

# **INVESTIGAÇÃO DE ANOMALIAS CARDÍACAS EM DATASETS DE ELETROCARDIOGRAMAS - PROPOSTA DE PESQUISA CIENTÍFICA -**

**Henrique L. Richa<sup>1</sup>, Renan B. Biavati<sup>1</sup>, Vitória Izabel M. Pinto<sup>1</sup>**

<sup>1</sup>Ciência da Computação – Pontifícia Universidade Católica do Paraná (PUCPR)  
Curitiba, PR – Brasil

henrique.levandoski@pucpr.edu.br, renan.belem@pucpr.edu.br,  
vitoria.pinto@pucpr.edu.br

**Abstract.** This study explores the application of machine learning techniques in the analysis of electrocardiograms (ECG), aiming to improve the detection of cardiac anomalies. With the increasing complexity of ECG data, machine learning emerges as a promising tool for more accurate and objective diagnostics. The research uses algorithms such as SVM, XGBoost, and MLP to identify patterns in data from the MIT-BIH Arrhythmia Database, focusing on the comparison of various models to maximize accuracy, specificity, and other evaluation metrics. The results indicate that supervised learning models are effective in identifying cardiac anomalies, contributing to the application of advanced technologies in healthcare and providing a solid foundation for future research in machine learning-assisted diagnostics.

**Resumo.** Este trabalho explora a aplicação de técnicas de aprendizado de máquina na análise de eletrocardiogramas (ECG), visando aprimorar a detecção de anomalias cardíacas. Com o aumento da complexidade dos dados de ECG, o aprendizado de máquina surge como uma ferramenta promissora para diagnósticos mais precisos e objetivos. A pesquisa utiliza algoritmos como SVM, XGBoost e MLP para identificar padrões em dados do MIT-BIH Arrhythmia Database, destacando-se pela comparação de diversos modelos para maximizar a acurácia, especificidade e outras métricas de avaliação. Os resultados indicam que modelos de aprendizado supervisionado são eficazes na identificação de anomalias cardíacas, contribuindo para a aplicação de tecnologias avançadas na área da saúde e oferecendo uma base sólida para futuras pesquisas no campo de diagnóstico assistido pelo aprendizado de máquina.

## **1. Introdução**

### **1.1. Sobre o conteúdo da Introdução**

O eletrocardiograma (ECG) é uma ferramenta essencial na medicina, fornecendo informações importantes sobre a saúde cardíaca do paciente. Tradicionalmente, sua interpretação depende da experiência do médico, mas, com o aumento de dados e a complexidade das condições cardíacas, abordagens mais objetivas se tornam necessárias [Schwarz 2009], [Wasimuddin et al. 2020]. Segundo [Breen et al. 2022], o ECG é o procedimento cardiovascular mais usado, e cardiologistas identificam corretamente entre

53% e 96% das anomalias. Como erros na interpretação podem ser fatais, o ideal seria alcançar 100% de acerto. Este trabalho explora a eficácia de métodos de aprendizado de máquina para detectar anomalias no ECG, revisando literatura e dados para identificar as melhores práticas, contribuindo com uma base para futuras pesquisas.

## 1.2. Objetivos

O objetivo geral do trabalho é aplicar e avaliar técnicas de aprendizado de máquina sobre dados de ECG, para maior eficácia nos diagnósticos de possíveis doenças cardíacas, e com isso, testar duas hipóteses:

- Hipótese nula: Nenhum método analisado apresenta melhoria possível a ser aplicada quanto aos resultados obtidos.
- Hipótese alternativa: Encontradas mudanças que promovem resultados melhores dos métodos utilizados.

Para alcançar este objetivo, o estudo tem os seguintes objetivos específicos:

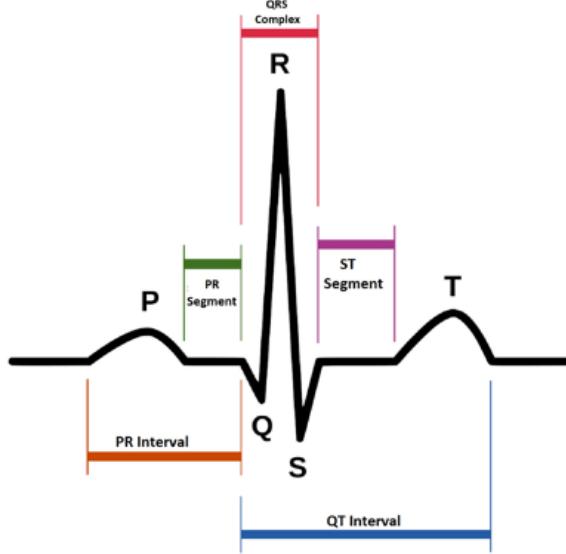
1. Realizar uma revisão da literatura para entender o campo da análise de ECG usando aprendizado de máquina;
2. Utilizar o pré-processamento de dados para configurar e refinar a aplicabilidade das técnicas de aprendizado de máquina.
3. Comparar e contrastar diferentes técnicas de aprendizado de máquina em termos de precisão, eficácia e aplicabilidade na detecção de anomalias em ECGs, investigando possíveis melhorias nos resultados obtidos com as técnicas de forma a garantir a correta análise das anomalias;
4. A partir das técnicas de aprendizagem de máquina empregadas, otimizá-las, afim de refinar ainda mais os resultados encontrados;
5. Com os resultados otimizados obtidos, propor experimentações que podem gerar ainda mais melhorias através de técnicas de otimização de modelos alternativas, como técnicas de ensemble.
6. Verificar se com a pesquisa e aplicação dos métodos e técnicas de otimização, foi possível testar as hipóteses iniciais.

## 2. Revisão da Literatura

### 2.1. Introdução ao Eletrocardiograma

O eletrocardiograma, desenvolvido no início do século XX, é uma ferramenta essencial para monitorar a atividade elétrica do coração, possibilitando a detecção de diversas condições cardíacas. Eletrodos na pele captam variações de íons cardíacos, gerando um gráfico com ondas e complexos (P, QRS e T) que representam fases da atividade cardíaca e ajudam a avaliar a saúde do paciente [Jambukia et al. 2015], podendo identificar diversas condições cardíacas.

**Figure 1. Morfologia de um ECG saudável**



Fonte: [Aziz et al. 2021]

## 2.2. Aprendizado de Máquina

Criado por Arthur Samuel em 1959, o aprendizado de máquina (ML) permite que computadores aprendam a partir de dados, diferentemente da inteligência artificial (IA), que abrange a capacidade de máquinas tomarem decisões automaticamente. Em ML, algoritmos analisam dados para prever saídas e identificar padrões [Handelman et al. 2018]. De acordo com [Netto and Maciel 2021], o ML usa métodos de classificação e detecção de padrões para gerar insights autonomamente. Contudo, a modelagem enfrenta problemas como underfitting, onde o modelo é simplista demais, e overfitting, em que o modelo se ajusta excessivamente ao treino. Técnicas de regularização, como Lasso e Ridge, ajudam a melhorar a generalização dos modelos [Kolluri et al. 2020]. Diversas métricas são empregadas para avaliar a performance dos modelos:

Para avaliar a eficácia dos modelos, diversas métricas são usadas:

- **Matriz de Confusão:** Apresenta verdadeiros e falsos positivos e negativos.
- **Precisão:** Proporção de previsões positivas corretas:

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (1)$$

- **Acurácia:** Proporção total de previsões corretas:

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + FN + VN} \quad (2)$$

- **F1-Score:** Média harmônica entre precisão e recall, útil para classes desbalanceadas:

$$\text{F1-Score} = 2 \times \frac{\text{Precisao} \times \text{Recall}}{\text{Precisao} + \text{Recall}} \quad (3)$$

- **Recall:** Proporção de positivos reais identificados corretamente:

$$\text{Recall} = \frac{VP}{VP + FN} \quad (4)$$

- **Especificidade:** Capacidade de detectar resultados negativos:

$$\text{Especificidade} = \frac{VN}{VN + FP} \quad (5)$$

Essas métricas ajudam a ajustar modelos e a equilibrar tipos de erro. Em áreas como a saúde, o recall alto é importante para identificar todos os casos positivos, embora outras métricas também sejam necessárias para evitar falsos positivos em excesso. O aprendizado de máquina se divide em três categorias principais: aprendizado supervisionado, não supervisionado e por reforço [Bonacorso 2018], [Wu and Nagahashi 2014].

### 2.2.1. Aprendizado Supervisionado

O aprendizado supervisionado consiste em treinar um algoritmo com dados rotulados para que ele aprenda a prever saídas corretas em novos dados. É amplamente utilizado para classificações (rótulos discretos) e regressões (rótulos contínuos) [Nasteski 2017]. Os principais métodos incluem:

- **XGBoost:** Algoritmo otimizado para árvores de decisão, conhecido pela rapidez e eficácia. Utiliza regularização para evitar overfitting e combina perda com regularização para equilibrar adaptação e generalização [Li et al. 2020].
- **AdaBoost:** Ajusta iterativamente pesos nos dados de treino para melhorar “aprendizes” fracos, priorizando os exemplos mais difíceis em cada iteração [Wang 2012].
- **RandomForest:** Combina várias árvores de decisão para tarefas de classificação ou regressão, capturando automaticamente interações complexas entre variáveis [Rigatti 2017].
- **K-Nearest Neighbors (KNN):** Classifica um ponto com base nos ”K” vizinhos mais próximos, mas torna-se lento em grandes conjuntos de dados. A distância é calculada, geralmente, usando a métrica euclidiana [Faceli et al. 2021].
- **Árvore de Decisão:** Estrutura de grafo onde nós representam decisões e ramos representam condições. Cada nó interno indica um atributo, facilitando a classificação de grupos [Nasteski 2017].
- **Support Vector Machine (SVM):** Algoritmo que maximiza a margem entre classes, aplicável tanto para classificações quanto para regressões. Seu uso em grandes datasets é limitado devido à complexidade computacional [Abdullah and Abdulazeez 2021].
- **GradientBoosting:** Semelhante ao XGBoost, mas cria uma árvore por vez, sendo mais lento por não suportar multithread [Ali et al. 2023].
- **Multilayer Perceptron (MLP):** Rede neural com múltiplas camadas de neurônios que utiliza o backpropagation para minimizar erros de previsão durante o treinamento [Popescu et al. 2009].

### 2.3. Otimização de Hiperparâmetros

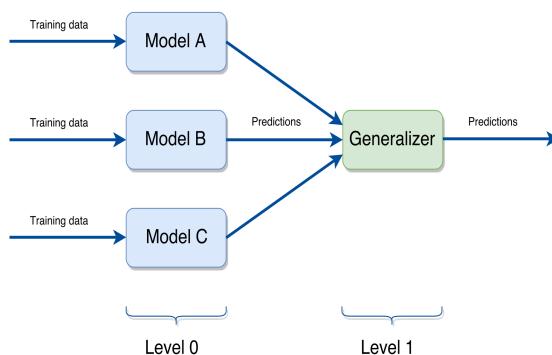
A otimização de hiperparâmetros é muito importante no desenvolvimento de modelos de aprendizado de máquina, visando maximizar o desempenho em variadas aplicações. Diferentes algoritmos de ML requerem a escolha cuidadosa de hiperparâmetros, que são

definidos antes da execução dos algoritmos e alguns casos distintos podem ser modificados durante o uso deles, como é o caso das redes neurais. Porém, não há uma regra certa na escolha dos hiperparâmetros, sua escolha pode variar de aplicação a aplicação. A escolha ideal destes pode influenciar significativamente na arquitetura e eficácia do modelo. Comumente, a escolha do hiper parâmetros é feita manualmente, testando seu desempenho geral no aprendizado dos dados [Yang and Shami 2020].

#### 2.4. Stacking

Stacking é uma técnica de ensemble amplamente usada no aprendizado de máquina que combina previsões de modelos à escolha do desenvolvedor, a fim de melhorar sua precisão. Inicialmente, estes modelos são treinados individualmente, com suas próprias previsões, e então, são passados como entrada para um modelo generalizador, conhecido como meta-modelo que aprende a melhor maneira de combinar tais previsões para fazer enfim sua previsão final [Divina et al. 2018], como pode-se notar na Figura 11.

**Figure 2. Representação Stacking**



**Fonte:** [Divina et al. 2018]

#### 2.5. Aprendizado de Máquina na Saúde

O aprendizado de máquina tem o potencial de aprimorar a precisão de diagnósticos e oferecer tratamentos mais personalizados aos pacientes [Handelman et al. 2018], se propondo a ser uma ferramenta auxiliar e potencializadora à saúde [Moreira et al. 2020].

À medida que a tecnologia evolui, o campo da detecção de anomalias em ECG tem se beneficiado significativamente do aprendizado de máquina. Diversas pesquisas têm explorado essa intersecção entre saúde e tecnologia. A seguir, estão os destaques de alguns estudos pertinentes sobre o assunto.

[Ahsanuzzaman et al. 2020] utilizaram os métodos KNN, Árvore de Decisão, ANN com pesos ponderados, SVM, LSTM com pesos ponderados e uma abordagem com Ensemble. Seus melhores resultados foram utilizando Ensemble com uma acurácia de 97.664%, precisão de 96.904% e recall de 97.062%.

[Tsai and Morshed 2022] utilizaram os métodos SVM, KNN, Ensemble Tree, Rede Neural, Árvore de Decisão, Naive Bayes e Regressão Logística. Porém, os pesquisadores utilizaram o dataset BIDMC Congestive Heart Failure (CHF). Seus melhores resultados foram com a Rede Neural, com acurácia de 98.64%, precisão de 99.3% e f1-score de 99.42%.

[Latif et al. 2020] utilizaram cinco métodos de aprendizado de máquina em sua pesquisa: Multilayer Perceptron, Radial Basis Function Network, Logistic Model, Naive Bayes e por último Random Forest. Seus melhores resultados foram obtidos a partir do método Random Forest, com uma acurácia de 97.45%.

[Mi et al. 2023] utilizaram os métodos Back Propagation Neural Network, SVM e KNN. Seus melhores resultados vieram da BP Neural Network, com uma acurácia próxima de 99.1%.

[Jambukia et al. 2015] realizaram uma survey de técnicas e algoritmos de classificação de batimentos cardíacos e encontraram pesquisas que utilizaram Redes Neurais, Máquinas de Vetores de Suporte (SVM), Transformada Discreta de Wavelet, entre outras, porém concluíram que redes neurais, especialmente MLP mostraram-se mais eficazes na classificação de ECG em termos de precisão.

[Apandi et al. 2018] fizeram um levantamento de vários métodos como SVM, redes neurais, etc. Focados em realizar a detecção de arritmias, especialmente usando o banco de dados MIT-BIH.

[Shoughi and Dowlatshahi 2021] implementaram uma classificação automática de doenças cardiovasculares com pré-processamento e um classificador que consiste em rede neural convolucional e rede BLSTM. Os resultados das avaliações sobre um conjunto de dados desequilibrados do MIT-BIH mostram que esta abordagem com treinamento de 50 épocas alcançou cerca de 99% de precisão para todas as cinco categorias, variando de 98,85% a 99,87%.

### 3. Método de Pesquisa

#### 3.1. Caracterização da Pesquisa

Esse trabalho será desenvolvido com abordagem descritiva e caráter experimental. Focado em analisar e identificar anomalias em ECG utilizando métodos de aprendizagem de máquina, com o propósito de auxiliar no diagnóstico prático e preciso de possíveis doenças cardíacas, comparando a eficácia de diferentes métodos de aprendizagem sobre o mesmo conjunto de dados. Os dados a serem analisados provêm do conjunto de dados utilizados são o MIT-BIH Arrhythmia Dataset e o PTB Diagnostic ECG Database, que são públicos e provenientes dos laboratórios no Beth Israel Hospital de Boston (hoje Beth Israel Deaconess Medical Center) e no Instituto de Tecnologia de Massachusetts (MIT).

#### 3.2. Método de Pesquisa

##### 3.2.1. Seleção de Dados

Como objeto para a experimentação foi selecionado um conjunto de dados disponibilizado pelo Kaggle, sendo formado por MIT-BIH Arrhythmia Dataset e o PTB Diagnostic ECG Database, que são dados públicos que podem ser encontradas individualmente no PhysioNet.

- **MIT-BIH Arrythmia Data<sup>1</sup>** Os dados foram recolhidos pela MIT-BIH na parte clínica do Beth Israel Hospital, entre os anos 1975 e 1979. O conjunto de dados

---

<sup>1</sup><https://www.physionet.org/content/mitdb/1.0.0/>

em questão contém 48 registros anotados de 47 indivíduos. Além disso, é dividido em séries de ‘100’ e ‘200’, escolhidos aleatoriamente entre um conjunto de 4000 fitas Holter de 24 horas com o objetivo de incluir uma variedade de fenômenos raros, mas clinicamente importantes. Os registros possuem em média 30 minutos de duração, com dois sinais amostrados a 360 Hz, totalizando aproximadamente 109.000 batimentos.

**Figure 3. Descrição das classes de dados**

AAMI Classes	Heartbeat Types
Normal beats (N)	Normal beats (N), Left bundle branch block (L), Right bundle branch block (R), Atrial escape beat (e), Nodal (junctional) escape beat (j)
Supraventricular ectopic Beats (S)	Atrial premature contraction (A), Aberrated atrial premature beat (a), Nodal (junctional) premature beat (J), Supraventricular premature beat (S)
Ventricular ectopic beats (V)	Ventricular premature contraction (V), Ventricular escape beat (E)
Fusion beats (F)	The fusion of ventricular and normal beat (F)
Unclassifiable beats (Q)	Paced beat (I), Fusion of paced and normal beat (f), Unclassified beat (Q)

**Fonte:** [Goldberger et al. 2000]

- **PTB Diagnostic ECG Database:** O conjunto de dados em questão apresenta 14.552 amostras com uma taxa de amostragem de 125 Hz. As amostras pertencem à 294 pacientes, com idades de 17 a 87 anos, englobando 209 homens e 81 mulheres. Entre os indivíduos em questão, parte deles demonstra uma análise saudável no eletrocardiograma e parte demonstra doenças cardíacas. A coleta foi uma iniciativa da Physikalisch-Technisch Bundesanstalt (PTB), o Instituto Nacional de Metrologia da Alemanha. Os eletrocardiogramas foram coletados sob a supervisão do Professor Michael Oeff, M.D., no departamento de cardiologia da Universidade Clínica Benjamin Franklin.

### 3.2.2. Pré-Processamento e Execução

Para a comprovação das hipóteses iniciais, se utiliza o conjunto de dados mencionados, que são amplamente reconhecidos na área de pesquisa médica. Para busca de resultados limpos e precisos será feito um pré-processamento, filtrando os elementos com limpeza de dados, visando remover registros com muitos elementos nulos, e redução de ruídos, para remoção de elementos que não apresentem sentido ao aspecto da pesquisa.

Com o pré-processamento realizado, se parte para a aplicação dos métodos selecionados KNN, XGBoost, RandomForest, MLPClassifier, GradientBoosting, SVC e AdaBoost. Os métodos serão comparados com seus resultados baseados nos atributos de Precisão, Recall, F1-Score e Acurácia e Especificidade.

Em visão de melhoria dos atributos e de resultados será feita uma experimentação de hiperparâmetros, com alguns conjuntos específicos como por exemplo learning rate (Taxa de aprendizado), learning rate decay (Declínio da taxa de aprendizado) e momentum (Dinâmica).

- **Exclusão do PTB Diagnostic ECG Database:** Após a análise dos datasets MIT-BIH Arrhythmia Database e PTB Diagnostic ECG Database, concluímos que seria muito custoso a tentativa de junção dos dois datasets pela diferença das classes, o PTB Diagnostic ECG Database possuindo apenas “normal” e “anormal” como elementos categóricos não suportaria a especificidade das 5 classes do MIT-BIH, e seria necessário um reducionismo ou contorcionismo de dados, que por sua vez, não garantiria melhoria para as classes do MIT-BIH Arrhythmia Database.

## 4. Análise dos Resultados

Nesta seção, serão discutidos os resultados de cada modelo, com foco em métricas-chave como acurácia, precisão, recall, f1-score, especificidade e a matriz de confusão. Os caderinos criados no Jupyter Notebook para cada modelo estão disponíveis no repositório do GitHub<sup>2</sup>.

### 4.1. Análise do Modelo SVC

O modelo SVC, inicialmente com seus parâmetros padrões apresentou bons resultados, como mostra a Tabela 1:

**Table 1. Resultados Padrões para SVC**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.969	0.97	0.97	0.97	0.9679
Aritmética	0.969	0.94	0.77	0.84	0.9679

**Fonte:** Autoria Própria (2024)

Para o SVC, foram testados diferentes valores para os parâmetros  $C$ ,  $gamma$ ,  $kernel$ ,  $max\_iter$  e  $shrinking$ , e os melhores entre eles provieram de:

- $C$ : 75
- $gamma$ : scale
- $kernel$ : rbf
- $max\_iter$ : 12500
- $shrinking$ : True

Com tais valores para os parâmetros testados, seus resultados estão apresentados na Tabela 2.

**Table 2. Resultados Otimizados para SVC**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.98	0.98	0.98	0.98	0.98
Aritmética	0.98	0.94	0.87	0.90	0.98

**Fonte:** Autoria Própria (2024)

<sup>2</sup><https://github.com/Viihizaa/Projeto-Transformador-I-Prototipo>

#### 4.2. Análise do Modelo KNN

Para começar a execução do KNN, assim como o SVC, foi executado primeiro com seus parâmetros padrões, como mostra a Tabela 3.

**Table 3. Resultados Padrões para KNN**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.974	0.972	0.973	0.972	0.983
Aritmética	0.974	0.922	0.846	0.879	0.983

**Fonte:** Autoria Própria (2024)

Sabendo dos resultados com os parâmetros padrões, uma variação dos valores de seus parâmetros foram experimentadas, e obtêm-se os melhores valores como:

- **n\_neighbors:** 1
- **weights:** uniform
- **algorithm:** auto
- **p:** 1
- **leaf\_size:** 20

Os resultados da combinação de valores dos parâmetros citados estão apresentados na Tabela 4.

**Table 4. Resultados Otimizados para KNN**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.977	0.976	0.977	0.976	0.976
Aritmética	0.977	0.908	0.878	0.892	0.976

**Fonte:** Autoria Própria (2024)

#### 4.3. Análise do Modelo XGBoost

O XGBoost é um modelo que desde sua fase crua, ou seja, com os parâmetros com os valores padrões do método já apresentam excelentes resultados, como mostra a Tabela 5.

**Table 5. Resultados Padrões para XGBoost**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada (Validação)	0.9800	0.98	0.98	0.98	0.9812
Ponderada (Teste)	0.9807	0.98	0.98	0.98	0.9820

**Fonte:** Autoria Própria (2024)

Os testes com o XGBoost inicialmente foram com o parâmetro *n\_estimators*, que, por sua vez, aceita apenas números inteiros, testado com todos os valores de 1 a 1000, tem-se que o melhor resultado para *n\_estimators* é de 944. Que apresentou uma pequena melhora de sua especificidade com média aritmética para validação e teste de 0.0021 e 0.0033, respectivamente. O próximo parâmetro testado foi o *learning\_rate* que dentre os valores testados, o melhor obtido foi de 0.5, com uma melhora minúscula de 0.002 para

especificidade do teste com média aritmética e 0.0006 para especificidade da validação com média aritmética.

Já o próximo parâmetro testado (*max\_depth*) não apresentou nenhuma melhora em seus resultados, então foi apenas utilizado para os próximos testes seu valor padrão. O último parâmetro que foi incluído na otimização do XGBoost que obteve resultados positivos, foi o *min\_child\_weight*, que foi executado com intervalos de 2, entre 1 e 9, ou seja, 1, 3, 5, 7 e 9. Esse foi o último parâmetro dentre os testados com melhores resultados. A Tabela 6 apresenta o final da otimização:

**Table 6. Resultados Otimizados para XGBoost**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada (Validação)	0.9454	0.98	0.98	0.98	0.9844
Ponderada (Teste)	0.9792	0.98	0.98	0.98	0.9855

**Fonte:** Autoria Própria (2024)

Pode-se concluir que o modelo XGBoost, por mais que seus valores padrões já eram muito bons, foi possível obter uma melhora que quando se trata da área da saúde, pode ser considerável.

#### 4.4. Análise do Modelo Random Forest

Para a fase de prototipação do estudo foi feita uma execução do método RandomForest utilizando a biblioteca disponível pelo SkLearn. Inicialmente o método foi executado em sua versão default. Nessa versão os resultados são listados na Tabela 7:

**Table 7. Resultados Padrões para RandomForest**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada(Validação)	0.9991	0.9605	0.9666	0.9688	0.9635
Ponderada(Teste)	0.9628	0.9603	0.9652	0.9610	0.9616

**Fonte:** Autoria Própria (2024)

Iniciando a experimentação foi realizada uma execução do GridSearch testando os hiperparâmetros da listados na Tabela 8.

**Table 8. Parâmetros utilizados no GridSearch**

Parâmetro	Valores
n_estimators	range 50-90
max_depth	None, 10, 20, 30
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4
bootstrap	True, False

**Fonte:** Autoria Própria (2024)

Dentro da execução já é realizada a comparação dos resultados, com isso foi estipulado que o melhor resultado veio da execução com os parâmetros: 'bootstrap': False,

'max\_depth': 30, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 84. Utilizando a matriz com a execução dos melhores parâmetros foi construída a Tabela 9:

**Table 9. Resultados Otimizados para RandomForest**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada(Validação)	0.9709	0.9705	0.9724	0.9661	0.9693
Ponderada(Teste)	0.9697	0.9702	0.9714	0.9651	0.9686

Fonte: Autoria Própria (2024)

#### 4.5. Análise do Modelo MLPClassifier

Mais um modelo de aprendizagem de máquina aplicado foi o MLPClassifier utilizando a biblioteca disponível pelo SkLearn. Primeiro, o método foi executado com seus atributos com valores padrão. Após a execução os resultados obtidos são listados na Tabela 10.

**Table 10. Resultados Padrões para MLPClassifier**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada(Validação)	0.9692	0.9674	0.9710	0.9648	0.9779
Ponderada(Teste)	0.9671	0.9668	0.9700	0.9636	0.9767

Fonte: Autoria Própria (2024)

Utilizando método do GridSearch, foi realizada uma experimentação dos hiperparâmetros, criando a grade com os valores abaixo:

**Table 11. Parâmetros utilizados no GridSearch**

Parâmetro	Valores
hidden_layer_sizes	50, 100, 150,
activation	tanh, relu
solver	sgd, adam
alpha	0.0001, 0.001. 0.01
learning_rate	constant, adaptative

Fonte: Autoria Própria (2024)

A execução determinou, pela comparação feita dentro do GridSearch, que os melhores valores nos parâmetros do modelo testado era: 'activation': 'tanh', 'alpha': 0.0001, 'hidden\_layer\_sizes': (150,), 'learning\_rate': 'constant' e 'solver': 'adam' . Utilizando os parâmetros encontrados como melhores foi construída a Tabela 12.

**Table 12. Resultados Otimizados para MLPClassifier**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada(Validação)	0.9719	0.9701	0.9711	0.9745	0.9783
Ponderada(Teste)	0.9711	0.9702	0.9721	0.9661	0.9785

Fonte: Autoria Própria (2024)

#### 4.6. Análise do Modelo AdaBoost

Para a fase de prototipação do estudo também foi feita uma execução do método AdaBoost utilizando a biblioteca disponível pelo SkLearn. Inicialmente o método foi executado em sua versão padrão, incluindo seus parâmetros. Nessa versão os resultados são listados na Tabela 13.

**Table 13. Resultados Padrões para AdaBoost**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.86	0.82	0.86	0.83	0.88

**Fonte:** Autoria Própria (2024)

Iniciando a experimentação, dentro dos próprios atributos foi realizada uma validação cruzada da acurácia da execução do método alterando o atributo “*n\_estimators*”, que é relativo ao número de árvores, de 50, originalmente, para os valores (50; 100; 150). Também foram alterados os parâmetros ”algorithm” de ”SAMME” para (’SAMME’; ’SAMME.R’), e o ”learning\_rate” de ”1.0” para (0.01; 0.1; 1.0).

Com o GridSearch foi possível extrair os melhores resultados, e foi estipulado que o melhor resultado veio da execução com 150 árvores, ’SAMME’ como método de combinação de classificadores e 1.0 como taxa de aprendizado.

Apesar dos resultados possuírem valores aproximados, notou-se que o valor das árvores mudou e teve uma acurácia maior em relação ao teste primário. Utilizando o valor das árvores encontrado foi construída a seguinte tabela com os novos resultados otimizados:

**Table 14. Resultados Otimizados para AdaBoost**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.89	0.85	0.89	0.87	0.8883

**Fonte:** Autoria Própria (2024)

#### 4.7. Análise do Modelo GradientBoost

Para a fase de prototipação do estudo também foi feita uma execução do método GradientBoost utilizando a biblioteca disponível pelo SkLearn. Inicialmente o método foi executado em sua versão default, incluindo seus parâmetros. Nessa versão os resultados são listados na Tabela 15.

**Table 15. Resultados Padrões para GradientBoost**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.96	0.96	0.96	0.96	0.96

**Fonte:** Autoria Própria (2024)

Iniciando a experimentação dentro dos próprios atributos foi realizada uma validação cruzada da acurácia da execução do método alterando alguns dos atributos, originalmente de: ’*n\_estimators*’: 100; ’*learning\_rate*’: 0.1; ’*max\_depth*’: 3;

`'min_samples_split': 2; 'min_samples_leaf': 1.` Foram alterados em teste para:  
`'n_estimators': 50, 100, 150; 'learning_rate': 0.01; 0.1; 0.2; 'max_depth': 3, 5, 7;`  
`'min_samples_split': 2, 5, 10; 'min_samples_leaf': 1, 2, 4.`

Ao comparar os resultados, foi estipulado que o melhor resultado veio da execução com os seguintes melhores parâmetros: `'learning_rate': 0.1; 'max_depth': 7;`  
`'min_samples_leaf': 4; 'min_samples_split': 10; 'n_estimators': 150.`

Notou-se então, que foi alterando em 0.1 a maioria das métricas em suas médias ponderadas. Utilizando os novos valores de parâmetros encontrados, foi construída a seguinte tabela com os novos resultados otimizados:

**Table 16. Resultados Otimizados para GradientBoost**

Média	Acurácia	Precisão	Recall	F1-Score	Especificidade
Ponderada	0.97	0.97	0.97	0.97	0.9683

**Fonte:** Autoria Própria (2024)

#### 4.8. Resultados do Stacking

Depois dos treinos dos modelos, parte da experimentação foi com o uso do método de ensemble stacking, onde foram treinados quatro diferentes experimentos:

- O primeiro experimento foi realizado utilizando os 2 melhores resultados dos modelos treinados como modelos base e o pior modelo como meta-modelo com os seguintes resultados:

**Table 17. Resultados Primeiro Stacking com Média Ponderada**

Conjunto	Acurácia	Precisão	Recall	F1-Score	Especificidade
Validação	0.8737	0.87	0.91	0.89	0.9646
Teste	0.8679	0.87	0.91	0.88	0.9623

**Fonte:** Autoria Própria (2024)

- O segundo experimento foi realizado utilizando os 2 piores resultados dos modelos treinados como modelos base e o melhor modelo como meta-modelo com os seguintes resultados:

**Table 18. Resultados Segundo Stacking**

Conjunto	Acurácia	Precisão	Recall	F1-Score	Especificidade
Validação	0.9683	0.97	0.97	0.97	0.9675
Teste	0.9659	0.97	0.97	0.96	0.9646

**Fonte:** Autoria Própria (2024)

- O terceiro experimento foi realizado utilizando todos os modelos treinados com exceção do pior deles, que foi utilizado como meta-modelo, com os seguintes resultados:

**Table 19. Resultados Terceiro Stacking**

Conjunto	Acurácia	Precisão	Recall	F1-Score	Especificidade
Validação	0.9450	0.94	0.91	0.89	0.9673
Teste	0.9438	0.94	0.91	0.89	0.9657

**Fonte:** Autoria Própria (2024)

4. O quarto e último experimento foi realizado de forma semelhante ao terceiro experimento com stacking, porém ao invés de utilizar o pior dos modelos como meta-modelo, foi usado o melhor dos modelos como meta-modelo e o resto dos modelos treinados como base, abaixo estão seus resultados:

**Table 20. Resultados Quarto Stacking**

Conjunto	Acurácia	Precisão	Recall	F1-Score	Especificidade
Validação	0.9786	0.98	0.98	0.98	0.9791
Teste	0.9785	0.98	0.98	0.98	0.9790

**Fonte:** Autoria Própria (2024)

A Tabela 20 evidencia que com os experimentos do stacking, é possível notar que os testes em que o meta-modelo foi o melhor entre os demais, foi notado melhores números em comparação com os experimentos em que foi usado o pior modelo como meta-modelo, ou seja, o experimento 2 e 4 com stacking, foram superiores aos experimentos 1 e 3, por conta do uso do melhor modelo treinado como meta-modelo. Vale articular também que uma maior quantidade de modelos-base trouxe melhores resultados em relação ao experimento em que foi utilizado apenas os dois piores modelos como base, concluí-se que o melhor experimento entre os utilizados com stacking foi o quarto. Tendo em vista, que todos os experimentos com stacking foram feitos utilizando tanto os modelos-base quanto os meta-modelos com seus parâmetros padrões, ainda há espaço para melhora.

#### 4.9. Resultados Gerais dos Modelos

Os modelos SVM, KNN, XGBoost, Random Forest, MLP, AdaBoost e GradientBoost foram analisados e submetidos ao método GridSearch para buscar os melhores valores de parâmetros e poder otimiza-los individualmente, promovendo uma melhorar seu desempenho. Em propósito de comparação foram construídas as tabelas com os valores de Acurácia, Precisão, Recall, F1-Score e Especificidade das execuções feitas. Foi levado em consideração a média ponderada nos resultados, como consequência do desbalanceamento existente na base de dados. As métricas em foco foram a Acurácia e a Especificidade, uma vez que se baseiam muito nos resultados verdadeiros positivos e verdadeiros negativos da matriz de confusão. Os resultados mostraram que a otimização dos hiperparâmetros teve um impacto significativo no desempenho dos modelos. Apesar de que nas comparações a diferença foi pequena, existe um aumento no valor dos resultados na grande maioria. O método que apresentou a melhor acurácia e melhor especificidade após as otimizações foi o XGBoost. E o método com maior diferença na especificidade e na acurácia entre o padrão e o otimizado foi o SVC, com diferença de cerca de 0.01 em ambos resultados.

**Table 21. Resultados Padrões com Média Ponderada**

<b>Modelo</b>	<b>Acurácia</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Especificidade</b>
KNN	0.974	0.972	0.973	0.972	0.983
SVC	0.969	0.97	0.97	0.97	0.9679
Random Forest	0.980	0.960	0.965	0.964	0.962
AdaBoost	0.86	0.82	0.86	0.83	0.88
GradientBoost	0.96	0.96	0.96	0.96	0.96
MLPClassifier	0.968	0.967	0.970	0.964	0.977
XGBoost	0.98	0.98	0.98	0.98	0.9812

**Fonte:** Autoria Própria (2024)

**Table 22. Resultados Otimizados com Média Ponderada**

<b>Modelo</b>	<b>Acurácia</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Especificidade</b>
KNN	0.977	0.976	0.977	0.976	0.976
SVC	0.98	0.98	0.98	0.98	0.98
Random Forest	0.970	0.970	0.971	0.965	0.968
AdaBoost	0.89	0.85	0.89	0.87	0.8883
GradientBoost	0.97	0.97	0.97	0.97	0.9683
MLPClassifier	0.971	0.970	0.971	0.970	0.978
XGBoost	0.98	0.98	0.98	0.98	0.9844

**Fonte:** Autoria Própria (2024)

## 5. Considerações Finais

Sendo a principal hipótese do estudo, a possibilidade de melhoria dos resultados com métodos de classificação, o demonstrado na trilha está se concretizando, e o objetivo vem sendo alcançado nos métodos, sendo principalmente com o XGBoost, RandomForest, KNN, etc. Nos resultados se tornou possível destacar os métodos com maior resultado, sendo o XGBoost, melhor deles, gerando uma possibilidade de recomendação quanto a um método que em geral já apresentou bons resultados em sua avaliação dos dados, mas após melhorias seguiu se destacando entre os demais, em comparação com os outros, e si mesmo em seu modelo padrão.

Em conclusão, com as otimizações atuais os métodos apresentaram, em sua maioria, melhorias. Mesmo sendo uma diferença pequena nos valores, essas melhorias feitas e comprovadas nos resultados abrem espaço para uso das técnicas e modelos apresentados, como o GridSearch, para melhorar a avaliação de bases de dados em diferentes modelos. Esse artigo traz a possibilidade para que outros trabalhos, que se fundamentem nas mesmas bases de dados ou nos mesmos princípios de experimentação e desenvolvimento, possam utilizá-lo como base e referência. E também, indicar possibilidades na construção de outros métodos de avaliação, como se constitui o ensemble, permitindo o aproveitamento de diferentes métodos para realizar a avaliação dos dados, é um caminho que proporciona uma liberdade de experimentação entre métodos e que pode ser demonstrada mesmo que brevemente neste projeto.

## 6. References

### References

- Abdullah, D. M. and Abdulazeez, A. M. (2021). Machine learning applications based on svm classification a review. *Qubahan Academic Journal*, 1(2):81–90.
- Ahsanuzzaman, S., Ahmed, T., and Rahman, M. A. (2020). Low cost, portable ecg monitoring and alarming system based on deep learning. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 316–319. IEEE.
- Ali, Z. A., Abduljabbar, Z. H., Taher, H. A., Sallow, A. B., and Almufti, S. M. (2023). Exploring the power of extreme gradient boosting algorithm in machine learning: A review. *Academic Journal of Nawroz University*, 12(2):320–334.
- Apandi, Z. F. M., Ikeura, R., and Hayakawa, S. (2018). Arrhythmia detection using mitbih dataset: A review. In *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pages 1–5. IEEE.
- Aziz, S., Ahmed, S., and Alouini, M.-S. (2021). Ecg-based machine-learning algorithms for heartbeat classification. *Scientific reports*, 11(1):18738.
- Bonacorso, G. (2018). *Machine Learning Algorithms: Popular algorithms for data science and machine learning*. Packt Publishing Ltd.
- Breen, C., Kelly, G., and Kernohan, W. (2022). Ecg interpretation skill acquisition: A review of learning, teaching and assessment. *Journal of electrocardiology*, 73:125–128.
- Divina, F., Gilson, A., Goméz-Vela, F., García Torres, M., and Torres, J. F. (2018). Stack-ing ensemble learning for short-term electricity consumption forecasting. *Energies*, 11(4).
- Faceli, K., Lorena, A. C., Gama, J., Almeida, T. A. d., and Carvalho, A. C. P. d. L. F. d. (2021). Inteligência artificial: uma abordagem de aprendizado de máquina.
- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220.
- Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Lee, M. J., and Asadi, H. (2018). ed octor: machine learning and the future of medicine. *Journal of internal medicine*, 284(6):603–619.
- Jambukia, S. H., Dabhi, V. K., and Prajapati, H. B. (2015). Classification of ecg signals using machine learning techniques: A survey. In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 714–721. IEEE.
- Kolluri, J., Kotte, V. K., Phridviraj, M., and Razia, S. (2020). Reducing overfitting problem in machine learning using novel 11/4 regularization method. In *2020 4th international conference on trends in electronics and informatics (ICOEI)(48184)*, pages 934–938. IEEE.
- Latif, G., Al Anezi, F. Y., Zikria, M., and Alghazo, J. (2020). Eeg-ecg signals classification for arrhythmia detection using decision trees. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pages 192–196. IEEE.

- Li, S., Yin, G., and Yang, T. (2020). Research on product iterative requirement analysis method based on internet review data and xgboost. In *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, volume 1, pages 179–184. IEEE.
- Mi, F., Li, B., Cheng, X., Zhao, Y., Li, M., and Jing, J. (2023). Classification and processing of mit-bih arrhythmia-based on bp algorithm. In *2023 International Conference on Intelligent Supercomputing and BioPharma (ISBP)*, pages 72–76. IEEE.
- Moreira, P. S. d. C., Salerno, B. N., Tsunoda, D. F., et al. (2020). Internet das coisas e aprendizado de máquina na área da saúde: uma análise bibliométrica da produção científica de 2009 a 2019.
- Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons*, b, 4(51-62):56.
- Netto, A. and Maciel, F. (2021). *Python Para Data Science: E Machine Learning Descomplicado*. Alta Books.
- Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., and Mastorakis, N. (2009). Multi-layer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588.
- Rigatti, S. J. (2017). Random forest. *Journal of Insurance Medicine*, 47(1):31–39.
- Schwarz, L. (2009). Artigo de revisão: eletrocardiograma. *Revista Ilha Digital*, 1:3–19.
- Shoughi, A. and Dowlatshahi, M. B. (2021). A practical system based on cnn-blstm network for accurate classification of ecg heartbeats of mit-bih imbalanced dataset. In *2021 26th international computer conference, Computer Society of Iran (CSICC)*, pages 1–6. IEEE.
- Tsai, I. H. and Morshed, B. I. (2022). Beat-by-beat classification of ecg signals with machine learning algorithm for cardiac episodes. In *2022 IEEE International Conference on Electro Information Technology (eIT)*, pages 311–314. IEEE.
- Wang, R. (2012). Adaboost for feature selection, classification and its relation with svm, a review. *Physics Procedia*, 25:800–807.
- Wasimuddin, M., Elleithy, K., Abuzneid, A.-S., Faezipour, M., and Abuzaghleh, O. (2020). Stages-based ecg signal analysis from traditional signal processing to machine learning approaches: A survey. *IEEE Access*, 8:177782–177803.
- Wu, S. and Nagahashi, H. (2014). Parameterized adaboost: introducing a parameter to speed up the training of real adaboost. *IEEE Signal Processing Letters*, 21(6):687–691.
- Yang, L. and Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316.