

Cerise

**CENTRO DE EXCELÊNCIA EM REDES INTELIGENTES
SEM FIO E SERVIÇOS AVANÇADOS - UFG**

**Centro de Excelência em Redes Inteligentes Sem Fio e
Serviços Avançados**

Estimação de Pose e Localização com CSI WiFi-Sensing

Grupo/Eixo Temático: Localização e Sensoriamento

Participante: Paulo Francisco da Conceição

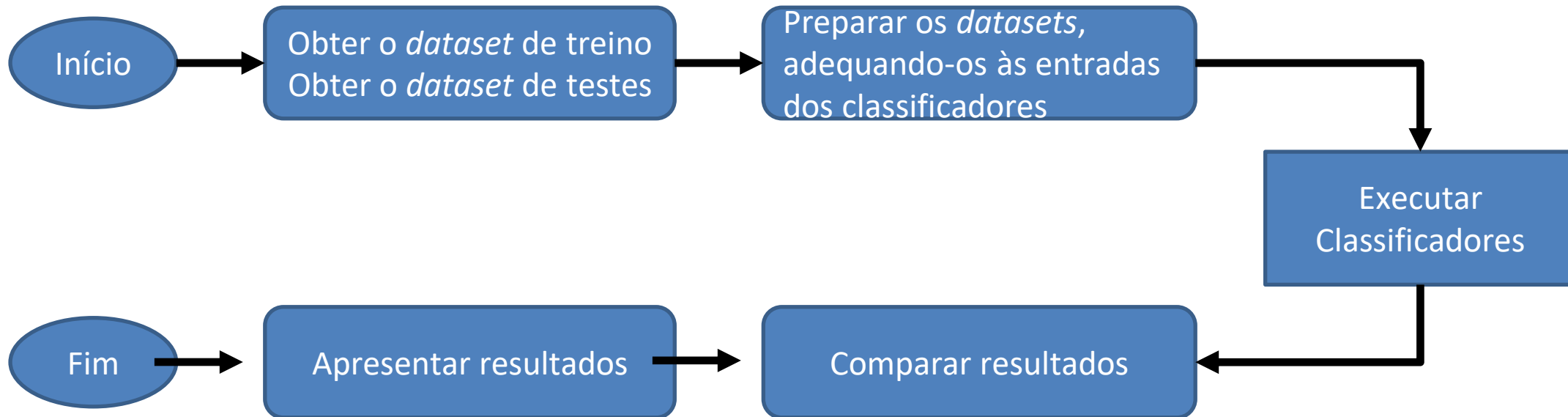
Coordenador do Grupo: Henrique Pires Corrêa

Sumário

1. Introdução
2. Aquisição do *dataset*
 1. Preparação do *dataset*
1. Modelos de classificação testados
1. Resultados
1. Conclusões e Desafios

Introdução

- Algoritmos em Python para preparar um *dataset*, contendo valores de treinamento e de testes. Modelos de classificação usados a partir da biblioteca *scikit-learn*.



Aquisição do CSI

- *DataSet* disponível em: <https://github.com/geekfeiw/CSI-Net>

Placa	Intel 5300
Subportadoras	30
Nr / Nt	1
Altura Tx / Rx	0,8 m
Distância Tx - Rx	3 m
Poses	Deitado / em pé
Localização	5 locais
Fc	5GHz



Instâncias de treinamento	21879
Instâncias de teste	2519

Ajuste dos dados de Treinamento

- Para os dados de treinamento
 1. Ler o arquivo .mat
 2. Ajustar para uma matriz 21879 x 30

	0	1	2	3	4	5	6	7	8	9	10
0	13.35716	16.55151	16.93782	17.53197	18.44881	18.83351	18.68419	18.48298	18.83190	19.23892	18.39928
1	13.35724	16.55163	16.93810	17.53229	18.44925	18.83398	18.68415	18.48355	18.83292	19.23998	18.40037
2	13.35720	16.55162	16.93822	17.53244	18.44955	18.83428	18.68393	18.48397	18.83379	19.24087	18.40132
3	13.35704	16.55148	16.93818	17.53244	18.44972	18.83441	18.68352	18.48422	18.83450	19.24160	18.40213
4	13.35675	16.55122	16.93798	17.53228	18.44974	18.83438	18.68295	18.48432	18.83508	19.24216	18.40280
5	13.35635	16.55085	16.93763	17.53197	18.44963	18.83419	18.68219	18.48426	18.83550	19.24256	18.40333
6	13.35583	16.55035	16.93713	17.53149	18.44938	18.83383	18.68127	18.48405	18.83578	19.24279	18.40373
7	13.35519	16.54974	16.93647	17.53087	18.44900	18.83332	18.68017	18.48368	18.83591	19.24286	18.40398
8	13.35443	16.54901	16.93567	17.53009	18.44848	18.83264	18.67891	18.48316	18.83590	19.24276	18.40410
9	13.35356	16.54817	16.93471	17.52916	18.44784	18.83181	18.67747	18.48248	18.83575	19.24251	18.40408
10	13.35257	16.54721	16.93361	17.52808	18.44706	18.83082	18.67587	18.48166	18.83545	19.24210	18.40393
11	13.35147	16.54614	16.93236	17.52685	18.44616	18.82968	18.67411	18.48068	18.83501	19.24152	18.40364
12	13.35025	16.54496	16.93097	17.52547	18.44512	18.82838	18.67218	18.47955	18.83443	19.24078	18.40320
13	13.34891	16.54366	16.92943	17.52394	18.44396	18.82692	18.67009	18.47827	18.83370	19.23989	18.40263
14	13.34747	16.54226	16.92774	17.52226	18.44266	18.82531	18.66783	18.47684	18.83283	19.23883	18.40192
15	13.34591	16.54074	16.92591	17.52044	18.44124	18.82355	18.66542	18.47526	18.83181	19.23761	18.40107
16	13.34423	16.53912	16.92394	17.51847	18.43968	18.82163	18.66285	18.47353	18.83066	19.23623	18.40008
17	13.34244	16.53738	16.92183	17.51635	18.43800	18.81956	18.66012	18.47165	18.82935	19.23469	18.39895
18	13.34054	16.53554	16.91957	17.51408	18.43619	18.81734	18.65723	18.46962	18.82790	19.23298	18.39767
19	13.33852	16.53358	16.91717	17.51166	18.43425	18.81496	18.65418	18.46744	18.82630	19.23111	18.39624

O mesmo deve ser feito com o *dataset* de testes.

Ajuste dos Rótulos de Treinamento

1. Ler o arquivo .mat
2. Ajustar para uma matriz 21879 x 2
3. A primeira coluna representa a pose (1 – em pé; 2 – deitado)
4. A Segunda coluna representa o local de 1 a 5
5. Ajustar para um único vetor de 21879 posições

241	1	1
242	1	1
243	1	1
244	1	1
245	1	1
246	1	1
247	1	1
248	1	1
249	2	1
250	2	1
251	2	1
252	2	1
253	2	1
254	2	1

246	247	248	249	250	251	252	253
1 - 1	1 - 1	1 - 1	2 - 1	2 - 1	2 - 1	2 - 1	2 - 1

O mesmo deve ser feito com o *dataset* de testes.

Modelos de Classificação

Testados

- Algoritmos de aprendizado supervisionado para classificação e regressão
 - SVM – *Support Vector Machine*
 - SVM é Ele trabalha encontrando o hiperplano que melhor separa os dados em diferentes classes;
 - Pode lidar com conjuntos de dados de alta dimensionalidade e é eficaz em espaços de grande dimensão.
 - KNN (*K-Nearest Neighbors*)
 - Ele faz previsões com base nas classes dos vizinhos mais próximos no espaço de características;
 - É um algoritmo simples e intuitivo, mas pode ser computacionalmente custoso para grandes conjuntos de dados.

Modelos de Classificação

Testados

– Árvore de Decisão

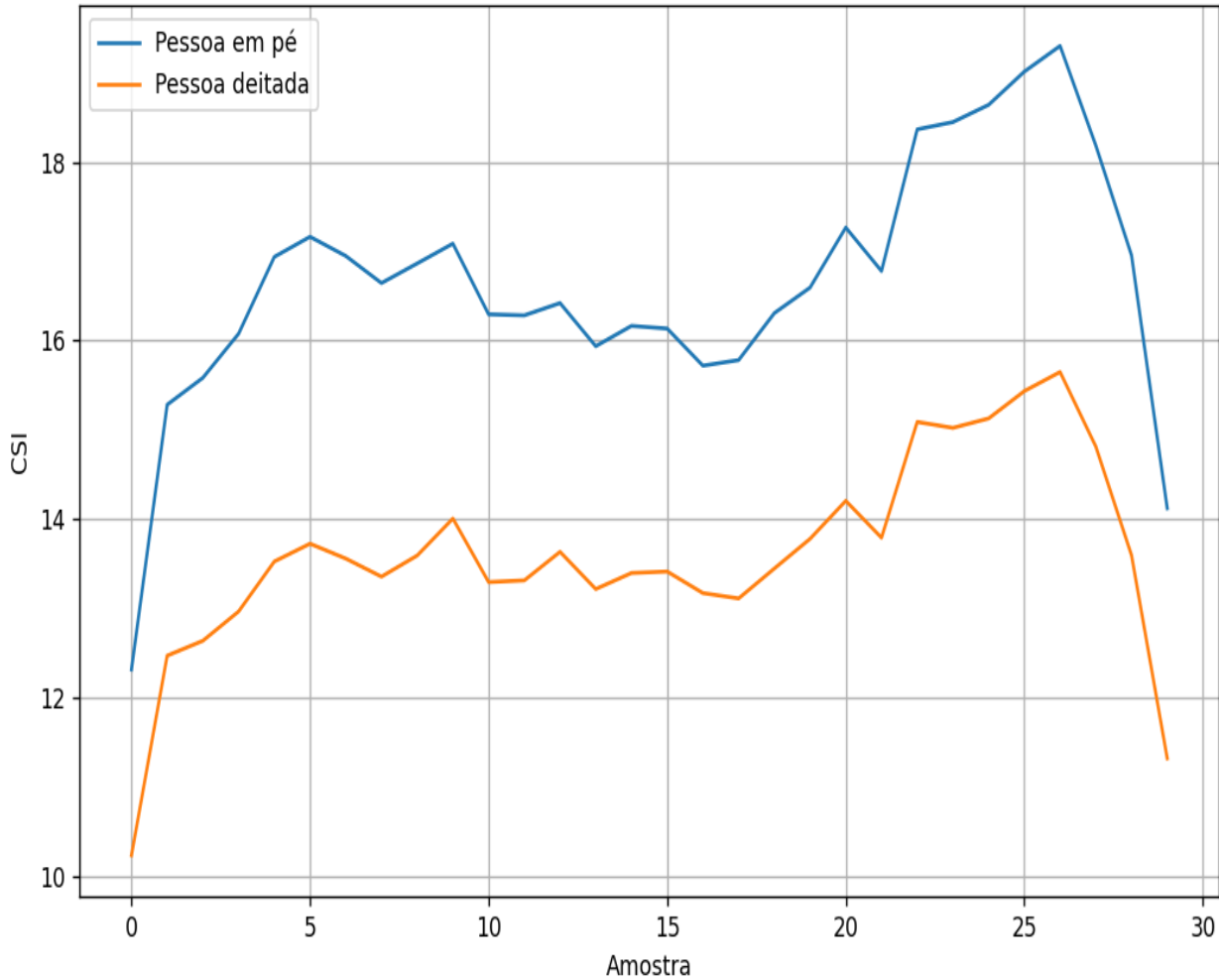
- Elas dividem o espaço de características em regiões retangulares e fazem previsões com base nas características dos dados;
- São fáceis de entender e interpretar, mas podem ser propensas a *overfitting* se não forem regularizadas adequadamente.

– *Random Forest* (Floresta Aleatória)

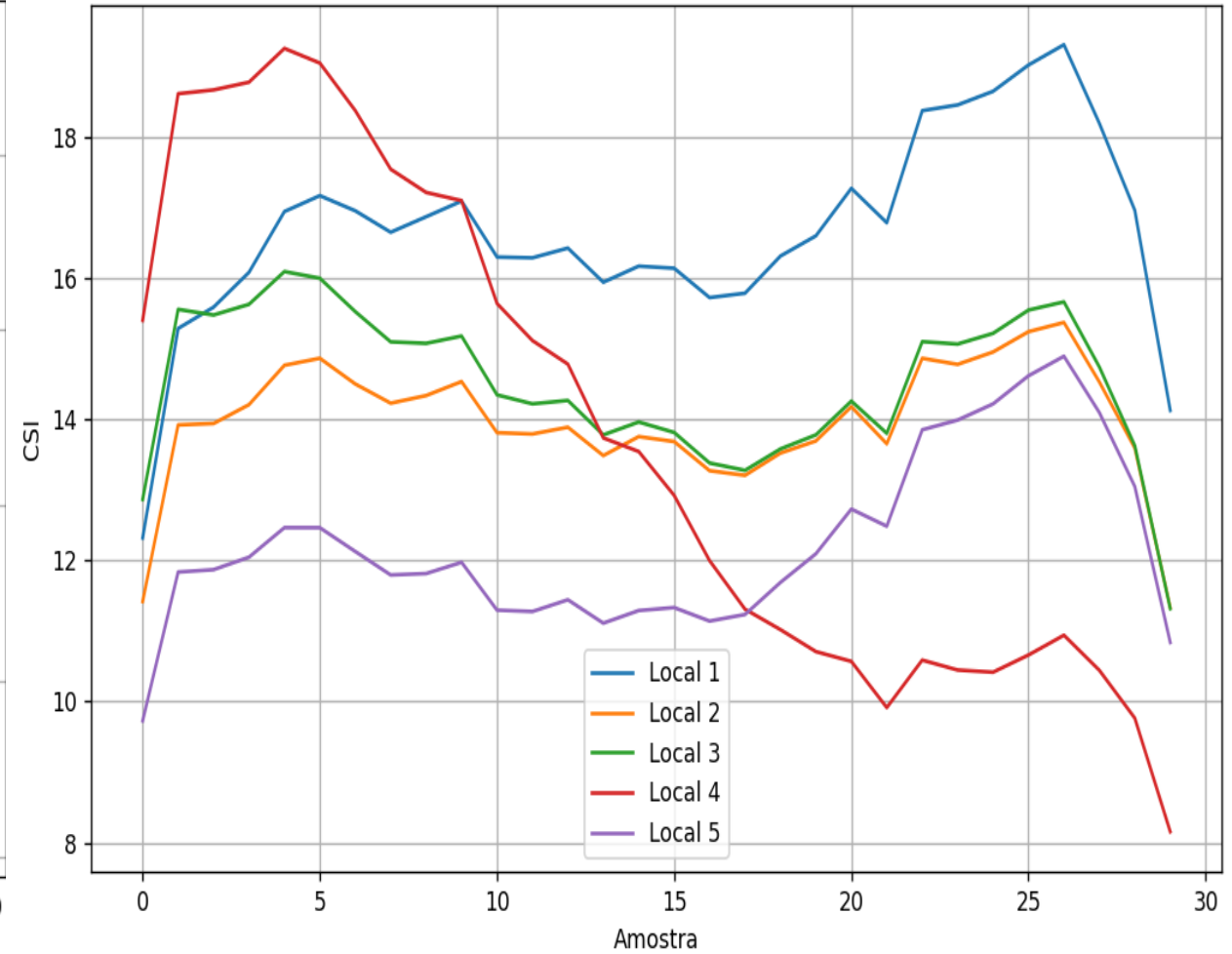
- *Random Forest* é um algoritmo de aprendizado supervisionado baseado em *ensemble*;
- Ele combina várias árvores de decisão para fazer previsões mais robustas e precisas;
- É eficaz para uma variedade de problemas de classificação e regressão e é menos propenso a *overfitting* do que uma única árvore de decisão.

Resultados

Medições de CSI para pessoa em pé e pessoa deitada



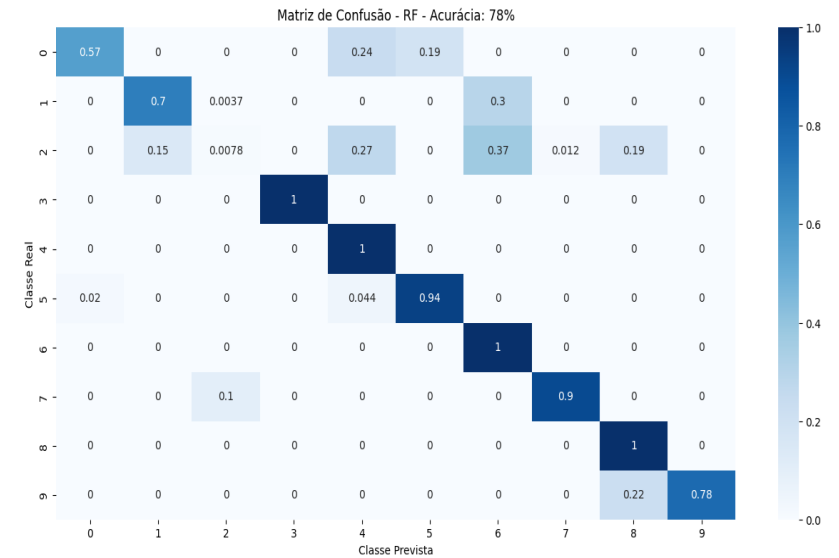
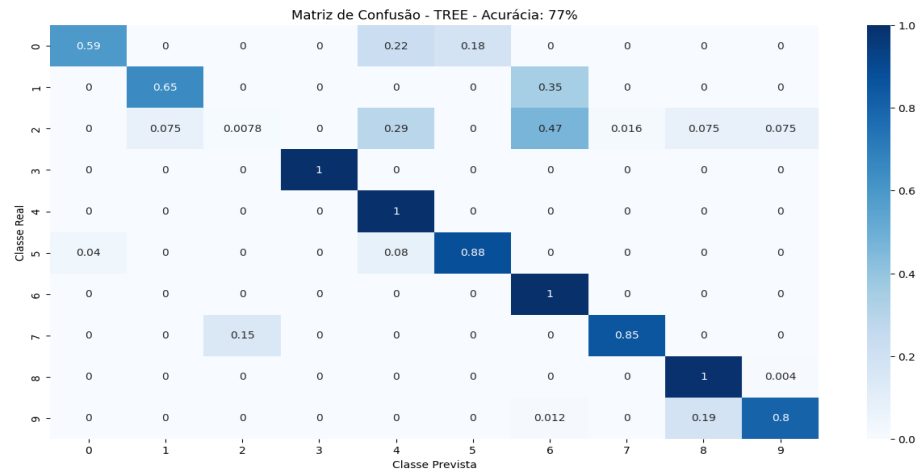
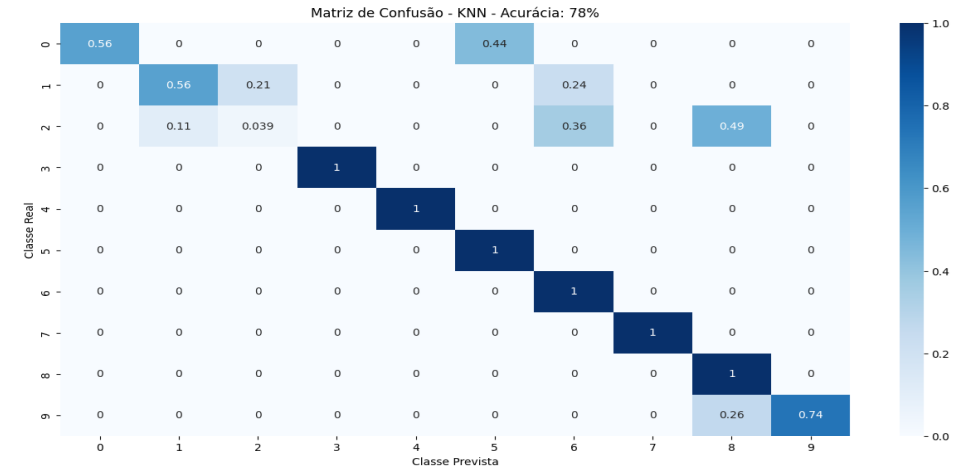
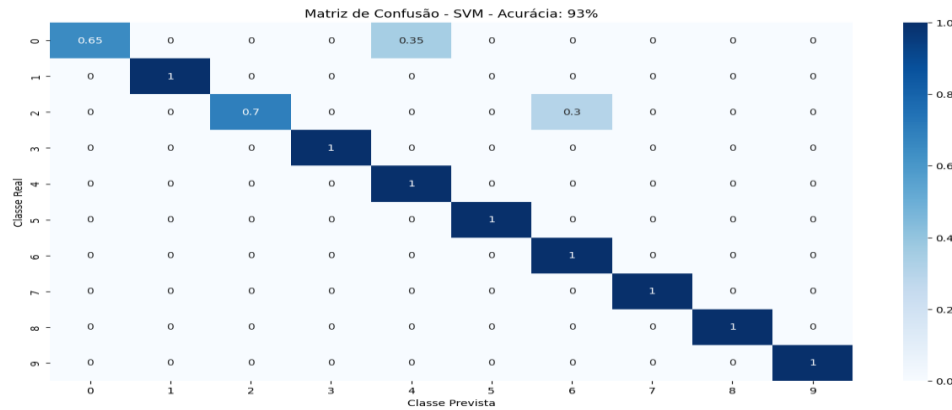
Medições de CSI para localização na sala



Resultados

Modelo / KPI	Acurácia (%)			Tempo (s)
	Pose	Local	Combinada	
SVM	96,94	93,48	93,48	0,41
KNN	84,47	85,47	78,56	4,30
Árvore de Decisão	85,31	85,11	77,45	0,23
Floresta Aleatória	87,45	85,03	78,64	8,97

Resultados



Conclusões e Desafios

- **Conclusões**
 - O CSI realmente permite a estimação de pose e localização
 - Mesmo os modelos de classificação simples, usando a configuração padrão do *Python*, produzem acurácia acima de 93%, como o SVM
- **Desafios**
 - Testar outros *datasets*
 - Coletar o próprio CSI
 - Testar outros algoritmos/modelos. Mais complexos talvez?
 - Produzir algoritmos próprios
 - KNN com distância de *Mahalanobis*?

Estimação de Pose e Localização com CSI WiFi-Sensing

Obrigado!

Participante: Paulo Francisco da Conceição

Coordenador do Grupo: Henrique Pires Corrêa