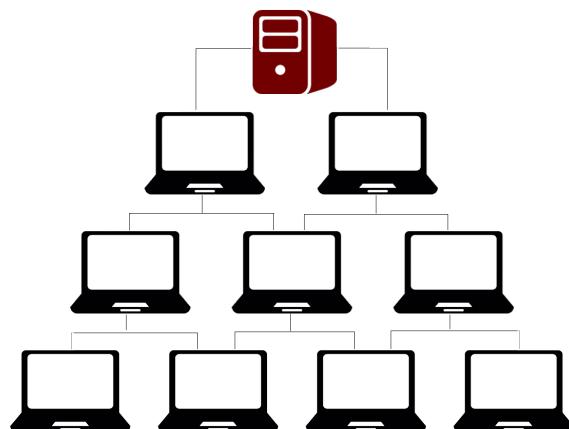
2 Fundamentação Teórica

Nas próximas seções será descrita a fundamentação teórica, a qual apresenta os conceitos abordados neste trabalho, tais como, fundamentos de redes de computadores, segurança em redes, anomalias, classificadores e otimizadores.

2.1 Redes de Computadores

Comumente, confunde-se *internet* com Rede de computadores, porém, apesar de estarem relacionados, os mesmos são distintos. De acordo com Tanenbaum e Wetherall (2011, p. 1), uma rede de computadores é "um conjunto de computadores autônomos interconectados por uma única tecnologia. Dois computadores estão interconectados quando podem trocar informações.", como ilustra a Figura 2. Já a *internet* é definida por Tanenbaum e Wetherall (2011, p. 33), como "um conjunto de redes diferentes que utilizam certos protocolos comuns e fornecem determinados serviços comuns.".

Figura 2: Rede de Computadores.



Fonte: Elaborada pela autora.

Atualmente, as redes de computadores podem ser classificadas de acordo com seu alcance, de tal forma em que os principais tipos são:

- PANs: Redes Pessoais (ou *Personal Area Networks*) que interligam periféricos para uso pessoal, geralmente sem fio, utilizando a tecnologia *Bluetooth*¹;
- LANs: Redes Locais (ou *Local Area Networks*) que interligam computadores de um mesmo espaço físico;

¹ *Bluetooth* é uma tecnologia que utiliza frequência de rádio de curto alcance.

- c) MANs: Redes Metropolitanas (ou *Metropolitan Area Networks*) são formadas por diversas redes locais compreendidas dentro de uma área com dezenas de quilômetros;
- d) WANs: Redes de Longa Distância (ou *Wide Area Networks*) que abrangem uma área maior que a MAN, como um país ou continente;
- e) *Internet*: sistema global de redes de computadores interligadas que utilizam protocolos comuns.

Contudo, nem sempre foi assim. O surgimento das redes de computadores é datado da década de 60, sendo mais simples do que hoje em dia.

2.1.1 História das Redes de Computadores

Em meados dos anos 60, pesquisadores de 4 universidades norte-americanas: Stanford Research Institute, Universidade da Califórnia, Universidade de Santa Barbara e Universidade de Utah, estudavam separadamente formas de comunicação entre *mainframes*² geograficamente isolados e de arquiteturas distintas. Assim, em 1969, surgia a ARPAnet, a primeira rede de computadores, formada por 4 nós: SRI, UCLA, UCSB e UTAH, representando as universidades citadas acima. Sua comunicação era feita através da comutação de pacotes³, ao invés da comutação de circuitos utilizada pela rede telefônica, tido como o maior meio de comunicação da época (FOROUZAN, 2009; LEINER et al., 2009; KUROSE; ROSS, 2010).

A rede tinha como objetivo inicial a defesa nacional, facilitando a comunicação militar e o estudo de novos ataques. Logo, a Agência de Projetos e Pesquisas Avançadas (*Advanced Research Projects Agency - ARPA*) do Departamento de Defesa dos Estados Unidos fomentou sua criação, dando nome à rede. Para uma rede estar ligada na ARPAnet, era necessário ter uma máquina (*host*) conectada a um processador de mensagens de interface (*Interface Message Processor - IMP*), de forma que os IMPs conseguissem se comunicar entre si, e cada IMP se comunicaria com o seu *host*. Nesse caso, o protocolo de controle de redes (*Network Control Protocol - NPC*) era o responsável pela comunicação.

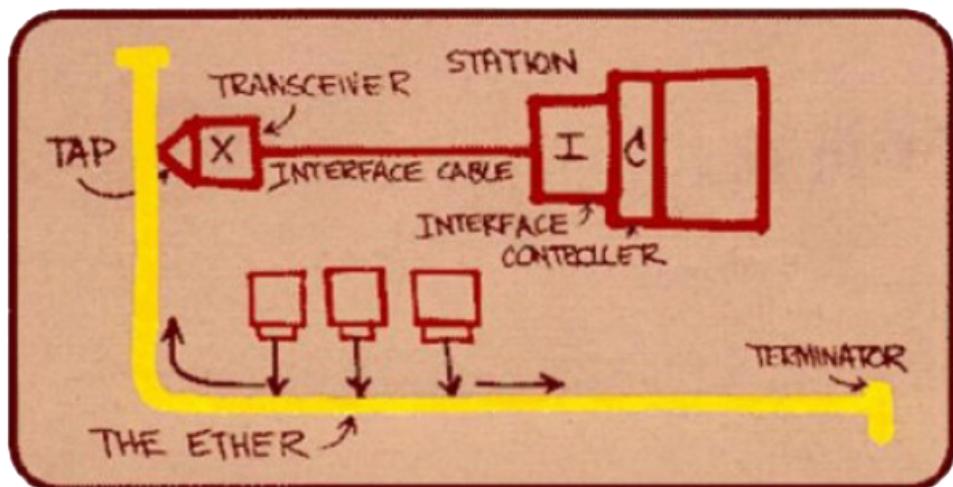
No início da década de 70, surgiu o protocolo de controle de transmissão (*Transmission Control Protocol - TCP*), o qual poderia atender às necessidades de um ambiente de rede de arquitetura aberta. Posteriormente, o TCP foi dividido em TCP e IP (Protocolo de Interligação de Redes - *Internetworking Protocol*). Sendo o primeiro responsável pela segmentação, remontagem e detecção de erros, e o segundo pelo envio (FOROUZAN, 2009). No final da década de 70, a ARPAnet possuía mais de 200 máquinas conectadas, interligando universidades, instituições militares e empresas. Contudo, a ARPAnet era de acesso restrito, e assim outras redes foram surgindo, tais como, a ALOHAnet (interligava universidades havaianas), TELEnet (rede

² Mainframes são computadores de grande porte dedicados ao processamento de um grande volume de informações.

³ Comutação de pacotes é baseada na teoria de filas para o tráfego em rajadas (pacotes em intervalos de tempo).

comercial), SNA (da IBM), dentre outras. O ALOHA foi o primeiro protocolo capaz de permitir múltiplos acessos de usuários geograficamente separados. Posteriormente, tal protocolo foi aprimorado, dando origem ao protocolo Ethernet, um método de transmissão em redes compartilhadas por fio (KUROSE; ROSS, 2010).

Figura 3: Concepção original da Ethernet.



Fonte: Kurose e Ross (2010)

A Figura 3 demonstra a ideia fundamental do protocolo Ethernet, no qual "Ether" fazia referência ao meio de transmissão de sinais. Originalmente, era utilizado um cabo coaxial, mas atualmente existem outras formas, tais como, a fibra óptica e até mesmo sem fio (*wireless*).

Com a crescente difusão da utilização das redes de computadores, a ARPAnet deixou de existir na década de 90, dividindo-se em Milnet (rede para uso doméstico e comercial) e a *Defense Data Network* (Rede de Dados de Defesa - destinada ao uso militar), sendo responsabilidade dos provedores de Internet a disponibilidade da mesma. A disponibilidade do acesso à rede fomentou sua utilização de tal forma que o mundo atual é amplamente conectado pela *internet*.

Ainda na década de 90, o desenvolvimento continuou crescendo, a fim de aprimorar a *internet*, Tim Berners-Lee criou o padrão *World Wide Web* (WWW), ou simplesmente Web, que é um sistema de disposição de documentos na Internet, através do formato hipertexto (WORLDWIDEWEB FOUNDATION, s.d.). Também surgiram os primeiros *browsers* (navegadores Web), provedores de *e-mail*, serviço mensagens instantâneas (ICQ). Berners-Lee desenvolveu as três tecnologias fundamentais para a Web atual:

- a) HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto): linguagem utilizada na construção de páginas na Web;
- b) URI (*Uniform Resource Identifier* - Identificador Uniforme de Recurso): endereço para identificação de recursos na Web, também conhecido como *URL* (*Uniform*

Resource Locator - Localizador Padrão de Recursos);

- c) HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto): protocolo de comunicação utilizado para sistemas de informação de hipermídia ou páginas Web.

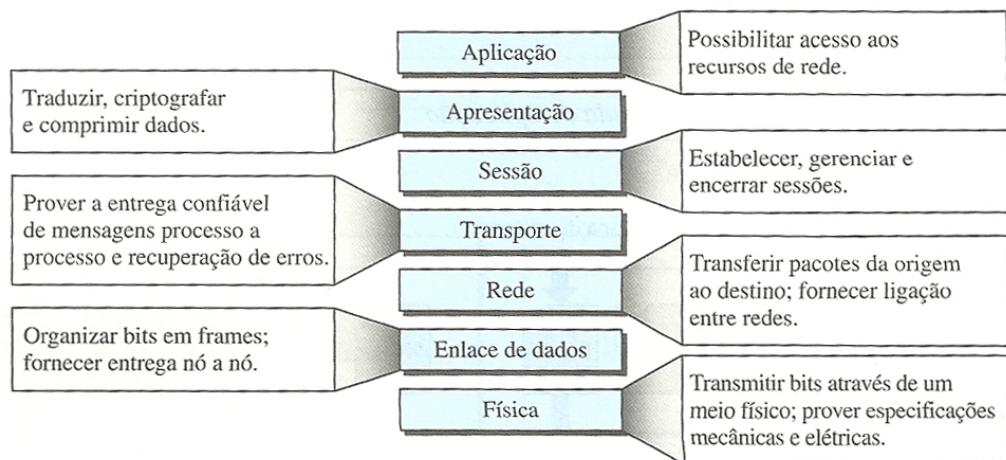
2.1.2 Protocolos de Redes

Com o intuito de reduzir a complexidade de implementação dos sistemas de rede, foi desenvolvido um padrão de hierarquia de protocolos, distribuídos entre camadas, modularizando o sistema. Cada camada utiliza o que é fornecido pela camada inferior e fornece serviços para camada subsequente (KUROSE; ROSS, 2010). No entanto, deve-se tomar cuidado para não confundir protocolo com serviço. Segundo Tanenbaum e Wetherall (2011, p. 25), serviço é um conjunto de operações primitivas e protocolo é um conjunto de regras que controla o formato e o significado dos pacotes.

Essa organização em camadas e protocolos é chamada de arquitetura de rede, e a implementação de serviços de rede é independente da mesma, ou seja, não importa como a camada ou protocolo foi implementada, devendo garantir o mesmo serviço. Tal independência da arquitetura facilita a criação de aplicações, já que não é preciso se preocupar com a arquitetura em si, mas apenas utilizar seus protocolos, mantendo oculto o que acontece nas camadas inferiores (FOROUZAN, 2009; KUROSE; ROSS, 2010; TANENBAUM; WETHERALL, 2011).

As camadas de protocolos de rede podem ser implementadas em todos os níveis de execução, como por exemplo, *software*, *hardware* ou híbrido. Atualmente existem duas arquiteturas principais: o modelo OSI e a TCP/IP.

Figura 4: Camadas Modelo OSI.



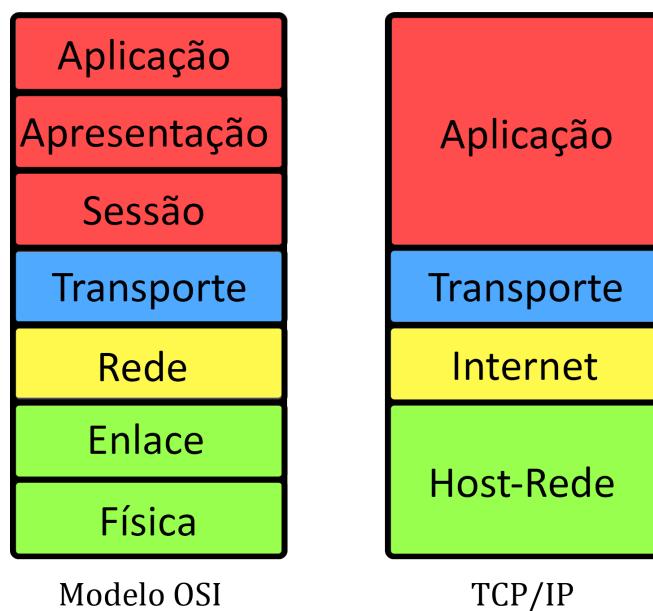
Fonte: Forouzan (2009)

O modelo OSI (*Open System Interconnection* - Interconexão de Sistemas Abertos) foi

desenvolvido pela ISO (*International Organization for Standardization - Organização Internacional para Padronização*) a fim de criar um padrão internacional de protocolos, facilitando a comunicação de dois sistemas diferentes sem alterações de *hardware* ou *software* (FOROUZAN, 2009). No entanto, o modelo OSI não especifica os protocolos em suas camadas, mas sim o que cada camada deve fazer, sendo utilizado como um modelo de referência. É composto por 7 camadas, sendo elas: física, de enlace, rede, transporte, sessão, apresentação e aplicação (TANENBAUM; WETHERALL, 2011), ilustradas pela Figura 4.

Já a arquitetura TCP/IP foi aprimorada a partir do período da ARPAnet, sendo que primeiramente criaram-se os protocolos e posteriormente foi criada a organização em camadas. Originalmente, é composta por 4 camadas: *host-rede*, *internet*, transporte e aplicação. A Figura 5 ilustra a diferença de camadas entre o modelo OSI e o TCP/IP.

Figura 5: Camadas Modelo OSI x TCP/IP.



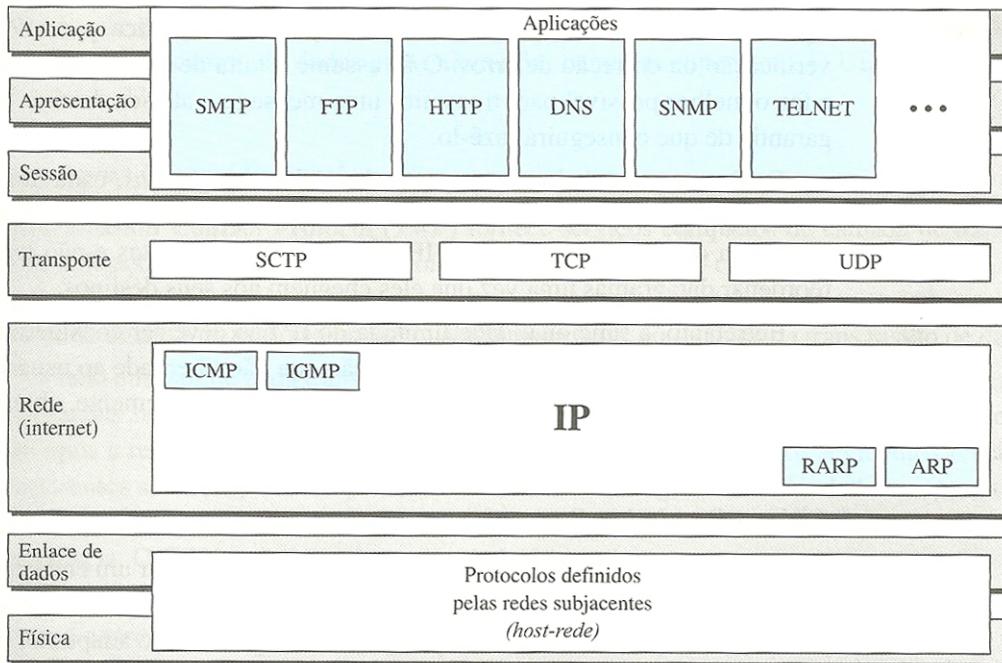
Fonte: Elaborada pela autora.

No entanto, o modelo OSI nunca foi implementado, sendo apenas um modelo teórico. Já a arquitetura TCP/IP tornou-se a arquitetura padrão em redes de computadores (FOROUZAN, 2009). Alguns dos protocolos, em suas respectivas camadas, são apresentados na Figura 6, a qual será melhor detalhada a seguir.

Camada de Aplicação

A camada mais externa é referente à camada de aplicação, e é onde reside todas as aplicações que utilizam a rede. Por ser a camada mais próxima do usuário, os protocolos são de maior nível e específicos ao tipo de aplicação desenvolvida.

Figura 6: Protocolos e Camadas TCP/IP em comparação ao Modelo OSI.



Fonte: Forouzan (2009)

O protocolo HTTP é responsável pela comunicação das páginas Web e é utilizado para requisição e transferência de arquivos. É implementado em dois programas: cliente e servidor. Assim, a troca de mensagens é realizada através da comunicação do protocolo presente no cliente e no servidor, onde o cliente faz a requisição de páginas *web* do servidor e o mesmo retorna as informações. Nessa comunicação é utilizado o protocolo TCP da camada de transporte devido à sua confiabilidade de entrega (KUROSE; ROSS, 2010).

No entanto, o HTTP é um protocolo sem estado⁴, ou seja, não armazena histórico de envios. Caso o cliente faça uma requisição duas vezes, o servidor retornará as duas solicitações (KUROSE; ROSS, 2010). São utilizados *cookies* para que as páginas web armazenem as informações dos usuários, os quais associam uma identificação ao usuário.

O protocolo FTP (*File Transfer Protocol* - Protocolo de Transferência de Arquivos) é o protocolo responsável pela transferência de arquivos entre dois sistemas distintos, geralmente um local e outro remoto. Para realizar a transferência, são utilizadas duas conexões: uma conexão de controle e uma conexão de dados. A primeira é responsável pela autenticação do usuário solicitante através de um login e senha. Após a autenticação, a conexão de dados realiza a transferência propriamente dita (KUROSE; ROSS, 2010). O FTP também utiliza um protocolo TCP da camada de transporte e, diferentemente do HTTP, é um protocolo com

⁴ Um protocolo sem estado considera requisições independentes, ou seja, não armazena informações entre uma requisição e outra. Já o protocolo com estado é o inverso, ou seja, há armazenamento de um estado interno no servidor para uma próxima requisição.

estado, pois armazena o usuário logado.

Uma das aplicações mais antigas ainda utilizadas é o correio eletrônico, o qual é responsável pelo envio de mensagens entre um remetente e destinatário. Sua ampla utilização deve-se ao fato da rapidez do envio de mensagens, como também do baixo custo.

O SMTP (*Simple Mail Transfer Protocol* - Protocolo de Transferência de Correio Simples) é o principal protocolo do sistema de correio eletrônico, o qual também é implementado em cliente/servidor. No entanto, por sua transmissão ser realizada no padrão ASCII de 7 bits, há uma transferência de dados limitada. Para o envio de mensagens multimídia, a mesma deve ser codificada no padrão ASCII. O protocolo TCP da camada de transporte também é utilizado devido a garantia de entrega sem falhas e até mesmo pela capacidade de recuperação de falhas (KUROSE; ROSS, 2010). Por ser um protocolo de envio de mensagens (*push*) é necessário um protocolo de acesso (*pull*) para a recuperação das mensagens.

O POP3 (*Post Office Protocol* - Protocolo dos Correios, versão 3) é um protocolo de acesso simples e limitado, sendo responsável pela conexão com o protocolo TCP. Seu funcionamento é realizado em três etapas, após a conexão com o TCP (FOROUZAN, 2009):

- a) autorização: realizada pela verificação do login/senha do usuário;
- b) transação: recuperação e marcação de mensagens a serem removidas;
- c) atualização: encerra a conexão TCP e apaga as mensagens marcadas.

O servidor POP3 tem duas opções de resposta para a requisição do cliente: +OK, quando a operação for realizada com sucesso e -ERR, quando ocorre um erro. Também há dois modos de operação: *1er-apagar*, onde não há leitura da mensagem de outra máquina, pois houve sua remoção do servidor. E *1er-armazenar*, no qual mesmo após serem lidas, as mensagens permanecem no servidor (KUROSE; ROSS, 2010; FOROUZAN, 2009). Devido à necessidade de marcação das mensagens, é um protocolo com estado. No entanto, o estado não permanece entre as sessões.

Já o protocolo IMAP (*Internet Message Access Protocol* - Protocolo de Acesso a Mensagem da Internet) é similar ao POP3, porém tem recursos adicionais, tais como:

- a) possibilidade de organização por pastas;
- b) pesquisa por conteúdo da mensagem (cadeia de caracteres);
- c) visualização do cabeçalho antes de fazer o *download* da mensagem;
- d) transferência parcial da mensagem caso a conexão não seja estável.

Por manter a organização das mensagens por pasta, ou seja, cada mensagem tem uma pasta associada, é um protocolo com estado, mantendo informações mesmo entre sessões.

O protocolo DNS (*Domain Name System* - Sistema de Domínio de Nomes) é o protocolo responsável por facilitar a denominação de endereços. Tal necessidade deve-se ao

fato de existirem duas formas possíveis para a representação de um *host* (KUROSE; ROSS, 2010):

- a) endereço IP: composto por 4 bytes (ou 32 bits), no qual cada byte é representado por um número entre 0 e 255, gerando um endereço do tipo 192.168.0.1, no qual cada byte é separado por . (ponto final). Este é o tipo de representação que o roteador interpreta;
- b) nome: representação expressa em linguagem natural (alfa-numérica), como por exemplo, www.unesp.br. Esta é a forma a qual os usuários possuem maior facilidade de compreensão.

No entanto, é necessário fazer a conversão de um sistema para outro, a fim de que o usuário e o roteador possam se comunicar de forma coerente. Para isso, foi criado o DNS, que é um banco de dados distribuído, permitindo a consulta no próprio banco (KUROSE; ROSS, 2010).

O protocolo Telnet (*Terminal Network* - Terminal de Rede) é uma aplicação que permite o acesso remoto. O FTP e o SMTP são protocolos de acesso remoto específicos, sendo necessário a criação de uma aplicação genérica. (FOROUZAN, 2009). O Telnet permite a conexão entre um terminal local e um sistema remoto, de tal forma em que o sistema local se comporte como um terminal do sistema remoto.

O SNMP (*Simple Network Management Protocol* - Protocolo Simples de Gerenciamento de Rede) é um *framework* de gerenciamento de dispositivos de rede composto por um gerente (servidor de gerenciamento), o qual controla os agentes (roteadores e *hosts*). É responsável pelo gerenciamento da rede independente das características físicas ou tecnológicas da mesma.

Camada de Transporte

A camada de transporte é responsável pela conexão de pares de *hosts*, para que a transferência dos dados seja eficiente e confiável (TANENBAUM; WETHERALL, 2011). Nesta camada, ocorre o controle de conexão, que determina o inicio da transmissão, podendo ser antes (não orientado à conexão) ou depois da mesma (orientado à conexão). Ocorre também a fragmentação dos pacotes, a fim de conseguir enviá-los pela rede.

Também é composta pelo protocolo TCP, um dos protocolos mais importantes para a comunicação de redes. É um protocolo orientado à conexões, que permite a entrega de um fluxo de bytes sem erros. O TCP do *host* de origem fragmenta as mensagens em segmentos e o TCP do destino remonta as mensagens, garantindo uma ordenação entre as mesmas (TANENBAUM; WETHERALL, 2011).

O protocolo UDP (*User Datagram Protocol* - Protocolo de Datagrama de Usuário) é um protocolo mais simples e não orientado à conexão, não garantindo a confiabilidade na entrega dos pacotes. Portanto, é utilizado para aplicações que não necessitem de uma sequência

de transmissão ou um controle de fluxo, ou seja, para quando a velocidade de entrega é o mais importante (TANENBAUM; WETHERALL, 2011).

O protocolo SCTP (*Stream Control Transmission Protocol* - Protocolo de Controle de Transmissão de Fluxo) é utilizado para aplicações mais recentes (voz), combinando as melhores funcionalidades do TCP e do UDP (FOROUZAN, 2009).

Camada de Internet

A camada de *internet* é responsável por integrar a arquitetura da rede, ou seja, deve garantir que os pacotes que trafegam na rede cheguem ao destino, independente da arquitetura utilizada (TANENBAUM; WETHERALL, 2011).

O protocolo IP é um protocolo não orientado à conexão e não confiável, objetivando interligar redes e enviar mensagens de forma eficiente (KUROSE; ROSS, 2010; TANENBAUM; WETHERALL, 2011). O IP divide os pacotes em datagramas, compostos por cabeçalho e dados. Existem duas versões para o protocolo IP.

O IP versão 4 ou IPv4 é o mais utilizado e seu endereço é composto por 32 bits divididos em 4 bytes separados por . (ponto final) e representados por um número decimal de 0 à 255. Seu endereço contém a parte de rede (fixa) e a parte de *host* (variável). Os tamanhos da parte de rede e de *host* são determinados pela máscara de rede, que também é um endereço de 32 bits divididos em 4 bytes, representadas por um número decimal de 0 à 255, sendo 255 a parte invariável. Por exemplo, o endereço 192.168.0.1 com máscara 255.255.255.0 tem como parte de rede 192.168.0 e parte de *host* 1, criando assim uma sub-rede onde o último byte identifica cada *host* e varia de 0 à 255.

O IP versão 6 ou IPv6 foi criado devido à necessidade de novos endereços, pois os endereços IPv4 estão se esgotando. Assim, algumas mudanças foram feitas no cabeçalho, tais como, a remoção e adição de campos e a forma de endereçamento. No protocolo IPv6, o endereço é composto por 128 bits, com 8 grupos no sistema hexadecimal separados por : (dois pontos), como por exemplo: 21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A ou simplificadamente 21DA:D3::2F3B:2AA:FF:FE28:9C5A. Esse aumento do endereço possibilita um número maior de endereços disponíveis. No entanto, o IPv6 é de difícil implementação, sendo pouco utilizado. Segundo Tanenbaum e Wetherall (2011, p. 285), apenas 1% da *internet* utiliza essa versão. Protocolos adicionais como ICMP, ARP e RARP foram modificados, excluídos ou adicionados à nova versão (FOROUZAN, 2009).

O protocolo ICMP (*Internet Control Message Protocol* - Protocolo de Mensagens de Controle de Internet) foi criado devido à falta de informações referentes às falhas do protocolo IP, sendo assim, responsável por relatar, principalmente, erros de comunicação com o roteador (TANENBAUM; WETHERALL, 2011). Algumas das mensagens enviadas são apresentadas na Tabela 1. Geralmente, o ICMP é considerado como parte do protocolo IP,

contudo, está logo acima, na visão hierárquica, do protocolo IP, pois sua mensagem é enviada dentro do datagrama IP (KUROSE; ROSS, 2010).

Tabela 1: Principais tipos de mensagens ICMP.

Tipo de Mensagem	Descrição
<i>Destination Unreachable</i>	O pacote não pode ser entregue
<i>Time Exceeded</i>	O campo TTL ⁵ atingiu 0
<i>Parameter problem</i>	Campo de cabeçalho inválido
<i>Source quench</i>	Restringe o envio de pacotes
<i>Redirect</i>	Ensina uma rota a um roteador
<i>Echo e echo reply</i>	Verificam se uma máquina está ativa
<i>Timestamp request/reply</i>	O mesmo que Echo, mas com registro de tempo
<i>Router advertisement/solicitation</i>	Encontra um roteador

Fonte: Tanenbaum e Wetherall (2011).

O protocolo ARP (*Address Resolution Protocol* - Protocolo de Resolução de Endereço) é responsável pela associação de endereços lógicos e físicos (FOROUZAN, 2009). Realiza a conversão de endereços IP (camada de rede) em endereços MAC (camada de enlace). Em uma mesma rede local, para se enviar dados para um *host* destino com segurança, é necessário além de seu endereço IP o seu endereço MAC, que é próprio da máquina. Para se obter esse endereço, o protocolo ARP no *host* de origem envia o endereço IP do destino do pacote a ser transferido e o *host* com aquele endereço retorna seu endereço MAC (KUROSE; ROSS, 2010). Para redes maiores, a solicitação entre protocolos ARP é realizada entre *host* de origem e roteador e, posteriormente, entre o roteador e o *host* de destino.

O protocolo RARP (*Reverse Address Resolution Protocol* - Protocolo de Resolução Reversa de Endereço) tem função inversa ao protocolo ARP, ou seja, transforma um endereço físico em endereço lógico. Essa conversão às vezes é necessária em *hosts* novos ou quando não é possível obter a informação do disco.

O RARP manda uma requisição para a rede local e o *host* que souber todos os endereços IPs da rede irá responder. No entanto, para essa operação, o *host* de origem deve rodar o programa RARP *Client* e todas os *hosts* da rede ou sub-rede devem rodar o RARP *Server*, logo, este protocolo encontra-se obsoleto.

Camada de *host-rede*

A camada de *host-rede* é onde ocorre a interligação das redes com serviços não orientados à conexão, já que inicialmente o objetivo das redes era para fins militares, os dados não poderiam ser perdidos caso ocorresse algum ataque (TANENBAUM; WETHERALL, 2011). Nesta camada não há um protocolo específico.

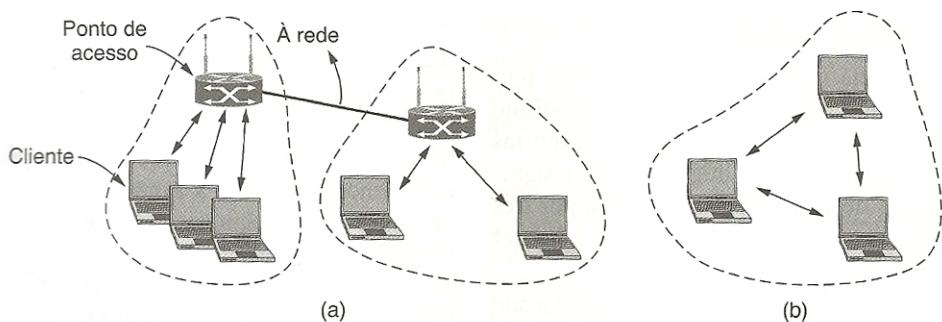
⁵ TTL (Time To Live) é o número máximo de pulos entre *hosts* ou dispositivos que um pacote pode dar antes de chegar ao seu destino.

Esta camada é referente às camadas físicas e de enlace do modelo OSI, portanto, abrange desde os meios de transmissão à comunicação de enlaces. De acordo com Kurose e Ross (2010, p. 318-319) enlaces são "os canais de comunicação que conectam nós adjacentes ao longo dos caminhos de comunicação". Assim, a camada é uma interface de conexão entre host e enlace (TANENBAUM; WETHERALL, 2011), ou seja, a comunicação de um enlace é implementada em uma placa de interface de rede (*Network Interface Controller - NIC*).

A transmissão dos dados é feita pela movimentação individual dos bits (KUROSE; ROSS, 2010), via os seguintes meios de transmissões: através de cabos como par trançado, coaxial e de fibra óptica, ou através de micro-ondas como a rede *Wireless* (FOROUZAN, 2009). Os meios de transmissão via cabo (guiados) utilizam a tecnologia Ethernet (LAN com fio) ou padrão IEEE 802.3 (TANENBAUM; WETHERALL, 2011). Atualmente há dois tipos de Ethernet, embora ambos sigam os princípios da Ethernet desenvolvida na década de 70.

A Ethernet clássica, que é a tecnologia desenvolvida em 1970 e com velocidade de 3 Mbps, e a Ethernet comutada que utiliza *switches*, que é um dispositivo para redistribuição dos pacotes, sendo a tecnologia utilizada atualmente (TANENBAUM; WETHERALL, 2011). No entanto, a maior mudança da Ethernet original para a atual é a velocidade, tendo: Ethernet-padrão (10 Mbps), *Fast Ethernet* (100 Mbps), *Gigabit Ethernet* (1 Gbps) e 10 *Gigabit Ethernet* (10 Gbps) (TANENBAUM; WETHERALL, 2011; KUROSE; ROSS, 2010; FOROUZAN, 2009).

Figura 7: Arquitetura LAN sem fio.



Fonte: Tanenbaum e Wetherall (2011). (a) Arquitetura com PAs e conexão à rede; (b) Arquitetura sem conexão à rede (*ad-hoc*).

Já os meios de transmissão *wireless* (não guiados) utilizam a tecnologia IEEE 802.11 ou LAN sem fio. É implementada na camada de rede e de enlace e possuem dois tipos de arquitetura: o conjunto básico de serviço (*Basic Service Set - BSS*), composto por estações sem fio e uma estação central ou ponto de acesso (PA) e o conjunto estendido de serviço (*Extended Service Set - ESS*), composto por dois ou mais BSS com PAs, onde BSSs são conectados por um sistema de distribuição, geralmente uma LAN com fio (FOROUZAN, 2009). Cada PA se conecta a um roteador que permite o acesso à *internet* (Figura 7.a.). Um BSS sem PA é denominada arquitetura *ad-hoc*, é isolada e não tem comunicação com outra BSS.

(Figura 7.b.) (FOROUZAN, 2009).

Outra tecnologia de LAN sem fio é o Bluetooth ou padrão IEEE 802.15.1 (KUROSE; ROSS, 2010), que através de ondas de rádio de curto alcance é capaz de conectar dispositivos com funcionalidades distintas. Por não ter conexão com a Internet, é uma rede *ad-hoc* (Figura 7.b.). É a tecnologia utilizada nas WPANs (redes de área pessoal sem fio).

A Tabela 2 apresenta algumas aplicações popularmente utilizadas e os protocolos utilizados nas camadas de aplicação e transporte, que são os protocolos mais variáveis, dado que dependem do tipo de aplicação.

Tabela 2: Aplicações e protocolos utilizados.

Aplicações	Protocolo (Cam. Aplicação)	Protocolo (Cam. Transporte)
Correio Eletrônico	SMTP	TCP
Acesso a terminal remoto	Telnet	TCP
Web	HTTP	TCP
Transferência de arquivos	FTP	TCP
Telefonia por Internet (exemplo Skype)	SIP, RTP ou proprietário	UDP

Fonte: Adaptada de Kurose e Ross (2010).

Com a arquitetura de hierarquia de protocolos, criou-se a ideia de encapsulamento. O encapsulamento é dado de acordo com a movimentação dos pacotes, que devem passar da camada superior até à de mais baixo nível, para ser transferida pelo meio de comunicação. Um protocolo da camada superior é encapsulado por um protocolo da camada inferior, composto por cabeçalho e dados, gerando um encapsulamento de protocolos (ou pilha de protocolos) até a camada mais inferior. O cabeçalho de um protocolo é responsável por armazenar informações específicas do mesmo, como endereço de origem e destino, tamanho, dentre outros, fazendo com que seja possível distingui-lo dentre os protocolos existentes. Já a parte de dados é onde são armazenadas as informações a serem utilizadas pela camada superior (FOROUZAN, 2009).

2.2 Segurança de Redes

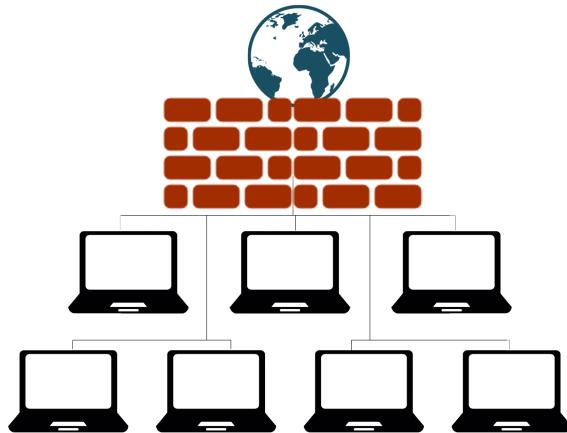
No início, as redes de computadores eram utilizadas, principalmente, por pesquisadores que tinham por objetivo compartilhar suas pesquisas a fim de fomentá-las. No entanto, a *internet* tornou-se ferramenta essencial do cotidiano, realizando diversas tarefas, tais como, transações bancárias, troca de mensagens entre usuários, negociações comerciais. Com isso, a segurança passou a ser fundamental na utilização da rede (TANENBAUM; WETHERALL, 2011).

A segurança de computadores foi definida por Stallings (2008, p. 3) como "o conjunto de ferramentas projetadas para proteger dados e impedir *hackers*⁶.", e a partir do momento em

⁶ *Hacker* é a denominação do indivíduo que pratica a quebra de um sistema sem fim lucrativos, apenas para demonstrar vulnerabilidades.

que os sistemas utilizam redes, a mesma pode ser chamada de segurança de redes ou segurança de inter-rede, conceito ilustrado pela Figura 8.

Figura 8: Segurança em Rede de Computadores.



Fonte: Elaborada pela autora.

A necessidade de proteção deve-se ao intenso volume de dados que trafega na rede e à possibilidade de serem acessados por pessoas mal-intencionadas, que buscam algum benefício próprio ou que procuram prejudicar alguém através do roubo ou modificação de informações (KUROSE; ROSS, 2010).

Em uma tentativa de suprir as vulnerabilidades das redes de computadores, foram definido alguns princípios que devem ser garantidos a fim de se obter uma utilização segura (STALLINGS; BROWN, 2014; FOROUZAN, 2009; KUROSE; ROSS, 2010; TANENBAUM; WETHERALL, 2011):

- confidencialidade ou sigilo: é a garantia de que apenas os participantes da comunicação tenham acesso às informações enviadas, sem a possibilidade de acesso não autorizado por terceiros;
- integridade: é a garantia de que as mensagens enviadas não sejam alteradas durante a transmissão, ou por interceptação de terceiros ou por falhas técnicas;
- autenticação: é a garantia de que os participantes sejam verdadeiros para determinada conexão.

Alguns autores como Stallings e Brown (2014), Tanenbaum e Wetherall (2011), ainda definem o princípio de não-repúdio ou determinação de responsabilidade, que é a garantia de autoria de uma operação, sem a possibilidade de negação ou isenção de responsabilidade pela mesma.

2.2.1 Ferramentas de Segurança

Para que esses princípios sejam garantidos na segurança de redes e impeçam ações não autorizadas, foram desenvolvidas ferramentas que auxiliassem na implementação da segurança em redes.

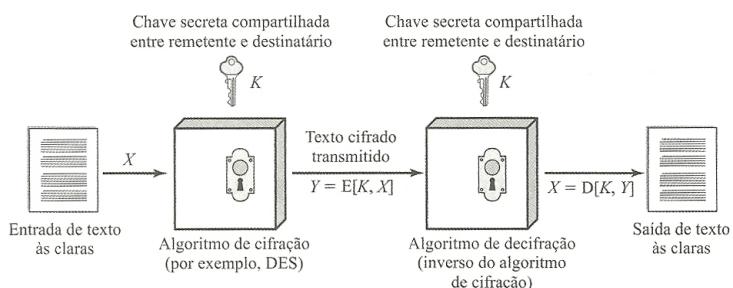
Criptografia

Uma das principais ferramentas utilizada na segurança de redes é a criptografia, utilizada desde o Império Romano de Júlio César. Do grego *kryptós* (escondido) e *gráphein* (escrita), criptografia é a capacidade de cifrar (esconder) uma mensagem, de tal forma que apenas o remetente e destinatário sejam capazes de compreendê-la. Atualmente, são utilizados algoritmos de criptografia de acesso público, sendo que a chave utilizada é o que torna a mensagem secreta (KUROSE; ROSS, 2010). A chave é passada como parâmetro para o algoritmo, podendo ser uma cadeia numérica ou de caracteres.

Assim, a criptografia de uma mensagem m com uma chave K pode ser representada por $K(m)$, e a decriptografia por $K_d(K_c(m)) = m$, sendo K_c a chave de cifração e K_d a de decifração. Existem dois tipos de chaves: simétrica e pública. Na criptografia com chave simétrica K_c e K_d são iguais e secretas, e o sistema é composto por 5 elementos (STALLINGS; BROWN, 2014):

- texto aberto;
- algoritmo de criptografia (encriptação);
- chave secreta;
- texto cifrado;
- algoritmo de decriptografia (decriptação).

Figura 9: Criptografia com chave simétrica.



Fonte: Stallings e Brown (2014)

A criptografia com chave simétrica, ilustrada pela Figura 9, pode ser feita de duas formas: em blocos ou em fluxo. Em blocos, a mensagem é dividida em blocos de tamanhos

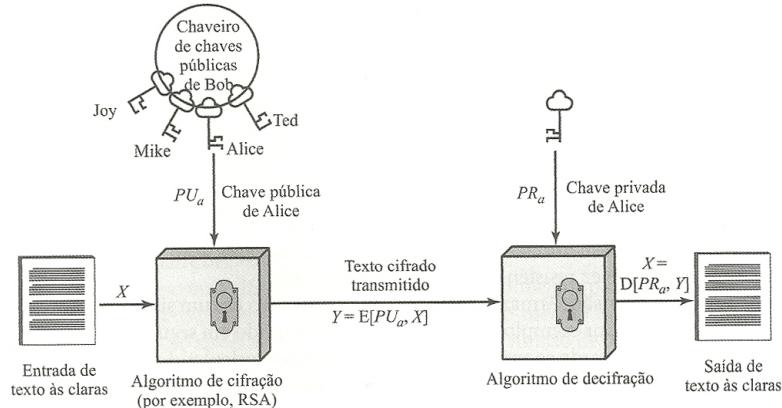
fixos e cada bloco passa por um processo de criptografia. Em fluxo, a criptografia é feita de forma contínua.

Os algoritmos mais utilizados de chave simétrica são: DES (*Data Encryption Standard*) e AES (*Advanced Encryption Standard*). O DES utiliza blocos de 64 bits e chaves com 56 bits, o que possibilita 2^{56} chaves diferentes. O AES é mais recente e surgiu da necessidade de chaves maiores, então utiliza blocos de 128 bits e chaves de 128, 192 ou 256 bits, possibilitando um número muito maior de combinações.

A criptografia com chave pública foi desenvolvida a partir da dificuldade de utilização de uma única chave conhecida, devido à interconectividade proporcionada pelas redes de computadores. Logo, ela utiliza duas chaves distintas (uma pública e outra secreta) e o sistema é composto por 6 elementos (STALLINGS; BROWN, 2014):

- a) texto aberto;
- b) algoritmo de encriptação;
- c) chave secreta;
- d) chave pública;
- e) texto cifrado;
- f) algoritmo de decriptação.

Figura 10: Criptografia com chave pública.



Fonte: Stallings e Brown (2014).

Como demonstra a Figura 10, o processo de cifração é realizado com a chave pública e o de decifração com a chave secreta, a qual apenas o indivíduo sabe. Um dos mais famosos algoritmos de chave pública é o RSA (do acrônimo de Ronald Rivest, Adi Shamir e Leonard Adleman, seus desenvolvedores) e utiliza a cifragem em blocos, gerando números inteiros de 0 a $n - 1$ para um dado n . As chaves inicialmente eram de 428 bits, mas atualmente são utilizados 1024, 2048 ou 3072 bits.

Existem duas formas de ataques à criptografia: por criptoanálise, no qual se exploram as características do algoritmo a fim de tentar descobrir a chave; e por força bruta, no qual todas as chaves possíveis são testadas. No entanto, a segurança de qualquer sistema criptográfico depende do comprimento da chave e do esforço computacional utilizado (STALLINGS; BROWN, 2014). Para garantir que o remetente e o destinatário sejam quem realmente dizem ser, e impedir que um terceiro seja capaz de mandar uma mensagem criptografada fingindo ser o remetente, é utilizado uma assinatura digital como forma de autenticação (STALLINGS; BROWN, 2014).

Assinatura Digital

A assinatura digital é uma das formas de garantir a autenticidade do usuário, já que a mesma deve ser própria. É baseada em criptografia, ou seja, um dado é criptografado tal que apenas o remetente da mensagem poderia ter feito (STALLINGS; BROWN, 2014).

A mensagem é criptografada utilizando uma chave secreta e, posteriormente, o destinatário utiliza a chave pública referente ao remetente para decifrar a mensagem, tendo como garantia sua procedência. Outro método de gerar uma assinatura digital é através de funções *hash* seguras, que são calculadas gerando um valor atribuído a mensagem. Posteriormente, este valor é criptografado com a chave secreta do remetente, e o destinatário testa sua autenticidade através da decriptação da chave, verificando o valor *hash* referente a mensagem.

Autenticação de Usuário

A autenticação de usuário é fundamental para a segurança e é definida na RFC⁷ 2828 como "o processo de verificação de uma identidade alegada por uma ou para uma entidade de sistema." (STALLINGS; BROWN, 2014).

O processo de autenticação é realizado em duas etapas: identificação e verificação. Na identificação é apresentado um identificador e a etapa de verificação é responsável por gerar alguma informação que comprove o vínculo entre a entidade e o identificador.

Os meios de autenticação são classificados de acordo com alguma característica do indivíduo, podendo ser (STALLINGS; BROWN, 2014):

- a) indivíduo sabe ou conhece: autenticação é dada a partir de alguma informação que o usuário saiba, por exemplo, senhas;
- b) indivíduo possui: autenticação é dada a partir de alguma informação que o usuário possua, por exemplo, um cartão de acesso;
- c) indivíduo é: autenticação é dada a partir de alguma informação inerente ao usuário, por exemplo biometria (impressão digital, facial, dentre outros);

⁷ Request for Comments são documentos técnicos desenvolvidos pelo IETF (*Internet Engineering Task Force*), que especificam os padrões utilizados na Internet.

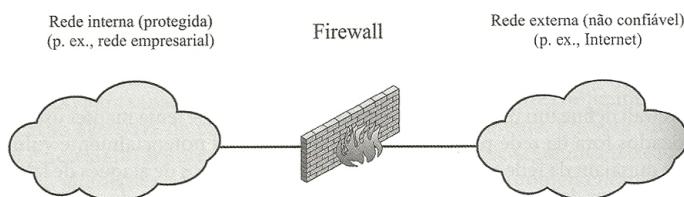
- d) indivíduo faz: autenticação é dada a partir de alguma informação que o usuário realize, por exemplo, voz ou características da escrita.

A autenticação baseada em conhecimento é realizada, geralmente, através da utilização de senhas. Essas senhas são atreladas à um usuário (ou *login*). Caso a senha fornecida seja compatível com a senha armazenada e associada ao ID, o acesso é permitido. No entanto o uso de senhas é considerado vulnerável pela possibilidade de adivinhação, ainda mais para senhas simples ou comuns. Já a autenticação baseada em posse utiliza *token* ou cartões, que permitem o acesso através de informações armazenadas. Uma forma de aumentar a segurança é a combinação com senha. Assim, mesmo em caso de perda ou roubo, o acesso não será permitido. A autenticação baseada no indivíduo é um sistema complexo e caro, pois é necessário verificar características muito particulares, tais como, a impressão digital, características da face ou ocular. A impressão digital é a forma de autenticação mais simples de verificação. Uma forma de autenticação baseada no que o indivíduo faz é a assinatura, pois em teoria, as características de escrita são particulares ao indivíduo.

Firewalls

Firewall é uma ferramenta que auxilia na segurança baseada em rede e que pode ser considerada uma camada de defesa adicional, isolando sistemas internos de sistemas externos (STALLINGS; BROWN, 2014), conceito ilustrado pela Figura 11.

Figura 11: *Firewall*.



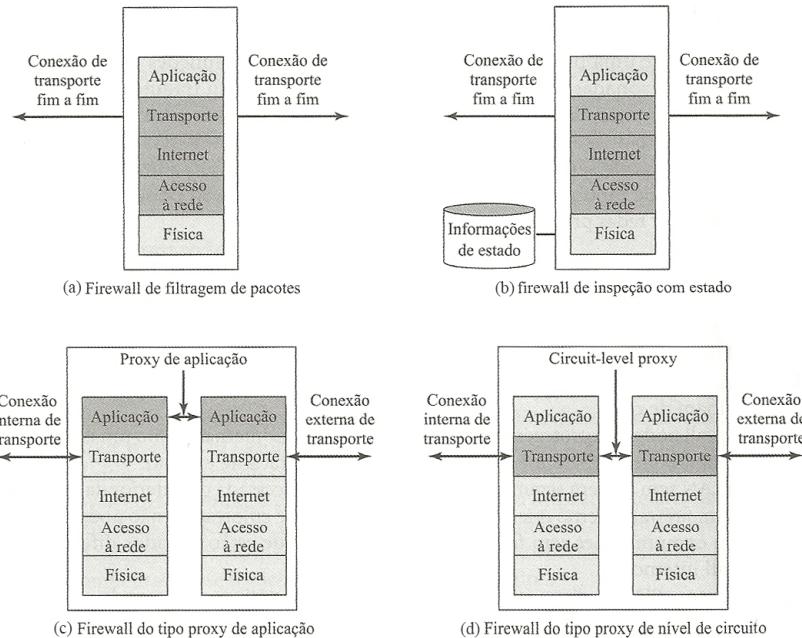
Fonte: Stallings e Brown (2014)

Todo o tráfego de origem interna para destino externo ou vice-versa deve passar pelo *firewall*, que é o responsável por autorizar ou não sua passagem.

A Figura 12 apresenta os tipos de *firewalls*. O tipo de filtragem de pacotes analisa os pacotes IPs recebidos da rede e segue um conjunto de regras para determinar quem será repassado ou descartado (Figura 12.a.). O *firewall* com inspeção de estados aprimora o *firewall* de pacotes com regras mais rígidas, criando um diretório de conexões TCP de saída, onde cada conexão em andamento tem uma entrada no diretório para que o *firewall* possa redirecionar os pacotes de acordo com o perfil de cada entrada (Figura 12.b.).

O *firewall* de *gateway* ou *proxy* de aplicação cria uma conexão em nível de aplicação, retransmitindo pacotes após a autenticação do usuário (Figura 12.c.). O *firewall* de *gateway* ou

Figura 12: Tipo de Firewalls.



Fonte: Stallings e Brown (2014).

proxy a nível de circuito estabelece duas conexões em nível de transporte: interno → *gateway* e *gateway* → externo, retransmitindo pacotes sem análise de conteúdo (Figura 12.d.).

Detecção de Intrusão

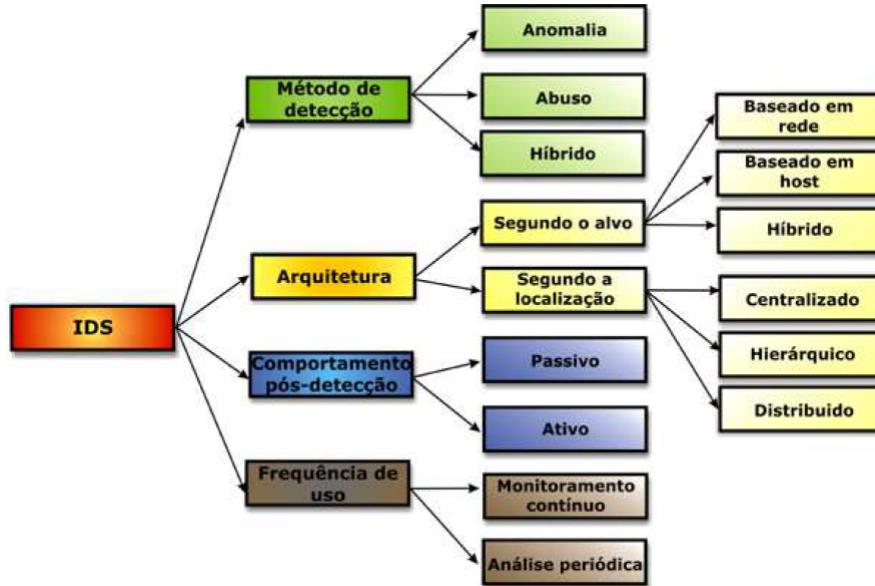
A detecção de intrusão foi definida na RFC 2828, conforme apresentado no livro de Stallings e Brown (2014, p. 235)

"Um serviço de segurança que monitora e analisa eventos de sistema com a finalidade de descobrir e avisar em tempo real ou quase real que estão ocorrendo tentativas de acesso a recursos de sistemas de modo não autorizado".

Os Sistemas de Detecção de Intrusões (SDIs) podem ser classificados a partir de características do alvo e método de detecção. Considerando os métodos de detecção mostrados na Figura 13, os SDIs mais importantes são: SDI baseado em assinatura (ou abuso) e SDI baseado em anomalia (PERLIN; NUNES; KOZAKEVICIUS, 2011).

O SDI baseado em assinatura utiliza um modelo para reconhecimento de intrusões, ou seja, o pacote analisado é comparado à uma base de dados e se sua assinatura é identificada, o pacote é então rotulado como intrusão. Devido à necessidade de um modelo, o SDI baseado em assinatura não é eficiente na análise de pacotes desconhecidos, entretanto, é de fácil implementação (MUDZINGWA; AGRAWAL, 2012).

Figura 13: Classificação dos SDIs.



Fonte: Campello e Weber (2001)

Já o SDI baseado em anomalia, não utiliza modelos de reconhecimento, porém, cria um perfil da rede utilizando as características dos pacotes que trafegam na mesma. Assim, os SDIs baseados em anomalias têm maior eficácia para detectar pacotes desconhecidos, entretanto, as anomalias podem ser definidas como falso-positivas ou falso-negativas (MUDZINGWA; AGRAWAL, 2012).

2.3 Anomalias

Uma anomalia pode ser definida como algo raro que difere de um comportamento definido como normal, ou seja, fora do padrão. Contudo, uma anomalia nem sempre é um ataque ou algo malicioso, mas sim uma informação que deve ser analisada com maior atenção.

As anomalias podem ser classificadas em dois grupos: falhas de rede e ataques à segurança. (LöF; NELSON, 2010). Independente do tipo, uma anomalia pode causar algum dano à rede, seja um congestionamento ou um roubo de informações. Anomalias de falhas de rede não são maliciosas e podem ocorrer devido à problemas físicos ou técnicos, tais como, queda de energia, erros de configuração, dentre outros. Uma anomalia de ataque à segurança é maliciosa e visa roubar informações ou causar algum dano.

Algumas anomalias de falhas de rede são:

- flash crowd*: Ocorre quando um grande volume de requisição de clientes passa pela rede de maneira desordenada podendo causar um congestionamento (JUNG; KRISHNAMURTHY; RABINOVICH, 2002);

- b) *babbling node*: Ocorre quando um nó da rede falha e envia pacotes aleatoriamente para vários pontos da rede (AL-KASASSBEH; ADDA, 2009);
- c) *bugs* de roteadores: Ocorre quando um roteador falha devido ao recebimento de um pacote com problemas.

Já as de ataque à segurança são as mais conhecidas e temidas pelos usuários, sendo algumas dos tipos:

- a) ataque de negação de serviço (*Denial of Service - DoS*): método de fatigar o sistema através do envio de diversas requisições. Um ataque distribuído de negação de serviço (*Distributed Denial of Service - DDoS*) utiliza uma máquina ("mestre") que domina um conjunto de máquinas ("escravas") a fim de utilizá-las para um ataque DoS (COLE, 2009);
- b) vermes (*worms*): programas capazes de se multiplicar na rede, infectando um grande número de computadores. Posteriormente, as cópias executam o ataque, criando um ataque em massa. (ELLIS, 2003);
- c) escaneamento de portas (*Port Scan*): o escaneamento de portas é realizado a fim de encontrar vulnerabilidades que possam ser utilizadas como portas de ataques subsequentes (COLE, 2009);
- d) *IP Spoofing*: é usado para convencer o sistema que a comunicação é verdadeira, permitindo acesso ao invasor. Para isso, é feita uma alteração no pacote TCP para simular um endereço de IP válido (COLE, 2009);
- e) *back door*: utilizado para garantir acesso remoto ao sistema ou à rede infectada, explorando falhas e conexões externas assíncronas para abrir portas do roteador (COLE, 2009);
- f) *dumpster diving*: envolve a aquisição de informações que são descartadas sem cuidados e que podem ser valiosas para um *cracker*⁸ (COLE, 2009).

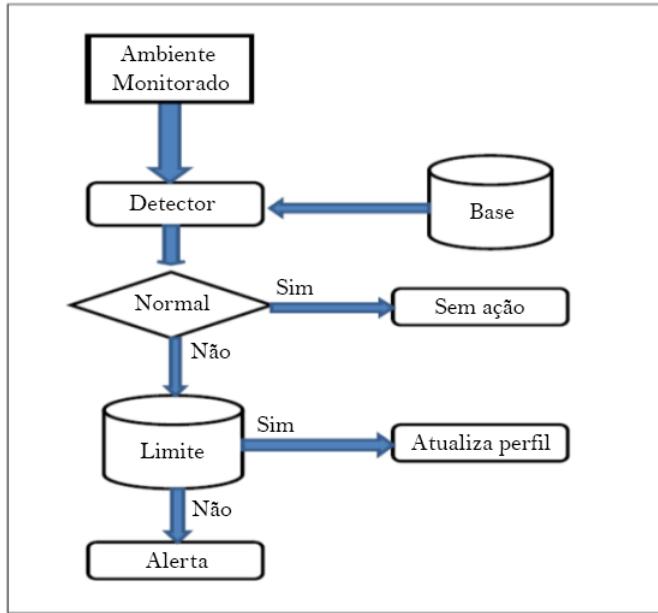
2.3.1 Detecção de Anomalias

Neste trabalho, será abordada a detecção de intrusão baseada em anomalias, devido à sua melhor efetividade de detecção de novos ataques. Esta abordagem segue o princípio de anomalia, ou seja, o que for diferente do comportamento normal é considerado anômalo.

Existem dois tipos de perfis para a detecção por anomalias (CHEBROLU; ABRAHAM; THOMAS, 2005): estático, a qual utiliza um perfil único e fixo, ou seja, no decorrer da análise o comportamento monitorado não é alterado; e dinâmico, na qual padrões são extraídos dos comportamentos habituais da interface de análise, alterando o perfil quando necessário.

⁸ *Cracker* é a denominação do indivíduo que pratica a quebra de um sistema de segurança de forma ilegal para fins lucrativos.

Figura 14: Arquitetura da metodologia de Detecção de Anomalia.



Fonte: Mudzingwa e Agrawal (2012). Traduzida e adaptada pela autora.

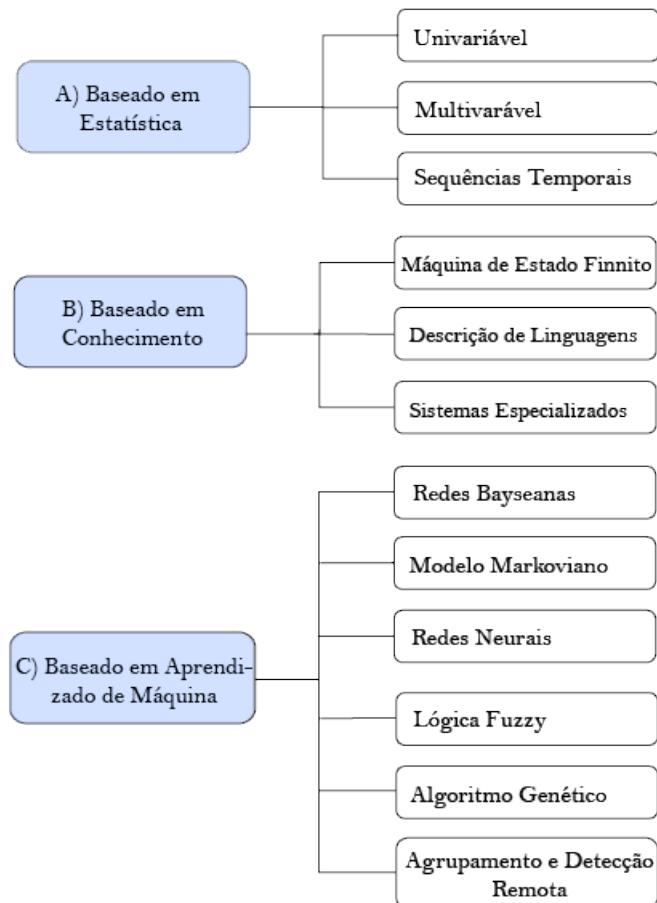
O funcionamento geral da detecção de anomalias pode ser simplificado como apresentado na Figura 14, na qual o ambiente é monitorado e as informações são enviadas a um detector que as compara com o perfil criado. Caso tenha algum traço anômalo e divirja do limite estipulado, um alerta é emitido, caso o limite não seja ultrapassado, o perfil deve ser atualizado. Caso contrário, nenhuma ação adicional é necessária e o pacote é definido como normal.

A detecção de anomalias em redes de computadores é uma área que tem sido amplamente pesquisada, utilizando-se das mais diversas técnicas de detecção. A grande variedade de técnicas estudadas dificulta a classificação das mesmas ao longo do processo de detecção. Contudo, García-Teodoro et al. (2009, p. 20) as classificou de acordo com a natureza do processamento envolvido: Análise Estatística, Conhecimento e Aprendizado de Máquina, conforme mostra a Figura 15.

Nos métodos estatísticos, o sistema observa a atividade das amostras e cria um perfil que represente um comportamento estocástico. Este perfil é baseado em métricas como, por exemplo, taxa de tráfego, número de pacotes por protocolo, utilização de processamento da máquina, dentre outras. Usualmente, dois perfis são criados: o perfil atual e o armazenado. No decorrer da atividade da rede, o sistema atualiza o perfil atual e calcula periodicamente uma pontuação de anomalia através da comparação do perfil normal e do armazenado, utilizando uma função de anormalidade com todas as métricas do perfil. Caso a pontuação seja muito alta em relação a um limite estabelecido, o sistema gera um alerta de anomalia (GARCÍA-TEODORO et al., 2009; PATCHA; PARK, 2007).

Haystack é um dos exemplos de detecção de anomalia baseada em análise estatística.

Figura 15: Classificação das técnicas de detecção.



Fonte: García-Teodoro et al. (2009). Traduzida e adaptada pela autora.

Smaha (1988) criou o protótipo modelando o sistema com parâmetros independentes através de variáveis gaussianas aleatórias. Também definiu um intervalo de valores que poderiam ser considerados normais para cada característica. Se durante a análise uma característica fugisse desse valores, uma pontuação era atribuída a esta amostra. Uma distribuição de probabilidade era calculada, e se a pontuação fosse muito alta, um alerta de intrusão era acionado.

Um dos primeiros sistemas de detecção de intrusão foi desenvolvido no Instituto de Pesquisas de Stanford (SRI) e foi chamado de *Intrusion Detection Expert System* (IDES). O IDES monitorava constantemente o comportamento do usuário e detectar eventos suspeitos (LUNT; JAGANNATHAN, 1988). Posteriormente, cientistas do SRI aprimoraram o sistema criando o *Next-Generation Intrusion Detection Expert System* (NIDES), o qual tinha um avançado sistema de análise estatística (ANDERSON et al., 1995).

As técnicas baseadas em conhecimento ou baseados em regras, utilizam um conjunto de regras e parâmetros aliado a um classificador, criando sistemas especialistas. Esta técnica consegue reduzir o número de falsos positivos. (GARCÍA-TEODORO et al., 2009).

Os sistemas especialistas têm por objetivo classificar os dados de acordo com um

conjunto de regras, sendo divididos em três etapas: Em primeiro lugar, diferentes atributos e classes são identificadas a partir dos dados de formação. Em segundo lugar, um conjunto de classificação de regras, parâmetros e procedimentos são deduzidos. Em terceiro lugar, os dados são classificados. Algumas ferramentas formais podem ser utilizadas no desenvolvimento de especificadores, como a máquina de estados finitos, que delimita uma sequência de estados e as transições entre eles. Algumas linguagens de descrição também podem ser consideradas como *N-grammars* e UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada).

O aprendizado de máquina pode ser definido como a capacidade de um programa aprender e melhorar sua performance de acordo com as atividades que realiza (GARCÍA-TEODORO et al., 2009; PATCHA; PARK, 2007). No contexto de detecção de anomalias baseado em aprendizado de máquina, podemos citar Mafra et al. (2008) cujo trabalho desenvolveu um sistema multicamadas utilizando Redes Neurais de Kohonen e Redes de Máquina de Vetores de Suporte (SVM). As redes de Kohonen eram responsáveis pela classificação genérica (anomalia ou não anomalia) e a SVM por especificar os tipos de ataque (DoS, Worm, Scan ou Normal), sendo que para cada caso específico existe uma SVM (MAFRA et al., 2008).

Yeung e Ding (2003) descreve o uso do modelo Markoviano oculto na detecção de anomalias baseado em chamadas sequenciais de sistemas de perfis. No treinamento, o modelo calcula a probabilidade de uma sequência observada utilizando o algoritmo *forward* ou *backward*. O limite de probabilidade, baseado na probabilidade mínima dentre todas as sequências de treinamento, é utilizado para determinar se o comportamento é normal ou anômalo (YEUNG; DING, 2003). Devido à habilidade de adaptação para novos reconhecimentos, além da automatização do processo, o aprendizado de máquina tem se mostrado um método altamente eficaz na detecção de anomalias.

2.3.2 Ferramentas de Detecção

Um dos tipos de ferramentas utilizada na obtenção de dados para a detecção são aquelas que analisam o tráfego da rede, denominadas Analisadores de Protocolos de Rede, mais popularmente conhecidas como *Sniffers*, ou seja, farejadoras. Tais ferramentas possibilitam avaliar e examinar os dados que trafegam na rede em busca de soluções de problemas, problemas de desempenho e identificação de falhas (CLINCY; ABU-HALAWEH, 2005). Apesar de ser uma forma de garantir a segurança e confiabilidade da rede, os *sniffers* também podem ser utilizados para fins maliciosos, monitorando a rede em busca de informações que possam ser roubadas e posteriormente utilizadas.

As ferramentas de análise de tráfego mais conhecidas são *tcpdump* e *wireshark*. A principal diferença entre elas é o ambiente de trabalho, pois o *tcpdump* possui um ambiente textual, sendo utilizado através de um terminal ou *prompt de comando*, e o *wireshark* possui um ambiente gráfico para manipulação. Neste trabalho, será utilizado o *tcpdump* devido à possibilidade de acesso remoto.

Tcpdump

O *tcpdump* (TCPDUMP, 2016) é, como dito anteriormente, uma ferramenta *sniffer* com ambiente textual sendo utilizada através de um terminal. Seu funcionamento é baseado na LIBPCAP, uma API para captação de pacotes de rede que faz a comunicação direta com a NIC, mostrando todo o tráfego da rede.

O tráfego da rede refere-se ao fluxo dos pacotes, pois quando um pacote é enviado de um computador (nó) à outro, este passa por diversos nós intermediários, possibilitando sua captação (MANIKOPOULOS; PAPAVASSILIOU, 2002). Contudo, a NIC deve estar em modo promíscuo. A NIC tem dois modos de funcionamento:

- a) não promíscuo: quando um pacote é recebido pela NIC, seu MAC *address* é verificado, e caso ele seja o destinatário, o pacote é então aceito, caso contrário, é descartado.
- b) promíscuo: qualquer pacote recebido, mesmo que não seja seu destino, é aceito pela NIC.

O funcionamento do *tcpdump* é dado através de parâmetros que são fornecidos para obter determinadas características dos pacotes, podendo ser consultadas através da página de manual (*manpage*) da ferramenta⁹. Alguns dos parâmetros mais importantes e utilizados são apresentados na Tabela 3.

Tabela 3: Comandos *tcpdump*.

Chave	Função
-D	Mostra as interfaces de redes disponíveis.
-i iface	Determina qual interface será utilizada.
-n	Faz resolução e não mostra o domínio do <i>host</i> .
-A	Mostra cabeçalho e <i>payload</i> dos pacotes em ASCII.
-X	Idem, mas em hexadecimal e caracteres ASCII.
-x	Idem, mas somente em sequências em hexadecimal.
-v	Aumenta a quantidade de informações extraídas do cabeçalho do pacote.
-vv	Idem ao anterior, com mais informações ainda.
-vvv	Idem ao anterior, com mais informações.
-w arq	Grava o resultado da captura em um arquivo.
-r arq	Lê um arquivo previamente gravado com -w.
-t	Não mostra a data e a hora na tela.
-ttt	Mostra a data e a hora utilizando o padrão yyyy-mm-dd hh:mm:ss.ssssss.
-e	Mostra também os dados referentes à camada 2 do Modelo OSI (enlace).
-S	Exibe os resultados TCP utilizando a sua sequência absoluta, em vez da sequência relativa.

Fonte: Mota Filho (2013)

O comando apresentado na Figura 16 demonstra uma utilização básica da ferramenta, podendo fornecer maiores informações de acordo com a necessidade e os parâmetros fornecidos.

⁹ http://www.tcpdump.org/tcpdump_man.html.

Figura 16: Exemplo *tcpdump*.

```

bruna@bruna-V5-171:~$ sudo tcpdump -i wlan0
[sudo] password for bruna:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:27:23.320005 ARP, Request who-has PowerBox.home tell LAPTOP-MOHR4V0A, length
28
12:27:33.358238 ARP, Request who-has PowerBox.home tell LAPTOP-MOHR4V0A, length
28
12:27:34.527996 IP bruna-V5-171.26714 > PowerBox.home.domain: 53037+ PTR? 255.25
5.255.in-addr.arpa. (46)
12:27:33.594506 IP bruna-V5-171.17500 > 255.255.255.255.17500: UDP, length 205
12:27:33.595317 IP bruna-V5-171.17500 > 192.168.25.255.17500: UDP, length 205
12:27:35.917790 IP PowerBox.home.35238 > 239.255.255.250.1900: UDP, length 336
12:27:35.921591 IP PowerBox.home.35238 > 239.255.255.250.1900: UDP, length 345
12:27:35.925666 IP PowerBox.home.35238 > 239.255.255.250.1900: UDP, length 388
12:27:35.929832 IP PowerBox.home.35238 > 239.255.255.250.1900: UDP, length 400
12:27:35.934260 IP PowerBox.home.35238 > 239.255.255.250.1900: UDP, length 402
12:27:35.938501 IP PowerBox.home.35238 > 239.255.255.250.1900: UDP, length 416
12:27:35.941659 IP PowerBox.home.49511 > 239.255.255.250.1900: UDP, length 272
^C
12 packets captured
41 packets received by filter
0 packets dropped by kernel
bruna@bruna-V5-171:~$ █

```

Fonte: Elaborado pela autora.

Elá captura os pacotes da interface `wlan0`, a qual é a interface da rede sem fio, sendo que o *tcpdump* deve ser utilizado como super-usuário (`sudo` no ambiente linux). Os resultados apresentados variam de acordo com o tipo de pacote, mas no geral são apresentados com hora (microssegundos), tipo do protocolo, endereço de origem, endereço de destino (indicado por `source` » `destination`) com as portas correspondentes e as informações específicas do protocolo.

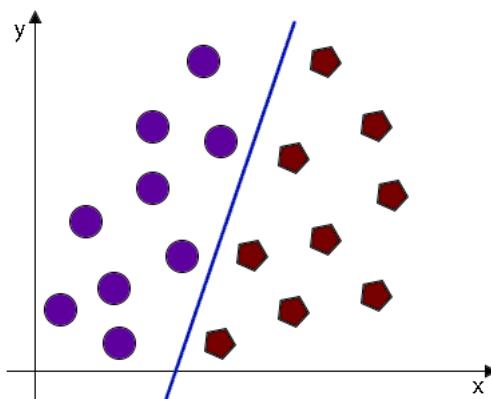
2.4 Classificação de Padrões

Classificação é o método de separar dados (amostras) em categorias ou classes, sendo uma de suas metodologias a classificação de padrões ou reconhecimento de padrões (*pattern recognition*), a qual é obtida através de técnicas de aprendizado de máquina. O classificador pode ser monoclasse (apenas uma possível classe para classificação), duas classes ou classificação binária (duas possíveis classes) ou multi-classes (DUTTON; CONROY, 1997; DUDA; HART; STORK, 2000). No entanto, é preciso ter cuidado com a diferença entre classe e característica. Uma classe é o conjunto final no qual o dado classificado pertence, já característica é uma propriedade da amostra a ser classificada, sendo cada amostra composta por um vetor de características.

As amostras são mapeadas para um espaço de busca, por exemplo, pontos em um plano, onde cada ponto é composto por seu vetor de características. Assim, o classificador de padrões tem por objetivo encontrar uma reta ou um conjunto de retas que melhor separe as amostras de cada classe, criando um hiperplano separador, ilustrado pela Figura 17. Para

que o melhor hiperplano seja encontrado, o classificador utiliza uma parte das amostras para treinamento e outra para teste. A fase de treinamento é responsável pelo aprendizado do classificador, ou seja, a cada iteração o classificador busca os parâmetros que melhor definem o hiperplano, atualizando-os sempre que for necessário. Na fase de teste, amostras desconhecidas são inseridas no espaço de busca e o classificador deve classificá-las juntamente aos elementos da classe que melhor as definem. Assim, é possível determinar a acurácia da classificação, ou seja, o valor percentual que indica a taxa de acerto do classificador.

Figura 17: Classificação binária linear.



Fonte: Elaborada pela autora.

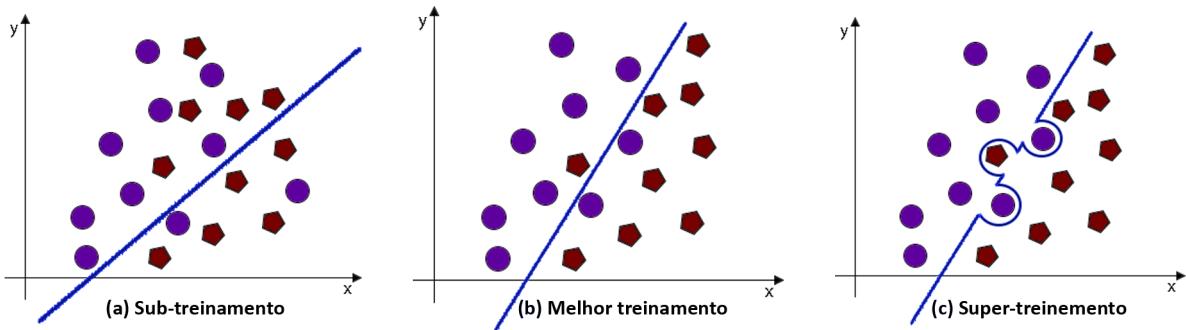
Contudo, é necessário observar a quantidade de amostras disponibilizadas para o treinamento e teste, a fim de evitar problemas de super-treinamento (*overfitting*) e sub-treinamento (*underfitting*), ilustrados pela Figura 18. Quando há um treinamento excessivo, em que o classificador se torna muito específico, ocorre o problema de *overfitting* (Figura 18.c), neste caso uma amostra desconhecida pode ser erroneamente classificada devido à essa especificação. O *underfitting* (Figura 18.a) refere-se ao contrário, ou seja, quando o conjunto de treinamento é composto por exemplos que não são suficientes para que o classificador aprenda corretamente as classes, apresentando uma baixa performance no processo de classificação (DUDA; HART; STORK, 2000). O melhor ajuste de treinamento do classificador ocorre quando há uma quantidade suficiente de amostras para que o mesmo aprenda, mas não sendo demasiado específico (Figura 18.b).

Os classificadores podem ser compreendidos de três modos:

- supervisionado: necessita de amostras rotuladas, ou seja, amostras pré-classificadas para que o classificador possa aprender como deve classificar (DUTTON; CONROY, 1997; DUDA; HART; STORK, 2000);
- semi-supervisionado: o conjunto de treinamento possui poucas amostras rotuladas, apenas para auxiliar no processo de aprendizado;

- c) não supervisionado: os dados não são rotulados, objetivando a separação das amostras em grupos similares, sem a pré-definição dos grupos (DUTTON; CONROY, 1997; DUDA; HART; STORK, 2000).

Figura 18: Ajuste de treinamento de uma classificação.



Fonte: Elaborado pela autora.

Atualmente, existem diversos algoritmos de classificação, os quais são baseados em métodos distintos, tais como, árvores de decisão, redes neurais, redes Bayesianas, k-vizinhos mais próximos, máquina de vetores de suporte, por exemplo. No entanto, a escolha do algoritmo de classificação depende de fatores como: tipo dos dados, acurácia desejada, tempo de execução disponível, parâmetros a serem escolhidos, dentre outros. Assim, não há como definir o melhor algoritmo, pois cada problema é específico, possuindo características particulares. Portanto, alguns dos mais conhecidos algoritmos de classificação serão brevemente apresentados.

Árvore de Decisão

Árvores de decisão (*Decision trees*) são árvores que classificam situações baseando-se no valor de suas características. Cada nó da árvore representa a característica a ser testada, e cada ramo é responsável por um possível valor que o nó pode assumir. A classificação inicia-se na raiz, na qual é testada sua característica e posteriormente sua ramificação, até que chegue em um nó-folha, contendo sua classificação. As árvores de decisão são geralmente univariadas, uma vez que sua divisão é baseada em uma única característica por nó interno (MITCHELL, 1997; MURTHY, 1998).

Redes Neurais Artificiais

Redes Neurais Artificiais (*Artificial Neural Networks - ANN*) ou Perceptron Multicamadas são compostas por um grande número de unidades (neurônios) conectados. Os neurônios são compostos por três camadas: entrada, escondida e saída. A camada de entrada é responsável por receber as informações que serão processadas, a camada escondida é responsável pelo processamento dos dados, e a camada de saída define onde os resultados são apresentados.

Durante a etapa de treinamento, a ANN visa encontrar os melhores valores para os pesos das conexões entre os neurônios, os quais são utilizados na função de ativação. Cada entrada tem um valor de ativação que representa alguma característica externa à rede. A função de ativação é responsável por propagar o sinal para o próximo neurônio, podendo ser baseada em uma função limite (0 ou 1) ou um valor no intervalo [0, 1] (KOTSIANTIS, 2007; MITCHELL, 1997).

Redes Neurais Artificiais com Função de Base Radial

Rede Neural Artificial com Função de Base Radial (*Radial Basis Function - ANN-RBF*) é um tipo de rede neural multicamadas também composta por três partes: entrada, escondida e saída. A camada de entrada recebe um vetor de entrada que é aplicado à uma transformação não-linear na camada escondida. Uma combinação linear dos resultados obtidos na camada escondida é realizada na camada de saída. Dentre as diversas funções de base radial, a mais empregada e conhecida é a Gaussiana, que possui uma simples e efetiva formulação (ORR, 1996).

Redes Bayesianas

A Rede Bayesiana (*Bayesian Network - BN*) é um modelo gráfico de probabilidade de relacionamento entre um conjunto de características. Sua estrutura é um grafo direcionado acíclico, onde cada nó representa uma característica e seus arcos representam a influência entre elas. A ausência de um arco representa a condição de independência. A força da dependência é dada por um valor probabilístico obtido através do teorema de Bayes (MITCHELL, 1997).

K-vizinhos mais próximos

O método do K-vizinhos mais próximos (*K-Nearest Neighbors - KNN*) considera cada caso dentro de um conjunto de dados como um ponto no espaço n -dimensional, onde cada uma das n -dimensões corresponde a uma das n -características do caso. Os pontos estão dispostos no espaço de modo que tenham propriedades semelhantes aos seus vizinhos mais próximos. Assim, um caso não classificado, busca dentre seus vizinhos, um que esteja classificado, e assim, define-se pela mesma classe. (MITCHELL, 1997)

Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (*Support Vector Machine - SVM*) foi inicialmente desenvolvida para uma classificação binária, baseada no aprendizado estatístico. O SVM visa criar um hiperplano de margem máxima, que é a distância máxima entre amostras de diferentes classes em relação ao hiperplano separador, entre os conjuntos de possíveis classes. É uma técnica bastante utilizada, pois tem boa capacidade de generalização e bom desempenho em grandes dimensões (CRISTIANINI; SHAWE-TAYLOR, 2000).

Floresta de Caminhos Ótimos

O classificador baseado em Florestas de Caminhos Ótimos (*Optimum-Path Forest - OPF*) utiliza uma metodologia baseada em grafos e seu funcionamento parte do princípio da seleção de protótipos (amostras mais representativas das classes) os quais devem competir entre si para conquistar os nós adjacentes, definindo, então, suas classes (PAPA; FALCÃO; SUZUKI, 2009). Uma explicação mais detalhada do OPF será apresentada a seguir, visto ser o classificador utilizado neste trabalho.

2.4.1 Floresta de Caminhos Ótimos

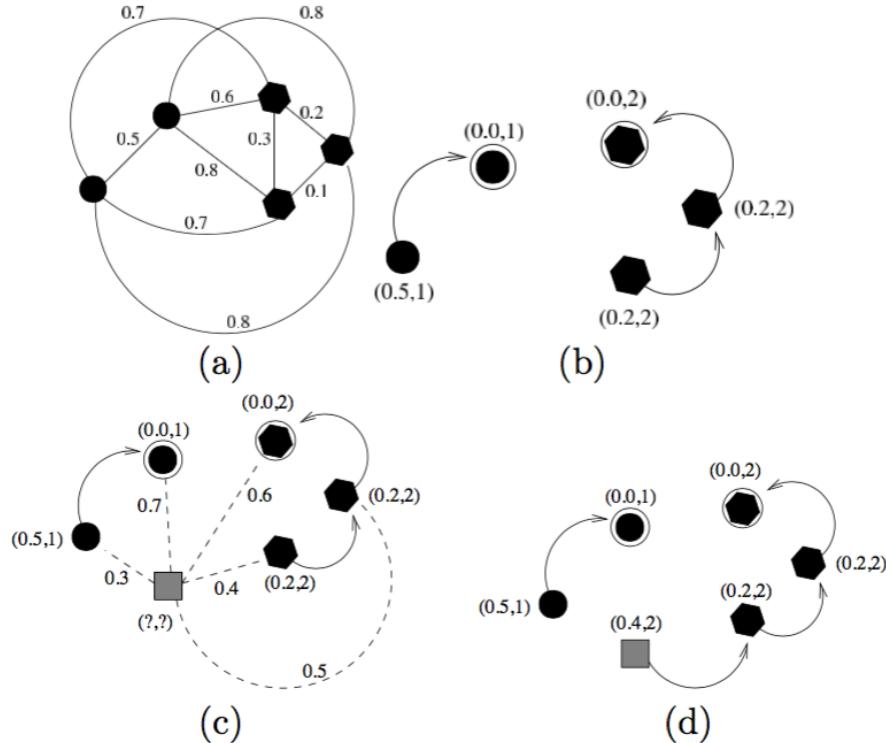
O classificador baseado em Floresta de Caminhos Ótimos é uma técnica multi-classes, desenvolvido por Papa, Falcão e Suzuki (2009) tendo como motivação a criação de um classificador eficiente na fase de treinamento e eficaz na fase de teste, sem a necessidade de um grande volume de dados.

O OPF tem como principal objetivo a segmentação do espaço de características sem perda de generalidade. Os espaços n -dimensionais identificados para uma classificação binária podem ser divididos em: (a) linearmente separáveis; (b) linearmente separáveis por partes; e (c) não linearmente separáveis (PAPA; FALCÃO; SUZUKI, 2009). A técnica do OPF pode ser supervisionada, semi-supervisionada ou não-supervisionada, sendo que neste trabalho será tratada a versão supervisionada. O classificador é baseado em um grafo, tendo duas possíveis relações de adjacências: grafo completo ou grafo Knn . A diferença entre as duas abordagens são a relação de adjacências, a metodologia para estimar os protótipos e a função de custo de caminho utilizada. Neste trabalho será abordada a relação de adjacências por grafo completo.

Nesta abordagem, a ideia é criar um grafo completo, no qual quaisquer duas amostras estão conectadas entre si. Neste caso, os nós representam o vetor de características das amostras e as arestas conectam todos os nós (Figura 19.a.). No caso dos protótipos, os mesmos são escolhidos através de Árvores de Espalhamento Mínimo (*Minimum Spanning Tree - MST*)¹⁰ a fim de encontrar os elementos mais próximos de diferentes classes, ou seja, selecionar amostras localizadas em regiões mais propensas à erros de classificação (fronteiras das classes). Após a definição dos protótipos, os mesmos competem entre si para conquistar nós adjacentes buscando sempre encontrar o melhor caminho (menor custo) definido pela função de custo de caminho, criando, ao final do processo, árvores de caminhos ótimos (*Optimum-Path Trees - OPTs*) (Figura 19.b.). O teste tem por objetivo a validação da técnica por meio do cálculo da acurácia de detecção, onde uma nova amostra é inserida no grafo (Figura 19.c) e o protótipo que oferece o caminho de menor custo deve conquistá-la, definindo sua classe (Figura 19.d.).

¹⁰ MSTs são subgrafos que conectam todos os vértices do conjunto com o menor custo possível.

Figura 19: Funcionamento OPF.



Fonte: Papa, Falcão e Suzuki (2009). (a) Representação do grafo completo; (b) Processo de conquista dos protótipos (amostras circuladas); (c) Inserção de amostra de teste no grafo; (d) Classificação da nova amostra.

OPF com Grafo Completo

Seja \mathcal{Z} uma base de dados, tal que $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$, onde \mathcal{Z}_1 e \mathcal{Z}_2 representam o conjunto de treinamento e teste, respectivamente. Cada amostra $s \in \mathcal{Z}$ pode ser representada por seu vetor de características $\vec{v}(s) \in \mathbb{R}^n$. O grafo do OPF_{cpl} é representado por $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, onde \mathcal{A} refere-se ao conjunto das arestas que conectam todos os pares de nós e \mathcal{V} o conjunto dos vetores de características $\vec{v}(s)$, $\forall s \in \mathcal{Z}$. Além disso, seja $\lambda(\cdot)$ uma função que atribui um rótulo verdadeiro para cada amostra em \mathcal{Z} .

Treinamento

Seja o grafo $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{A})$ induzido do conjunto de treinamento, onde \mathcal{V}_1 contém todos os vetores de características das amostras pertencentes ao conjunto de treinamento. O primeiro objetivo da fase de treinamento é obter um conjunto de protótipos \mathcal{S} , onde $\mathcal{S} \subset \mathcal{Z}_1$.

Seja um caminho π_s em \mathcal{G} com término em s e uma função $f(\pi_s)$ que associa um valor à esse caminho. A fim de um protótipo conquistar as amostras adjacentes, o propósito é minimizar $f(\pi_s)$ através de uma função de custo de caminho f_{max} dada por:

$$\begin{aligned} f_{max}(\langle \mathbf{s} \rangle) &= \begin{cases} 0 & \text{se } \mathbf{s} \in S, \\ +\infty & \text{caso contrário} \end{cases} \\ f_{max}(\pi \cdot \langle \mathbf{s}, \mathbf{t} \rangle) &= \max\{f_{max}(\pi), d(\mathbf{s}, \mathbf{t})\}. \end{aligned} \quad (1)$$

em que $f_{max}(\pi \cdot \langle \mathbf{s}, \mathbf{t} \rangle)$ computa a distância máxima entre as amostras adjacentes \mathbf{s} e \mathbf{t} ao longo do caminho $\pi \cdot \langle \mathbf{s}, \mathbf{t} \rangle$. Um caminho π_s é dito como ótimo se $f(\pi_s) \leq f(\tau_s)$ para qualquer outro caminho τ_s .

A minimização de f_{max} atribui a cada amostra $\mathbf{t} \in \mathcal{Z}_1$ um caminho ótimo $P^*(\mathbf{t})$, cujo custo mínimo $C(\mathbf{t})$ é dado por:

$$C(\mathbf{t}) = \min_{\forall \pi_t \in (\mathcal{Z}_1, \mathcal{A})} \{f_{max}(\pi_t)\}. \quad (2)$$

Teste

O grafo do conjunto de teste $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A})$ é composto por amostras $\mathbf{t} \in \mathcal{V}_2$. Cada amostra \mathbf{t} é conectada à uma amostra $\mathbf{s} \in \mathcal{V}_1$ tornando \mathbf{t} parte do grafo original. O objetivo é encontrar um caminho ótimo $P^*(\mathbf{t})$ de \mathcal{S} até \mathbf{t} com a classe $\lambda(R(\mathbf{t}))$ de seu protótipo $R(\mathbf{t}) \in \mathcal{S}$. Ao final do processo, a amostra \mathbf{t} é removida do grafo. Esse caminho pode ser identificado avaliando o valor de custo ótimo $C(\mathbf{t})$:

$$C(\mathbf{t}) = \min\{\max\{C(\mathbf{s}), d(\mathbf{s}, \mathbf{t})\}\}, \forall \mathbf{s} \in \mathcal{Z}_1. \quad (3)$$

2.5 Otimização

Otimização é o processo de encontrar o melhor valor para um problema modelado matematicamente, minimizando-o ou maximizando-o. Assim, uma função f é dita minimizada, se, $f(x_{otimo}) \leq f(x), \forall x$, e maximizada, se, $f(x_{otimo}) \geq f(x), \forall x$ do problema. No entanto, nem todo problema pode ser facilmente solucionado, podendo encontrar soluções sub-ótimas ou ficar preso em ótimos locais. Assim, métodos baseados em programação matemática, tais como, baseados em gradiente ou métodos analíticos, podem não ser eficientes em determinados problemas.

A fim de buscar soluções que resolvam problemas mais complexos, surgiu a otimização com algoritmos bio-inspirados, que são baseados em populações e meta-heurística. A bio-inspiração tem como base o comportamento de animais em uma "sociedade", já a meta-heurística é um conjunto de procedimentos que visam encontrar uma boa solução, possivelmente a ótima, através de procedimentos de intensificação (*exploitation*) e diversificação (*exploration*) (CREPINSEK; LIU; MERNIK, 2013; EIBEN; SCHIPPERS, 1998).

Diversificação consiste no processo de visitar novas regiões do espaço de busca e a intensificação em visitar regiões no espaço de busca cuja vizinhança já foi anteriormente visitada. Tais processos tem como objetivo obter uma completa abrangência de todo o espaço de busca. Para tal, é necessário alternar entre ambos processos durante a otimização, geralmente, iniciando-se com a diversificação para que as regiões do espaço de busca sejam exploradas, e então, a intensificação é aplicada para refinar tais regiões. No entanto, é aconselhável que mesmo na fase de intensificação ainda exista um baixo nível de diversificação.

Uma das abordagens baseadas em populações é a inteligência de enxame (*Swarm Intelligence*), ou inteligência de colônias, que são técnicas baseadas no comportamento auto-organizado (autônomo). O algoritmo é inspirado na capacidade de observação e alteração do ambiente de indivíduos pertencentes à uma sociedade (BONABEAU; DORIGO; THERAULAZ, 1999; SERAPIÃO, 2009). Alguns dos algoritmos mais conhecidos são: algoritmo de colônia artificial de abelhas, otimização por colônia de formigas e a otimização por enxame de partículas.

Algoritmo de Colônia Artificial de Abelhas

O algoritmo de colônia artificial de abelhas (*Artificial Bee Colony algorithm* - ABC) é baseado no comportamento de colmeias de abelhas, e é composto por três tipos de abelhas: trabalhadoras (*employed*), oportunistas (*onlookers*) e exploradoras (*scouts*). As abelhas trabalhadoras são alocadas uma para cada fonte de néctar próximo à colmeia, e passam as informações referentes à fonte através de uma dança, que é analisada pelas abelhas oportunistas, responsáveis por escolher a melhor fonte. Quando uma fonte de néctar é esgotada, a abelha trabalhadora se torna uma abelha exploradora e parte em busca de uma nova fonte de néctar, reiniciando o ciclo (KARABOGA; BASTURK, 2007).

Otimização por Colônia de Formigas

A otimização por Colônia de Formigas (*Ant Colony Optimization* - ACO) é baseada no comportamento coletivo de algumas espécies de formigas e é realizada através da comunicação indireta, utilizando-se feromônios. Quando vão em busca de alimento, as formigas liberam feromônios pelo caminho, criando uma trilha. Com isso, as formigas tendem a seguir caminhos com uma maior quantidade de feromônios e um menor espaço, objetivando encontrar a menor trilha possível. A quantidade de feromônios tende à aumentar mais rapidamente, atraindo as formigas para a mesma trilha (DORIGO; BIRATTARI; STUTZLE, 2006).

Otimização por Enxame de Partículas

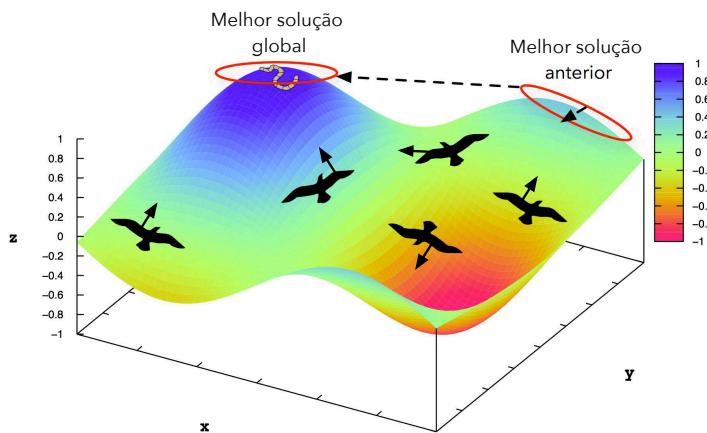
A otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) é baseada no comportamento social biológico, especificamente no comportamento social de bandos de pássaros e cardumes de peixes, os quais têm capacidade de aprender e transmitir conhecimento para outros indivíduos da população. Este compartilhamento de conhecimento influencia as

decisões dos indivíduos, que buscam seguir a melhor opção (KENNEDY; EBERHART, 1995). Uma explicação mais detalhada do PSO será apresentada a seguir, visto ser a técnica de otimização utilizada neste trabalho.

2.5.1 Otimização por Enxame de Partículas

A otimização por Enxame de Partículas foi desenvolvida por Kennedy e Eberhart (1995) e baseia-se, como dito anteriormente, no comportamento social de bandos de pássaros e cardumes de peixes. Assim, seu intuito é modelar computacionalmente tal comportamento a fim de encontrar os melhores valores para o problema, conceito representado pela Figura 20, na qual o objetivo é que os pássaros encontrem o alimento (máximo global). Outras definições consideram o PSO como um algoritmo de pesquisa baseado em processos estocásticos e populacionais, onde a aprendizagem do comportamento social permite a cada solução possível (partícula) mover-se dentro desse espaço (enxame) à procura de outras partículas que possuem melhores características e, assim, minimizar a função objetivo. Esse mecanismo sócio-recognitivo pode ser resumido em três princípios (KENNEDY; EBERHART; SHI, 2001): (a) avaliação, (b) comparação e (c) imitação.

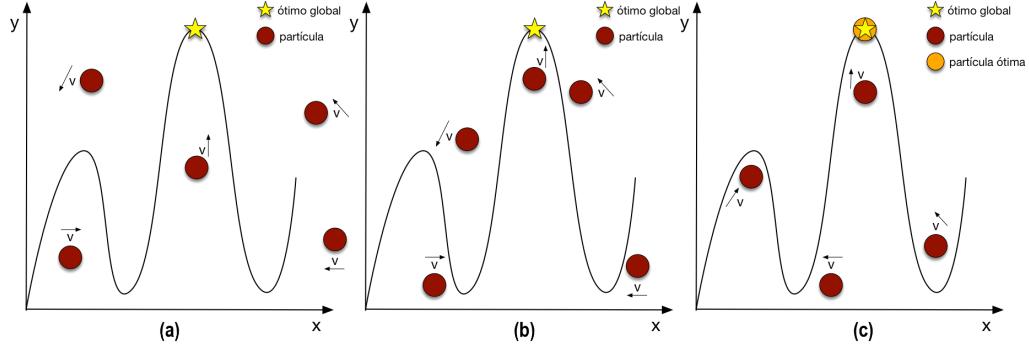
Figura 20: Espaço de busca com bando de pássaros (representação PSO).



Fonte: Elaborado pela autora.

Desta forma, as partículas são dispostas em um espaço de busca delimitado, onde cada partícula possui seu valor em relação à uma função de aptidão, além de sua velocidade e posição (Figura 21.a). A tendência das partículas é movimentar-se dentro do espaço de busca, influenciando as partículas mais próximas (vizinhas) encontrando, assim, o melhor valor para a função (Figura 21.b). Em outras palavras, cada partícula busca o melhor caminho para alcançar o objetivo (Figura 21.c) imitando o comportamento da melhor partícula (KENNEDY; EBERHART, 1995).

Figura 21: Sistematização PSO.



Fonte: Elaborada pela autora. (a) Partículas dispostas no espaço; (b) Movimentação das partículas; (c) Influência da partícula de melhor valor.

Dado um espaço n -dimensional R^n composto por partículas, cada partícula $p_i = (x_i, v_i) \in R^n$ possui dois parâmetros: (a) sua posição x_i e (b) velocidade v_i . Também são conhecidas a melhor solução (posição no enxame) local \hat{x}_i e global \hat{s} .

O processo é iniciado com valores aleatórios de velocidade e posição, onde cada partícula é avaliada em relação à uma função objetivo f , também chamada de função aptidão (*fitness*). A movimentação da partícula é dada por:

$$v_i = wv_i + c_1r_1(\hat{x}_i - x_i) + c_2r_2(\hat{s} - x_i). \quad (4)$$

onde w refere-se à força de inércia que controla o poder de interação entre as partículas, r_1 e r_2 são variáveis aleatórias entre $[0, 1]$ que trazem a ideia de comportamento social, e as constantes c_1 e c_2 são fatores de aprendizado utilizados para guiar as partículas. A posição das partículas é dada por:

$$x_i = x_i + v_i. \quad (5)$$

3 Desenvolvimento

A detecção de intrusão é uma das técnicas mais conhecidas e utilizadas para prevenção e bloqueio de ataques às redes de computadores. A abordagem de detecção por anomalias é uma alternativa que pode trazer uma maior eficácia ao processo, devido à sua flexibilidade de detecção.

Em sua maioria, é utilizada a abordagem estatística ou baseada em regras. Contudo, a abordagem por aprendizado de máquina pode trazer um panorama mais inovador. Destarte, este trabalho propõe a utilização de ferramentas baseadas em aprendizado de máquina, tais como, um classificador de padrões e um otimizador baseado em inteligência de enxame.

Entretanto, outro desafio enfrentado na detecção de anomalias é a escassa diversidade de dados disponíveis para análise, além da existência de falso-positivos, dificultando sua classificação. Com a grande exploração das mesmas bases de dados (KDDCup¹, NSL-KDD², ICSX³ e DARPA⁴) em diversas pesquisas, seus resultados apresentam-se desgastados, carecendo de novas bases para experimentos na área. Portanto, foi criada uma nova base de dados que pudesse ser utilizada para aplicar as técnicas estudadas e ser disponibilizada para posteriores pesquisas.

Neste capítulo, serão descritas em maiores detalhes as etapas de desenvolvimento do projeto, sistematizadas pela Figura 22, sendo explicada na Seção 3.1 a metodologia de pesquisa, na Seção 3.2 o processo de criação da base de dados, a implementação e aplicação das técnicas de classificação e otimização nas Seções 3.3 e 3.4, respectivamente. A Seção 3.5 informa as configurações dos dados referentes ao processamento realizado, já a Seção 3.6 exibe, através de gráficos e tabelas, os resultados obtidos acompanhado de uma análise a respeito da eficácia das técnicas aplicadas. Por fim, a Seção 3.7 mostra a interface gráfica criada para o trabalho em questão a fim de ilustrar, visualmente, o procedimento realizado.

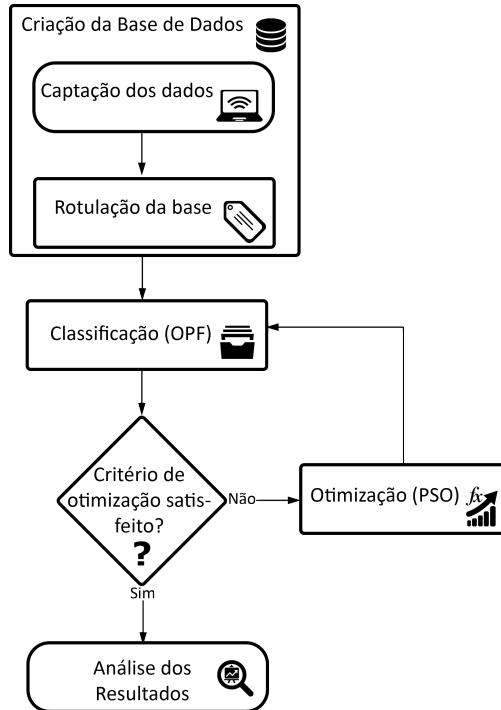
¹ <http://www.sigkdd.org/kddcup/index.php>

² <http://nsl.cs.unb.ca/NSL-KDD/>

³ <http://www.uvic.ca/engineering/ece/isot/datasets/>

⁴ <http://www.ll.mit.edu/ideval/data/>

Figura 22: Fluxograma das etapas de desenvolvimento do trabalho.



Fonte: Elaborado pela autora.

3.1 Método de Pesquisa

Esta seção explica a metodologia aplicada neste trabalho. Conforme descrito na Seção 3, foi utilizado um classificador de padrões baseado em floresta de caminhos ótimos, fundamentado na Seção 2.4.1, e um otimizador baseado em inteligência de enxames, fundamentado na Seção 2.5.1. A escolha das técnicas fundadas em relação às outras sintetizadas nas Seções 2.4 e 2.5 deve-se a diversos fatores.

Em relação ao classificador, foi escolhido o OPF devido à vantagem de processamento, já que o mesmo obtém um bom resultado em um menor tempo de execução. Uma comparação do OPF em relação a outras técnicas de classificação de padrões pode ser vista no trabalho de Pereira (2012) sobre detecção de intrusão em redes de computadores utilizando floresta de caminhos ótimos (OPF). A Tabela 4 apresenta uma comparação dos resultados obtidos através de diversos métodos de classificação em relação ao OPF na base de dados KDDCup.

Tabela 4: Comparação das técnicas de classificação de padrões na KDDCup.

Classificadores/Métricas	Acurácia	Tempo de treinamento (s)	Tempo de teste (s)
OPF	$93.19\% \pm 2.49$	13.109	20.134
Bayes	$91.38 \% \pm 2.49$	4.9268	263.349
SVM-RBF	$89.59\% \pm 3.91$	5018.6606	1.9853
SOM	$89.95\% \pm 1.97$	3121.4089	8.05812

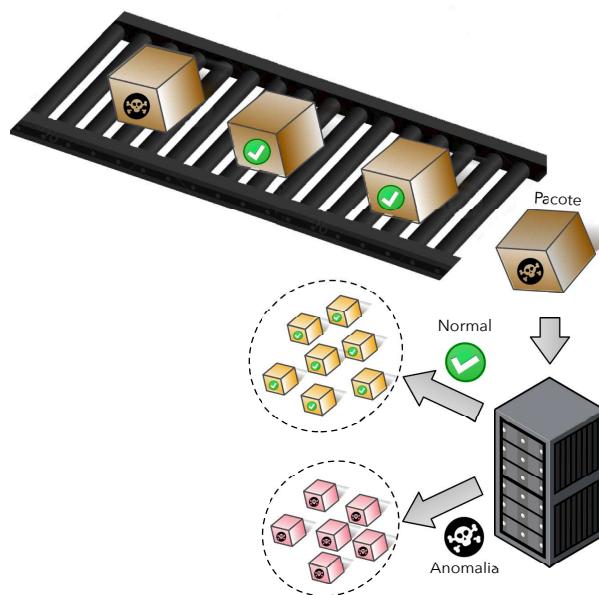
Fonte: Adaptada de Pereira (2012).

Pode-se perceber que o OPF, além de apresentar a melhor taxa de acurácia de classificação, possui menor tempo geral de processamento do que as demais técnicas analisadas. No entanto, como já discutido na Seção 2.4, não é possível dizer qual é o melhor algoritmo de classificação, pois o mesmo depende do tipo de problema a ser analisado.

A escolha da otimização através do PSO deve-se à sua simplicidade e eficácia. Por ser uma técnica meta-heurística que se utiliza do processo de *exploitation* e *exploration*, consegue obter uma melhor avaliação do espaço de busca, encontrando um ótimo valor para problemas de otimização, mesmo em casos de elevado número de características. Selvi e Umarani (2010) apresentam a multi-aplicabilidade como mais uma vantagem do PSO em relação à outros processos de otimização por inteligência de enxame.

Assim, a aplicação das técnicas selecionadas tem por objetivo melhorar a taxa de acurácia no processo de classificação de anomalias, conforme ilustra a Figura 23, criando um sistema de detecção inteligente e otimizado para a área de segurança de redes.

Figura 23: Representação gráfica do problema proposto.

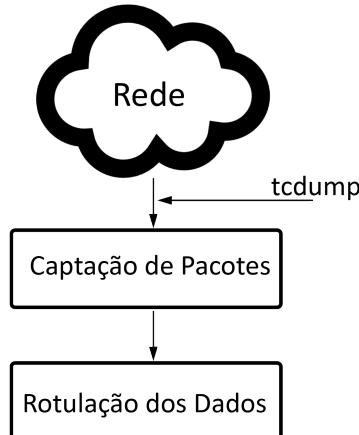


Fonte: Elaborada pela autora.

3.2 Criação da Base de Dados

O processo de criação da base de dados foi executado através de duas etapas: a captação de pacotes que trafegam na rede analisada e a rotulação dos dados em duas classes: anomalia ou não anomalia, obtendo assim, uma base robusta e rotulada para aplicar as técnicas estudadas. Uma sistematização do processo de criação da base pode ser vista na Figura 24.

Figura 24: Fluxograma de criação da Base de Dados.



Fonte: Elaborado pela autora.

3.2.1 Captação

A base de dados a ser analisada foi criada através da captação de pacotes que circulam na rede oferecida pela universidade, a *wi-fi UNESP WFU*, sendo a captação realizada pela ferramenta *tcpdump*. A captação dos dados foi realizada em uma máquina com adaptador de rede *Wireless TP-Link TL-WN725N V2*.

Os dados foram capturados em dias úteis, no período de um mês. Cabe ressaltar que a referida rede possui um grande tráfego de dados, pois esta é disponibilizada para alunos, professores e funcionários. Conforme dados quantificados pelo Serviço Técnico de Informática da Faculdade de Ciências da Universidade Estadual Paulista (STI/FC), o tráfego gerado pela rede sem fio *WFU* obtém aproximadamente 1.500 usuários conectados concomitantemente em um dia de atividade.

Assim, foi definido um período de 6 horas diárias para captação, sendo estas divididas em 2h e distribuídas entre os períodos: matutino, vespertino e noturno, a fim de abranger todos os períodos de aulas realizadas no campus.

Para conseguir obter a maior quantidade de informações possíveis, a ferramenta *tcpdump* foi rodada com parâmetros que disponibilizassem o maior número possível de características. A sintaxe do comando de captação pode ser vista a seguir, e a explicação de cada parâmetro utilizado é informado na Tabela 5.

Código: Comando de captação *tcpdump*.

```
# sudo tcpdump -G 7200 -W 1 -i wlan0 -A -n -vvv -tttt -w database-k.i.pcap
```

Tabela 5: Parâmetros do comando *tcpdump*.

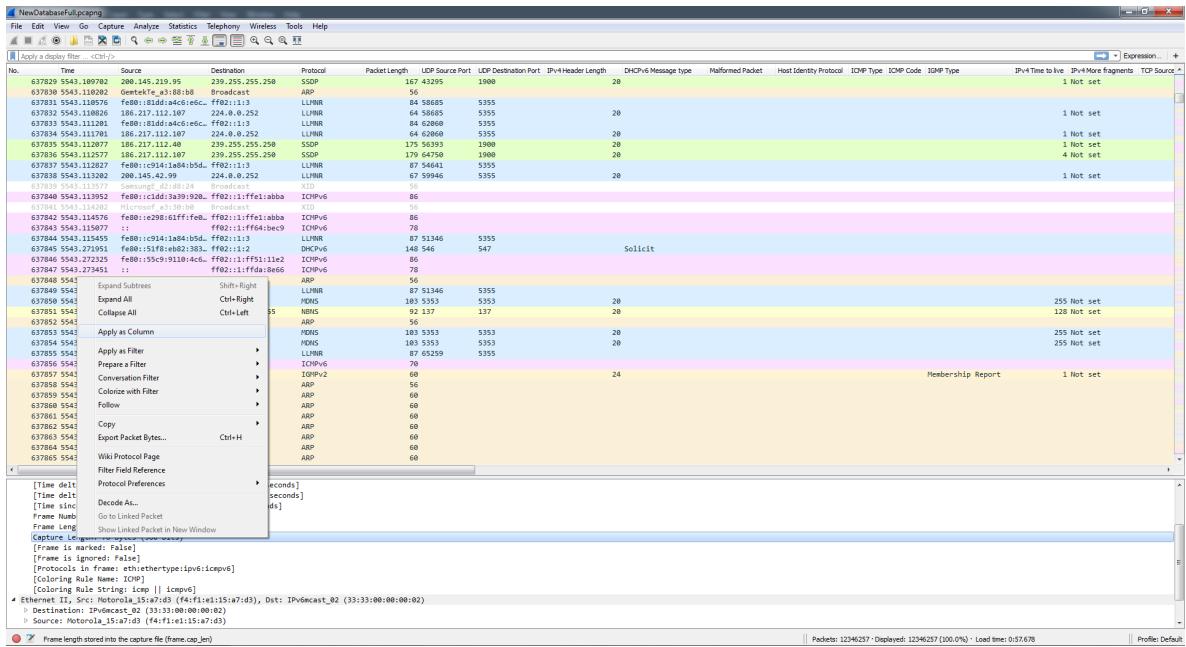
Parâmetro	Função
-G	Define o tempo de execução do <i>tcpdump</i> em segundos. Nesse caso, os 7200 são referentes ao período de 2 horas
-W	Define a quantidade de arquivos gerados na gravação. Neste caso, será um arquivo por chamada ao <i>tcpdump</i>.
-i iface	Determina qual interface será utilizada. wlan0 faz referência à interface wireless.
-A	Mostra o cabeçalho e payload dos pacotes em código ASCII.
-n	Faz resolução e não mostra o domínio do host, quando está opção não for bloqueada.
-vvv	Aumenta ainda mais a quantidade de informações extraídas do cabeçalho do pacote.
-tttt	Mostra a data e a hora utilizando o padrão yyyy-mm-dd hh:mm:ss.ssssss. Neste caso, a data e hora serão utilizadas apenas para controle de captação.
-w arq	Grava o resultado da captura em um arquivo. Neste caso, cada dia k gerará arquivos i: um por período.

Fonte: Elaborado pela autora.

Após a captação dos dados, os arquivos foram concatenados e manipulados na ferramenta *Wireshark*⁵, que conforme explicado na Seção 2.3.2, é uma ferramenta de análise de tráfego com ambiente gráfico, o que facilita a manipulação dos dados. Assim, informações importantes contidas nos pacotes foram transformadas em características (colunas) da base, conforme ilustra a Figura 25. Após a manipulação, a base foi exportada para um banco de dados, sendo rotulada manualmente.

⁵ <https://www.wireshark.org/>

Figura 25: Manipulação da base no Wireshark.



Fonte: Elaborado pela autora.

3.2.2 Rotulação

A importância da etapa de rotulação dar-se-á devido a utilização de um classificador supervisionado, sendo necessário informar as possíveis classes. Para realizar a rotulação manual, é necessário conhecer um padrão que define o que é uma anomalia e o que não é uma anomalia, denominado assinatura de intrusão. Uma assinatura é uma característica particular do ataque, que o difere das demais intrusões. Neste trabalho, foi utilizada a documentação de análise de detecção de intrusão da DARPA, realizada pelo Lincoln Laboratory do MIT (LINCOLN LABORATORY, s.d), que foi baseada na tese de mestrado de Kendall (1998). A Tabela 6 apresenta algumas assinaturas dos principais tipos de ataques: negação de serviço (*Denial of Service*), usuário para máquina (*User to Root*), remoto para local (*Remote to local*) e de sondas (*Probes*).

Tabela 6: Assinatura de intrusões.

Tipo	Nome	Protocolo	Descrição
Negação de Serviço			
	Apache2	HTTP	Alto número (> 1000) requisições HTTP para o mesmo endereço IP de destino.
	Arppoison	ARP	Retorno errado de endereço MAC da máquina que executa o ataque após uma requisição "who has".
	Back	URL	Requisição de URL com alto número (>100) de barra (slash - "/").
	Crashii	FTP e TELNET	Requisição GET malformada através da porta 80 (Porta padrão: FTP - 20/21, TELNET - 23).
	Dosnuke	NETBIOS	Pacote contém uma bandeira (flag) "urg".
	Land	IP	Pacote com mesmo endereço IP de origem e destino.
	Mailbomb	SMTP, POP3 e IMAP	Vários emails enviados com o mesmo remetente e destinatário em um curto período de tempo.
	Neptune	TCP	Pacotes SYN enviados simultaneamente destinado a uma única máquina vindo de uma máquina inacessível.
	Ping of Death	ICMP	Pacotes com tamanho maior que 64000 bytes.
	Process Table	-	Grande número (> 100) de conexões ativadas em uma porta particular. Porta particular: >1024 .
	Selfping	ICMP	Ping no broadcast ⁶ antes da máquina morrer.
	Smurf	TCP	Retorno de echo, sem ter realizado nenhuma requisição.
	Syslogd	UDP	Pacote destinado à uma porta syslog (514) com endereço de origem inacessível.
	Tcprequest	TCP	Sessão TCP com pacotes Request vindo de máquinas que, inicialmente, tentaram conexão.
Usuário-Máquina			
	Casesen	-	Transferência de 3 arquivos executáveis (.exe), sendo um deles o "psxss.exe".
	Loadmodule	-	Uma sessão que contenha "loadmodule".
	Perl	-	Pacote que contém a string " $\$ \geq 0; \$ \leq 0$ ".
Remoto para Local			
	Dictionary	-	Diversas tentativas de login sem êxito em um determinado período de tempo.
	Imap	IMAP	String de autenticação Imap muito grande (oversized).
	Named	DNS	Solicitações de DNS reverso maiores que o buffers de 4096 bytes.
	Netbus	TCP e UDP	Informação "netbus".
	Sendmail	SMTP	Emails recebidos que contenham um cabeçalho MINE muito grande (overflow).
	Xlock	-	Tráfego suspeito vindo de uma máquina desconhecida (unknown).
Sondas			
	Ipsweep	ICMP e DNS	Diversos pings da mesma máquina (de origem) para cada máquina disponível na rede.
	Mscan	STATD, IMAP, POP e DRIX	Máquina externa conectada as portas: (STATD), (IMAP), (POP) e (DRIX).
	NTinfoscan	FTP	Requisições de HTML GET em diretórios como "/cgi-bin" e "/scripts" com usuário anônimo e senha "gestacct@compuserve.com".
	Resetscan	ARP	Pacotes enviados à endereços IPs sem conexões anteriores com "who has" solicitando endereços MAC para IP inexistente.

Fonte: Adaptada de Lincoln Laboratory (s.d.).

Através de um Sistema de Gerenciamento de Banco de Dados (SGBD) SQL, foi averiguada a presença de anomalias na base real desenvolvida. No entanto, devido ao aspecto acadêmico da rede analisada, poucas anomalias puderam ser identificadas. Isto posto, foi necessário injetar anomalias de acordo com as assinaturas descritas pela Tabela 6 criando, portanto, uma base de dados semi-sintética. As anomalias inseridas foram escolhidas empiricamente, baseando-se na maior probabilidade de ocorrência em um caso real, e podem ser vistas na Tabela 13, presente no Apêndice A. Um fragmento exemplificando a base pode ser visualizado

⁶ Broadcast é um endereço capaz de enviar uma mensagem a todos os hosts existentes na rede.

pela Figura 26.

Figura 26: Amostra da Base de Dados criada.

The screenshot shows a database interface with the following details:

- Database:** databasefull @ database (Local) - Table - Navicat Premium
- Table:** databasefull
- Columns:** No, Label, Attack Code, Time, Source, Destination, Same IP, Broadcast Destination, Protocol, Packet Length, UDP Source Port, UDI
- Data:** A large table containing 1000 rows of network traffic data. Each row includes a timestamp (Time), source IP (Source), destination IP (Destination), and various protocol-related metrics (Protocol, Packet Length, UDP Source Port, UDI).
- Bottom Status Bar:** SELECT * FROM 'databasefull' ORDER BY 'Time' LIMIT 0, 1000 | Record 23 of 1000 in page 1

Fonte: Elaborado pela autora.

Durante o processo de rotulação, foi avaliada a viabilidade de aplicação da técnica, e então, para que a mesma fosse possível, a base de dados foi reduzida para aproximadamente 1 milhão de amostras com 23 características, sendo 10% anômala, criando assim, a base denominada uneSPY⁷.

3.3 Classificação

A implementação utilizada do OPF foi desenvolvida por Papa, Falcão e Suzuki (2009), na forma de uma biblioteca, a LibOPF (PAPA; FALCÃO; SUZUKI, 2015), disponível em um repositório do *github*⁸. Essa versão foi escolhida por ser a original e implementada em linguagem C, o que traz uma maior facilidade de compreensão e adaptação por parte da autora.

No entanto, para utilizar o OPF, é necessário utilizar um formato específico na base de dados, já que o classificador comprehende características numéricas. O Apêndice A apresenta as tabelas de codificação utilizadas para transformar os dados alfanuméricos. O formato aceito é o binário, ou um arquivo texto que será convertido internamente. A Tabela 7 apresenta a formatação da base de dados necessária para o OPF.

⁷ http://wwwp.fc.unesp.br/~papa/recogna/network_detection

⁸ <https://github.com/jppbsi/LibOPF>

⁹ <https://github.com/jppbsi/LibOPF/wiki/OPF-file-format-for-datasets>

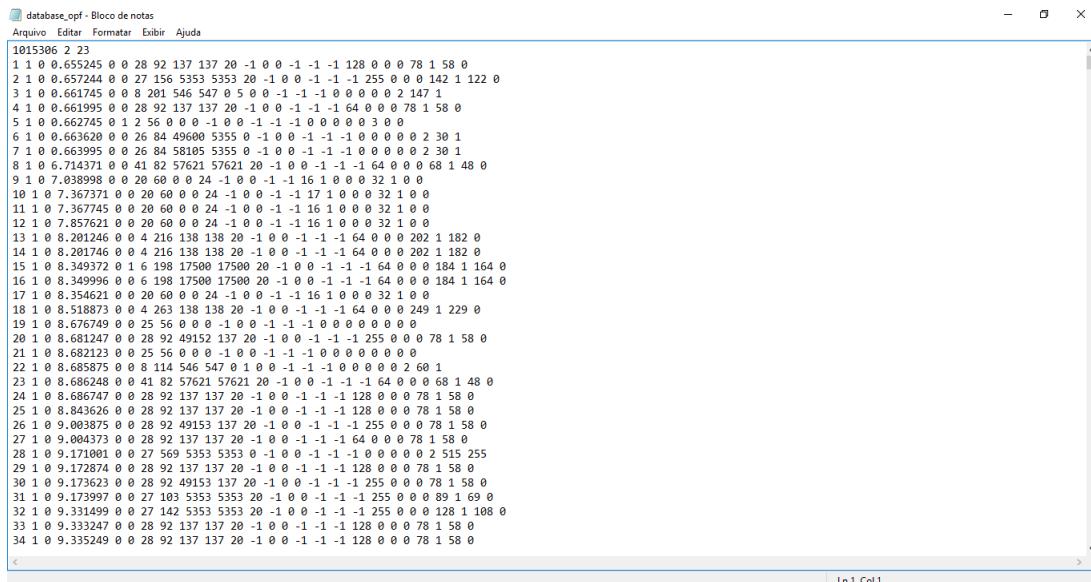
Tabela 7: Formato base de dados

<# número de amostras> <# número de classes> <# número de características>
<0> <rótulo> <característica 1 para a amostra 0> <característica 2 para a amostra 0> ...
...
<n-1> <rótulo> <característica 1 para a amostra n-1> <característica 2 para a amostra n-1> ...

Fonte: Adaptada de *LibOPF Wiki*⁹.

Um exemplo do formato da base de dados utilizada pelo OPF pode ser visto na Figura 27, no qual a primeira linha apresenta o número de amostras, o número de classes e o número de características, respectivamente, e as linhas subsequentes apresentam os dados das amostras.

Figura 27: Amostra da Base de Dados criada no formato entendido pelo OPF.



```

database.opf - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
1015306 2 23
1 1 0 0,655245 0 0 28 92 137 137 20 -1 0 0 -1 -1 128 0 0 0 78 1 58 0
2 1 0 0,657244 0 0 27 156 5353 5353 20 -1 0 0 -1 -1 255 0 0 0 142 1 122 0
3 1 0 0,661745 0 0 291 546 547 0 5 0 0 -1 -1 0 0 0 0 0 2 147 1
4 1 0 0,661995 0 0 28 92 137 137 20 -1 0 0 -1 -1 64 0 0 0 78 1 58 0
5 1 0 0,662745 0 1 2 56 0 0 0 -1 0 0 -1 -1 0 0 0 0 0 3 0 0
6 1 0 0,663620 0 0 26 84 49600 5355 0 -1 0 0 -1 -1 0 0 0 0 0 2 30 1
7 1 0 0,663995 0 0 28 84 58198 5355 0 -1 0 0 -1 -1 0 0 0 0 0 2 30 1
8 1 0 0,714371 0 0 41 82 57621 57621 20 -1 0 0 -1 -1 64 0 0 0 68 1 48 0
9 1 0 0,738998 0 0 28 69 57621 57621 20 -1 0 0 -1 -1 16 1 0 0 0 32 1 0 0
10 1 0 0,7367371 0 0 29 60 0 0 24 -1 0 0 -1 -1 17 1 0 0 0 32 1 0 0
11 1 0 0,7367745 0 0 29 60 0 0 24 -1 0 0 -1 -1 16 1 0 0 0 32 1 0 0
12 1 0 0,7857621 0 0 28 60 0 0 24 -1 0 0 -1 -1 16 1 0 0 0 32 1 0 0
13 1 0 0,8201246 0 0 4 216 138 138 20 -1 0 0 -1 -1 64 0 0 0 202 1 182 0
14 1 0 0,8201746 0 0 4 216 138 138 20 -1 0 0 -1 -1 64 0 0 0 202 1 182 0
15 1 0 0,8349372 0 1 6 198 17500 17500 20 -1 0 0 -1 -1 64 0 0 0 184 1 164 0
16 1 0 0,8349996 0 0 6 198 17500 17500 20 -1 0 0 -1 -1 64 0 0 0 184 1 164 0
17 1 0 0,8354621 0 0 29 60 0 0 24 -1 0 0 -1 -1 16 1 0 0 0 32 1 0 0
18 1 0 0,8358873 0 0 4 263 138 138 20 -1 0 0 -1 -1 64 0 0 0 249 1 229 0
19 1 0 0,8367649 0 0 25 56 0 0 0 -1 0 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
20 1 0 0,8681247 0 0 28 92 49152 137 20 -1 0 0 -1 -1 255 0 0 0 78 1 58 0
21 1 0 0,8682123 0 0 25 56 0 0 0 -1 0 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 1 0 0,8685875 0 0 114 546 547 0 1 0 0 -1 -1 0 0 0 0 0 2 68 1
23 1 0 0,8686248 0 0 41 82 57621 57621 20 -1 0 0 -1 -1 64 0 0 0 68 1 58 0
24 1 0 0,8686747 0 0 28 92 137 137 20 -1 0 0 -1 -1 128 0 0 0 78 1 58 0
25 1 0 0,8843626 0 0 28 92 137 137 20 -1 0 0 -1 -1 128 0 0 0 78 1 58 0
26 1 0 0,9003875 0 0 28 92 49153 137 20 -1 0 0 -1 -1 255 0 0 0 78 1 58 0
27 1 0 0,9004373 0 0 28 92 137 137 20 -1 0 0 -1 -1 64 0 0 0 78 1 58 0
28 1 0 0,9171081 0 0 27 569 5353 5353 0 -1 0 0 -1 -1 0 0 0 0 0 2 515 255
29 1 0 0,9172874 0 0 28 92 137 137 20 -1 0 0 -1 -1 128 0 0 0 78 1 58 0
30 1 0 0,9173623 0 0 28 92 49153 137 20 -1 0 0 -1 -1 255 0 0 0 78 1 58 0
31 1 0 0,9173997 0 0 27 103 5353 5353 20 -1 0 0 -1 -1 255 0 0 0 89 1 60 0
32 1 0 0,9331499 0 0 27 142 5353 5353 20 -1 0 0 -1 -1 255 0 0 0 128 1 108 0
33 1 0 0,9333247 0 0 28 92 137 137 20 -1 0 0 -1 -1 128 0 0 0 78 1 58 0
34 1 0 0,9335249 0 0 28 92 137 137 20 -1 0 0 -1 -1 128 0 0 0 78 1 58 0

```

Fonte: Elaborado pela autora.

3.4 Otimização

O algoritmo do PSO utilizado foi implementado em uma biblioteca, a LibOPT-plus, disponível em um repositório do *github*¹⁰. Essa versão foi escolhida por ser implementada em linguagem C, seguindo o padrão de implementação do OPF, além de facilitar a adaptação por parte da autora.

O algoritmo do PSO utilizado foi baseado no trabalho de Kennedy, Eberhart e Shi (2001), no qual além dos parâmetros apresentados na Equação 4 na página 51, o número de partículas (agentes) e o número máximo de iterações também são necessários, mas variam de acordo com o problema, dependendo do número de características do mesmo.

¹⁰ <https://github.com/jppbsi/LibOPT-plus>

Neste trabalho, a otimização tem como objetivo selecionar as características que melhor definem o problema, neste caso a definição do comportamento anômalo, ou seja, visa diminuir o número de características sem perda significativa da acurácia de classificação, gerando uma otimização do processo. A seleção de características utilizada foi baseada na *Swarmed Feature Selection* de Firpi e Goodman (2004), assim, um vetor armazena o estado das características no espaço binário, sendo 0 uma característica inativa e 1 ativa.

No algoritmo de seleção de características utilizado o espaço de busca do PSO é iniciado, a otimização é aplicada com n -dimensões, sendo n o número de características do problema, retornando partículas com valores reais. Em seguida é aplicado uma normalização no vetor de características de cada partícula para transformá-la em um valor binário (0 ou 1), sendo realizada através de uma função de transferência, que conforme definido por Firpi e Goodman (2004) é a função sigmoide. Após a normalização, o OPF é aplicado a fim de obter a acurácia de classificação da nova base de dados formada pelas características selecionadas. Essa acurácia é utilizada como função de aptidão no PSO, que refaz o procedimento no ambiente real, objetivando encontrar a melhor solução para o problema até o número máximo de iterações definido.

A união das técnicas de classificação e otimização ocorreram de duas formas: durante o processo de otimização e como avaliação da eficácia da otimização. No primeiro caso, o OPF foi utilizado como função de avaliação do processo de otimização, e no segundo, foi realizada uma classificação em cima da base de dados otimizada, a fim de avaliar sua efetividade.

3.5 Experimentos

Para a realização dos experimentos, foi utilizada uma das máquinas disponibilizadas pela universidade, sendo realizados em Sistema Operacional Ubuntu 15.04, processador Intel Xeon de 2.4Ghz e 128GB de memória RAM.

Os experimentos foram realizados de forma a analisar a acurácia de detecção antes e depois da otimização, com diferentes proporções de treinamento e teste com o objetivo de obter o melhor resultado. A fim de evitar uma classificação ineficiente, cada conjunto foi rodado 10 vezes, gerando uma média dos resultados. Inicialmente, a classificação foi rodada em diferentes porcentagens da base de dados original, conforme demonstrado pela Tabela 8.

Após uma avaliação inicial dos resultados, foi possível observar certa redundância na base de dados, ou seja, amostras muito semelhantes ou iguais. Em virtude disso, não é necessário aplicar a otimização na base completa, o que diminui o tempo de processamento. Assim, o PSO foi executado em 10% da base de dados, já que é possível obter resultados satisfatórios com um menor tempo de execução. O tempo de execução do PSO varia de acordo com os dados de entrada e o tamanho da base de dados. Como dados de entrada tem-se: número de partículas, número máximo de iterações (força a convergência), w ou força de

Tabela 8: Programação de execução do OPF na base original.

% da Base	% de Treino/Teste
10%	10/90
10%	30/70
10%	50/50
10%	70/30
<hr/>	
50%	10/90
50%	30/70
50%	50/50
50%	70/30
<hr/>	
100%	10/90
100%	30/70
100%	50/50
100%	70/30

Fonte: Elaborada pela autora.

inércia, c_1 e c_2 ou fatores de aprendizado e as constantes r_1 e r_2 , sendo que as duas últimas assumem um valor aleatório entre [0, 1].

O número de partículas varia de acordo com o número de características ou espaço de busca, sendo que um valor pequeno não será capaz abranger todo espaço, e um número muito alto vai superlotar. Assim, o mesmo é definido empiricamente, sendo definido como 15 para o problema de 23 características. O número máximo de iterações também é relativo, e influencia diretamente no tempo de execução junto com o número de agentes. Desta forma, foi definido o número máximo de iterações em 25.

A força de inércia ou w é definido empiricamente, sendo usado neste problema, como 0.7, um valor que não afeta demasiadamente o resultado, mas auxilia na iteração entre as partículas. As constantes c_1 e c_2 de acordo com Kennedy e Eberhart (1995), Kennedy, Eberhart e Shi (2001) e Firpi e Goodman (2004) devem ter valores aproximados de 2, assim ambos foram definidos como 1.7.

No processo de seleção de características, é necessário utilizar um conjunto de treino e de validação, para não comprometer o conjunto de teste. Então, para isso a porcentagem de treino foi dividida igualitariamente em treino e validação, de forma que quando ocorresse a classificação final, através do OPF, a mesma fosse feita em cima das porcentagens inicialmente definidas, unificando novamente as partes de treino e validação. Tal divisão foi necessária, pois segundo Hastie, Tibshirani e Friedman (2001, p.222) "o conjunto de treinamento é usado para ajustar os modelos; o conjunto de validação é usado para estimar erro de predição para a seleção de modelo; o conjunto de teste é usado para avaliação do erro de generalização do modelo final escolhido".

A partir dessas configurações, o PSO foi rodado em todas as porcentagens de treino e teste utilizadas no OPF: 10/90, 30/70, 50/50 e 70/30, 10 vezes para cada. Sendo que após cada iteração o OPF foi rodado para verificar a acurácia do processo. Portanto, a partir desses

resultados foi feita uma análise a respeito da eficácia da técnica de otimização empregada em relação aos resultados da base de dados original.

3.6 Resultados

Conforme descrito na Seção 3.5, os experimentos realizados foram divididos em duas categorias: classificação (pura) e classificação otimizada (otimização). Os resultados apresentados na presente seção tem por objetivo comparar o comportamento de ambas categorias, a fim de validar a utilização das técnicas propostas neste trabalho.

Inicialmente, foi feita uma avaliação da base de dados original através da classificação pura, seguindo a descrição da Tabela 8; obtendo os resultados para cada configuração de treino/teste por meio dos valores médios das iterações. A Tabela 9 apresenta os resultados para cada porcentagem da base de dados utilizada: 10%, 50% e 100%, detalhando também as configurações de treino/teste. Note que os melhores resultados estão em negrito de acordo com o teste de Wilcoxon (1945).

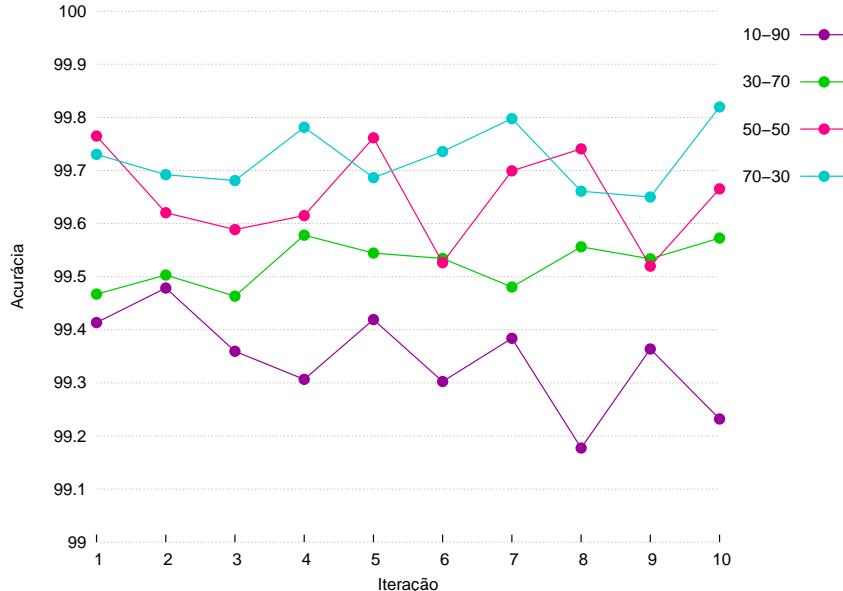
Tabela 9: Média dos resultados da classificação pura da base de dados.

Base de dados (%)	% Treino/% Teste	Tempo Treino (s)	Tempo Teste (s)	Acurácia (%)
10%				
	10/90	10.06 ± 1.47	51.75 ± 9.52	99.34 ± 0.09
	30/70	102.03 ± 11.21	238.49 ± 28.02	99.52 ± 0.04
	50/50	300.57 ± 18.61	325.81 ± 20.14	99.65 ± 0.09
	70/30	602.78 ± 20.12	266.47 ± 14.17	99.72 ± 0.06
50%				
	10/90	299.16 ± 15.27	2851.99 ± 88.77	99.61 ± 0.01
	30/70	2480.29 ± 227.62	6534.88 ± 317.05	99.76 ± 0.01
	50/50	6601.03 ± 676.26	7503.06 ± 527.27	99.81 ± 0.02
	70/30	12628.76 ± 1309.16	6628.84 ± 329.90	99.86 ± 0.01
100%				
	10/90	1055.86 ± 85.45	10492.93 ± 588.60	99.70 ± 0.02
	30/70	8912.15 ± 876.38	24599.03 ± 1381.76	99.84 ± 0.01
	50/50	24505.70 ± 2188.45	29427.51 ± 758.88	99.87 ± 0.01
	70/30	47112.92 ± 1737.22	24600.99 ± 545.35	99.88 ± 0.01

Fonte: Elaborada pela autora.

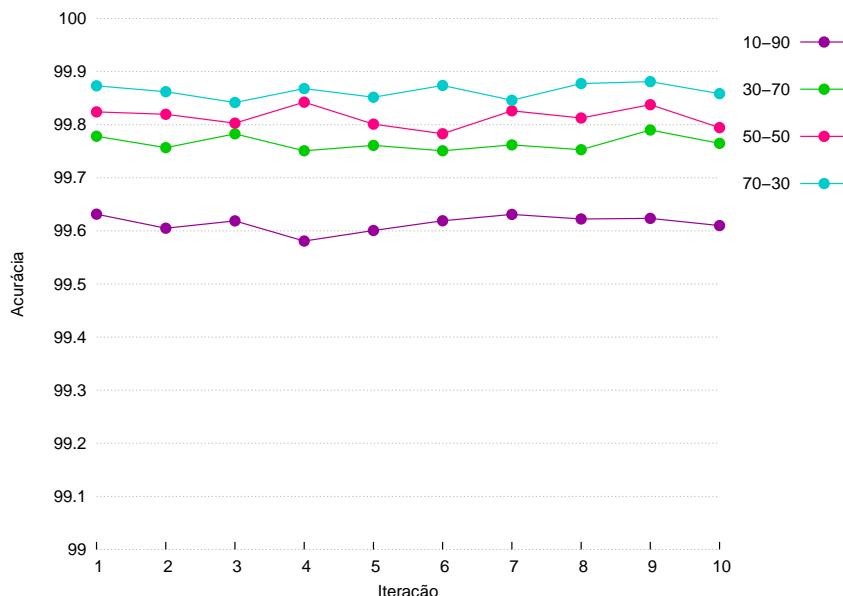
A Figuras 28, 29 e 30 apresentam o comportamento da classificação pura de cada porcentagem da base de dados a cada iteração: 10%, 50% e 100%, respectivamente.

Figura 28: Gráfico da acurácia de classificação pura x iteração em 10% da base de dados.



Fonte: Elaborado pela autora.

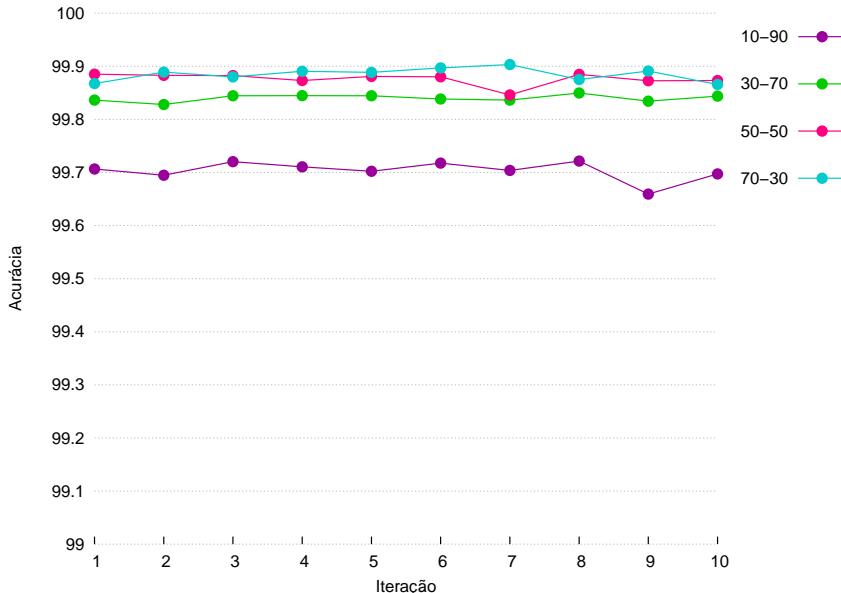
Figura 29: Gráfico da acurácia de classificação pura x iteração em 50% da base de dados.



Fonte: Elaborado pela autora.

O comportamento demonstrado pelas Figuras 28, 29 e 30 indica que quanto maior a base de dados melhor será a acurácia de classificação, apresentado também pela última coluna da Tabela 9. No entanto, através de uma análise relativa à porcentagem da base de dados, pode-se perceber que a diferença da acurácia média torna-se mínima (0.26 % - diferença de acurácia entre 10% e 100%) em proporção à diferença do tempo médio de classificação (20142.8 s - diferença do tempo de treinamento entre 10% e 100%), como demonstra a

Figura 30: Gráfico da acurácia de classificação pura x iteração em 100% da base de dados.



Fonte: Elaborado pela autora.

Tabela 10. Assim, pode-se dizer que há uma redundância dos dados.

Tabela 10: Média dos resultados da classificação pura pela porcentagem da base de dados.

% Base de Dados	Tempo Treino (s)	Tempo Teste (s)	Acurácia (%)
10%	253.86	220.63	99.56
50%	5502.31	5879.69	99.76
100%	20396.66	22280.12	99.82

Fonte: Elaborada pela autora.

Deste modo, a fim de diminuir o tempo de execução dos experimentos, conforme descrito na Seção 3.5, a otimização foi realizada em apenas 10% da base de dados, sem que ocorresse uma perda significativa de acurácia no processo proposto por este trabalho. Para poder realizar a comparação, a otimização também utilizou as mesmas configurações de treino/teste, conforme apresentado pela Tabela 11.

Tabela 11: Média dos resultados da otimização.

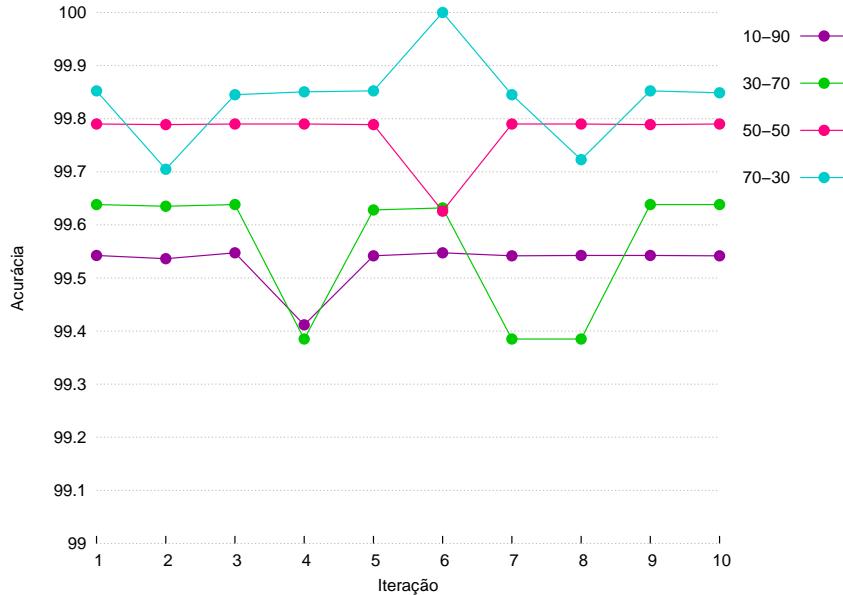
% Treino/% Teste	# de Características	Tempo Otimização (s)	Tempo Classificação (s)	Acurácia (%)
10/90	14.30	1613.98 ± 72.75	47.89 ± 2.66	99.53 ± 0.04
30/70	14.40	13808.09 ± 752.54	163.66 ± 26.91	99.56 ± 0.11
50/50	15.30	46412.15 ± 3116.70	234.15 ± 18.15	99.77 ± 0.05
70/30	15.60	99682.64 ± 4037.70	209.58 ± 13.26	99.84 ± 0.08

Fonte: Elaborada pela autora.

A Figura 31 exibe a acurácia de classificação da base otimizada por iteração, na qual é possível perceber uma melhora não apenas dos valores médios apresentados pela Tabela 11,

mas também de cada iteração do processo.

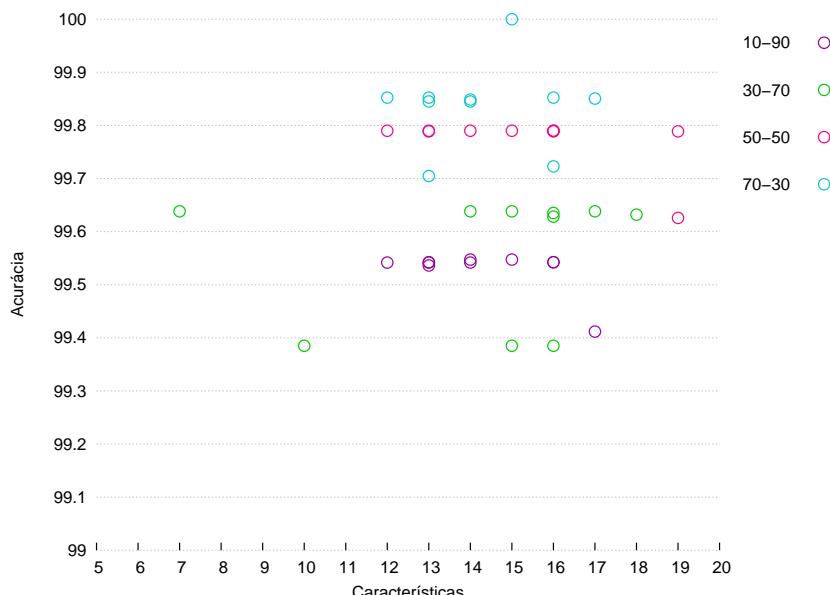
Figura 31: Gráfico da acurácia x iteração do processo de otimização.



Fonte: Elaborado pela autora.

Conforme mencionado na Seção 3.5, o processo de otimização teve por objetivo encontrar um menor conjunto de características, sem a perda significante de acurácia. Destarte, a Figura 32 aponta o número de características selecionadas e suas respectivas acuráncias durante o processo de otimização.

Figura 32: Gráfico da acurácia x quantidade de características selecionadas durante o processo de otimização.



Fonte: Elaborado pela autora.

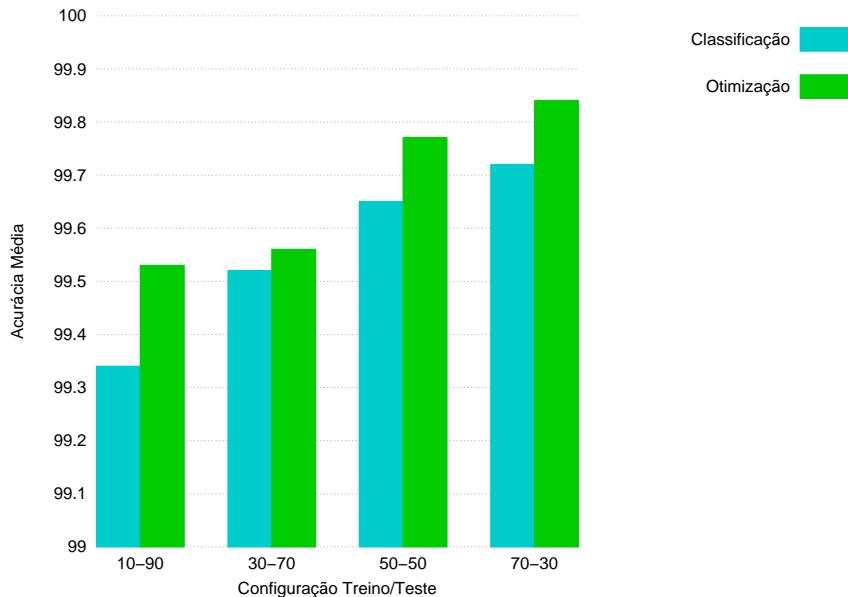
A Tabela 12 apresenta a comparação dos resultados médios de classificação pura e otimizada, detalhando as configurações de treino/teste, comportamento também ilustrado pela Figura 33.

Tabela 12: Comparação dos resultados da classificação com a otimização.

% Treino/% Teste	Acurácia Classificação (%)	Acurácia Otimização (%)
10/90	99.34 ± 0.09	99.53 ± 0.04
30/70	99.52 ± 0.04	99.56 ± 0.11
50/50	99.65 ± 0.09	99.77 ± 0.05
70/30	99.72 ± 0.06	99.84 ± 0.08

Fonte: Elaborada pela autora.

Figura 33: Gráfico de comparação de acurácia da classificação com a otimização.



Fonte: Elaborado pela autora.

Baseado nas informações apresentadas, pôde-se perceber que, com um menor número de características encontradas, ao invés de obter uma acurácia de classificação menor do que a obtida pela base de dados original, obteve-se uma melhora de desempenho, demonstrando assim que o comportamento foi melhor do que o esperado para o processo de otimização.

Conclui-se então, que a proposta de utilização da otimização para o processo de detecção de anomalias em redes de computadores foi eficaz e obteve resultados superiores ao esperado, validando, desta forma, a técnica estudada no presente trabalho; além de fortalecer o estudo de utilização de técnicas de aprendizado de máquina na área de segurança de redes de computadores, uma vez que a mesma requer melhorias constantes.

3.7 Interface Gráfica

Com o intuito de exemplificar, de forma mais visual, as técnicas aplicadas no trabalho, foi criado um *software* que auxilia o usuário a visualizar o procedimento, sendo desenvolvido em linguagem C# para ambiente *linux*. Seu funcionamento é simples, criando apenas um *front-end*¹¹ para o processamento realizado em *back-end*¹² através do terminal de comando. O projeto do *software* está disponível em um repositório do *github*¹³, sendo público para quem tem interesse em utilizá-lo ou estudá-lo.

Entretanto, para o correto funcionamento da interface há alguns pré-requisitos, tais como, o compilador de C# MonoDevelop integrado com o .NET Framework (necessária versão 4.5.2), que pode ser obtido através do comando:

Código: Comando de instalação Mono.

```
# sudo apt-get install mono-complete
```

Também são necessárias as bibliotecas: LibOPF¹⁴, LibOPT Plus¹⁵, LibDEEP¹⁶ e LibDEV¹⁷, que devem ser baixadas e instaladas de acordo com as instruções presentes em suas respectivas *wikis*¹⁸.

Inicialmente, o usuário deve fornecer as informações básicas para que o processo seja realizado, tais como, o arquivo da base de dados, operação a ser realizada e porcentagem de utilização da base para as etapas de treinamento e de teste, conforme ilustra a Figura 34. O *software* conta com duas possíveis opções de procedimento: apenas a classificação (OPF) e a classificação otimizada (OPF + PSO).

¹¹ *Front-end* é a parte do sistema visível para o usuário, tal como, a interface.

¹² *Back-end* é a parte do sistema invisível para o usuário, tal como, o processamento de arquivos ou dados.

¹³ <https://github.com;brurubio/TCCInterface>

¹⁴ <https://github.com/jppbsi/LibOPF>

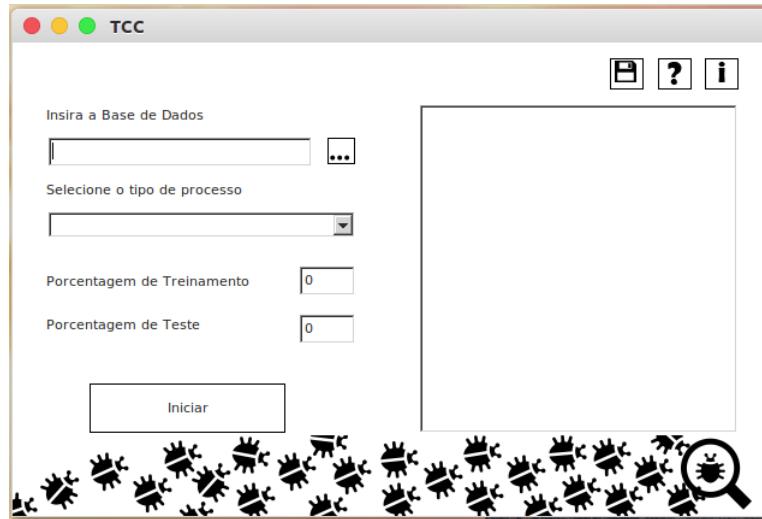
¹⁵ <https://github.com/jppbsi/LibOPT-plus>

¹⁶ <https://github.com/jppbsi/LibDEEP>

¹⁷ <https://github.com/jppbsi/LibDEV>

¹⁸ *GitHub Wiki* é um lugar (página) em um repositório com o objetivo de compartilhar conteúdo de longa duração sobre o projeto, tais como a forma de usá-lo, como ele foi projetado, dentre outros.

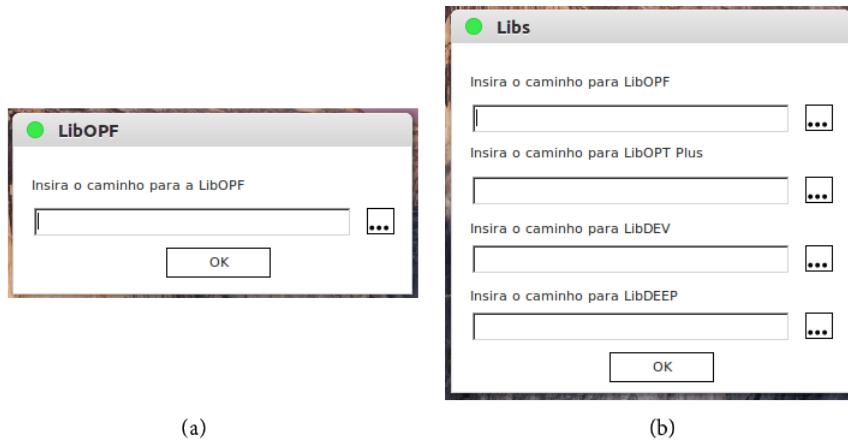
Figura 34: Janela principal da interface.



Fonte: Elaborado pela autora.

O botão Iniciar é responsável por acionar o procedimento, no entanto, é necessário informar o caminho das bibliotecas instaladas na janela seguinte. Para o procedimento de classificação, apenas o caminho onde a LibOPF está instalada é necessário, conforme ilustra a Figura 35.a. Já no procedimento de classificação otimizada, os caminhos da LibOPF, LibOPT Plus, LibDEV e LibDEEP são necessários, conforme ilustrado pela Figura 35.b.

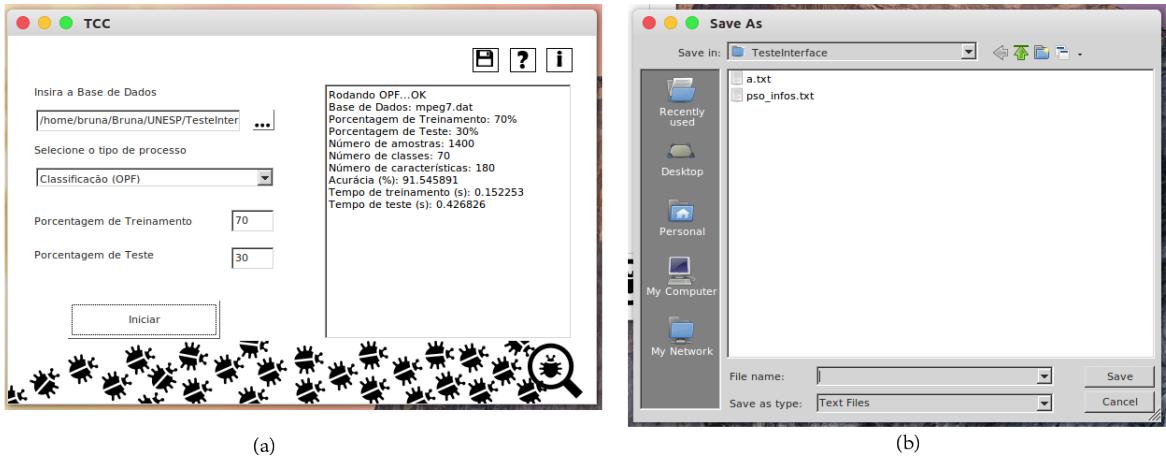
Figura 35: Caminho para as pastas de instalação das Bibliotecas.



Fonte: Elaborado pela autora. (a) Caminho da LibOPF; (b) caminho das LibOPF, OPT-Plus, DEV e DEEP.

Os resultados serão exibidos à direita, e podem ser salvos em um arquivo texto através do botão Salvar, onde é possível selecionar o lugar em disco e o nome do arquivo a ser salvo. A Figura 36.a apresenta os resultados de um processo de classificação pura e a Figura 36.b mostra a opção de salvar.

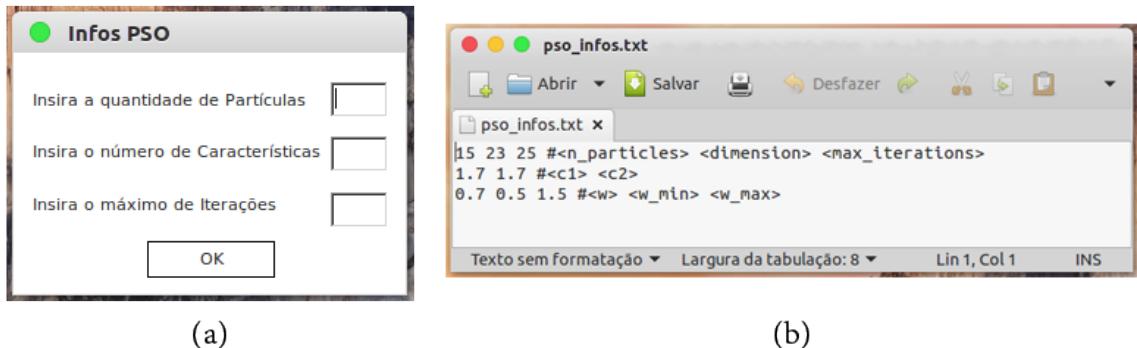
Figura 36: Resultados da operação Classificação e Opção de salvar resultados.



Fonte: Elaborado pela autora. (a) Resultados de operação Classificação (OPF); (b) Opção de salvar os resultados.

Para o processo OPF + PSO, são necessárias informações adicionais, tais como, o número de partículas, o número de características do problema e o número de iterações máxima, conforme apresenta a Figura 37.a, onde as informações inseridas são armazenadas em um arquivo, ilustrado pela Figura 37.b, que será utilizado durante o processamento.

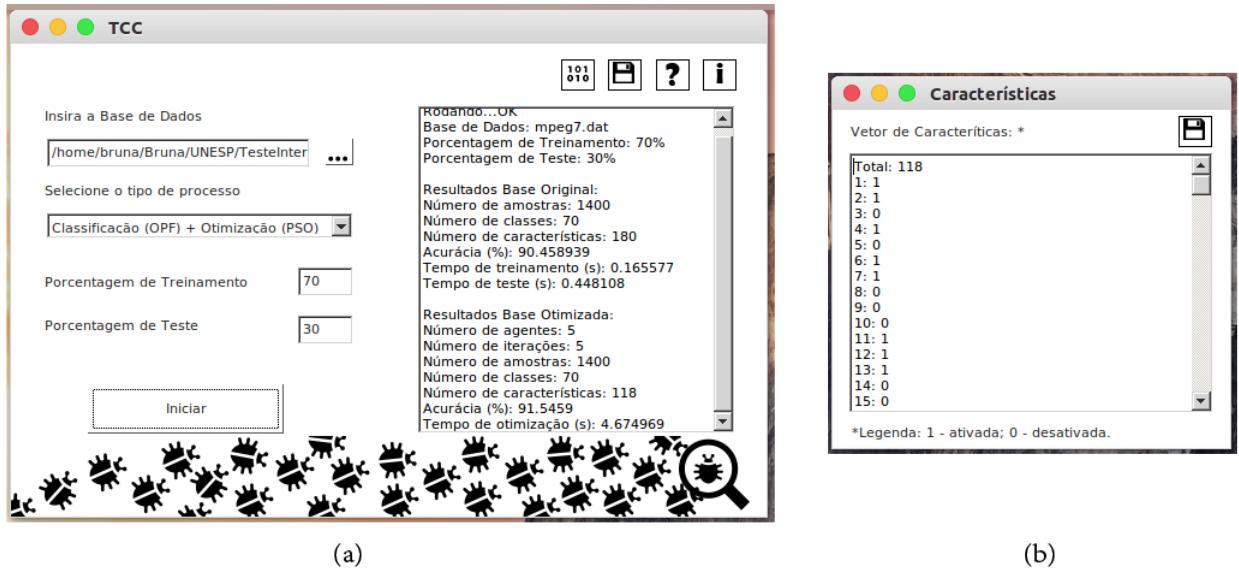
Figura 37: Informações adicionais para o PSO.



Fonte: Elaborado pela autora. (a) Janela de Inserção de Informações; (b) Arquivo gerado com as informações adicionais do PSO.

Após a finalização do processo OPF + PSO, um novo botão é mostrado na janela principal: Melhores Características, no qual é possível ver quais características foram selecionadas no processo de otimização, sendo também possível salvar os resultados através do botão Salvar. A Figura 38.a mostra os resultados obtidos através de um processo de classificação otimizada e a Figura 38.b apresenta as melhores características obtidas.

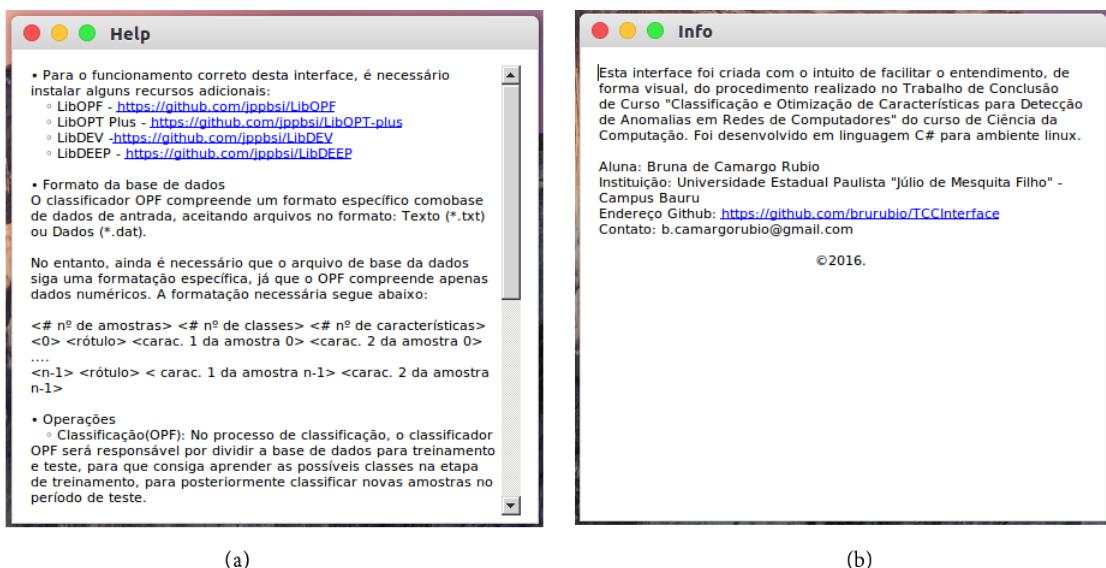
Figura 38: Resultados da operação Classificação + Otimização e melhores características.



Fonte: Elaborado pela autora. (a) Resultados da operação Classificação (OPF) + Otimização (PSO); (b) Melhores características selecionadas.

Na janela principal existe ainda dois botões informativos: Ajuda e Informações. O botão de Ajuda apresenta uma nova janela com as informações necessárias para utilização da interface (Figura 39.a), já o botão de Informações apresenta uma nova janela com as informações referentes ao trabalho vinculado, além de informações de contato (Figura 39.b).

Figura 39: Janelas auxiliares: Ajuda e Informações



Fonte: Elaborado pela autora. (a) Janela de ajuda; (b) Janela de Informações.

O *software* criado contém uma interface de utilização simples e intuitiva atendendo o objetivo de ilustrar graficamente o processo realizado no presente no trabalho de conclusão de curso. No entanto, é importante ressaltar que por não ser o foco do projeto, o mesmo é tratado como um protótipo simplificado com operações reduzidas, que podem ser aprimoradas posteriormente.

4 Conclusão

A segurança de redes tornou-se essencial devido à evolução das redes de computadores, em especial da *internet* o que acarretou na centralização das informações e na facilidade de acesso por pessoas autorizadas ou não.

Em virtude disto, empresas e pesquisadores investem em abordagens inovadoras a fim de criar ferramentas mais eficazes de detecção. Dentre as ferramentas existentes, uma das mais conhecidas pelos usuários em geral são os sistemas de detecção de intrusão, o qual contempla a abordagem por detecção de anomalias. Nesta abordagem, além do desafio da variedade e complexidade dos ataques, ainda existe a dificuldade da definição do que é ou não uma anomalia. No entanto, tem como vantagem uma eficácia maior na detecção de ataques desconhecidos.

Uma área de pesquisa em ascensão utilizada para buscar uma solução deste problema é a inteligência artificial. Desta forma, este trabalho propôs uma abordagem baseada em aprendizado de máquina, através da utilização de um classificador de padrões que seja capaz de aprender a definição de anomalia e não anomalia, aliado a um processo de otimização meta-heurística que melhore o processo de detecção através da seleção de características necessárias para tal, tornando o processo ainda mais vantajoso.

Estudo este mostrou-se eficaz através dos resultados obtidos que apresentaram a melhora de desempenho na detecção após o processo de otimização. Os resultados obtidos também fortalecem pesquisas sobre técnicas de aprendizado de máquina aplicadas na área de segurança de redes, considerando a alta taxa de acurácia obtida por este estudo. Impulsionando também o estudo na área de segurança de redes, visto que novas abordagens tem se mostrado eficazes e que futuramente, se corretamente desenvolvido, possa alcançar precisão máxima e ser realizado em tempo real.

Portanto, pode-se concluir que este trabalho tenha agregado de forma significativa para a área em questão. Contribuindo à diversas áreas que cercam o aprendizado na área da ciência da computação, à formação do aluno, vislumbrando um exemplo prático das inúmeras áreas de atuação profissional na área da computação.

Referências

- AL-KASASSBEH, M.; ADDA, M. Network fault detection with wiener filter-based agent. *Journal of Network and Computer Applications*, v. 32, n. 4, p. 824 – 833, 2009. ISSN 1084-8045.
- ANDERSON, D.; LUNT, T.; JAVITZ, H.; TAMARU, A.; VALDES, A. Detecting unusual program behavior using the statistical components of NIDES. may 1995.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. *Swarm intelligence: from natural to artificial systems*. [S.I.]: Oxford university press, 1999.
- CAI, C.; PAN, H.; CHENG, G. Fusion of bvm and elm for anomaly detection in computer networks. In: *Computer Science Service System (CSSS), 2012 International Conference on*. [S.I.: s.n.], 2012. p. 1957–1960.
- CAMPELLO, R. S.; WEBER, R. F. Sistemas de detecção de intrusão. In: *Livro Texto dos Minicursos: Simpósio Brasileiro de Redes de Computadores*. [S.I.: s.n.], 2001. p. 1–43.
- CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 41, n. 3, p. 15:1–15:58, jul 2009. ISSN 0360-0300.
- CHEBROLU, S.; ABRAHAM, A.; THOMAS, J. P. Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, v. 24, n. 4, p. 295 – 307, 2005. ISSN 0167-4048.
- CLINCY, V. A.; ABU-HALAWEH, N. A taxonomy of free network sniffer for teaching and research. *J. Comput. Sci. Coll.*, Consortium for Computing Sciences in Colleges, USA, v. 21, n. 1, p. 64–75, out. 2005. ISSN 1937-4771.
- COLE, E. *Network Security Bible*. 2. ed. [S.I.]: Wiley Publishing, 2009. ISBN 0470502495, 9780470502495.
- CORTES, C.; VAPNIK, V. Support-vector networks. In: *Machine Learning*. [S.I.: s.n.], 1995. p. 273–297.
- CREPINSEK, M.; LIU, S.-H.; MERNIK, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, ACM, New York, NY, USA, v. 45, n. 3, p. 35:1–35:33, jul. 2013. ISSN 0360-0300.
- CRISTIANINI, N.; SHAWE-TAYLOR, J. *An introduction to support vector machines and other kernel-based learning methods*. [S.I.]: Cambridge university press, 2000. ISBN 0521780195.
- DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. *IEEE Computational Intelligence Magazine*, v. 1, n. 4, p. 28–39, Nov 2006. ISSN 1556-603X.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. 2. ed. [S.I.]: Wiley-Interscience, 2000. ISBN 0471056693.
- DUTTON, D. M.; CONROY, G. V. A review of machine learning. *Knowl. Eng. Rev.*, Cambridge University Press, New York, NY, USA, v. 12, n. 4, p. 341–367, dez. 1997. ISSN 0269-8889.

- EIBEN, A. E.; SCHIPPERS, C. A. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, v. 35, p. 35–50, 1998.
- ELLIS, D. Worm anatomy and model. In: *Proceedings of the 2003 ACM Workshop on Rapid Malcode*. New York, NY, USA: ACM, 2003. (WORM '03), p. 42–50. ISBN 1-58113-785-0.
- FIRPI, H. A.; GOODMAN, E. Swarm feature selection. In: *33rd Applied Imagery Pattern Recognition Workshop*. Washington, DC, USA: IEEE Computer Society, 2004. p. 112–118. ISBN 0-7695-2250-5.
- FOROUZAN, B. *Comunicação de Dados e Redes de Computadores*. 4. ed. [S.I.]: McGraw Hill Brasil, 2009. ISBN 9788563308474.
- GARCÍA-TEODORO, P.; DÍAZ-VERDEJO, J.; MACIÁ-FERNÁNDEZ, G.; VÁZQUEZ, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, v. 28, n. 1–2, p. 18 – 28, 2009. ISSN 0167-4048.
- GUANGMIN, L. Modeling unknown web attacks in network anomaly detection. In: *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*. [S.I.: s.n.], 2008. v. 2, p. 112–116.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning*. [S.I.]: Springer series in statistics Springer, Berlin, 2001. v. 1.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501.
- JUNG, J.; KRISHNAMURTHY, B.; RABINOVICH, M. Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites. In: *Proceedings of the 11th International Conference on World Wide Web*. New York, NY, USA: ACM, 2002. (WWW '02), p. 293–304. ISBN 1-58113-449-5.
- KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, Springer, v. 39, n. 3, p. 459–471, 2007.
- KENDALL, K. *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*. Dissertação (Mestrado) — Massachusetts Institute of Technology, 1998.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. [S.I.: s.n.], 1995. v. 4, p. 1942–1948 vol.4.
- KENNEDY, J.; EBERHART, R. C.; SHI, Y. *Swarm intelligence*. [S.I.]: Morgan Kaufmann, 2001.
- KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. In: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 3–24. ISBN 978-1-58603-780-2.
- KUROSE, J.; ROSS, K. *Redes de Computadores e a Internet: uma abordagem top-down*. 5. ed. [S.I.]: Pearson Education Br, 2010. ISBN 9788588639973.

LEINER, B. M.; CERF, V. G.; CLARK, D. D.; KAHN, R. E.; KLEINROCK, L.; LYNCH, D. C.; POSTEL, J.; ROBERTS, L. G.; WOLFF, S. A brief history of the internet. *ACM SIGCOMM Computer Communication Review*, ACM, v. 39, n. 5, p. 22–31, 2009.

LöF, A.; NELSON, R. Comparing anomaly detection methods in computer networks. In: *Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on*. [S.I.: s.n.], 2010. p. 7–10.

LI, L.; LEE, G. Ddos attack detection and wavelets. In: *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*. [S.I.: s.n.], 2003. p. 421–427. ISSN 1095-2055.

LINCOLN LABORATORY. *Intrusion Detection Attacks Database*. s.d. Disponível em <<https://www.ll.mit.edu/ideval/docs/attackDB.html>>. Acessado em 19 de Junho de 2016.

LUNT, T. F.; JAGANNATHAN, R. A prototype real-time intrusion-detection expert system. In: *Proceedings of the 1988 IEEE Conference on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 1988. (SP'88), p. 59–66. ISBN 0-8186-0850-1.

MAFRA, P. M.; FRAGA, J. da S.; MOLL, V.; SANTIN, A. O. Polvo-iids: Um sistema de detecção de intrusão inteligente baseado em anomalias. In: *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*. [S.I.: s.n.], 2008. p. 61–72.

MANIKOPOULOS, C.; PAPAVASSILIOU, S. Network intrusion and fault detection: a statistical anomaly approach. *IEEE Communications Magazine*, v. 40, n. 10, p. 76–82, Oct 2002. ISSN 0163-6804.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

MOTA FILHO, J. E. *Guia Análise de tráfego em redes TCP/IP com tcpdump e WinDump*. 2013. Disponível em <http://eriberto.pro.br/files/guia_tcpdump.pdf>. Acessado em 07 de Abril de 2016.

MUDZINGWA, D.; AGRAWAL, R. A study of methodologies used in intrusion detection and prevention systems (idps). In: *Southeastcon, 2012 Proceedings of IEEE*. [S.I.: s.n.], 2012. p. 1–6. ISSN 1091-0050.

MURTHY, S. K. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Min. Knowl. Discov.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, n. 4, p. 345–389, dez. 1998. ISSN 1384-5810.

ORR, M. J. *Introduction to Radial Basis Function Networks*. Centre for Cognitive Science, University of Edinburgh, 1996. Disponível em <<http://www.anc.ed.ac.uk/rbf/intro/intro.html>>. Acessado em 02 de Abril de 2016.

PAPA, J.; FALCÃO, A.; SUZUKI, C. *LibOPF: A library for the design of optimum-path forest classifiers*. [S.I.], 2015.

PAPA, J. P.; FALCÃO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, Wiley Subscription Services, Inc., A Wiley Company, v. 19, n. 2, p. 120–131, 2009. ISSN 1098-1098.

PATCHA, A.; PARK, J.-M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, v. 51, n. 12, p. 3448 – 3470, 2007. ISSN 1389-1286.

PEREIRA, C. R. *Detecção de Intrusão em Redes de Computadores Utilizando Floresta de Caminhos Ótimos*. Dissertação (Mestrado) — Universidade Estadual Paulista, São José do Rio Preto, São Paulo, 2012.

PERLIN, T.; NUNES, R. C.; KOZAKEVICIUS, A. d. J. Detecção de anomalias em redes de computadores e o uso de wavelets. *Revista Brasileira de Computação Aplicada*, v. 3, n. 1, p. 2–15, 2011.

SECURELIST. *History of malicious programs*. 2013. Disponível em <<https://securelist.com/threats/history-of-malicious-programs/>>. Acessado em 23 de Março de 2016.

SELVAKANI, S.; RAJESH, R. Genetic algorithm for framing rules for intrusion detection. *IJCNS International Journal of Computer Science and Network Security*, v. 7, n. 1, p. 285–290, 2007.

SELVI, V.; UMARANI, R. Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications*, v. 5, n. 4, p. 1–6, 2010.

SERAPIÃO, A. B. d. S. Fundamentos de otimização por inteligência de enxames: uma visão geral. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 20, n. 3, p. 271–304, 2009.

SMAHA, S. E. Haystack: an intrusion detection system. In: *Aerospace Computer Security Applications Conference, 1988., Fourth*. [S.I.: s.n.], 1988. p. 37–44.

STALLINGS, W. *Criptografia e segurança de redes: princípios e práticas*. 4. ed. [S.I.]: PRENTICE HALL BRASIL, 2008. ISBN 9788576051190.

STALLINGS, W.; BROWN, L. *Segurança de Computadores*. 2. ed. [S.I.]: Elsevier Brasil, 2014. ISBN 9788535264500.

TANENBAUM, A. S.; WETHERALL, D. *Redes de computadores*. 5. ed. [S.I.]: Pearson Prentice Hall, (SP), 2011. ISBN 9789702601623.

TCPDUMP. 2016. Disponível em <<http://www.tcpdump.org/manpages/tcpdump.1.html>>. Acessado em 14 de Março de 2016.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, v. 1, n. 6, p. 80–83, 1945.

WINSTON, P. H.; PENDERGAST, K. A. Artificial intelligence: A perspective! In: *The AI Business: Commercial Uses of Artificial Intelligence*. [S.I.]: MIT Press, 1984. p. 1–12.

WORLDWIDEWEB FOUNDATION. *History of the Web*. s.d. Disponível em <<http://webfoundation.org/about/vision/history-of-the-web/>>. Acessado em 20 de Abril de 2016.

WU, S. X.; BANZHAF, W. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, v. 10, n. 1, p. 1 – 35, 2010. ISSN 1568-4946.

YEUNG, D.-Y.; DING, Y. Host-based intrusion detection using dynamic and static behavioral models. *Pattern recognition*, Elsevier, v. 36, n. 1, p. 229–243, 2003.

Apêndices

APÊNDICE A – Codificação no Processo de Rotulação

Conforme mencionado na Seção 3.3, o classificador utiliza uma base de dados numérica como entrada, assim foi necessário transformar os dados alfanuméricos. Para manter o controle e padrão sobre os dados, tabelas com os códigos numéricos foram criadas. Desta forma, a base de dados mantém suas informações, podendo ser utilizada de forma numérica. Para se obter quaisquer informações a respeito dos dados alfanuméricos, as tabelas podem ser consultadas. A Tabela 13 apresenta a codificação das anomalias injetadas, a Tabela 14 os dados do campo 'Tipos IGMP', as mensagens DHCPv6 foram codificadas de acordo com a Tabela 15, a Tabela 16 indica a codificação dos tipos dos pacotes, já os protocolos presentes na base de dados foram codificados segundo a Tabela 17.

Tabela 13: Codificação dos Tipos de Anomalias.

Nome	Código	Qtde
Normal	0	913723
<i>Flooding</i>	1	12206
<i>Apache2</i>	2	12714
<i>Land</i>	3	12628
<i>Ping Of Death</i>	4	11334
<i>Selfping</i>	5	12593
<i>Mscan</i>	6	12219
<i>Crashis</i>	7	14778
<i>Ipsweep</i>	8	13111

Fonte: Elaborado pela autora.

Tabela 14: Codificação dos Tipos IGMP.

Protocolo	Tipo IGMP	IANA ¹	Código
-	Vazio	-	-1
IGMPv1	<i>Membership Report</i>	0x12	12
IGMPv2	<i>Membership Query</i>	0x11	11
IGMPv2	<i>Membership Report</i>	0x16	16
IGMPv2	<i>Leave Group</i>	0x17	17
IGMPv3	<i>Membership Report</i>	0x22	22

Fonte: Elaborado pela autora.

Tabela 15: Codificação das mensagens DHCPv6.

Tipos de mensagem DHCPv6	Código (IANA)
Vazio	- (-1) ²
Solicit	1
Advertise	2
Request	3
Confirm	4
Renew	5
Rebind	6
Reply	7
Release	8
Information-request	11

Fonte: Elaborado pela autora.

Tabela 16: Codificação dos Tipos de Pacotes.

Tipos	Código
Vazio	0
IPv4	1
IPv6	2
ARP	3
802.1X Authentication	4

Fonte: Elaborado pela autora.

¹ IANA - *Internet Assigned Numbers Authority* ou Autoridade para Atribuição de Números da Internet.
Disponível em: <http://www.iana.org/>

² Na definição do IANA, o código 0 é considerado reservado, por isso será utilizado -1, para que o sentido real não seja alterado.

Tabela 17: Codificação dos Protocolos.

Protocolo	Código
ALLJOYN-NS	1
ARP	2
BJNP	3
BROWSER	4
CUPS	5
DB-LSP-DISC	6
DHCP	7
DHCPv6	8
DNS	9
EAP	10
EAPOL	11
Elasticsearch	12
ESP	13
FTP	14
HIP	15
HPEXT	16
HTTP	17
ICMP	18
IGMPv1	19
IGMPv2	20
IGMPv3	21
IMAP	22
IPv4	23
IPv6	24
LLC	25
LLMNR	26
MDNS	27
NBNS	28
NetBIOS	29
OSPF	30
POP3	31
QUIC	32
SMB Mailslot	33
SNMP	34
SRVLOC	35
SSDP	36
TCP	37
TELNET	38
TLSv1	39
TLSv1.2	40
UDP	41
WOL	42
XID	43

Fonte: Elaborado pela autora.