Apresentação Banca

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

Bruna de Camargo Rubio

UNESP Bauru
Universidade Estadual Paulista
"Júlio de Mesquita Filho"
Faculdade de Ciências - Campus Bauru

Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

2016

Sumário

Introdução

A crescente quantidade e variedade de dados que trafegam nas redes de computadores atualmente, principalmente na Internet, requer uma **segurança de redes** eficaz e aprimorada.

No entanto, as ferramentas existentes nem sempre conseguem acompanhar a complexidade dos ataques criados.

Informação Cert.br[1]:

Em 2014 ocorreu um aumento de 197% de incidentes de segurança em redes conectadas à *internet* em relação ao ano de 2013.

Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil.

Estudos na área estão sendo fomentados por empresas e pesquisadores com o intuito de aprimorar tais ferramentas.

Em ferramentas como os **Sistemas de Detecção de Intrusões** (SDIs), estão sendo aplicadas técnicas de **Inteligência Artificial** (IA), que tem por objetivo imitar o comportamento da mente humana, a fim de aprimorá-las. Utilizando abordagens como Mineração de Dados (LI; LEE, 2003), Redes Neurais Artificiais (HAYKIN, 1998), Máquinas de Vetores de Suporte (CORTES; VAPNIK, 1995) e outros.

A fim de encontrar uma abordagem que seja eficiente no processo de detecção de intrusões, este trabalho propôs a utilização de um **classificador de padrões** aliado à uma seleção de características através de uma **otimização meta-heurística**.

No entanto, outro problema enfrentado é a **escassa diversidade de dados** disponíveis para análise, gerando resultados desgastados pela utilização das mesmas bases de dados (KDDCup, NSL-KDD, ICSX e DARPA).

A fim de solucionar este problema, foi criada uma nova base de dados: a **uneSPY**, com pacotes captados na rede da universidade UNESP Bauru.

Conceitos Fundamentais

Anomalia:

Pode ser definida como **algo raro** que difere de um comportamento definido como normal.

Classificação de Padrões:

É o método de separar dados (amostras) semelhantes, através de um **hiperplano separador** criado por uma reta ou um conjunto de retas, em categorias ou classes.

Otimização Meta-heurística:

Tem por objetivo encontrar o melhor valor para um problema através de procedimentos de **diversificação** (*exploration*) e de **intensificação** (*exploitation*) (EIBEN; SCHIPPERS, 1998).

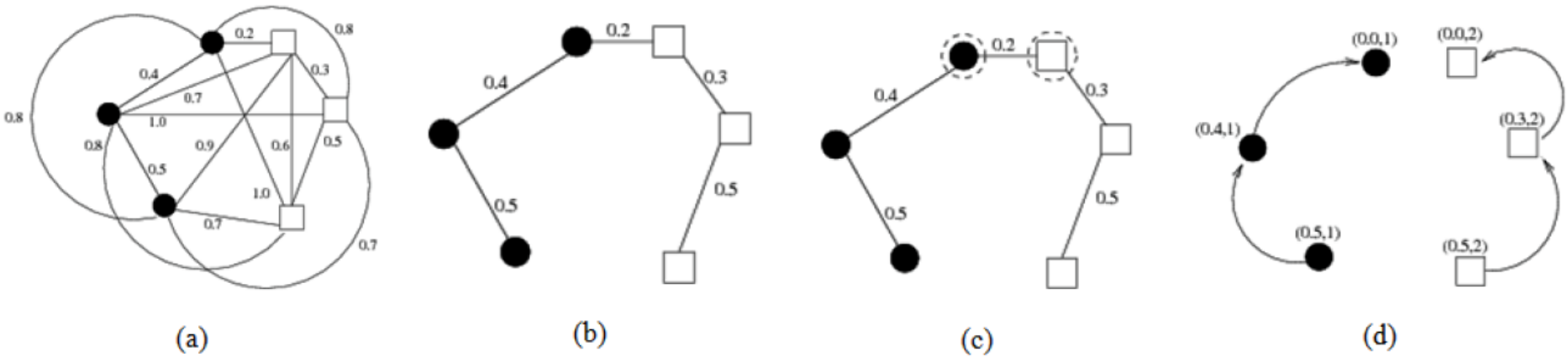
Floresta de Caminhos Ótimos (*Optimum-Path Forest* - OPF)

É uma técnica de classificação multi-classes, desenvolvido por Papa, Falcão e Suzuki (2009). A versão utilizada é supervisionada e baseada em grafo completo e seu funcionamento é dividido em duas etapas:

- **treinamento**;
- **teste**.

OPF - Treinamento

Figura 1: OPF - Treinamento.



A função de custo de caminho é dada por:

f_{max}:

$$f_{max}(\langle \mathbf{s} \rangle) = \begin{cases} 0 & \text{se } \mathbf{s} \in S, \\ +\infty & \text{caso contrário} \end{cases}$$

$$f_{max}(\pi \cdot \langle \mathbf{s}, \mathbf{t} \rangle) = \max\{f_{max}(\pi), d(\mathbf{s}, \mathbf{t})\}.$$

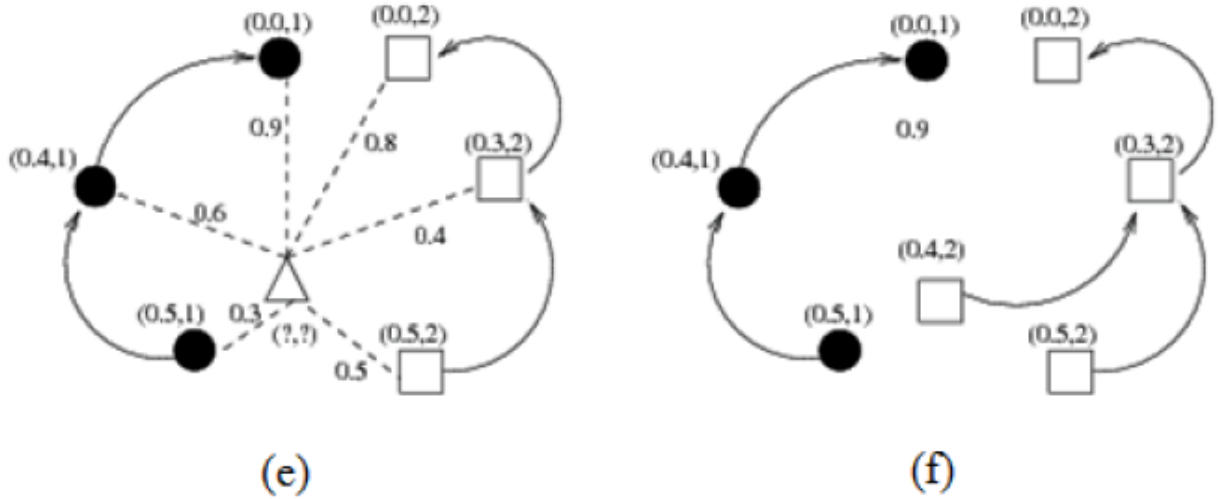
Sendo o custo mínino de um caminho dado por:

Custo Mínimo:

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

OPF - Teste

Figura 2: OPF - Teste.



Custo Ótimo:

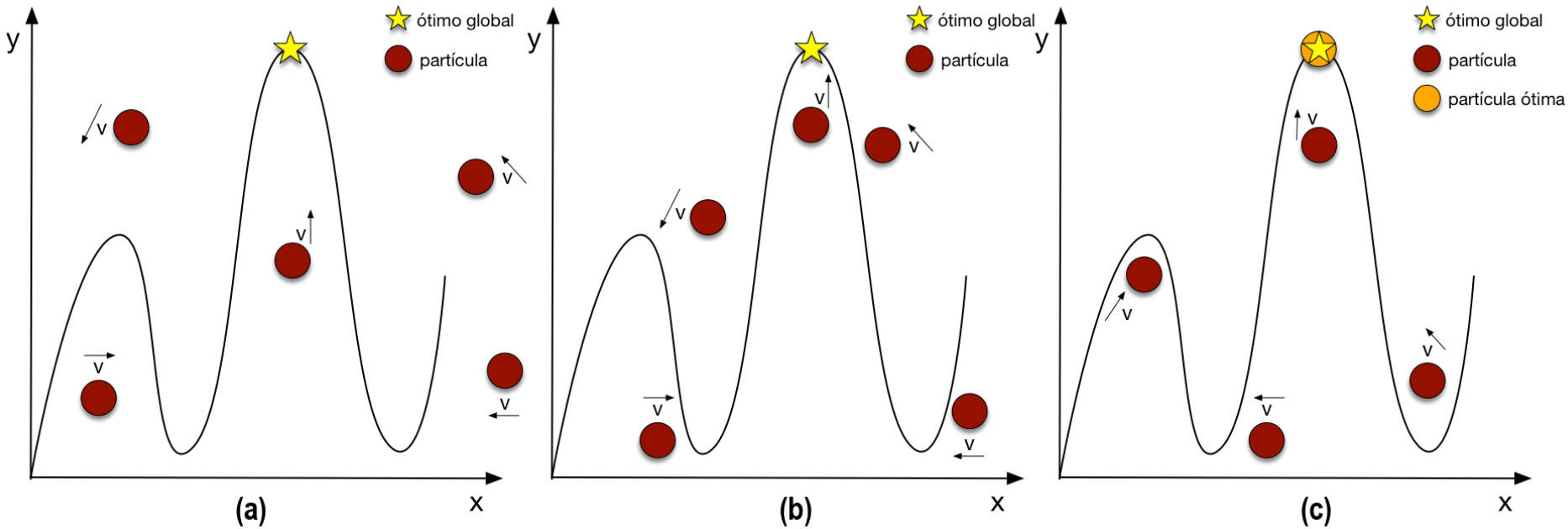
$$C(\mathbf{t}) = \min\{\max\{C(\mathbf{s}), d(\mathbf{s},\mathbf{t})\}\}, \forall \mathbf{s} \in \mathcal{Z}_1.$$

Otimização por Partículas de Enxame (*Particle Swarm Optimization* - PSO)

A otimização por Enxame de Partículas foi desenvolvida por Kennedy e Eberhart (1995) e baseia-se no comportamento social de bandos de pássaros e cardumes de peixes. Esse mecanismo sócio-recognitivo pode ser resumido em três princípios (KENNEDY; EBERHART; SHI, 2001):

- **avaliação**;
- **comparação**;
- **imitação**.

Figura 3: Sistematização do PSO.



A movimentação das partículas é dada por:

Movimentação das partículas:

$$v_i = wv_i + c_1 r_1(\hat{x}_i - x_i) + c_2 r_2(\hat{s} - x_i).$$

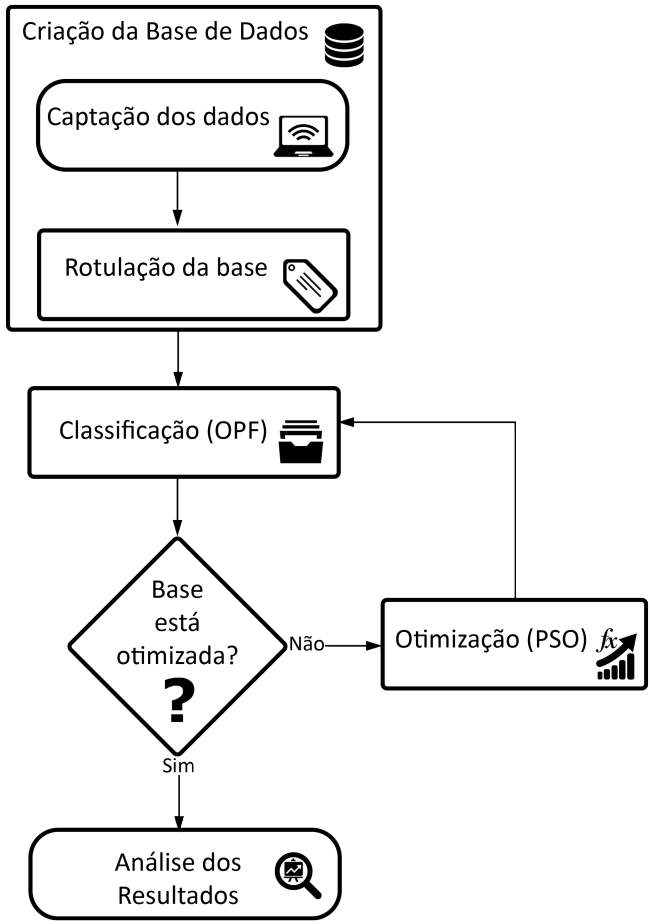
E a posição é dada por:

Posição das partículas:

$$x_i = x_i + v_i.$$

Desenvolvimento

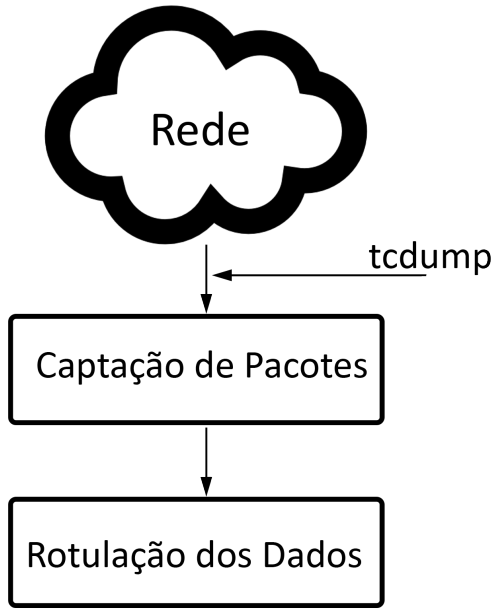
Figura 4: Fluxograma das etapas de desenvolvimento.



Criação da Base de Dados

O processo de criação da base é dividido em duas etapas: **captação** e **rotulação**.

Figura 5: Fluxograma de criação da base de dados.



Captação dos Dados

Os pacotes foram captados na rede *wi-fi* da UNESP Campus Bauru (WFU) através da ferramenta ***tcpdump***, um analisador de protocolo de rede.

O processo foi realizado em, aproximadamente, um mês nos dias úteis (segunda à sexta) dividido em três períodos: matutino, vespertino e noturno.

O extenso tempo de captação deve-se à instabilidade da rede, ocasionada pela grande quantidade de usuários (alunos, professores e funcionários).

Rotulação dos Dados

A importância da etapa de rotulação dar-se-á devido a utilização de um classificador supervisionado, sendo necessário informar as possíveis classes.

Na rotulação manual, é necessário conhecer um padrão responsável por definir o que é ou não uma anomalia, processo denominado **assinatura de intrusão**.

As assinaturas utilizadas foram baseadas na documentação de análise de detecção de intrusão da DARPA, realizada pelo Lincoln Laboratory do MIT (LINCOLN LABORATORY, s.d).

Tabela 1: Exemplos de assinaturas de intrusões.

Nome	Protocolo	Descrição
Apache2	HTTP	Alto número (> 1000) requisições HTTP para o mesmo endereço IP de destino.
Land	IP	Pacote com mesmo endereço IP de origem e destino.
Ping of Death	ICMP	Pacotes com tamanho maior que 64000 bytes.
Ipsweep	ICMP e DNS	Diversos pings da mesma máquina (de origem) para cada máquina disponível na rede.

Foi então criada uma base de dados **semi-sintética** denominada **uneSPY** com aproximadamente 1 milhão de amostras com 23 características, sendo 10% anômala e disponibilizada para estudos posteriores.

Configuração Experimental

Os experimentos foram realizados de forma a analisar a acurácia de detecção antes e depois da otimização, com **diferentes proporções de treinamento e teste** (10/90, 30/70, 50/50, 70/30), 10 vezes cada com o objetivo de obter o melhor resultado.

A classificação pura foi avaliada, além da configuração de treino/teste, em **porcentagem da base de dados** (10%, 50% e 100%).

Como foi observado uma redundância na base, o PSO foi executado em 10% da base de dados com as mesmas configurações de treino e teste utilizadas.

Implementação das técnicas

Implementação OPF:

Biblioteca LibOPF (PAPA; FALCãO; SUZUKI, 2015), disponível em um repositório do *github*.

Implementação PSO:

Biblioteca LibOPT-plus, disponível em um repositório do *github*.

Configuração do OPF

Para utilizar o OPF, é necessário utilizar um **formato específico** na base de dados, já que o classificador compreende características numéricas, assim os dados alfanuméricos foram convertidos e codificados em tabelas, para verificação, caso necessário.

Parâmetros do PSO

PSO - Movimentação das partículas:

$$v_i = wv_i + c_1 r_1(\hat{x}_i - x_i) + c_2 r_2(\hat{s} - x_i).$$

Tabela 2: Parâmetros do PSO.

Parâmetro	Descrição	Valor
w	força de inércia	0.7
r_1 e r_2	trazem a ideia de comportamento social	aleatório entre [0, 1]
c_1 e c_2	fatores de aprendizado	1.7 (KENNEDY; EBERHART; SHI, 2001)
Número de partículas	-	15
Número de características	Depende da base utilizada	23
Número de máximo de iterações	-	25

PSO - Seleção de Características

Neste trabalho, a otimização tem como objetivo selecionar as características que melhor definem o problema, ou seja, visa diminuir o número de características sem perda significativa da acurácia de classificação.

Desta forma, o processo de seleção de características foi baseado no trabalho de Firpi e Goodman (2004), sendo utilizado um vetor auxiliar o que armazena o estado das características de forma binária, definindo **0 uma característica inativa e 1 uma ativa.**

No processo de seleção de características, é necessário utilizar um conjunto de treino e de validação (HASTIE; TIBSHIRANI; FRIEDMAN, 2001), para não comprometer o conjunto de teste.

Resultados

Os resultados apresentados tem por objetivo comparar o comportamento do processo de classificação (pura) e classificação otimizada (otimização), a fim de validar a utilização das técnicas propostas neste trabalho.

Tabela 3: 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**

Tabela 4: 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**

Figura 6: Gráfico da acurácia x quantidade de características selecionadas durante o processo de otimização.

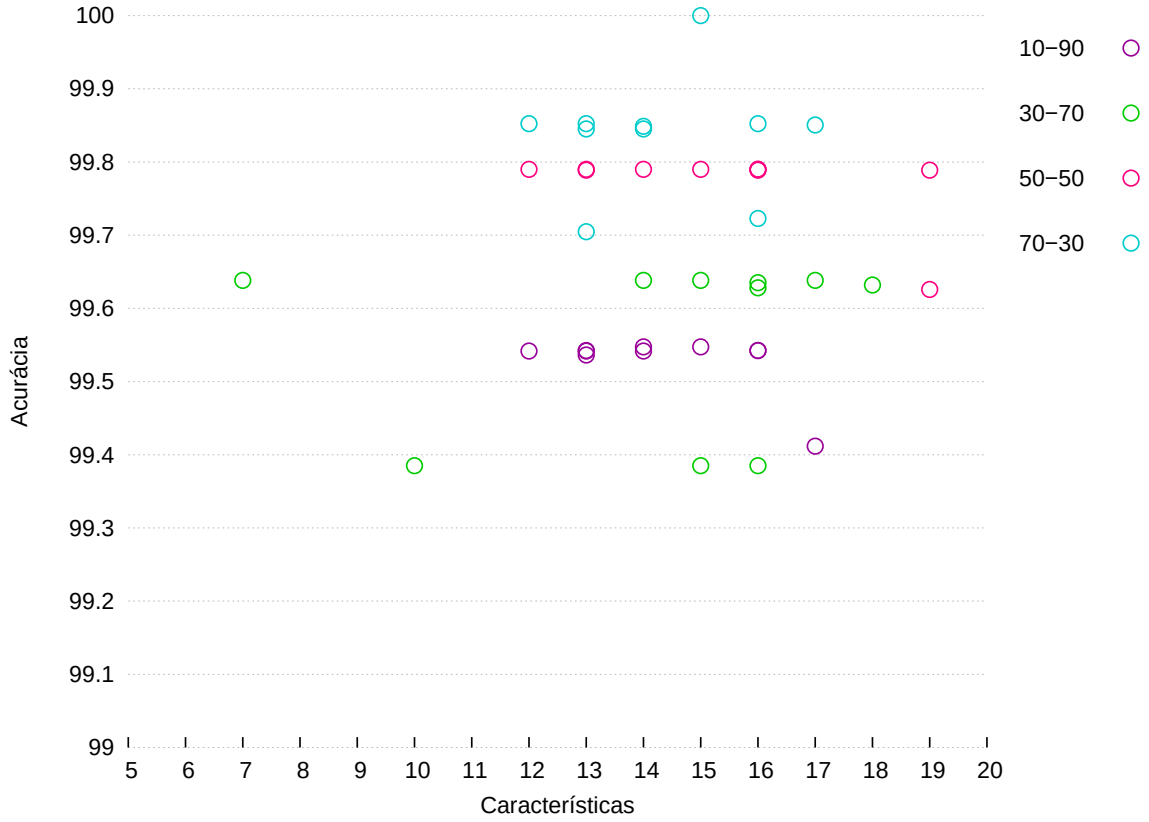
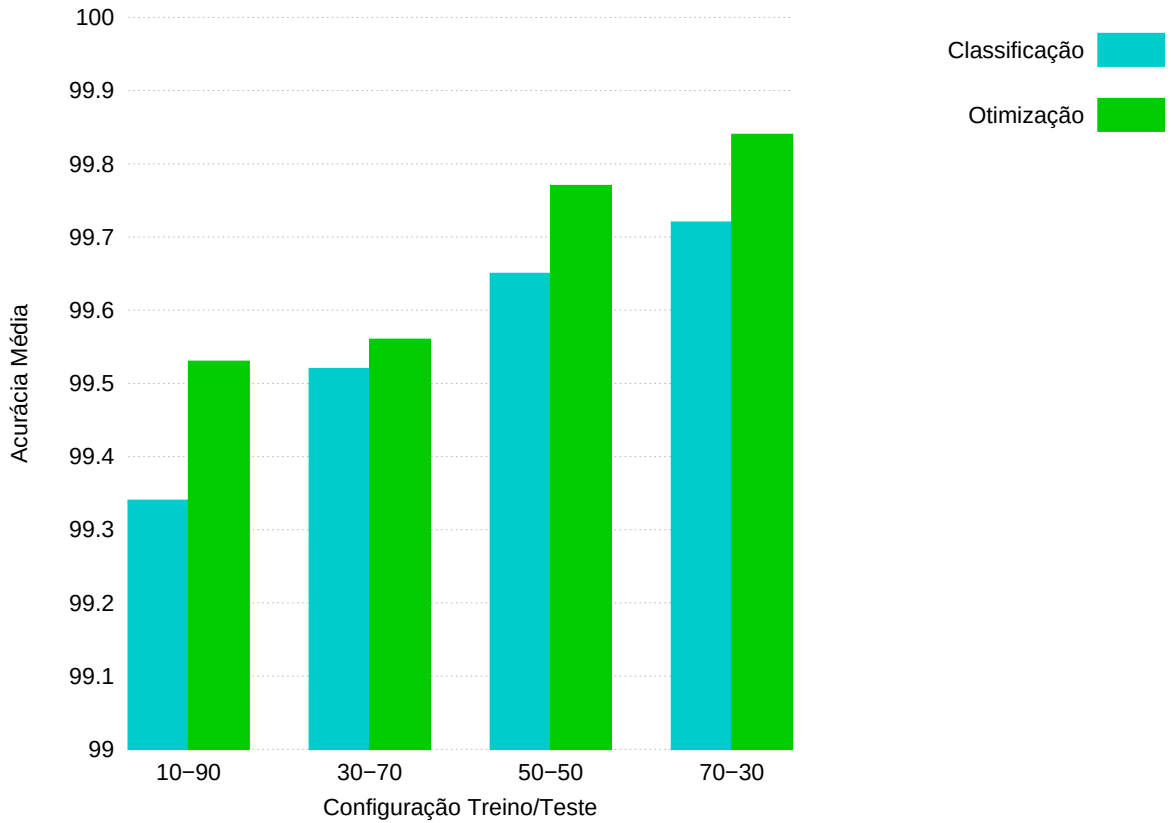


Tabela 5: Comparação dos resultados da classificação com a otimização.

% Treino/% Teste	Acurácia Classificação (%)	Acurácia Otimização (%)
10/90	99.34 ± 0.09 (23 c)	99.53 ± 0.04 (14.3 c)
30/70	99.52 ± 0.04 (23 c)	99.56 ± 0.11 (14.4 c)
50/50	99.65 ± 0.09 (23 c)	99.77 ± 0.05 (15.3 c)
70/30	**99.72 ± 0.06 (23 c)**	**99.84 ± 0.08 (15.6 c)**

Figura 7: Gráfico de comparação de acurácia de classificação com a de otimização.



Interface Gráfica

Uma interface gráfica foi desenvolvida em linguagem C# para ambiente *linux*, com o objetivo de exemplificar, de forma visual, as técnicas aplicadas no trabalho em questão, no entanto, é importante ressaltar que por não ser o foco do projeto, o mesmo foi tratado como um protótipo simplificado com operações reduzidas, que podem ser aprimoradas posteriormente.

Vídeo:

Apresentação do funcionamento da interface gráfica.

Conclusão

- A abordagem proposta se mostrou eficaz;
- Importante contribuição à área de segurança de redes de computadores;
- A inteligência artificial, ou o aprendizado de máquina, tem oportunidades na área de segurança;
- Como complemento, é possível testar outras técnicas de classificação e de otimização, para comparar resultados.

Referências I

CORTES, C.; VAPNIK, V. Support-vector networks. In: *Machine Learning*. [S.l.: s.n.], 1995. p. 273–297.

EIBEN, A. E.; SCHIPPERS, C. A. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, v. 35, p. 35–50, 1998.

FIRPI, H. A.; GOODMAN, E. Swarmed feature selection. In: *33rd Applied Imagery Pattern Recognition Workshop*. Washington, DC, USA: IEEE Computer Society, 2004. p. 112–118. ISBN 0-7695-2250-5.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning*. [S.l.]: 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.

Referências II

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4.

KENNEDY, J.; EBERHART, R. C.; SHI, Y. *Swarm intelligence*. [S.l.]: Morgan Kaufmann, 2001.

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

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

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

Referências III

 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.