Pós-Graduação Lato Sensu
Curso de Especialização em Inteligência Artificial

Introdução a Inteligência Artificial

Prof. Dr. **Lucas Dias Hiera Sampaio**

PARTE 07

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO

Inteligência Artificial em Cenários Reais

1. Apresentação

A inteligência artificial é uma realidade no nosso dia-a-dia: seja nos assistentes pessoais ou na propaganda direcionada que recebemos, as ferramentas de IA possibilitam a solução de diferentes problemas, o aprimoramento e a criação de produtos. Na última semana, estudamos como funcionam diferentes produtos comercializados hoje que embarcam no seu núcleo IA. Nesta semana, nosso foco é entender como pesquisadores aplicam em diferentes contextos as ferramentas de IA e observar que os detalhes desta aplicação são bem mais aguerridos quando comparados aos produtos que estudamos.

2. Os Problemas de Classificação

Os problemas de classificação envolvem utilizar as informações para determinar em qual categoria ou classe as informações pertencem: na prática eles se distinguem dos problemas de categorização por possuírem à priori quais são as classes ou categorias existentes. Os exemplos de aplicação de inteligência artificial para solução de problemas de classificação são muitos e alguns destes são apresentados abaixo.

O trabalho de Hamamoto et al. (2018)[1], por exemplo, utiliza lógica difusa/nebulosa (Fuzzy) associada a Algoritmos Genéticos para criar um sistema de detecção de anomalias de tráfego de rede de computadores. Neste problema clássico de redes, o sistema inteligente deve ser capaz de distinguir entre duas classes: o tráfego comum de rede gerado pelos usuários legítimos e o tráfego anormal/anômalo que é gerado por usuários adversários com a intenção de comprometer o bom funcionamento da rede. O trabalho comparou os resultados obtidos em termos de taxa de detecção e taxa de falso positivo com outras técnicas que haviam sido publicadas na literatura como *Support Vector Machine* (SVM), *Ant Colony Optimization* (ACO) e *Continuous k Nearest Neighbours* (CkNN) por exemplo. Os resultados apresentados pelos autores são condensados na Figura 1.

A Figura 1 apresenta uma curva ROC (*Receivers Operation Characteristics*), uma das formas mais simples de se analisar resultados de problemas de classificação binária, isto é, aqueles em que existem apenas duas classes - neste caso normal ou anômalo. Esta curva relaciona a taxa de detecção (quando o sistema afirma que um dado pertence a classe anômalo e de fato ele pertence) com a taxa de falso positivo (quando o dado pertence a classe normal mas o sistema o classifica como anômalo).

[1] https://www.sciencedirect.com/science/article/abs/pii/S095741741730619X

O objetivo do problema de detecção de anomalias é obter a maior taxa de detecção com a menor taxa possível de falso positivo. Uma das formas mais comuns de análise da curva ROC, além de observar o melhor valor de taxa de detecção, é a área sob a curva (*area under the curve* - AUC). Quanto maior o valor da área sob a curva melhor é o desempenho geral do algoritmo.

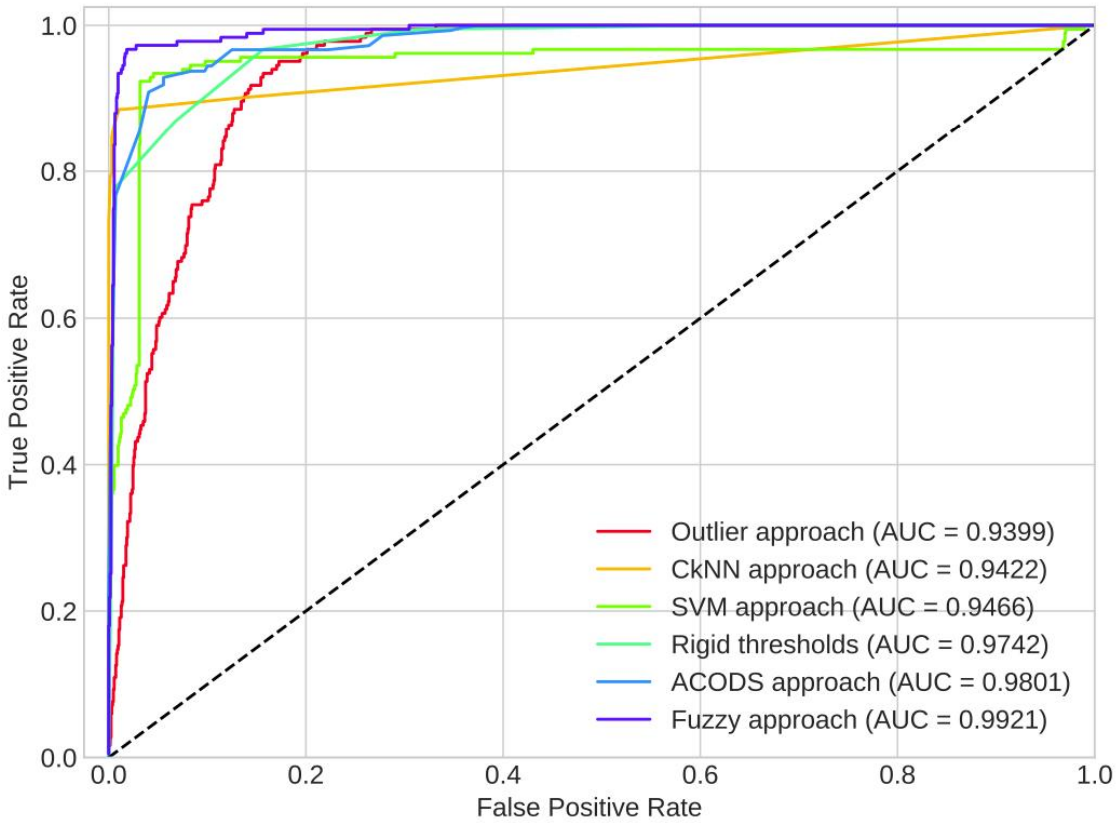

Figura 1: Curvas ROC para diferentes algoritmos de detecção de anomalias em redes de computadores. Fonte: Hamamoto et al. (2018)

A solução proposta pelos autores envolve duas etapas: num primeiro momento é necessário estabelecer o que se chama de *digital signature of network segment using flow analysis* (DSNSF) - isso é feito levando em conta diferentes dados que são coletados na rede como número de bits por segundo em um *flow*, número de pacotes por segundo em cada *flow*, IP de origem e destino, Portas de origem e destino. Como destas 6 métricas apenas bits por segundo e pacotes por segundo são quantitativas, utilizou-se a entropia de Shannon como forma extrair quantitativamente as informações de IP e porta.

Com as leituras dessas 6 características para intervalos de 60 segundos na rede, a criação do DSNFS é feita com base no seguinte problema de otimização: para cada um dos 1440 minutos de um dia e para cada uma das 6 características, defini-se como DSNFS o ponto cuja distância Euclidiana é a menor - para resolver este problema, utilizou-se algoritmos genéticos.

O segundo passo é aplicar nos valores lidos instantaneamente e com base no DSNFS construído, lógica Fuzzy para realizar a detecção das anomalias. A Figura 2 resume o processo de detecção proposto no trabalho.

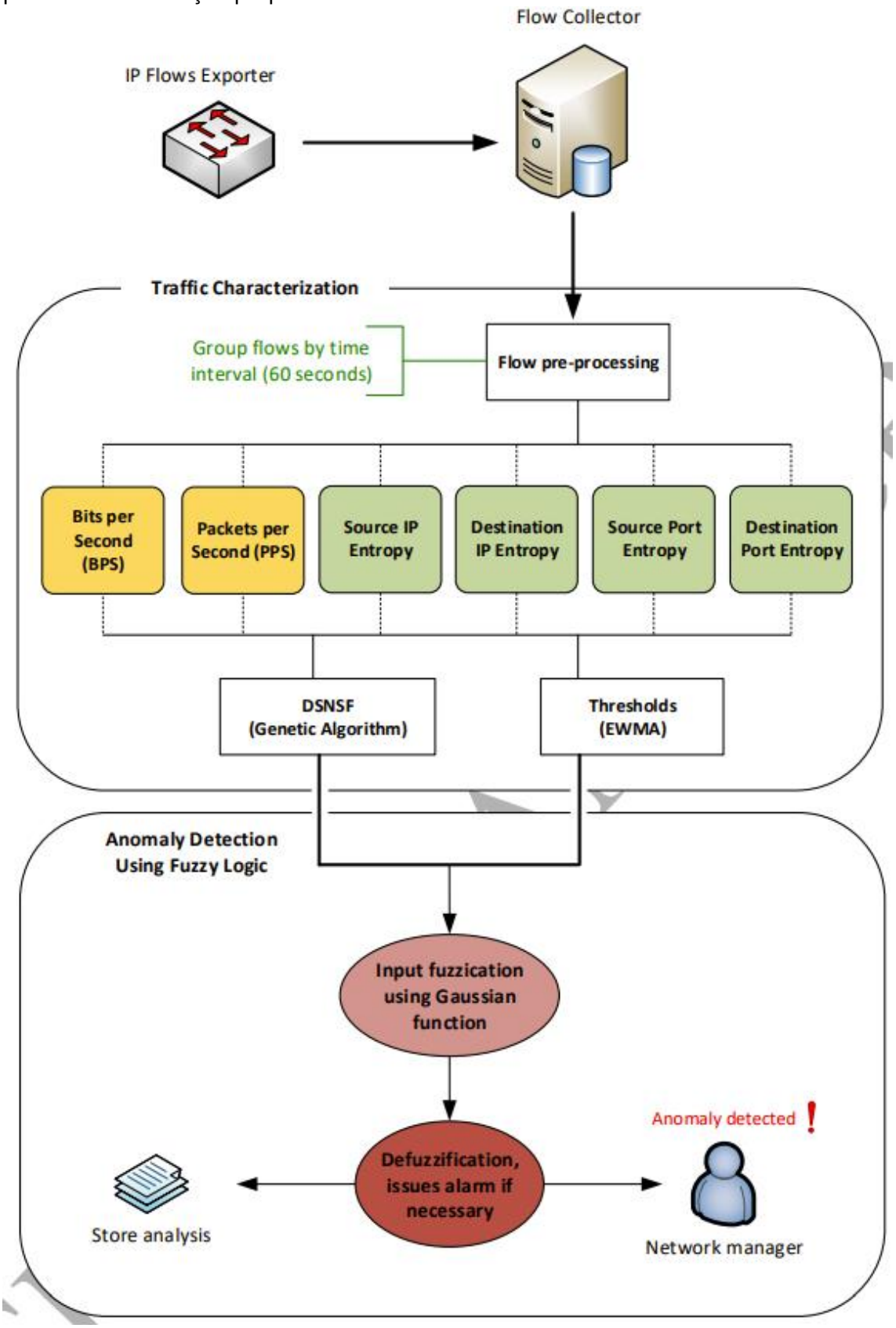

Figura 2: Arquitetura do Sistema de Detecção de Anomalias proposto utilizando Lógica Fuzzy e Algoritmos Genéticos. Fonte: Hamamoto et al. (2018).

Existem ainda problemas de classificação que incorporam mais do que duas classes para serem solucionados: são os denominados problemas multiclasse. O problema discutido no trabalho de Hammad et al. (2022)[2] é um exemplo deste tipo de classificação: no artigo os autores propõe o uso de técnicas de processamento de imagens aliadas aos classificadores *Naive Bayes* (NB) e *Gaussian Discriminant Analysis* (GDA) para classificar os *malwares* do seu Dataset em 26 classes distintas.

O processo de detecção proposto passa por 4 etapas:

- A primeira etapa consiste em converter o código do malware em uma imagem em escala de cinzas.
- A segunda etapa consiste em extrair características utilizando *Segmentation-based Fractal Texture Analysis* (SFTA) e *Gabor Features* (GF).
- A terceira etapa envolve o treinamento dos classificadores NB e GDA utilizando aprendizado supervisionado (os dados do dataset são rotulados) e a utilização da metodologia de validação cruzada K-Fold onde o dataset é separado em dois conjuntos - treinamento e teste - para realizar as duas etapas (treinamento e teste). Este processo é repetido K vezes (no caso do artigo são 10 vezes).
- A última etapa é a análise dos resultados observando a média da metodologia K-Fold.

Todas as etapas descritas acima são sintetizadas na Figura 3 que inclui também o processo de classificação binário.

[2] https://www.mdpi.com/2076-3417/12/24/12528

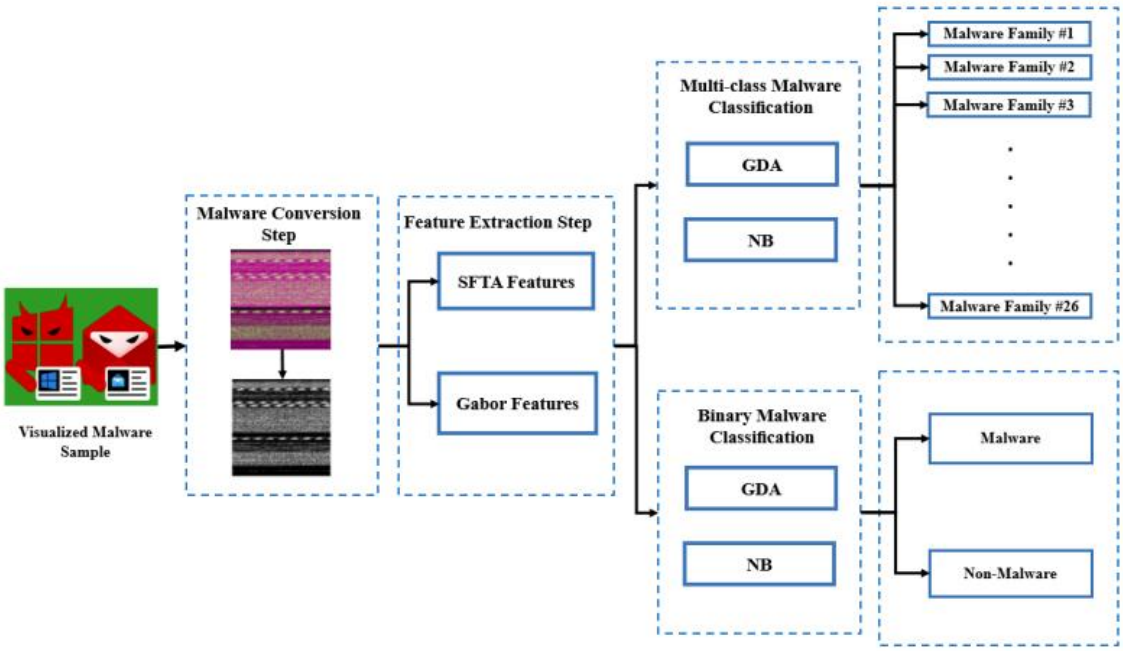

Figura 3: Etapas do processo de detecção multiclasse de malwares. Fonte: Hammad et al. (2022)

Os resultados encontrados pelos autores demonstraram que o melhor resultado é obtido utilizando o classificador GDA com o método de extração de características SFTA chegando a 98% de acurácia média, enquanto o pior resultado foi apresentado pelo método de classificação NB utilizando GF para extração de características. O valor médio de acurácia ao longo das 26 classes foi de 82%. A Tabela 1 sintetiza os resultados obtidos pelos autores.

Tabela 1: Acurácia de classificação para as 26 classes para cada um dos classificadores e métodos de extração de características. Fonte: Hamad et al. (2022)

Class ID	Family Name	Classification Accuracy (%)			
		Naive Classifier		GDA Classifier	
		SFTA Feature	Gabor Feature	SFTA Feature	Gabor Feature
#1	Adposhel	97	90	99	94
#2	Agent	89	52	99	96
#3	Allaple	76	78	98	97
#4	Amonetize	89	62	99	96
#5	Androm	84	52	97	95
#6	Autorun	84	94	98	96
#7	BrowseFox	61	95	99	95
#8	Dinwod	89	55	97	96
#9	Elex	95	80	98	95
#10	Expiro	66	58	99	95
#11	Fasong	98	94	96	95
#12	HackKMS	97	99	99	98
#13	Hlux	99	98	99	99
#14	Injector	73	88	99	95
#15	InstallCore	99	99	96	96
#16	MultiPlug	88	66	97	96
#17	Neoreklami	87	70	99	96
#18	Neshta	95	95	99	97
#19	Regrun	94	91	99	95
#20	Sality	95	80	99	95
#21	Snarasite	99	99	99	95
#22	Stantinko	83	85	99	93
#23	VBA	85	95	99	98
#24	VBKrypt	62	96	97	95
#25	Vilsel	99	99	99	99
Average		87	82%	98%	95%

Os autores ainda comparam seus achados com diferentes artigos da literatura de detecção de malwares (binários e multiclasse) indicando que o método proposto atinge o maior nível de acurácia entre todos.

3. Os Problemas de Otimização

Os problemas de otimização são uma realidade do cotidiano de quase todas as pessoas: seja para decidir o menor caminho entre sua casa e o trabalho, qual marca de um determinado produto comprar, a sequência de atividades a serem

desenvolvidas no trabalho, de forma geral, estamos o tempo todo tentando encontrar a resposta para um problema de otimização muito simples cuja resposta é difícil de encontrar: como alocar nosso tempo da melhor forma possível maximizando nossa felicidade.

Em termos de pesquisa, diferentes problemas, com diferentes domínios (variáveis de decisão) e diferentes funções custo existem em praticamente todas as áreas. Para solucionar estes problemas podemos aplicar diferentes técnicas de inteligência artificial, todavia, de forma geral, a maioria das publicações opta por técnicas dentro do paradigma sub-simbólico.

Neste sentido, os algoritmos genéticos (GA) e o *particle swarm optimization* (PSO) foram aplicados no trabalho de Matos et al. (2022)[3] para solucionar o problema de alocação de sequências piloto em sistemas de comunicação 5G. Este é um problema de análise combinatória onde o número de possibilidades de soluções é muito grande e uma busca exaustiva poderia levar milênios para ser finalizada. Além disso, o problema deve ser solucionado múltiplas vezes por segundo para poder ser utilizado na prática.

No artigo em questão a função objetivo que deseja-se maximizar é a capacidade do sistema, isto é, a quantidade de bits por segundo trafegando em toda rede de telefonia móvel. As restrições são simples: cada sequência piloto é alocada utilizando uma variável binária (a sequência pode estar alocada ou não para um usuário de uma torre), a sequência só pode ser utilizada por um único usuário em cada uma das torres e cada usuário só pode utilizar uma sequência.

Ao todo, 7 algoritmos foram testados para o problema: o RA (*random allocation*) que atribuí as sequências de forma aleatória, utilizado como pior caso, o GA, o BPSO (*Binary PSO*), o PSO utilizando *Smallest Position Value* (SPV) e *variable neighbourhood search* (VNS). O cenário onde o problema foi analisado conta com 7 células hexagonais de celular com a torre posicionada sempre no meio da célula conforme mostra a Figura 4.

[3] https://www.mdpi.com/2076-3417/12/10/5117

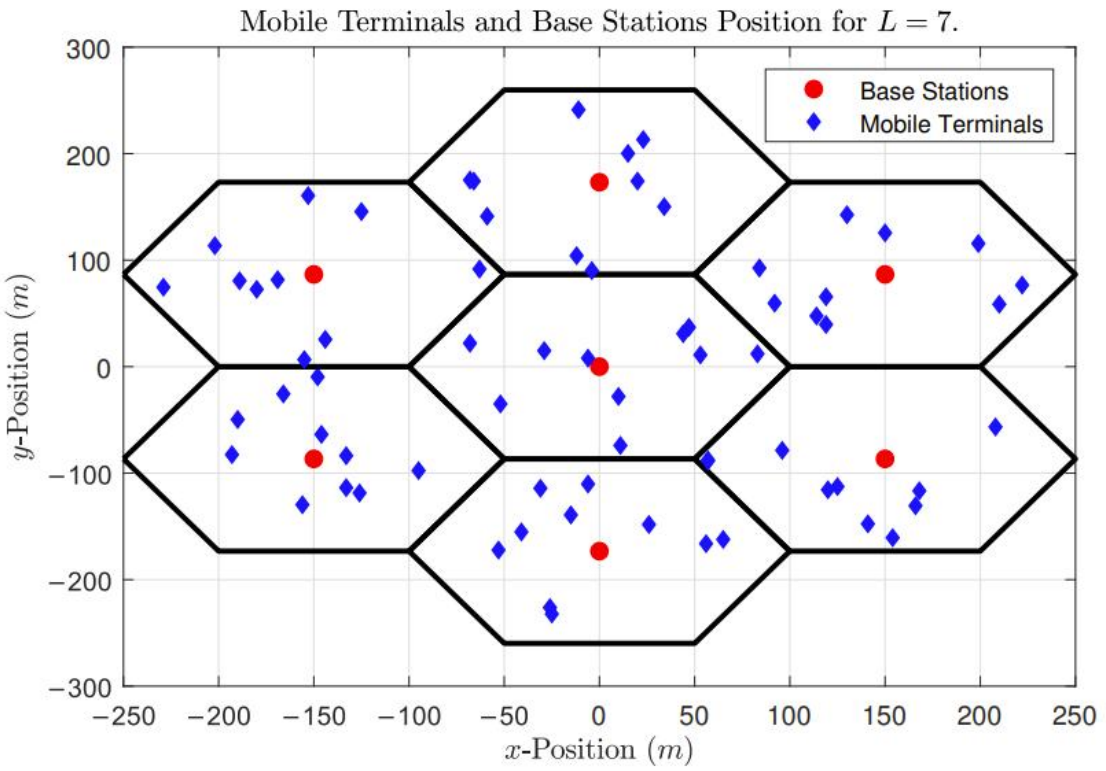

Figura 4: Posição das torres (*Base Stations*) e dos usuários (*Mobile Terminals*) no cenário com 7 células. Fonte: Matos et al. (2022)

Os resultados do trabalho foram obtidos por meio da simulação computacional de todo o cenário e subsequente resolução do problema de otimização no cenário simulado. Os resultados foram obtidos por meio da simulação de 1000 cenários diferentes e são apresentados pelos autores no formato de CDF (*Cumulative Distribution Function*) uma métrica estatística que permite verificar a probabilidade de ocorrência de diferentes intervalos das variáveis.

Nos resultados, a capacidade do sistema (bits/segundo) é analisada utilizando a métrica da relação sinal ruído mais interferência (SINR): quanto maior a SINR maior a taxa de transmissão do sistema em bits/segundo. A Figura 5 apresenta a CDF empírica para os 7 algoritmos utilizados no trabalho.

Vale apontar que alguns algoritmos não melhoram consideravelmente o desempenho em relação à alocação aleatória (RA) como apresentado na Figura 5 na curva mais à esquerda. Enquanto que na curva mais à direita há um empate técnico no desempenho de diferentes algoritmos (SPV-PSO, SPV-VNS-PSO, SPVG-PSO e SPVG-VNS-PSO).

Se por um lado os resultados apresentados mostram que a utilização de metaheurísticas melhora o desempenho do sistema, uma análise do tempo de simulação pode tornar o uso desses algoritmos infactível. A Figura 6 apresenta o

tempo médio de execução de cada algoritmo em termos do número de iterações. Veja que na melhor das hipóteses o tempo ultrapassa 100 segundos.

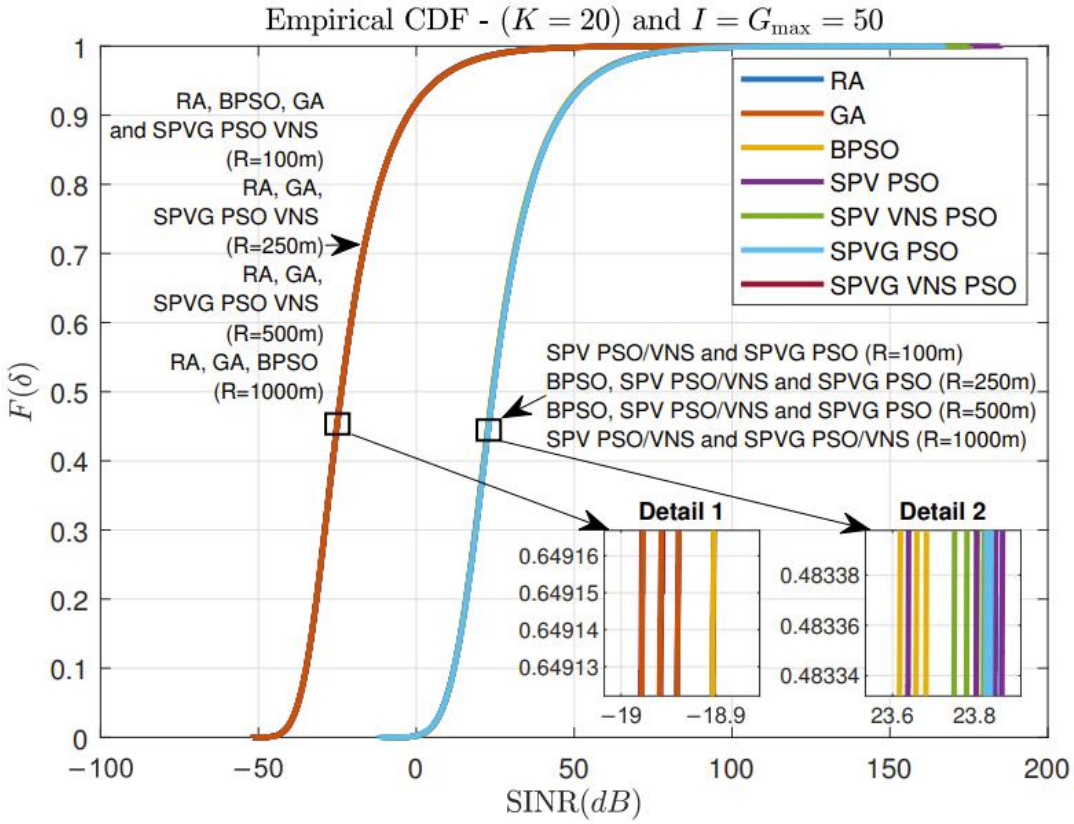

Figura 5: CDF empírica para os cenários simulados. K é o número de usuários por célula, I e Gmax são o número máximo de iterações dos algoritmos de otimização, R indica o raio das células.

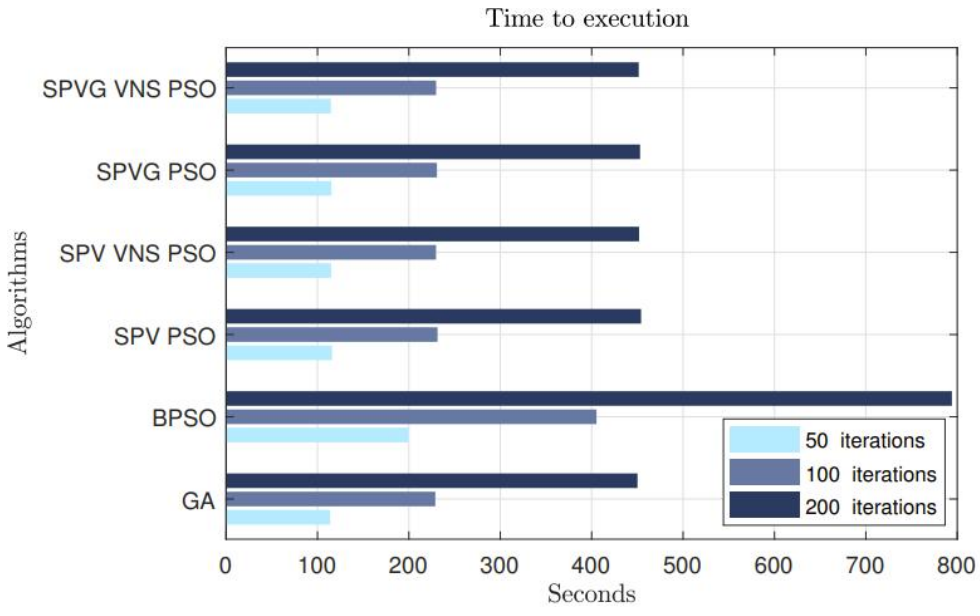

Figura 6: Tempo médio de execução de cada algoritmo

4. Outros Problemas

Existem diferentes problemas além da classificação e otimização que podem ser solucionados por meio de técnicas e ferramentas de IA. Um destes é o problema de categorização: como podemos separar os dados em categorias diferentes se não temos ideia do que estes dados significam ou quais são as categorias possíveis.

Para solucionar este tipo de problema podemos utilizar diferentes técnicas como o algoritmo das K-médias (K-means), Mean-Shift Algorithm (MSA) e Self-Organizing Maps (SOM). A categorização pode surgir como problema em diferentes cenários como determinar quais clientes têm preferências comuns ou podem ser classificados como bons pagadores, por exemplo, são aplicações de categorização.

Outro problema muito comum é a regressão: tentar determinar a partir dos dados o modelo matemático que define os dados ou então tentar determinar a relação entre as diferentes variáveis de um modelo são exemplos de problemas de regressão. Em muitos casos, a regressão pode ser encarada como problema de otimização e solucionada com diferentes técnicas do paradigma sub-simbólico. Em outros casos podemos utilizar técnicas de aprendizado de máquina ou até mesmo ferramentas do paradigma simbólico para resolver problemas relacionados à regressão.

Ao longo do curso de especialização vocês serão introduzidos a diferentes técnicas, ferramentas e métodos de inteligência artificial que podem ser aplicados nos problemas que abordamos nesta apostila.