

Detecção de doenças cardiovasculares através de sinais de ECG utilizando ferramentas de inteligência artificial

Aluno: João Pedro de Oliveira Pagnan

Professor: Prof. Dr. José Wilson Magalhães Bassani [CEB/UNICAMP]

Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e Computação
EA997 - Introdução à Engenharia Biomédica

24 de junho de 2022

Introdução

O diagnóstico de arritmias cardíacas através de sinais de eletrocardiograma (ECG) é de extrema importância para monitorar a saúde do coração através de um método não invasivo^[1]. Devido a isso, uma boa etapa de interpretação de sinais computadorizados de ECG é fundamental para que o diagnóstico seja feito de forma precisa e, caso exista, a arritmia cardíaca seja detectada corretamente.

Embora que esta análise seja tradicionalmente feita por cardiologistas, trabalhos recentes indicam que ferramentas computacionais de aprendizado de máquina¹ podem obter métricas de **f-medida** e **acurácia** melhores que as alcançadas por grande parte dos cardiologistas^[2,3].

Desta forma, estas ferramentas podem detectar diversos tipos de arritmias cardíacas a partir de uma única derivação com desempenho comparável ao de cardiologistas e, em contextos clínicos, podem reduzir a chance de diagnósticos incorretos e melhorar a interpretação do sinal de ECG de um especialista humano que já terá uma indicação da provável arritmia que o paciente possui^[3].

¹Do inglês: *Machine Learning*.

Objetivos

Este projeto visa implementar e comparar quatro tipos de classificadores para identificar arritmias cardíacas através de sinais de ECG: um modelo baseado em máquinas de vetores-suporte (SVM²), outro baseado nos k vizinhos mais próximos (KNN³), um terceiro baseado em florestas aleatórias (RF⁴) e, por fim, um baseado no tipo de rede neural LSTM ⁵[4].

Neste caso, planeja-se também analisar qual a melhor representação para os sinais de ECG, isto é, se representaremos os sinais no domínio do tempo ou da frequência, bem como se o uso de filtros para remoção de ruído pode aprimorar o desempenho dos classificadores.

²Do inglês: *Support Vector-Machine*.

³Do inglês: *K-Nearest Neighbors*.

⁴Do inglês: *Random Forest*.

⁵Do inglês: *Long-Short Term Memory*.

Foi utilizada a linguagem **Python 3**, mais precisamente, as bibliotecas **Scikit-Learn** e **TensorFlow** e a base de dados⁶ aberta **MIT-BIH**^[1], que contém dados de 47 pessoas de 23 a 89 anos, incluindo homens e mulheres.

Esta base foi construída entre 1975 e 1979 e contém amostras de cinco padrões diferentes de batimento cardíaco: um padrão com o coração saudável, outro com bloqueio do ramo esquerdo, um terceiro com bloqueio do ramo direito, um quarto com contração atrial prematura e, por fim, um último com contração ventricular prematura. Estes padrões estão ilustrados na figura 1.

O dado foi digitalizado com uma frequência de 360 Hz e cada sinal dura 1 s, ou seja, cada sinal é um conjunto de 360 valores e, por fim, foi utilizada a derivação precordial (V2).

⁶Do inglês: *Dataset*.

Metodologia

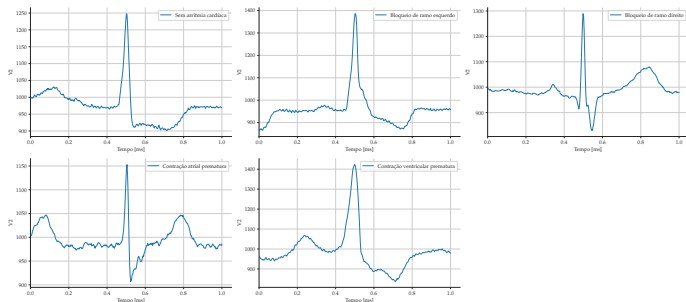


Figura 1: Tipos de batimento cardíaco presentes nos sinais de ECG presentes na base de dados selecionada.

A base de dados foi balanceada para que todas as classes tenham a mesma proporção no conjunto final, ou seja, haja o mesmo número de sinais (2546) em todos os tipos de batimentos cardíacos presentes nos sinais de ECG.

Após carregar as séries temporais, os dados foram normalizados utilizando a técnica *z-score*, definida na equação (1).

$$x' = \frac{x - \mu}{\sigma}, \quad (1)$$

onde x é o valor original, x' é o valor normalizado, μ é a média dos valores do sinal e σ é o desvio-padrão.

Após este processo de normalização, foram criados os seguintes conjuntos de representações para os sinais:

- 1 Séries temporais normalizadas;
- 2 Séries temporais normalizadas e filtradas;
- 3 Módulos dos espectros de frequência das séries temporais normalizadas;
- 4 Módulos dos espectros de frequência das séries temporais normalizadas e filtradas;

Metodologia

Com as representações, os modelos foram ajustados com base em um conjunto de treinamento que continha 75% do número total de sinais e avaliados em um conjunto de teste composto por 25% do total.

No caso da rede neural, como o ajuste dos pesos dos neurônios é iterativo, pois não há solução em forma fechada, o conjunto de treinamento possui 67.5% do total de amostras para que haja um conjunto de validação composto por 7.5% das coletas. Este conjunto será usado durante o treinamento da rede para evitar o problema do sobreajuste.

Para medir o desempenho dos modelos nesta tarefa, foram utilizadas as métricas numéricas de acurácia e f-medida definidas em (2):

$$\text{Acurácia } (\hat{y}(x)) = \frac{TP + TN}{N}, \quad (2a)$$

$$\text{Precisão } (\hat{y}(x)) = \frac{TP}{TP + FP}, \quad (2b) \quad \text{Sensibilidade } (\hat{y}(x)) = \frac{TP}{TP + FN}, \quad (2c)$$

$$\text{F-medida } (\hat{y}(x)) = 2 \cdot \frac{\text{Sensibilidade } (\hat{y}(x)) \cdot \text{Precisão } (\hat{y}(x))}{\text{Sensibilidade } (\hat{y}(x)) + \text{Precisão } (\hat{y}(x))}, \quad (2d)$$

sendo N o total de amostras, TP e TN o número de verdadeiros positivos e negativos, respectivamente, e FP e FN o número de falsos positivos e negativos.

Resultados

Utilizando as métricas indicadas anteriormente, foram obtidas as seguintes métricas para as representações utilizadas:

Tabela 1: Métricas obtidas para os sinais temporais não-filtrados.

Modelo	Acurácia	F-medida
SVM	0.9654	0.9658
KNN	0.9538	0.9541
RF	0.9627	0.9623
LSTM	(0.9709 ± 0.0029)	(0.9711 ± 0.0029)

Tabela 2: Métricas obtidas para os sinais temporais filtrados.

Modelo	Acurácia	F-medida
SVM	0.9544	0.9549
KNN	0.9560	0.95640
RF	0.9365	0.9372
LSTM	(0.9683 ± 0.0041)	(0.9685 ± 0.0041)

Tabela 3: Métricas obtidas para os módulo dos espectros em frequência não-filtrados.

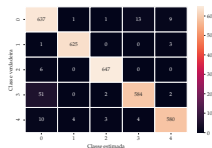
Modelo	Acurácia	F-medida
SVM	0.9061	0.9068
KNN	0.9538	0.9541
RF	0.9189	0.9189
LSTM	(0.9176 ± 0.0066)	(0.9190 ± 0.0067)

Tabela 4: Métricas obtidas para os módulo dos espectros em frequência filtrados.

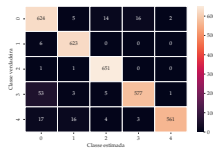
Modelo	Acurácia	F-medida
SVM	0.8960	0.8966
KNN	0.9010	0.9013
RF	0.9139	0.9139
LSTM	(0.9247 ± 0.0037)	(0.9253 ± 0.0039)

Resultados

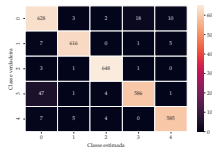
Utilizando a representação que obteve os melhores valores para as métrica (sinais temporais não-filtrados) foram obtidas as seguintes matrizes de confusão para os classificadores utilizados:



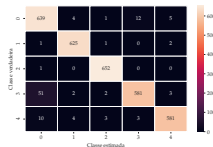
(a) SVM



(b) KNN



(c) RF



(d) LSTM

Figura 2: Matrizes de confusão obtidas pelos quatro modelos avaliados utilizando os sinais temporais não-filtrados.

Discussão

Analisando os resultados obtidos pode-se perceber que:

- 1 Representar os dados de ECG no formato de séries temporais favorece a classificação;
- 2 A filtragem utilizada prejudicou a classificação;
- 3 O melhor modelo classificador (em todas as representações) foi o baseado em redes LSTM;
- 4 A classificação incorreta que mais prejudicou o desempenho de todos os modelos foi a confusão da classe 0 com a classe 3, ou seja, os classificadores indicaram que um sinal de ECG de uma pessoa com contração atrial prematura é o de uma pessoa com coração saudável.

Devido ao quarto ponto mencionado acima, recomenda-se que, caso seja determinado pelo modelo que o indivíduo não possui as arritmias analisadas, esta pessoa faça mais exames para confirmar este fato.

Conclusão

Com este projeto conclui-se que ferramentas de aprendizado de máquina e

Referências

- [1] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [2] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature medicine*, vol. 25, no. 1, pp. 65–69, 2019.
- [3] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *arXiv preprint arXiv:1707.01836*, 2017.
- [4] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

Muito Obrigado!