IDENTIFICAÇÃO DE ATIVIDADE FÍSICA ATRAVÉS DE MODELOS DE INTELIGÊNCIA ARTIFICIAL NO PROCESSAMENTO DE SINAIS DE ECG

Leonardo Ferreira,* Luis de Deus,† and Tiago Knorst‡
Centro de tecnologia, Universidade Federal de Santa Maria.
(Dated: 11 de julho de 2019)

This work aims to identify physical activity, through the application of artificial intelligence models in the analysis of heart rate variability. The HRV is extracted by the processing of electrocardiogram signals acquired through a signal acquisition board connected to an Arduino. The classification of the indicators is done through an algorithm described in high-level programming language that runs on a Raspberry Pi device.

Keywords: Physical Activity, HRV, Artificial Intelligence.

I. INTRODUÇÃO

A Variabilidade da Frequência Cardíaca (HRV) é uma fonte poderosa para a avaliação do sistema cardíaco, sendo objeto de diversas pesquisas para determinar e encontrar problemas crônicos no coração. O avanço tecnológico permitiu o desenvolvimento de dispositivos móveis capazes de quantizar estes dados em intervalos inter-batimentos (Intervalos RR).

Dispositivos comerciais de empresas como Polar e Garmin, por exemplo, são soluções economicamente viáveis para que usuários comuns façam uso em campos de pesquisa como em esporte, medicina e outros campos de pesquisa, inclusive com resultados promissores em repouso, se comparado a equipamentos de ECG como descrito em [1].

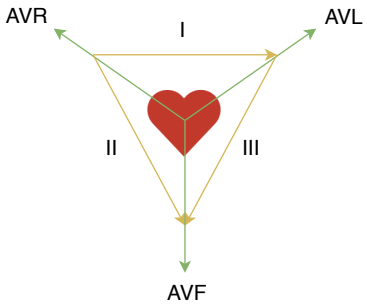
Entretanto, a maior parte dos estudos são relacionados a aquisição de dados em repouso, devido a grande quantidade de ruído presente no monitoramento em atividade física, associado a dificuldade no processamento e filtragem dos dados. Este trabalho tem por objetivo a identificação e classificação de estados de atividade física, como repouso e atividade física.

II. REVISÃO TEÓRICA

Segundo [2] alterações na frequência cardíaca, definidas como variabilidade da frequência cardíaca (HRV), são normais e esperadas ao iniciar e terminar atividades físicas, elas indicam a habilidade do coração em responder aos múltiplos estímulos fisiológicos, e também compensar desordens induzidas por doenças.

Para medição de um sinal ECG, a posição dos eletrodos varia conforme o tipo de informação clínica necessária, assim, são 12 as derivações eletrocardiográficas, sendo 6 periféricas e 6 precordiais. Na Fig. 1 é possível observar as derivações periféricas, sendo apenas estas utilizadas no trabalho pela sua facilidade de ligação e menor uso no número de eletrodos.

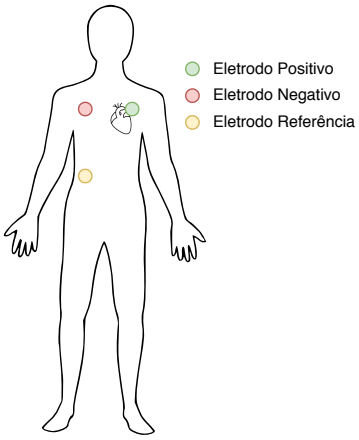
Figura 1. Derivações periféricas do eletrocardiograma.



Fonte: Autor

A posição dos eletrodos para medição do sinal ECG utilizadas neste trabalho é demonstrado na Fig. 2, sendo dois para medição e um atuando como referência para o sinal.

Figura 2. Posição dos eletrodos em derivação I.



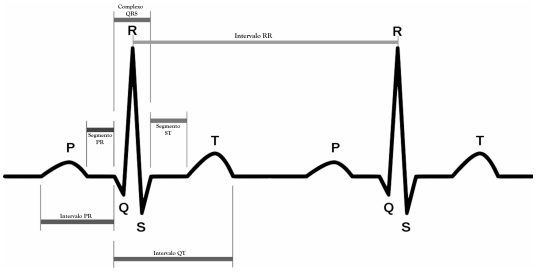
Fonte: Autor

Em um sinal de eletrocardiograma (ECG) de derivação I, como demonstra a Fig.3 são observados cinco ondas, a onda "P", o complexo "QRS" e a onda "T". Como abordado em [3], a onda "P" é resultado da despolarização

* leonardo_lferreira@hotmail.com
† felipe.deus@ecomp.ufsm.br
‡ tiago.knorst@ecomp.ufsm.br

dos átrios antes da contração do coração e, como os átrios possuem pouco músculo, a variação da tensão elétrica é pequena. Já o complexo "QRS" é causado pela despolarização ventricular sendo a porção de maior amplitude do ECG. O tempo durante o qual ocorre a contração ventricular é referido como sístole e por fim, a onda "T" é causado pela repolarização ventricular, sendo o tempo entre as contrações ventriculares referido com diástole.

Figura 3. Sinal característico de ECG.



Fonte: Revista brasileira de cardiologia v17 n3

O intervalo de tempo entre duas ondas R é definido como intervalo RR, a maneira mais precisa para mensurar a frequência cardíaca é determinar a média do intervalo RR, ou seja, o período entre batimentos e assim dividir 60 segundos por esta média afim de se obter batimentos por minuto (bpm).

No trabalho proposto pelo autor, em [4], investigou-se a premissa de que indivíduos com maior variabilidade cardíaca possuem um melhor nível de aptidão cardiorespiratória por meio da análise de HRV no domínio da frequência, e a comparação dos sinais de ECG, com os pacientes sob atividade física alta, média e baixa.

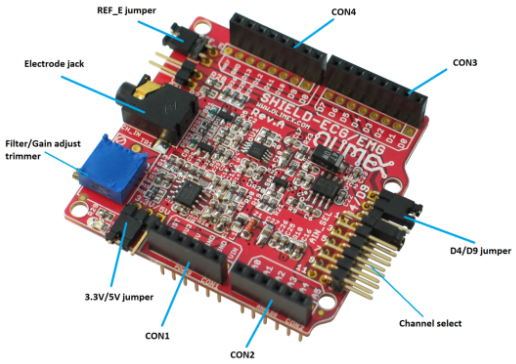
III. METODOLOGIA

Para obtenção do sinal de ECG, foi utilizado uma placa de aquisição de sinais (*shield* de bio-*feedback*) Fig.4, desenvolvida pela fabricante de *hardware* búlgara Olimex LTD, o qual tem a finalidade de obter os sinais dos eletrodos. A placa é composta por uma série de filtros para rejeição de altas frequências, um amplificador com ganho controlado pelo trimmer "TR1" e no último estágio, um filtro Besselworth de 3^{a} ordem, com frequência de corte em 40 Hz.

A placa da Olimex dispõe o sinal adquirido em uma de suas portas analógicas, para isso, fez-se necessário um Arduino Uno para efetuar a conversão analógico-digital dos dados obtidos e estabelecer uma comunicação serial com uma placa de desenvolvimento Raspberry PI 3 model B+ Fig.5, responsável pelo processamento de dados dos algoritmos descritos na linguagem de programação Python. Foi escolhido a plataforma Raspberry PI, devido a ser uma solução com custo-benefício alto, devido ao seu valor de mercado, e por ser uma aproximação interessante de um sistema embarcado, para o qual este trabalho tem o seu direcionamento. A placa que é classificada com um microcomputador, abaixo são citadas algumas de suas características:

- Processador: Broadcom BCM2837B0 64bits ARM Cortex-A53 Quad-Core;
- Clock: 1.4 GHz;
- Memória RAM: 1 GB;
- Wifi 802.11 b/g/n/AC 2.4GHz;
- GPIO: 40 pinos.

Figura 4. *shield* de bio-*feedback*.



Fonte:[5]

Figura 5. Raspberry PI 3 model B+.

Fonte: [6]

A. Método de Hamilton e Tompkins

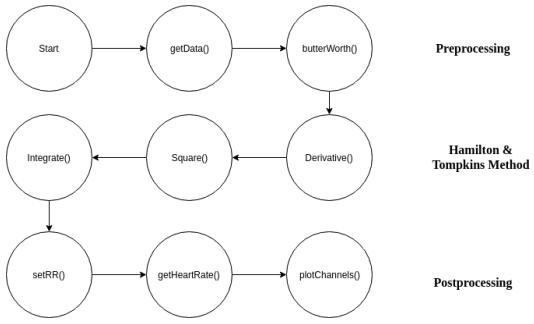
Como já descrito, a principal medida necessária à determinação do HRV é o intervalo R-R, assim, existem diversas técnicas descritas na literatura para detecção do complexo QRS. Neste trabalho, primeiramente, foi utilizado a técnica implementada em [7], desenvolvida por Hamilton e Tompkins que consiste em etapas ordenadas

a que são sujeitos os dados coletados, com objetivo de isolar a onda R e evitar falsas detecções. As etapas são:

- Filtragem do sinal através de filtro passa-faixas (5-15 Hz), a fim de eliminar ruídos referentes a rede elétrica (60 Hz), bem como baixas frequências causadas pela respiração $\approx 1\ Hz$;
- Derivação do sinal, com intuíto de destacar variações no mesmo;
- Elevação do sinal ao quadrado para enfatizar altas frequências e garantir que todos dados sejam positivos;
- Integração do sinal através de uma média móvel, com objetivo de determinar como a energia é distribuída no ECG, permitindo a localização do ponto de referência.

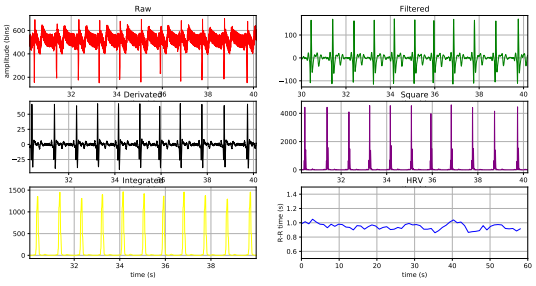
Na Figura 6 é demonstrado um fluxograma da implentação do método no algoritmo criado e na Figura 7 é exibido um exemplo do resultado obtido.

Figura 6. Diagrama de estados de algoritmo Hamilton e Tompkins.



Fonte: Autor

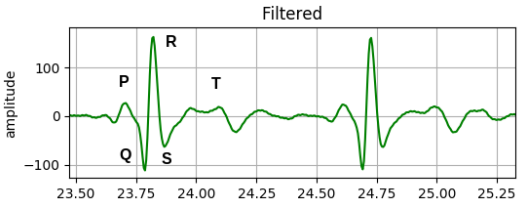
Figura 7. Detecção QRS com algoritmo Hamilton e Tompkins.



Fonte: Autor

Na Fig. 7 foram coletados dados por 60 segundos com o paciente em repouso, sabendo que o intervalo R-R é o período entre batimentos, a curva em azul possui um tendência em torno de 1, o que significa que o paciente possui uma média de 60 bpm. É possível observar que após a integração do sinal, os picos referentes a onda R são completamente isolados das demais componentes, facilitando a obtenção do intervalo R-R para cada batimento detectado. Na Fig.8 é possível identificar as ondas características do ECG em um resultado obtido pelo autor em repouso.

Figura 8. Sinal de ECG do autor em repouso.



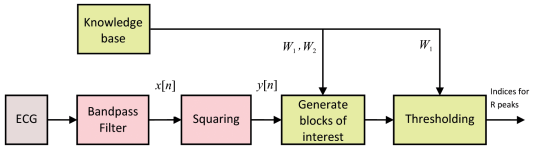
Fonte: Autor

Como o foco do trabalho se mantém na atividade física, o sinal coletado possui muito ruído originado do movimento dos cabos presos aos eletrodos, assim este método demonstrou problemas na decteção neste tipo de sinal. Por tal razão, estudou-se a possibilidade de implementação de um novo método descrito a seguir.

B. Método de Elgendi

Um segundo método foi implementado para detecção do complexo QRS, este tem por autor Mohaemd Elgendi e é descrito em [8]. Este método usa de dois filtros do tipo média movel com eventos relacionados apoiados a uma base de conhecimento. Na Fig. 9 é possível observar o diagrama que descreve seu funcionamento.

Figura 9. Sinal de ECG do autor em repouso.



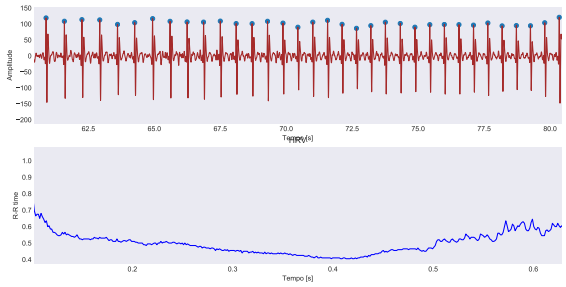
Fonte: [8]

A principal ideia do algoritmo é usar uma base de conhecimento para auxiliar nas tomadas de decisão, tanto na geração de blocos de interesse quanto nos *thresholds*. Desta forma w_1 e w_2 representam o período de duração do complexo QRS e a duração de um batimento cardíaco, respectivamente.

O algoritmo procura pelo pico da onda R dentro de uma janela de tempo dada pelos valores de w_1 e w_2, o que aumenta a precisão para sinais ruidosos, já que o método estima o tempo médio para acontecer um batimento, impossibilitando que batimentos sejam negligenciados.

Na Fig. 10 é demonstrado um exemplo da implementação do algoritmo Elgendi, para este caso, o autor inicialmente estava sob atividade física intensa e posteriormente mais leve.

Figura 10. Detecção QRS com algoritmo Elgendi.



Fonte: Autor

Com a Fig. 10 é possível observar o aumento do batimento cardíaco inicialmente pela diminuição do intervalo R-R e, posteriormente uma leve queda no batimento com o aumento do intervalo causado pela redução da intensidade do exercício.

IV. *MACHINE LEARNING*

Até o momento o modelo proposto tem a capacidade de dado um sinal de ECG, extrair a variabilidade cardíaca e assim retornar a frequência cardíaca do indivíduo, entretanto, uma frequência cardíaca na casa dos 110 bpm pode indicar atividade física em um grupo de pessoas e em outro grupo não se caracterizar, devido a fatores como idade e indíce de gordura corporal como descrito em [9]. Portanto, com o objetivo de efetuar inferências sobre o estado do indivíduo, foram utilizadas técnicas de *Machine Learning* neste trabalho.

Machine Learning é um campo da da área de computação que visa o "aprendizado de máquina", que faz parte do conceito de inteligência artificial, a qual estuda meios para que máquinas possam fazer tarefas que seriam executadas por pessoas. Neste trabalho será utlizada a abordagem de aprendizado de máquina supervisionado.

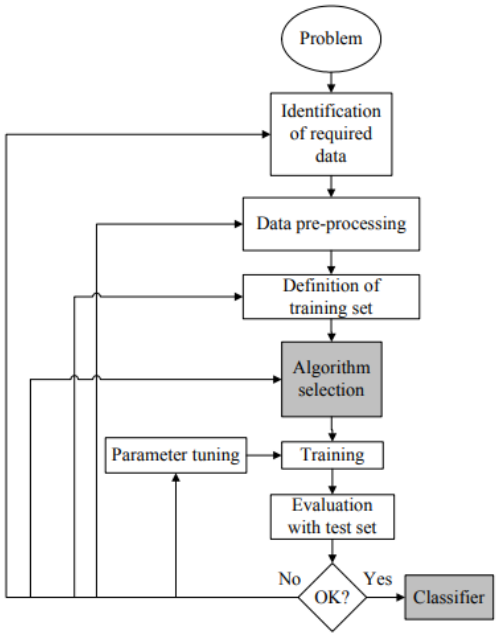
Como descrito por [10], o aprendizado de máquina supervisionado busca algoritmos que raciocinam a partir de instâncias fornecidas na fase de treinamento para produzir hipóteses gerais, que então fazem previsões sobre instâncias futuras, na Fig.11 é possível vizualizar um fluxograma geral de um algoritmo de aprendizado de máquina supervisionado.

Para este trabalho foram implementados dois algoritmos de aprendizado de máquina com abordagem supervisonada, *Random Forest* e *K-Nearest Neighbors*, ambos com o objetivo de classificar as amostras entre repouso e atividade física

A. *Random Forest*

Florestas Aleatórias (*Random Forest*) é um algoritmo de aprendizado supervisionado que cria uma combinação

Figura 11. Algoritmo geral de aprendizado de máquina supervisionado.



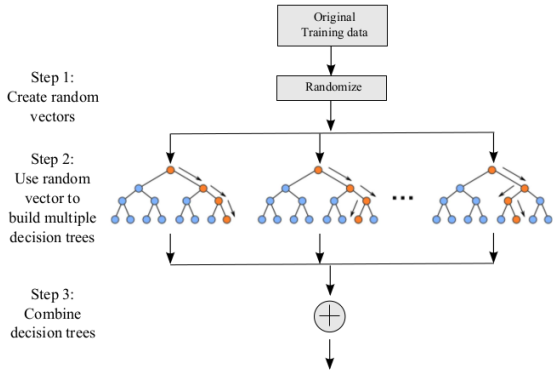
Fonte: [10]

(*ensemble*) ou "floresta" do algoritmo Árvore de Decisão criado por Breiman (2001). Enquanto o algoritmo de Árvore de Decisão, visa a construção total da estrutura baseado nos dados do modelo, o algoritmo de Florestas Aleátorias, parte da antiga premissa de dividir para consquistar, construindo "n" árvores com um conjunto menor de dados, selecionado aleatoriamente dos dados originais, este subconjunto é chamado de *Bootstrap*.

Como descrito em [11], Florestas Aleatórias correlacionam as árvores de decisão na floresta via randomização de atributos o que leva a uma melhoria em relação às árvores tradicionais e reduz a variância quando calculada a média das árvores (Breiman, 2001). *Random Forest* tem como principais características ser considerado um algoritmo muito fácil e acessível, é menos suceptível a um dos maiores problemas de *machine learning*, que é o *overfitting* ou sobreajuste, devido ao fato de geralmente a população de árvores ser na casa de centenas, e pela aleatoriedade na escolha dos dados usados em cada árvore. Em contraponto a maior limitação do Floresta Aleatória é que uma quantidade muito grande de árvores pode tornar o algoritmo lento.

Com o algoritmo treinado, a amostra a ser classificada será submetida a todas as árvores de decisão, o grupo pertencente da amostra será a classe com o maior número de votos das árvores, como demonstra a Fig.12.

Figura 12. Metodologia do algoritmo *Random Forest*.



Fonte: [11]

B. *K-Nearest Neighbors (KNN)*

O segundo método implementado foi o algoritmo de aprendizado supervisonado *K-Nearest Neighbors* (KNN) ou em tradução livre o "K vizinho mais próximo", este algoritmo é um dos mais fundamentais e simples métodos, como descrito em [12]. O KNN é um classificador onde o aprendizado é baseado "no quão similar" é um dado (um vetor) do outro.

O algoritmo efetua uma medida de distância entre o dado a ser classificado, e os dados já classificados, esta medida entre dois pontos pode ser a distância Euclidiana, Manhattan, Minkowski, Ponderada entre outras. As estapas de execução do algoritmo são:

1. Recebe um novo dado não classificado;
2. Mede a distância (Euclidiana, Manhattan, Minkowski ou Ponderada) do novo dado com todos os outros dados que já estão classificados;
3. Obtém as K menores distâncias;
4. Verifica a classe de cada um dos dados que tiveram a menor distância e conta a quantidade de cada classe;
5. Adota como resultado a classe que mais apareceu dentre os dados que tiveram as menores distâncias;
6. Classifica o novo dado com a classe tomada como resultado da classificação.

V. MÉTRICAS DE DESEMPENHO

Uma das questões mais corriqueiras da área de *Machine Learning* é como determinar o desempenho do método proposto, para avaliar este tipo de atributos existem alguns indicadores, como a acurácia, precisão, *recall*, entre outros. O entendimento de alguns conceitos faz-se necessário para a avaliação destes indicadores, são eles:

- *True positive* (TP): Casos em que o classificador retornou atividade e o estado realmente era de atividade;
- *False positive* (FP): Casos em que o classificador retornou atividade e o estado era de repouso;
- *True Negative* (TN): Casos em que o classificador retornou repouso e o estado realmente era repouso;
- *False Negative* (FN): Casos em que o classificador retornou repouso e o estado era de atividade.

A. Matriz de Confusão

A matriz de confusão é uma tabela que demonstra a frequência de classificação para cada classe do modelo. Usando um exemplo com 100 dados hipotéticos que mantêm as mesmas características deste trabalho, apenas para facilitar o entendimento da matriz, é apresentado na Fig. 13 um exemplo de seu funcionamento.

Figura 13. Tabela-exemplo de matriz de confusão utilizando 100 dados.

Matriz de confusão		Valores preditos	
		Atividade	Repouso
Valores reais	Atividade	52 (TP)	23 (FP)
	Repouso	15 (FN)	10 (TN)

Fonte: Autor.

Para os dados apresentados na Fig. 13, são 52 acertos do algoritmo, ou seja, na identificação do estado de atividade, de 100 dados, 52 foram classificados corretamente.

B. Acurácia

A acurácia é a frequência em que o classificador acerta, considerando todas as classes e é descrita pela Eq.1

$$Acurácia = \frac{TP + TN}{Total} \tag{1}$$

C. Precisão

A precisão é a taxa de acerto para a classe que é dita como correta, ou *true positive*, a Eq.2 descreve este indicador.

$$Precisão = \frac{TP}{TP + FP} \tag{2}$$

D. Sensibilidade

A sensibilidade ou *recall* indica a relação entre as previsões positivas realizadas corretamente e todas as previsões que realmente são positivas, na Eq. 3 é possível observar esta relação.

$$Sensibilidade = \frac{TP}{TP + FN} \tag{3}$$

E. Especificidade

A especificidade indica a relação entre as previsões negativas realizadas corretamente e todas as previsões que realmente são negativas, na Eq. 4 é demonstrado esta relação.

$$Especificidade = \frac{TN}{TN + FP} \tag{4}$$

F. *F1-Score*

A *F1-Score* é um modo de visualizar as métricas precisão e sensibilidade juntas. A Eq. 5 descreve o cálculo para esta métrica.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

G. Curva AUC-ROC

A curva AUC-ROC, do inglês *Area Under the Curve - Receiver Operating Characteristic*, é um método gráfico para avaliação, organização e seleção de sistemas de diagnóstico ou predição. Esta curva é a representação gráfica da especificidade pela sensibilidade. Segundo [13], ROC é a curva de probabilidade e AUC representa o grau ou medida de separabilidade, assim, quanto maior o AUC, simplificadamente, o modelo distingui melhor uma classe da outra.

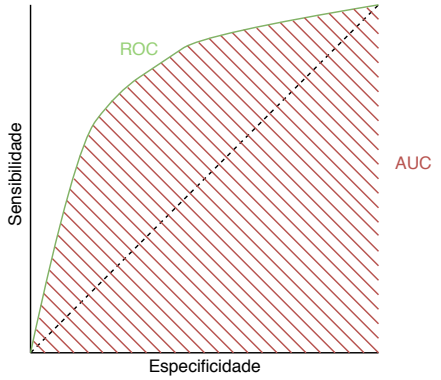
Na Fig. 14 é demonstrado um exemplo de curva AUC-ROC, onde é possível visualizar ambas partes do gráfico (ROC e AUC).

VI. RESULTADOS

Com a etapa de aquisição de sinais concluída, foi então submetido o sinal de ECG ao método de detecção de QRS de Elgendi [8] para obtenção do sinal do HRV, como demonstra a Fig.10. A partir do sinal HRV foram extraídas as *features*, que são descritas na Tab.I.

O método utilizado para extração de *features* consiste em efetuar uma janela móvel ao longo do sinal HRV, calculando-as de dez em dez pontos. Dado um sinal de HRV, formado por "n" pontos, onde os pontos são os batimentos de um determinado indivíduo, do 1° até o 10° ponto formam uma amostra, do 2° até o 11° formam outra amostra, até o final do sinal, onde a amostra é composta pelos valores de todas as *features*.

Figura 14. Exemplo de curva AUC-ROC.



Fonte: Autor.

Para este trabalho, foram selecionadas apenas *features* no domínio do tempo, visto que seria possível a aquisição do domínio da frequência, bem como *features* não-lineares. Foi seguido esta metodologia, devido dada a aplicação desejável, as *features* no domínio do tempo já eram suficientes, e tendo em vista que a extração de *features* em diferentes domínios demandaria um aumento no custo computacional, o que para esta aplicação não é interessante.

Tabela I. Descrição das *features*

Features	Descrição
Mean	Média do intervalo R-R
SDNN	Desvio Padrão do intervalo R-R
RMSSD	Raiz quadrada média do intervalo R-R
Median	Mediana do intervalo R-R
Var	Variância do intervalo R-R
Range	Maior variação do intervalo R-R
CVSD	Coeficiente de variação do intervalo R-R

Fonte: Autor

Para este trabalho foram obtidos sete sinais de ECG dos autores, com o seguinte protocolo: um minuto de repouso, dois minutos em atividade fisíca intensa em uma bicicleta ergométrica e dois minutos em repouso.

Com os dados obtidos para todos os sete sinais resultaram em 3,342 amostras, estas amostras formam um *frame* de dados, onde as linhas são as *features* e as colunas são todas as amostras, um exemplo deste *frame* é mostrado na Tab.II.

A. Fase de Treino

Com a preparação dos dados concluída, é então passada à fase de treino dos classificadores, como já des-

crito este trabalho optou pelos classificadores *Random Forest* e KNN. Para o algoritmo *Random Forest* alguns parâmetros setados foram a criação de 100 árvores sem profundidade máxima, para o KNN foi escolhido o *range* K de 3 vizinhos e a distância Euclidiana. Para a fase de treino o conjunto de amostras foi separado aleatoriamente em 70% para treino e 30% para teste.

Na fase de treino é adicionado mais uma linha na tabela de dados denominada "activity", que diz respeito a classificação da amostra, uma vez que, estes algoritmos de *machine learning* seguem a abordagem de aprendizado supervisionado, a codificação adotada foi zero (0) para repouso e um (1) para atividade.

Tabela II. *Frame* de dados utilizados nos classificadores

Features	s1	s2	s3	sn	s3342
mean	0.777346	0.784846	0.785846	...	0.800180
sdnn	0.092493	0.081944	0.080988	...	0.027652
rmssd	0.074407	0.069519	0.067096	...	0.016400
sdsd	0.073906	0.069464	0.067039	...	0.016269
median	0.802513	0.802513	0.802513	...	0.802513
var	0.008555	0.006715	0.006559	...	0.000765
range	0.440007	0.345006	0.345006	...	0.100002
cvsd	0.095719	0.088577	0.085381	...	0.020496
activity	0	0	0	...	1

Fonte: Autor

B. Fase de Teste

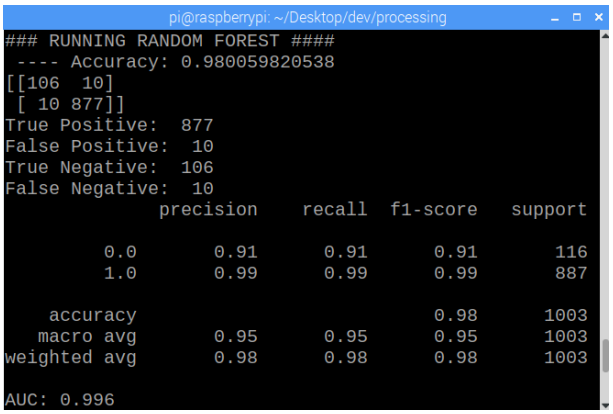
Na fase de teste, o classificador com seu modelo já ajustado, é submetido aos outros 30% dos dados que não foram utilizados na fase de treino, para avaliar o seu desempenho, é importante ressaltar que nesta etapa o classificador não tem acesso ao campo "activity" anteriormente descrito.

Exemplos de execução são demonstrados pela Fig.15 e Fig.16, onde o algoritmo está executando os classificadores no terminal de um Raspberry PI, é mostrado a acurácia do classificador, bem como a matriz de confusão e o *score* AUC.

As execuções da Fig.15 e Fig.16, foram apenas testes de execução de *frames* de dados. A Tab.III demonstra os resultados reais obtidos no *Random Forest*, enquanto a Tab.IV os resultados do algoritmo KNN. É possível verificar que o desempenho foi satisfatório, tendo 94,44% de acurácia no *Random Forest* e 95,11% no KNN, com uma precisão de 97% e 96% ao detectar *True Positive*, ou seja, ao detectar que o indivíduo está em atividade física. A maior taxa de erro se dá na detecção de repouso, com uma precisão de 80% para o *Random Forest*, entretanto justifica-se devido a disparidade de proporção nos dados em atividade para dados em repouso.

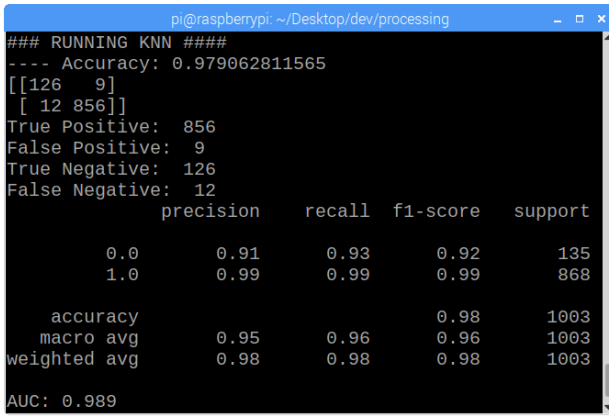
Dado os resultados obtidos, foi avaliado a importância de cada *feature* para o algoritmo *Random Forest* e plotado no gráfico da Fig.17. É possível verificar que as *features* tem importância similar, não havendo grande disparidade de importância, uma vez que, caso houvesse

Figura 15. Execução do algoritmo *Random Forest* em um Raspberry PI



Fonte: Autor

Figura 16. Execução do algoritmo KNN em um Raspberry PI



Fonte: Autor

Tabela III. *Random Forest*: Resultados com todas as *features*

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.80	0.79	0.80	0.9444
1	0.97	0.97	0.97	0.9444

Fonte: Autor

Tabela IV. KNN: Resultados com todas as *features*

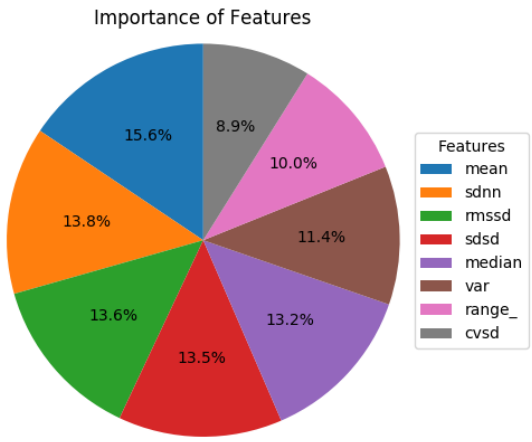
Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.88	0.78	0.82	0.9511
1	0.96	0.98	0.97	0.9511

Fonte: Autor

um grupo de *features* com grande disparidade de importância, seria ideal a não utilização das mesmas, com o objetivo de economizar processamento.

Porém, tendo em vista que este trabalho tem como objetivo um modelo que possa ser usado em um sistema embarcado, foram executados os algoritmos com apenas as três *features* mais importantes, reduzindo a dimensionalidade e consequentemente o custo computacional no processamento, para verificar o impacto que a remoção

Figura 17. Importância de cada *feature*



Fonte: Autor

de *features* teria no desempenho dos classificadores.

Os resultados obtidos são descritos na Tab.V e Tab.VI.

Tabela V. *Random Forest*: Resultados com três *features*

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.83	0.76	0.79	0.9425
1	0.96	0.97	0.97	0.9425

Fonte: Autor

Tabela VI. KNN: Resultados com três *features*

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.71	0.80	0.75	0.9301
1	0.97	0.95	0.96	0.9301

Fonte: Autor

Como demonstrado nos resultados da Tab.V e Tab.VI, o modelo não perdeu muito desempenho reduzindo o número de *features*, o que é interessante para uma possível implementação embarcada, podendo ser reduzido as características de capacidade de processamento do *hardware* necessário.

Entretanto, como descrito na literatura o maior problema da área de *machine learning* é o *overfitting*, ou seja, o classificador só ter desempenho alto para os indíviduos os quais ele foi treinado. Para avaliar este fenômeno, foi selecionado um outro indivíduo para a coleta de dados de ECG. O protocolo de teste foi de três (3) minutos de repouso, doze (12) minutos de atividade física moderada e três minutos (3) de recuperação, assim será modificado tanto o indivíduo selecionado, como o protocolo e tipo de atividade.

Para a nova aquisição de resultados, foram treinados os classificadores com todos as amostras dos sete sinais iniciais de ECG dos autores, onde os mesmos são submetidos a atividade física intensa, e será avaliada a performance do modelo com um outro indivíduo, o qual foi submetido a atividade física moderada. O modelo foi executado com três *features*, e os resultados podem ser visualizados nas tabelas Tab.VII e Tab.VIII, respectivamente para o algoritmo *Random Forest* e KNN.

Tabela VII. *Random Forest*: Resultados teste de *overfitting*

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.70	0.78	0.74	0.9365
1	0.97	0.96	0.96	0.9365

Fonte: Autor

Tabela VIII. KNN: Resultados teste de *overfitting*

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.54	0.73	0.62	0.8997
1	0.96	0.92	0.94	0.8997

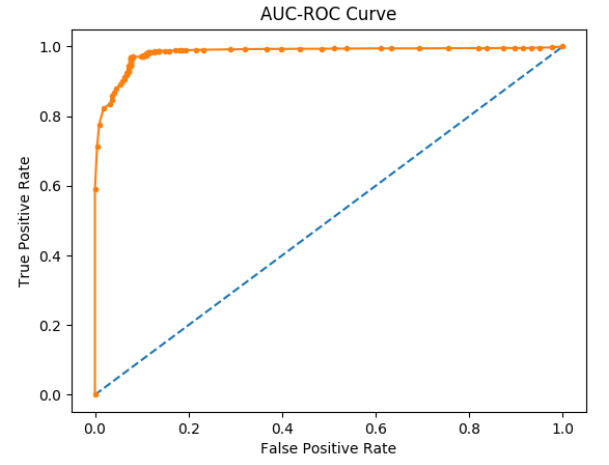
Fonte: Autor

É possível observar que a performance diminuiu, no entanto ainda está dentro dos padrões desejados para este trabalho, tendo em vista que o sinal coletado seria uma situação de dia-a-dia de atividade física moderada.

Para investigar as propriedades da janela móvel adotada para extração de *features*, foi então alterado o número de pontos que compõe uma amostra, que anteriormente eram de 10, agora foi aumentado este valor para 20 e verificado o desempenho do modelo, o resultado pode ser visualizado na Tab.IX e Tab.X, respectivamente para o algoritmo *Random Forest* e KNN.

Por fim, foi plotado a curva ROC, de um resultado obtido com o *Random Forest*, que é destacada na Fig.18 bem como calculado o índice AUC, que para este resultado foi de 0.98, como descrito anteriormente a curva ROC relaciona a taxa de verdadeiros positivos com a taxa de falsos positivos, ou seja, numero de vezes que o classificador acertou a predição conta o número de vezes que o classificador errou a predição.

Figura 18. Curva AUC-ROC para o classificador *Random Forest*



Fonte: Autor

É possível destacar que o algoritmo *Random Forest* se

mostrou mais eficiente, dado que a construção de árvores de decisão com dados aleatórios favorece este tipo de aplicação, outro ponto a se destacar é que aumentando o tamanho da janela de pontos por amostra, o desempenho de ambos os classificadores aumentaram, foram realizados testes com tamanho de janela maior, que teve como resultado um leve aumento na performance, no entanto, a medida que se aumenta o número de pontos, diminui a velocidade de predição do modelo, tendo em vista que cada ponto é um pico de onda R, por consequência, um batimento cardíaco, para aplicações embarcadas de tempo real não é interessante um tamanho de janela muito grande.

Tabela IX. *Random Forest*: Resultados teste de tamanho da janela

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.78	0.90	0.84	0.9614
1	0.99	0.97	0.98	0.9614

Fonte: Autor

Tabela X. KNN: Resultados teste de tamanho da janela

Class	*Precision*	*Recall*	F1	*Accuracy*
0	0.70	0.88	0.78	0.9456
1	0.98	0.95	0.97	0.9456

Fonte: Autor

VII. CONCLUSÕES

Este trabalho teve por objetivo a identificação e classificação de estados de atividade física a partir da Variabilidade da Frequência Cardíaca (HRV), calculada a partir dos intervalos inter-batimentos (Intervalos RR). Tomou-se como base que alterações no HRV são normais e esperadas ao se iniciar e terminar atividades físicas, conforme detalhado por [2].

A aquisição dos sinais a serem utilizados se deu inicialmente com os autores em repouso e posteriormente se exercitando de maneira gradual em bicicleta ergométrica. Os dados foram coletados e armazenados através do conjunto de eletrodos conectados ao *shield* do microcontrolador.

A metodologia utilizada para a detecção dos picos R, para o cálculo do intervalo RR, foi o a técnica desenvolvida por Mohaemd Elgendi, a qual evita falsas detecções devido às etapas de filtragem e processamento do sinal, o que possibilitou a construção do sinal de HRV relativo aos ECGs. Em seguida separou-se o sinal em pequenas amostras a partir de uma média móvel, além de se extrair dados estatísticos relativos às amostras, assim como a pré-classificação de cada amostra, necessária para o aprendizado supervisionado.

Uma vez extraídas e rotuladas as amostras de HRV, passou-se para a etapa de treinamento e classificação dos dados utilizando algoritmos de aprendizado de máquina supervisionados. Foi utilizado o algoritmo de Florestas Aleatórias, assim como o algoritmo KNN, utilizando amostras separadas para treino e validação de desempenho.

Os resultados obtidos a partir da combinação entre o método de Elgendi juntamente à classificação do algoritmo de Florestas Aleatórias foram satisfatórios, uma vez que tiveram bom desempenho nas métricas utilizadas, em especial em relação à curva ROC, indicado pelo índice AUC de 0,98.

O baixo custo de computação do algoritmo de classificação de Florestas Aleatórias, aliado à eficácia da técnica de Elgendi, permitiram que o processamento pudesse ser feito em dispositivos embarcados mesmo com suas restrições de processamento, o que indica que a metodologia aplicada ao trabalho tem condições de atuar em situações práticas, como em dispositivos esportivos ou aplicações médicas.

Como trabalho futuro se pretende aumentar o número de amostras a serem utilizadas no treinamento, de modo a abranger pessoas de diferentes idades e condições de saúde, de modo a melhorar o desempenho do algoritmo.

[1] D. Hernando, N. Garatachea, J. A. Casajús, and R. Bailón, "Comparison of Heart Rate Variability Assessment During Exercise from Polar RS800 and ECG," in *2017 Computing in Cardiology Conference (CinC)*, 2017.

[2] L. C. M. Vanderlei, C. M. Pastre, I. F. Freitas, and M. F. de Godoy, "Geometric indexes of heart rate variability in obese and eutrophic children," *Arquivos brasileiros de cardiologia*, 2010.

[3] G. D. Clifford, "Signal Processing Methods for Heart Rate Variability," Ph.D. dissertation, 2002.

[4] S. L. Lin, T. Y. Lin, C. Y. Huang, Y. L. Hsu, and H. C. Chang, "The comparison study between distinct physical fitness norms and their ECG signals under graded exercise intensities and recovery," in *Proceedings of the 2017 IEEE International Conference on Applied System Innovation: Applied System Innovation for Modern Technology, ICASI 2017*, 2017.

[5] "Olimex," Olimex LTD. [Online]. Available: www.olimex.com

[6] "Raspberry pi," Raspberry. [Online]. Available: www.raspberrypi.org/

[7] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE Transactions on Biomedical Engineering*, 1985.

[8] M. Elgendi, "Fast QRS Detection with an Optimized Knowledge-Based Method: Evaluation on 11 Standard ECG Databases," *PLoS ONE*, 2013.

[9] S.-W. Niu, J.-C. Huang, C. Szu-Chia, H. Y.-H. Lin, I.-C. Kuo, P.-Y. Wu, Y.-W. Chiu, and J.-M. Chang, "Association between Age and Changes in Heart Rate Variabi-

lity after Hemodialysis in Patients with Diabetes," *Aging Neurosci.*, 2018.

[10] S. Kotsiantis, "Supervised Machine Learning: A Review of ClassificationTechniques," 2007.

[11] M. Malekipirbazari and V. Aksakalli, "Risk Assessment in Social Lending via Random Forests," *Expert Systems with Applications*, 2015.

[12] J. D. Novakovic, M. Papic, S. S. Ilic, and A. Veljovic, "EXPERIMENTAL STUDY OF USING THE K-NEAREST NEIGHBOUR CLASSIFIER WITH FILTER METHODS," 2016.

[13] S. Narkhede, "Understanding AUC - ROC Curve - Towards Data Science," 2018. [Online]. Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5