

João Victor Sousa

**TCC**

Ilhéus - Bahia  
2025

João Victor Sousa

# TCC

Trabalho apresentado ao Curso de Y da Universidade X como requisito parcial para obtenção do título de Bacharel.

Universidade Estadual de Santa Cruz  
Curso de Graduação em Ciênciā da Computação

Orientador: Otacílio José Pereira

Ilhéus - Bahia  
2025

## RESUMO

O presente trabalho teve como objetivo principal aplicar modelos de aprendizagem profunda para a classificação de arritmias cardíacas ventriculares em sinais de ECG. O foco não foi alcançar o estado da arte, mas sim compreender o processo metodológico envolvido , investigando as técnicas de pré-processamento , as arquiteturas de redes neurais e as métricas de avaliação adequadas ao problema.

Para isso, a metodologia utilizou o banco de dados MIT-BIH Arrhythmia Database , focando na classe ventricular (V). O pré-processamento incluiu a limpeza de ruídos do sinal , segmentação e reamostragem dos batimentos para 288 amostras , além da inclusão dos intervalos RR como features. Foram comparadas duas arquiteturas: um modelo GRU puro e um modelo híbrido (CNN + GRU). Ambos foram treinados com sequências de 16 batimentos e avaliados usando uma estratégia interpaciente com validação cruzada de 5 folds no conjunto DS1.

Os resultados indicaram que o modelo híbrido (CNN + GRU) teve um desempenho superior ao do GRU puro , atingindo um F1-Score médio de 0,8593 e recall de 0,8726, em comparação com 0,8060 de F1-Score do modelo GRU. Embora ambos os modelos tenham demonstrado sobreajuste , o fenômeno foi menos intenso no modelo híbrido. Uma análise de erros do pior cenário revelou que 98,46% das falhas se concentraram em um único paciente (203), notório por sua complexidade. Além disso, identificou-se um problema de superconfiança nas previsões do modelo.

Em conclusão, o trabalho atingiu seu objetivo de explorar e compreender o processo de aplicação de deep learning ao ECG. A superioridade do modelo híbrido sugere que a estratégia de usar CNNs para extração de características morfológicas combinada com GRUs para análise temporal é vantajosa. A análise dos erros, do overfitting e da superconfiança evidenciou os desafios práticos e as limitações atuais para a aplicação desses modelos em cenários clínicos reais, principalmente devido à alta variabilidade dos sinais entre pacientes.

**Palavras-chave:** aprendizado de máquina. eletrocardiograma. redes neurais

# Abstract

**Keywords:**

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	Contextualização	10
1.2	Problema de pesquisa	11
1.3	Objetivos	11
1.3.1	Objetivo geral	11
1.3.2	Objetivos específicos	11
1.4	Justificativa	12
1.5	Organização	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
2.1	Funcionamento do coração	13
2.2	O eletrocardiograma	14
2.3	Arritmias	14
2.3.1	Arritmias clínicas	15
2.3.2	Padrões de classificação para algoritmos (AAMI)	15
2.3.2.1	Arritmias Ventriculares	15
2.3.2.2	Arritmias Supraventriculares	17
2.3.2.3	Batimentos normais	17
2.3.2.4	Conclusão da Fundamentação sobre arritmia e definição do objetivo	17
2.4	Redes Neurais Artificiais	18
2.4.1	Treinamento de redes neurais	21
2.5	Redes Neurais Recorrentes e suas variações	23
2.5.1	Long Short-Term Memory (LSTM)	25
2.5.2	GRU	26
2.6	Redes Neurais Convolucionais	27
2.7	Trabalhos correlatos	29
<b>3</b>	<b>METODOLOGIA</b>	<b>35</b>
3.1	O banco de dados	36
3.2	Pré-processamento	38
3.2.1	Features	39
3.3	Arquiteturas	40
3.3.1	Tamanho da sequência	42
3.4	Estratégia de avaliação	43
3.4.1	Métricas	44

<b>4</b>	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>49</b>
<b>4.1</b>	<b>Resultados do modelo GRU . . . . .</b>	<b>49</b>
4.1.0.1	Resultados no pior e melhor caso (Modelo GRU) . . . . .	50
<b>4.2</b>	<b>Resultados do modelo híbrido GRU e CNN . . . . .</b>	<b>55</b>
4.2.0.1	Resultados no pior e melhor caso (Modelo Híbrido) . . . . .	57
4.2.1	Conclusão dos resultados . . . . .	60
<b>5</b>	<b>ANÁLISE DE ERROS NO PIOR <i>FOLD</i> . . . . .</b>	<b>62</b>
<b>5.1</b>	<b>Análise de erros do modelo híbrido CNN com GRU . . . . .</b>	<b>62</b>
5.1.0.1	Comparação morfológica entre o paciente 203 e 119 . . . . .	63
5.1.0.2	Comparação temporal entre o paciente 203 e 119 . . . . .	66
5.1.0.3	Conclusão da análise de erros . . . . .	69
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>71</b>
<b>6.1</b>	<b>Trabalhos futuros . . . . .</b>	<b>71</b>
<b>REFERÊNCIAS . . . . .</b>		<b>72</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>73</b>

# Listas de tabelas

Tabela 1 – Metodologias de Classificação de Batimentos Cardíacos: Modelos, Partitionamento, Features, Segmentação e Limpeza . . . . .	33
Tabela 1 – Metodologias de Classificação de Batimentos Cardíacos: Modelos, Partitionamento, Features, Segmentação e Limpeza . . . . .	34
Tabela 2 – Mapeamento das anotações originais do MIT-BIH para as classes AAMI.	36
Tabela 3 – Distribuição das classes nos conjuntos Ds1 e Ds2 . . . . .	43
Tabela 4 – Distribuição dos IDs dos Pacientes por Fold para Conjuntos de Treinamento e Validação . . . . .	44
Tabela 5 – Exemplo de matriz de confusão binária . . . . .	46
Tabela 6 – Média das métricas do GRU para a classificação normal vs. ventricular na validação . . . . .	49
Tabela 7 – Média das métricas do GRU para a classificação normal vs. ventricular no treino . . . . .	50
Tabela 8 – Média das métricas do modelo híbrido CNN e GRU para a classificação normal vs. ventricular na validação . . . . .	55
Tabela 9 – Média das métricas do modelo híbrido CNN e GRU para a classificação normal vs. ventricular no treino . . . . .	56
Tabela 10 – Resumo dos desempenho dos modelos para a classificação de arritmia (ventricular) . . . . .	61
Tabela 11 – Total dos erros e acertos por paciente no <i>fold</i> de validação . . . . .	62

# Listas de ilustrações

Figura 1 – Sistema de condução do coração . . . . .	13
Figura 2 – Exemplo de ECG com sua morfologia destacada . . . . .	14
Figura 3 – Exemplo de uma ANN <i>feed-forward</i> . . . . .	19
Figura 4 – Funções de ativação Sigmoid, Tanh e ReLU . . . . .	21
Figura 5 – Exemplo de uma RNN . . . . .	24
Figura 6 – Esquema de uma célula LSTM . . . . .	25
Figura 7 – Esquema de uma célula GRU . . . . .	26
Figura 8 – Esquema de como uma CNN identificaria um tigre . . . . .	27
Figura 9 – Arquitetura típica de uma CNN . . . . .	29
Figura 10 – Esquema a metodologia adotada. . . . .	35
Figura 11 – Trecho ECG com ritmo normal do paciente 100 com arritmia classe A	37
Figura 12 – Trecho ECG com ritmo normal do paciente 100 com com arritmia classe V	37
Figura 13 – Trecho ECG com ritmo normal do paciente 106 com com arritmia classe V	38
Figura 14 – Comparação entre o ECG original, segmentado e reamostrado . . . . .	39
Figura 15 – Arquitetura GRU pura. . . . .	41
Figura 16 – Arquitetura híbrida CNN e GRU. . . . .	42
Figura 17 – Curva ROC de um Classificador Perfeito: Comparação com Modelo Aleatório. . . . .	45
Figura 18 – Curva Precisão–Recall de um Classificador Perfeito: Comparação com Baseline. . . . .	46
Figura 19 – Diagrama de calibração mostrando curvas de modelos perfeitamente calibrado, superconfiante e subconfiante. . . . .	47
Figura 20 – Métricas do modelo GRU por <i>fold</i> . . . . .	51
Figura 21 – Matriz de confusão do modelo GRU em seu melhor <i>fold</i> . . . . .	52
Figura 22 – Matriz de confusão do modelo GRU em seu pior <i>fold</i> . . . . .	52
Figura 23 – Curva <i>ROC</i> modelo GRU em seu melhor <i>fold</i> . . . . .	53
Figura 24 – Curva <i>ROC</i> do modelo GRU em seu pior <i>fold</i> . . . . .	53
Figura 25 – Curva precisão vs <i>recall</i> do modelo GRU em seu pior <i>fold</i> . . . . .	54
Figura 26 – Curva precisão vs <i>recall</i> do modelo GRU em seu melhor <i>fold</i> . . . . .	55
Figura 27 – Métricas do modelo híbrido CNN e GRU por <i>fold</i> . . . . .	56
Figura 28 – Matriz de confusão modelo híbrido CNN e GRU em seu melhor <i>fold</i> .	57
Figura 29 – Matriz de confusão modelo híbrido CNN e GRU em seu pior <i>fold</i> . .	58
Figura 30 – Curva <i>ROC</i> modelo híbrido CNN e GRU em seu melhor <i>fold</i> . . . . .	58
Figura 31 – Curva precisão vs <i>recall</i> do modelo híbrido CNN e GRU em seu melhor <i>fold</i> . . . . .	59

Figura 32 – Curva <i>ROC</i> modelo híbrido CNN e GRU em seu pior <i>fold</i> . . . . .	59
Figura 33 – Curva precisão vs <i>recall</i> do modelo híbrido CNN e GRU em seu melhor <i>fold</i> . . . . .	60
Figura 34 – Matriz de confusão do paciente 203 . . . . .	63
Figura 35 – ECG normal do paciente 203 . . . . .	64
Figura 36 – ECG normal do paciente 119. . . . .	64
Figura 37 – ECG arrítmico do paciente 203: acerto e erro . . . . .	65
Figura 38 – ECG arrítmico do paciente 119. . . . .	66
Figura 39 – Scatter plot do paciente 203 de um falso positivo e um verdadeiro positivo.	67
Figura 40 – Scatter plot da sequência normal do paciente 203 e 119 . . . . .	68
Figura 41 – Curva de calibração para o paciente 203 e 119. . . . .	69

# **Lista de Siglas**

AUC	Area Under Curve
AP	Average Precision
RNN	Redes Neurais Recorrentes
CNN	Redes Neurais Convolucionais
LSTM	Long Short-Term Memory
AAMI	Association for the Advancement of Medical Instrumentation
GRU	Gated Recurrent Unit
ECG	Eletrocardiograma
TP	Verdadeiro Positivo
FP	Falso Positivo
TN	Verdadeiro Negativo
FN	Falso Negativo

# 1 Introdução

## 1.1 Contextualização

Segundo a Organização Mundial da Saúde (World Health Organization (2025)), as doenças cardiovasculares estão entre as principais causas de morte no mundo. Em 2022, foram responsáveis por aproximadamente 32% de todos os óbitos globais. Ainda segundo Agência Brasil (2024), as arritmias cardíacas causam cerca de 320 mil mortes súbitas anualmente.

Um dos principais exames utilizados no diagnóstico de doenças cardiovasculares é o eletrocardiograma (ECG), obtido por meio de eletrodos posicionados na superfície da pele para medir as correntes elétricas que controlam as contrações do músculo cardíaco. A partir desse exame, é possível identificar diversas alterações, incluindo arritmias, caracterizadas por perturbações na condução ou na origem dos impulsos elétricos cardíacos.

Nos últimos anos, métodos de aprendizagem de máquina vêm sendo empregados para auxiliar no diagnóstico de arritmias com o objetivo de otimizar o processo e apoiar o trabalho dos especialistas. Entretanto, a aplicação dessas técnicas não é trivial. O ritmo cardíaco apresenta grande variabilidade entre indivíduos — e até dentro de um mesmo indivíduo — tanto em condições normais quanto patológicas. Além disso, o sinal de ECG é naturalmente sujeito a ruídos e distorções.

Essa variabilidade torna a detecção automática de arritmias um desafio pois o modelo precisa aprender características que sejam generalizáveis a diferentes pacientes e condições. Soma-se a isso o desafio de projetar o próprio sistema de detecção: é necessário avaliar cuidadosamente decisões relacionadas a pré-processamento, seleção de *features*, arquitetura do modelo e escolha de métricas, garantindo que estas sejam adequadas ao contexto clínico do problema.

Dante desse cenário, o presente trabalho tem como objetivo aplicar um modelo de aprendizado profundo para a detecção de arritmias em sinais de ECG, investigando as decisões metodológicas envolvidas e analisando como interpretar adequadamente os resultados obtidos.

Vale destacar que o foco deste trabalho não é propor um novo método ou alcançar desempenho de estado da arte, mas compreender o processo de modelagem, as decisões envolvidas e como interpretar os resultados obtidos.

## 1.2 Problema de pesquisa

O desafio da aplicação de modelos de aprendizagem de máquina em sinais de ECG envolve tanto a busca por um bom desempenho quanto a definição das decisões metodológicas adequadas. Para isso, é fundamental compreender o domínio do problema, identificar quais *features* são relevantes, selecionar as métricas apropriadas e avaliar como essas escolhas influenciam o processo de modelagem.

Grande parte da complexidade surge da necessidade de conciliar dois domínios distintos: a cardiologia, que fornece o conhecimento clínico necessário para interpretar o sinal, e a aprendizagem de máquina, que disponibiliza as ferramentas computacionais para modelá-lo. Essa interseção exige entendimento suficiente de ambos os campos para formular o problema de modo coerente e avaliar corretamente os resultados.

## 1.3 Objetivos

Dante desses desafios, estabeleceram-se os seguintes objetivos.

### 1.3.1 Objetivo geral

Aplicar um modelo de aprendizagem profunda para a classificação de arritmias cardíacas em sinais de ECG e compreender os aspectos metodológicos envolvidos nesse processo.

### 1.3.2 Objetivos específicos

- **Compreender o eletrocardiograma e as arritmias cardíacas:** estudar a morfologia do ECG e sua relação com o funcionamento cardíaco, identificando padrões associados a diferentes arritmias.
- **Analizar uma base de dados de ECG:** identificar uma base de dados e entender a sua estrutura, incluindo suas anotações e a relação destas com as patologias estudadas.
- **Investigar técnicas de pré-processamento:** avaliar quais procedimentos são necessários para preparar o sinal antes de sua utilização em modelos baseados em redes neurais.
- **Estudar arquiteturas de redes neurais aplicadas ao problema:** analisar como diferentes tipos de redes podem explorar características do ECG.

- **Avaliar métricas adequadas ao desempenho do modelo:** compreender como cada métrica contribui para a interpretação do modelo no contexto da classificação de arritmias.
- **Realizar experimentos de classificação de arritmias em ventricular e normal:** aplicar uma arquitetura de rede neural ao problema e avaliar seu desempenho por meio de experimentação.

## 1.4 Justificativa

O uso de modelos de aprendizagem de máquina para auxiliar no diagnóstico de arritmias cardíacas apresenta grande potencial, tanto pela relevância clínica dessas condições quanto pela capacidade desses modelos de lidar com grandes volumes de dados e com a variabilidade inerente ao sinal de ECG. No entanto, aplicar tais métodos de forma eficaz exige uma compreensão adequada do domínio clínico e de como ele se relaciona com as técnicas de inteligência artificial.

No caso do ECG, a diversidade de arritmias, a variabilidade fisiológica entre pacientes e a presença de ruídos tornam o problema especialmente desafiador. Esses fatores reforçam a necessidade de compreender não apenas o desempenho de um modelo, mas também o processo de construção, preparação e avaliação envolvidos.

Dessa forma, a justificativa deste trabalho reside na análise desse processo: entender quais decisões metodológicas influenciam o comportamento de modelos de aprendizagem profunda aplicados à classificação de arritmias. Vale destacar que o foco não está em propor um novo método ou atingir resultados de estado da arte, mas em discutir os desafios envolvidos e as possíveis soluções adotadas ao longo da modelagem.

Espera-se que as reflexões apresentadas contribuam para uma compreensão mais clara das dificuldades práticas do problema e auxiliem futuros estudos ou aplicações na área.

## 1.5 Organização

O presente trabalho está organizado da seguinte forma. No capítulo 2 é explicado sobre arritmias cardíacas, começando sobre o funcionamento do coração e a caracterização das arritmias e, em seguida, sobre os modelos de redes neurais usadas. Além de uma revisão de outros trabalhos sobre o tema. No capítulo 3 é explicado sobre a metodologia usada e o banco, o processo de limpeza e arquiteturas das redes. No capítulo 4, é discutido os resultados alcançados pelos modelos através das métricas empregadas e quais problemas foram identificados. Por fim, no capítulo 5 é feita uma breve investigação do pior caso do melhor modelo.

## 2 Fundamentação Teórica

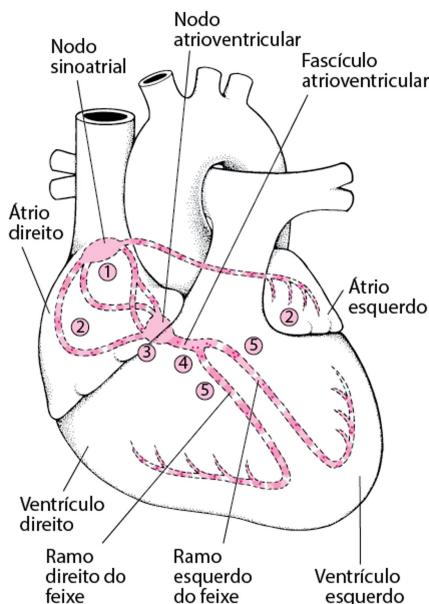
Como o trabalho envolve o reconhecimento de arritmias cardíacas, o primeiro ponto é compreender como o coração funciona, mais especialmente, como o funciona o mecanismo do impulso elétrico responsáveis pela contração do mesmo. Em seguida, entender o princípio de funcionamento das redes neurais, que integra a solução para o problema. Por fim, será apresentado alguns trabalhos que correlacionaram as duas áreas.

### 2.1 Funcionamento do coração

O coração é um órgão muscular composto por quatro câmaras — átrio direito e esquerdo, e ventrículo direito e esquerdo — que se contraem de forma rítmica, bombeando sangue para o corpo. Essas contrações são controladas por correntes elétricas que percorrem o coração de maneira precisa e em velocidade controlada.

Na Figura 1, o sistema de condução elétrica do coração é ilustrado.

Figura 1 – Sistema de condução do coração



Fonte: Adaptado de Mitchell (2025)

Segundo Mitchell (2025), o batimento cardíaco normal se inicia no nódulo sinusal (1), localizado no átrio direito, que atua como o marcapasso natural do coração. A corrente elétrica propaga-se do átrio direito para o esquerdo (2), promovendo sua contração e o bombeamento do sangue para os ventrículos. Em seguida, o impulso atinge o nódulo atrioventricular (3) — conexão entre os átrios e ventrículos — onde é temporariamente

retardado, permitindo que os átrios se contraiam completamente e enchem as câmaras inferiores.

Posteriormente, a corrente percorre o feixe de His (4), que se divide e conduz o impulso para ambos os ventrículos (5), promovendo sua contração e o bombeamento do sangue para o restante do corpo.

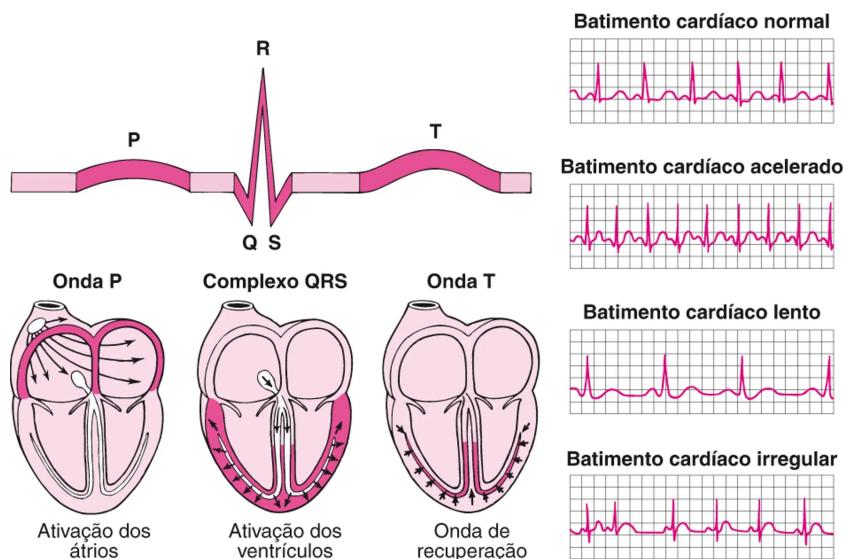
## 2.2 O eletrocardiograma

O eletrocardiograma, ECG, é um exame não invasivo usado para medir a atividade elétrica do coração. Ele é feito a partir do contato de eletrodos, chamados de derivações ou *leads*, sobre a pele. A quantidade de eletrodos varia, mas geralmente são 12 (CASCINO; SHEA, 2025).

Ao registrar a magnitude e direção da corrente, as derivações geram uma onda que representa a atividade elétrica do coração. O passo a passo descrito em 2.1 é refletido em sua morfologia.

Na Figura 2, é ilustrado um ECG de um batimento, observe que ele é subdividido em: onda P, complexo QRS e onda T (CASCINO; SHEA, 2025). Note que cada uma dessas partes se refere a um estágio do batimento.

Figura 2 – Exemplo de ECG com sua morfologia destacada



Fonte: Adaptado de Cascino e Shea (2025)

## 2.3 Arritmias

As doenças cardíacas podem ser diversas. Dentre elas, as arritmias são um grupo que podem ser diagnósticas via ECG

### 2.3.1 Arritmias clínicas

As arritmias são alterações no ritmo cardíaco que podem ter diversas causas, incluindo alterações hormonais, uso de medicamentos, toxinas (como álcool ou cafeína), anomalias eletrolíticas ou doenças cardíacas. Segundo Mitchell (2025), em adultos em repouso, a frequência cardíaca normal varia entre 60 e 100 batimentos por minuto (bpm). Frequências mais baixas, conhecidas como bradicardia sinusal, são comuns em atletas, crianças pequenas, adolescentes, jovens adultos e durante o sono. Por outro lado, a taquicardia sinusal ocorre quando a frequência se eleva, podendo ser observada durante o esforço físico, doenças, estimulação neural simpática ou emoção intensa.

Variações no ritmo cardíaco são fenômenos fisiológicos normais. Durante a respiração, por exemplo, é comum que a frequência aumente e diminua levemente, comportamento conhecido como arritmia sinusal respiratória. O autor ainda observa que um ritmo cardíaco perfeitamente regular pode indicar patologias no sistema nervoso autônomo, como ocorre em casos de diabetes avançado. Dessa forma, ainda não existe um indicador global e definitivo do que seria um ritmo sinusal considerado saudável.

As arritmias podem ser classificadas de forma simplificada em três tipos principais:

1. **Taquicardia** — frequência excessivamente rápida;
2. **Bradicardia** — frequência excessivamente lenta;
3. **Irregular** — quando os impulsos percorrem o coração por vias irregulares.

Observe na Figura 2, exemplos desses três tipos arrítmicos ilustrados em um ECG.

### 2.3.2 Padrões de classificação para algoritmos (AAMI)

A *Association for the Advancement of Medical Instrumentation*, AAMI, define cinco classes de arritmia: normal (N), ventricular (V), supraventricular (S), fusão (F) e não classificado (Q) (SILVA et al., 2025; SAADATNEJAD; OVEISI; HASHEMI, 2020), que são normalmente usadas para classificação.

#### 2.3.2.1 Arritmias Ventriculares

As arritmias ventriculares são classificadas conforme sua origem e manifestação clínica, que variam de batimentos compensatórios a condições de alto risco. A seguir, são apresentados alguns tipos de arritmia ventricular

- **Batimento Ventricular de Escape:** Atua como um mecanismo compensatório, funcionando como um "backup" protetivo do coração quando o marcapasso natural falha temporariamente.

- **Contração Prematura Ventricular (PVC):** De acordo com Sattar e Hashmi (2025), as PVCs são batimentos originários dos ventrículos que podem ocorrer mesmo em indivíduos saudáveis. Sua morfologia é variável, dependendo da origem do impulso, de doenças estruturais ou ainda de uso de medicamentos. Quando frequentes, podem causar fadiga e palpitações, evoluindo para disfunções ventriculares e, em alguns casos, representando a primeira manifestação de cardiopatias estruturais. Mitchell (2024c) cita ainda outras causas potenciais de PVCs, como doenças da artéria coronária (especialmente durante ou após infarto), dilatação ventricular decorrente de insuficiência cardíaca e alterações nas válvulas cardíacas.
- **Taquicardia Ventricular (TV):** Em Mitchell (2024b), Mitchell (2024d), TV é descrito como uma arritmia originada nos ventrículos, produzindo uma frequência cardíaca de até 120 bpm. É formada por uma sequência de contrações ventriculares prematuras (PVCs). Quando persiste por mais de 30 segundos, recebe a denominação de taquicardia sustentada. Costuma ocorrer em indivíduos com alterações estruturais cardíacas, como infarto do miocárdio, falha cardíaca ou cardiomiopatia. Os sintomas incluem fraqueza, tontura e desconforto torácico. Caso persista por mais de 30 segundos, o tratamento é indicado mesmo na ausência de sintomas, uma vez que pode evoluir para fibrilação ventricular.
- **Flutter Ventricular:** Segundo o U.S. National Library of Medicine (2025), o *flutter* ventricular é caracterizado por uma taquicardia extremamente rápida e instável hemodinamicamente (150 a 300 bpm). É potencialmente fatal e, tipicamente, evolui para a fibrilação ventricular.
- **Fibrilação Ventricular (FV):** É caracterizada por batimentos rápidos e desordenados, resultantes de sinais elétricos caóticos nos ventrículos. Essa condição leva à perda de consciência em poucos segundos e à morte caso não haja intervenção imediata, configurando-se como um tipo de parada cardíaca. Entre suas causas estão afogamentos, choques elétricos e falha cardíaca Mitchell (2024b), Mitchell (2024d).

Apesar de ocorrerem também em indivíduos saudáveis, as PVCs possuem significância clínica, pois podem estar associadas a outros tipos de arritmias ventriculares graves. É importante considerar o contexto do batimento: no caso da taquicardia ventricular, por exemplo, o diagnóstico é estabelecido a partir de uma sequência de PVCs que produz uma frequência cardíaca elevada ( $> 20$  bpm). Assim, o diagnóstico depende de uma combinação de características temporais e morfológicas, observando-se o contexto dos batimentos e, naturalmente, os sintomas clínicos.

### 2.3.2.2 Arritmias Supraventriculares

As arritmias supraventriculares (S) se originam acima dos ventrículos, como nos átrios ou nos caminhos de condução (Texas Heart Institute, s.d.). Geralmente, são consideradas mais benignas que as arritmias ventriculares podendo ocorrer, assim como os PVCs, em resposta ao consumo de cafeína, tabaco, álcool, tosse ou remédio para resfriado. Outras causas incluem problemas na tireoide. Elas podem causar palpitações, falta de ar ou aperto no peito.

Dentre alguns exemplos de arritmia supraventricular, incluem:

- **Contrações Supraventriculares Prematuras (CSPs):** Ocorrem quando o átrio se contrai de forma antecipada.
- **Taquicardias Supraventriculares (TSV):** Segundo Patti, Horenstein e Ashurst (2025), TSVs são formadas por desordens rítmicas rápidas e caracterizadas por um complexo QRS mais estreito (menor que 120 ms) e uma frequência cardíaca alta. Em adultos, essa frequência é superior a 100 bpm, enquanto que em crianças, varia de 180 a 220 bpm. Os autores apontam como as TSV mais comuns a taquicardia atrioventricular nodal reentrante e a taquicardia atrioventricular recíproca.
- **Fibrilação Atrial (FA):** (Texas Heart Institute, s.d.) caracteriza FA como batimentos rápidos e irregulares, resultantes de contrações desordenadas das fibras musculares. É uma das principais causas de Acidente Vascular Cerebral (AVC) em idosos, pois o ritmo desordenado causa o acúmulo de sangue nos átrios, o que pode levar à formação de coágulos que viajam até o cérebro.

### 2.3.2.3 Batimentos normais

Dentro da classe dos batimentos normais, além do batimento típico descrito na 2.1, incluem-se também os batimentos atriais de escape e os bloqueios do ramo esquerdo e direito. Estes últimos, embora inofensivos por si só, podem indicar condições cardíacas subjacentes mais graves, como doença da artéria coronária ou infarto do miocárdio prévio (MITCHELL, 2024a).

### 2.3.2.4 Conclusão da Fundamentação sobre arritmia e definição do objetivo

As arritmias cardíacas são classificadas, primariamente, de acordo com sua origem: ventriculares (quando iniciadas nos ventrículos) ou supraventriculares (originadas acima deles, como nos átrios). Tais alterações se manifestam no eletrocardiograma (ECG) como um ritmo cardíaco acelerado, retardado ou com condução elétrica anormal, refletindo-se na morfologia do traçado.

Apesar dessa classificação, a detecção automática enfrenta desafios cruciais devido à alta variabilidade do ECG:

- Variações Fisiológicas: Não há padrões universais de atividade cardíaca normal, pois o ECG varia conforme a faixa etária, o nível de atividade física ou as condições clínicas individuais.
- Diversidade Intra-classe: A amplitude de manifestações é alta, como o contraste entre as contrações ventriculares prematuras (PVCs) e as taquicardias ventriculares (TVs).

Dadas a maior significância clínica e o potencial risco de morte súbita associado à progressão de arritmias ventriculares, o presente trabalho foca na classificação binária da arritmia ventricular (contração ventricular prematura e batimento ventricular de escape), em contraste com a classe de batimentos normais. Apesar de ocorrerem em indivíduos saudáveis, PVCs estão associados a tipos arrítmicos mais graves, como a taquicardia ventricular e fibrilação atrial, representando um risco maior a vida.

## 2.4 Redes Neurais Artificiais

Devido a complexidade dos dados — em partes devido a variabilidade observada tanto intra classe quanto entre pacientes — e também ao volume que costuma ser gerado, ECGs são um bom material para a aplicação de algoritmos de aprendizado de máquina, como redes neurais que serão discutidas na seguinte seção

Segundo Géron (2022), as redes neurais artificiais (ANN) foram introduzidas em 1943 pelo neurofisiologista Warren McCulloch e pelo matemático Walter Pitts como um modelo simplificado de como os neurônios biológicos podem realizar cálculos complexos por meio de lógica proposicional.

O sucesso inicial levou à crença de que tais modelos poderiam ser usados no desenvolvimento de sistemas verdadeiramente inteligentes. No entanto, em parte devido às limitações na capacidade de processamento e à escassez de dados disponíveis, as ANNs foram gradualmente abandonadas em favor de outras técnicas, como as máquinas de vetor de suporte (*support vector machines* — SVMs).

Nos últimos anos, entretanto, com a criação de grandes conjuntos de dados, o aumento do poder computacional proporcionado pelas GPUs e o desenvolvimento de novas arquiteturas, as ANNs passaram por um reavivamento, consolidando-se como uma das técnicas mais promissoras da inteligência artificial.

O autor descreve os neurônios biológicos como células formadas por um corpo celular e uma longa extensão chamada axônio. A partir do corpo celular, partem diversas ramificações denominadas dendritos. Na extremidade do axônio, ocorrem ramificações

menores chamadas telodendriais, cujas pontas possuem estruturas denominadas terminais sinápticos.

O axônio de um neurônio conecta-se ao dendrito de outro, formando uma sinapse — o ponto de comunicação entre as células.

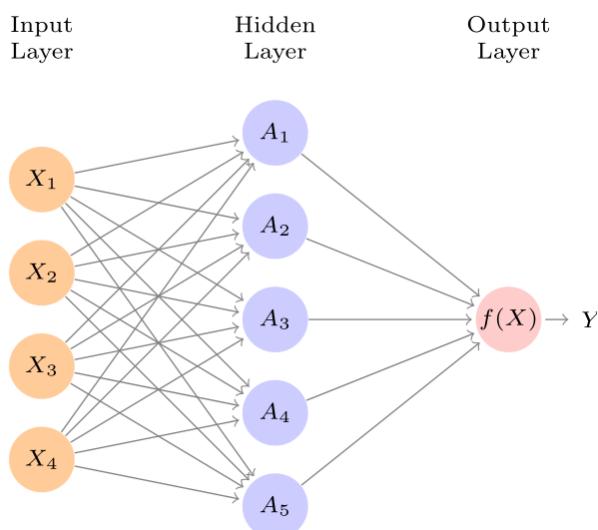
Esses neurônios transmitem sinais elétricos chamados potenciais de ação (*action potentials* — APs), que percorrem o axônio até os terminais sinápticos, onde estimulam a liberação de substâncias químicas conhecidas como neurotransmissores. Esses neurotransmissores permitem a comunicação entre os neurônios, podendo estimular ou inibir o disparo de novos potenciais de ação.

Bilhões de neurônios interligam-se por meio de sinapses, formando redes altamente complexas. A ação coordenada dessas redes possibilita a realização de cálculos e processos cognitivos sofisticados, que inspiraram o desenvolvimento das redes neurais artificiais.

Esse é o princípio em que se baseiam as redes neurais artificiais (ANNs): pequenas unidades computacionais simples que, ao se conectarem, são capazes de realizar funções complexas. É importante destacar que um neurônio artificial, no contexto das ANNs, não é uma réplica de um neurônio biológico, mas sim uma abstração matemática inspirada em seu funcionamento.

Segundo James et al. (2023), uma ANN pode ser vista como uma função que recebe um conjunto de p preditores e, por meio de transformações não lineares, busca prever uma variável resposta Y. Na Figura 3, é ilustrada uma arquitetura simples de rede neural *feedforward*.

Figura 3 – Exemplo de uma ANN *feed-forward*



Fonte: Adaptado de James et al. (2023)

Essa rede é composta por três camadas principais: a camada de entrada (*Input layer*), a camada oculta (*Hidden layer*) e a camada de saída (*Output layer*). Cada neurônio

na camada oculta é conectado a todos os neurônios da camada de entrada, assim como os neurônios da camada de saída são conectados aos da camada oculta.

A cada conexão é atribuído um peso, que representa a força da relação entre os neurônios — em analogia à intensidade das sinapses no cérebro biológico. Além disso, cada neurônio possui um viés (*bias*), que desloca o ponto de ativação.

A ativação de um neurônio é determinada por uma função de ativação não linear, essencial para permitir que a rede aprenda relações complexas entre as variáveis de entrada. Sem essa função, uma ANN de múltiplas camadas seria equivalente a uma simples regressão linear.

Formalmente, uma ANN pode ser descrita como uma função  $f(X)$  que recebe um vetor de  $p$  preditores  $X = (X_1, X_2, \dots, X_p)$  e produz uma previsão para a variável resposta  $Y$ .

No caso da rede ilustrada na Figura 3, essa função  $f$  pode ser expressa como:

$$f(X) = \beta_0 + \sum_{k=1}^K \beta_k h_k(X) \quad (2.1)$$

Onde  $K$  refere-se a quantidade de neurônios na camada oculta.  $h$  é expandido em:

$$h_k = g(w_{k0} + \sum_{j=1}^p w_{kj} X_j) \quad (2.2)$$

Assim,  $f$  é construída em dois passos; primeiro,  $K$  ativações da camada oculta são calculados:

$$A_k = h_k(X) = g(w_{k0} + \sum_{j=1}^p w_{kj} X_j) \quad (2.3)$$

Onde  $g(z)$  é a função de ativação não linear. Os parâmetros  $\beta_0, \dots, \beta_K$  e  $w_0, \dots, w_{Kp}$  precisam ser estimados — isto é, aprendidos a partir dos dados.

Existem diversas funções de ativação propostas ao longo do tempo. A sigmoide, por exemplo, foi amplamente utilizada nas primeiras redes neurais:

$$g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}} \quad (2.4)$$

Segundo Géron (2022), essa função é uma aproximação do comportamento de ativação observado em neurônios biológicos. Entretanto, atualmente ela é pouco utilizada nas camadas intermediárias, uma vez que é propensa ao problema do *vanishing gradient* e não é centrada em zero, o que prejudica a convergência do modelo. Apesar disso, a sigmoide ainda é frequentemente empregada na camada de saída em problemas de classificação binária, pois seu contradomínio é  $[0, 1]$ ; o que torna simples interpretar a saída dos neurônios como probabilidades.

Outra função comumente utilizada é a tangente hiperbólica:

$$g(z) = \tanh(z) = 2\sigma(2z) - 1 \quad (2.5)$$

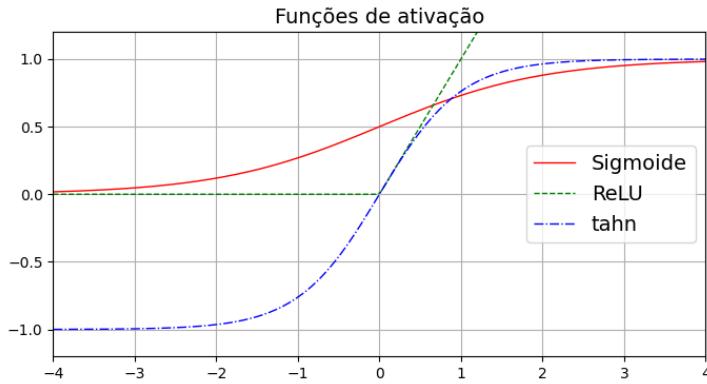
Essa função é centrada em zero e possui uma derivada mais acentuada, o que ajuda a mitigar o problema do *vanishing gradient*.

Atualmente, a função de ativação mais popular é a ReLU (*Rectified Linear Unit*), definida como:

$$g(z) = z_+ = \begin{cases} 0, & \text{se } z < 0 \\ z, & \text{caso contrário} \end{cases} \quad (2.6)$$

A ReLU apresenta um cálculo simples e eficiente, além de acelerar o treinamento de redes profundas. Embora não seja centrada em zero, o viés dos neurônios auxilia a compensar esse deslocamento. Na Figura 4, são ilustradas as três funções de ativação discutidas.

Figura 4 – Funções de ativação Sigmoid, Tanh e ReLU



Fonte: Elaborado pelo autor.

#### 2.4.1 Treinamento de redes neurais

ANNs aprendem através da minimização de uma função de custo. Uma função muito utilizada para problemas de classificação binária é a *binary cross-entropy*:

$$-\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (2.7)$$

Onde  $N$  é a quantidade de amostras.

Como já discutido, na camada de saída costuma-se utilizar a função de ativação sigmoide. Assim, uma interpretação para a Equação 2.7 é que ela penaliza o modelo quando a probabilidade estimada para uma amostra se distancia de sua classe real. Por exemplo, considere que uma amostra pertença à classe positiva ( $y = 1$ ); se o modelo prevê uma probabilidade de apenas 0,01, o erro será significativamente maior do que se tivesse previsto 0,6.

Para minimizar essa função de custo, redes neurais artificiais utilizam algoritmos baseados no *gradiente descendente*. Segundo Géron (2022), o gradiente descendente busca encontrar o mínimo de uma função ajustando iterativamente seus parâmetros. Intuitivamente, a função de custo define uma superfície de erro, e o modelo inicia em um ponto aleatório dessa superfície — isto é, com pesos inicializados aleatoriamente.

A cada iteração, o gradiente da função é calculado no ponto atual, indicando a direção de maior crescimento da função de custo. Os parâmetros são então atualizados na direção oposta, ou seja, na direção de maior decaimento. O tamanho desse passo é controlado por um hiperparâmetro chamado taxa de aprendizado (*learning rate*), denotado por  $\eta$ . Formalmente, a atualização é dada por:

$$\theta^{(i+1)} = \theta^{(i)} - \eta \nabla f(\theta) \quad (2.8)$$

onde  $\theta$  representa o vetor de parâmetros do modelo (pesos e vieses), e  $\nabla f(\theta)$  é o gradiente da função de custo em relação a esses parâmetros.

Na prática, utiliza-se variações do algoritmo, como o *stochastic gradient descent* (SGD), que atualiza os parâmetros a partir de amostras individuais ou pequenos subconjuntos (*mini-batches*) dos dados, e otimizadores mais sofisticados como o Adam, que ajusta automaticamente a taxa de aprendizado ao longo do treinamento.

O processo de cálculo dos gradientes é realizado por outro algoritmo, chamado de *backpropagation*. Segundo Géron (2022), um dos principais obstáculos históricos para o uso eficiente de redes neurais artificiais era a dificuldade de ajustar seus parâmetros quando a rede possuía muitas camadas. Aprofundar a rede, entretanto, é essencial para que ela aprenda representações (*features*) mais complexas dos dados.

Em 1985, David Rumelhart, Geoffrey Hinton e Ronald Williams demonstraram que um algoritmo proposto anteriormente por Seppo Linnainmaa, em 1970, poderia ser aplicado para esse fim. Esse algoritmo, atualmente conhecido como *backpropagation*, tornou possível o treinamento eficiente de redes com múltiplas camadas.

De forma simplificada, o *backpropagation* é composto por duas etapas principais: *forward pass* e *backward pass*.

Na fase *forward*, os dados de entrada percorrem a rede camada por camada. Cada neurônio realiza seu cálculo, multiplicando os valores de entrada pelos pesos correspondentes, somando o viés e aplicando a função de ativação. O resultado dessa sequência de operações é a saída da rede, que é então comparada com o valor real (*ground truth*) por meio de uma função de perda, que mede o erro do modelo.

Na fase *backward*, o algoritmo utiliza a regra da cadeia do cálculo diferencial para determinar quanto cada parâmetro (peso e viés) contribuiu para o erro total. Esse cálculo é feito de trás para frente, da camada de saída até a camada de entrada, ajustando os gradientes de forma eficiente. Dessa maneira, cada peso recebe uma medida precisa de sua influência no erro observado.

Por fim, os parâmetros são atualizados aplicando-se o gradiente descendente — ou outra técnica de otimização —, movendo os pesos na direção que reduz o erro da rede.

Redes profundas, entretanto, podem ser propensas a um problema denominado genericamente como gradiente instáveis que se manifesta de duas formas distintas: desvanecimento do gradiente — *vanishing gradient* — e a explosão do gradiente — *exploding gradient* — que ocorre quando os pesos ficam exponencialmente pequenos ou exponencialmente grandes respectivamente.

No primeiro caso, a atualização torna-se cada vez menor conforme o gradiente flui para as camadas mais iniciais. Na prática a rede para de aprender. Por exemplo, considere uma rede com três camadas. Se a taxa de contribuição do gradiente da camada três for 0,5 e o gradiente local da camada dois for 0,3, a contribuição do erro nesta camada (devido a regra da cadeia) será de  $0,5 \times 0,3 = 0,15$ . Na primeira camada que possui um gradiente local de 0,1, o erro seria de  $0,15 \times 0,1 = 0,015$ . Com mais camadas, esse erro seria multiplicado por gradientes cada vez menores, chegando a zero nas camadas iniciais. Como pode ser visto na equação 2.8, isso significa que esses pesos não seriam atualizados.

## 2.5 Redes Neurais Recorrentes e suas variações

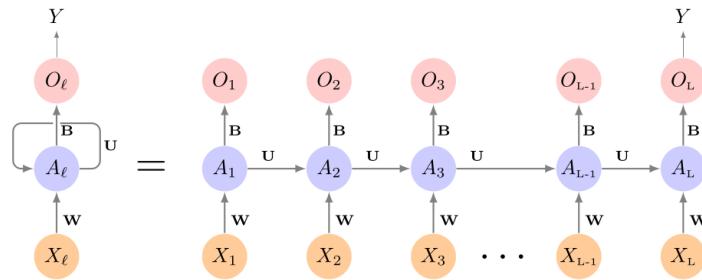
Uma rede neural recorrente — *Recurrent Neural Network* (RNN) — pode ser entendida, de forma intuitiva, como uma rede que possui memória. Segundo James et al. (2023), as RNNs se destacam em tarefas que envolvem dados com natureza sequencial, como:

- **Documentos textuais** — como resenhas de livros, filmes, artigos jornalísticos ou *tweets*. Nesses casos, a sequência e a posição relativa das palavras ajudam a capturar a narrativa, os temas e o tom do texto. Entre as aplicações estão a análise de sentimento, a tradução automática e a sumarização de textos.
- **Séries temporais** — como temperatura, precipitação, velocidade do vento, qualidade do ar, entre outros. As RNNs podem ser usadas para prever o comportamento dessas variáveis em diferentes horizontes de tempo, de dias a décadas.
- **Sinais sonoros** — como voz gravada e música. Aplicações incluem legendagem automática, tradução de fala e classificação de sons.

A entrada de uma RNN é uma sequência. Por exemplo, em uma tarefa de classificação de documentos, um texto pode ser representado como uma sequência  $X = \{X_1, X_2, \dots, X_L\}$  de  $L$  elementos, onde cada  $X_l$  representa uma palavra (ou vetor de características da palavra).

Na Figura 5, é ilustrada uma RNN simples com apenas um neurônio em cada camada — de entrada, oculta e saída.

Figura 5 – Exemplo de uma RNN



Fonte: Adaptado de James et al. (2023)

O *loop* à esquerda representa a retroalimentação da rede. À direita, a RNN é “desenrolada” no tempo: a rede processa um elemento  $X_l$  de cada vez e produz a ativação  $A_l$ , calculada a partir tanto da entrada atual quanto da ativação anterior  $A_{l-1}$ . Essa ativação é então usada para gerar a saída  $O_l$ . Neste exemplo, apenas a última ativação da sequência é utilizada como saída final.

Mais formalmente, suponha que cada entrada  $X_l$  possua  $p$  componentes:  $X_l^T = (X_{l1}, X_{l2}, \dots, X_{lp})$ , e que a camada oculta tenha  $K$  unidades:  $A_l^T = (A_{l1}, A_{l2}, \dots, A_{lK})$ . A matriz  $\mathbf{W}$  representa os  $K \times (p + 1)$  pesos da camada de entrada,  $\mathbf{U}$  representa os  $K \times K$  pesos recorrentes (entre ativações sucessivas), e  $\mathbf{B}$  é o vetor de pesos da camada de saída. A ativação é calculada por:

$$A_{lk} = g \left( w_{k0} + \sum_{j=1}^p w_{kj} X_{lj} + \sum_{s=1}^K u_{ks} A_{l-1,s} \right) \quad (2.9)$$

A saída é dada por:

$$O_l = \beta_0 + \sum_{k=1}^K \beta_k A_{lk} \quad (2.10)$$

Em problemas de classificação binária, a função sigmoide pode ser usada na camada de saída. Note que as matrizes  $\mathbf{U}$ ,  $\mathbf{W}$  e  $\mathbf{B}$  são compartilhadas entre todas as etapas temporais, o que permite que a rede capture dependências sequenciais nos dados.

Segundo Géron (2022), a entrada e saída de uma RNN pode ser configurada de três maneiras:

- **Sequencia para sequencia (sequence-to-sequence)** — A entrada e a saída são sequencias. Um exemplo seria a previsão do consumo de energia onde a rede recebe o consumo nos últimos  $N$  dias e prever os próximos  $N - 1$  dias.
- **Sequencia para vetor (sequence-to-vector)** — Apenas a entrada é uma sequência, como na figura 5. Uma aplicação é análise de sentimentos de uma resenha, onde a rede recebe uma sequência de palavras e deve classificar em: gostei ou não gostei.

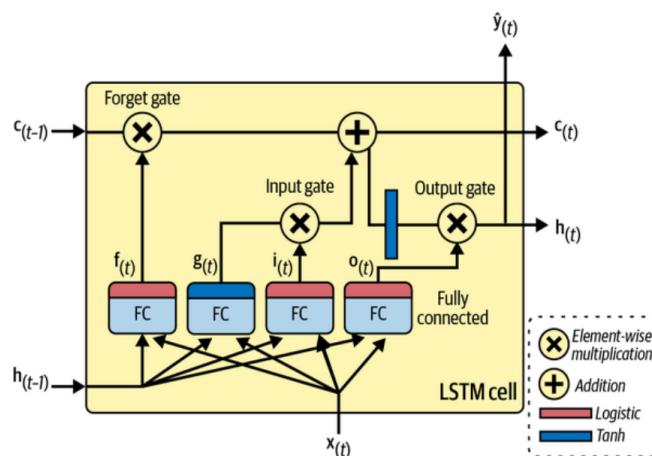
- **Vetor para sequência (vector-to-sequence)** — Apenas a saída é uma sequência. A rede recebe o mesmo vetor em cada passo de tempo. Um exemplo é gerar a legenda de uma imagem. Assim, a rede receberia a imagem ou mesmo a saída de uma CNN e baseado nisso, ela gera a legenda.

Existem duas variações de RNN muito utilizadas: a LSTM — *long short term memory* e a GRU — *gated recurrent unit*. Ambas foram desenvolvidas para amenizar o problema do *vanishing gradient* que ocorre na camada recorrente da RNN, o que faz com que a rede "esqueça" sequências longas.

### 2.5.1 Long Short-Term Memory (LSTM)

Géron (2022) descreve a LSTM (*Long Short-Term Memory*) como uma célula recorrente que combina três componentes principais: uma memória de curto prazo, uma memória de longo prazo e um mecanismo de esquecimento. Na Figura 6, é apresentado o diagrama de uma célula LSTM típica.

Figura 6 – Esquema de uma célula LSTM



Fonte: Adaptado de Géron (2022)

A célula LSTM é composta por quatro unidades *fully connected* (FC), formando uma estrutura semelhante à apresentada na Seção 2.4. As saídas dessas unidades são representadas por  $f_t$ ,  $i_t$ ,  $o_t$  e  $g_t$ , onde as três primeiras correspondem às portas de controle — *forget* (**f**), *input* (**i**) e *output* (**o**) — enquanto  $g_t$  é a saída da unidade principal (**g**). As variáveis  $h_t$  e  $c_t$  representam, respectivamente, a memória de curto prazo (*hidden state*) e a memória de longo prazo (*cell state*).

De forma simplificada, o funcionamento de uma LSTM ocorre da seguinte maneira: a entrada atual  $X_t$  e a memória de curto prazo anterior  $h_{t-1}$  são fornecidas à célula. Essas duas informações são processadas pelas quatro camadas FC, que calculam os vetores  $f_t$ ,  $i_t$ ,  $o_t$  e  $g_t$ .

O vetor  $f_t$  atua como um filtro de esquecimento, controlando qual fração da memória anterior  $c_{t-1}$  deve ser mantida. Em seguida, o vetor  $i_t$  define o quanto da nova informação  $g_t$  será incorporada à memória. O novo estado de memória de longo prazo é então atualizado conforme:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

Por fim, a saída da célula é obtida aplicando uma função tangente hiperbólica sobre  $c_t$  — cuja fórmula é dada em 2.4 —, modulada pela porta de saída  $o_t$ :

$$h_t = o_t \odot \tanh(c_t)$$

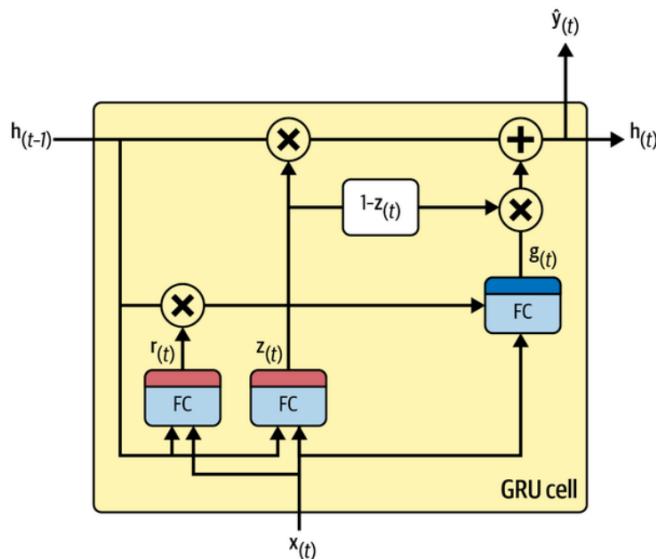
onde o operador  $\odot$  indica a multiplicação elemento a elemento (*element-wise*).

O controle exercido por cada porta ocorre porque suas funções de ativação são do tipo logística (sigmoide), cujo contradomínio é o intervalo  $[0, 1]$ . Assim, o valor produzido por cada porta funciona como um coeficiente de passagem: se a ativação for próxima de 0, a informação é bloqueada; se for próxima de 1, ela é totalmente transmitida. Essa estrutura permite que a LSTM controle de forma adaptativa o fluxo de informação ao longo do tempo, mitigando o problema do *vanishing gradient* presente em RNNs tradicionais.

### 2.5.2 GRU

Segundo Géron (2022), a Gated Recurrent Unit (GRU) — proposta por Kyunghyun Cho et al. (2014) — é uma versão simplificada da LSTM que mantém desempenho equivalente em diversas tarefas. Na figura 7, é apresentado um esquema ilustrativo dessa célula.

Figura 7 – Esquema de uma célula GRU



Fonte: Adaptado de Géron (2022)

Os vetores de estado  $r_t$ ,  $z_t$  e  $g_t$  correspondem às três unidades totalmente conectadas (**r**, **z** e **g**), enquanto  $h_t$  representa a memória da célula. A unidade **r** controla a quantidade de informação de  $h_{t-1}$  que é transmitida para **g**, enquanto **z** desempenha o papel do mecanismo de esquecimento. Note que há multiplicações elemento a elemento entre  $z_t$  e  $h_{t-1}$ , e entre  $(1 - z_t)$  e  $g_t$ ; desse modo, para que novas informações sejam incorporadas à memória, parte das informações antigas precisa ser descartada.

O mecanismo de controle da GRU é, portanto, mais simples que o da LSTM, o que se traduz em um número menor de parâmetros a serem aprendidos durante o treinamento.

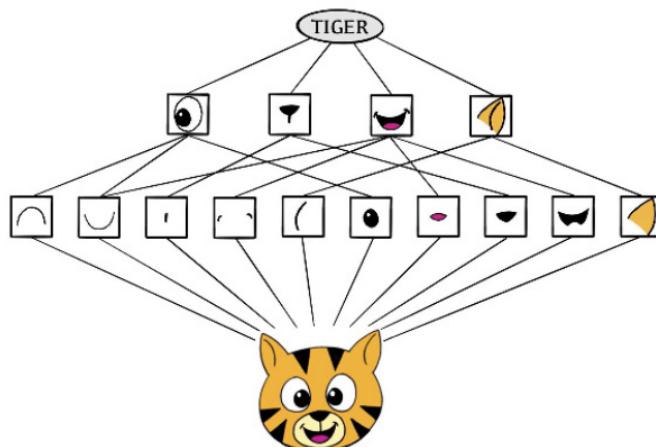
Cabe ressaltar, entretanto, que a simplicidade da GRU não implica necessariamente em melhor desempenho. Estudos como Narotamo et al. (2024) e Saadatnejad, Oveisi e Hashemi (2020) relatam resultados distintos em diferentes contextos e conjuntos de dados.

## 2.6 Redes Neurais Convolucionais

Segundo James et al. (2023), as redes neurais convolucionais — ou CNNs (*Convolutional Neural Networks*) — tiveram grande destaque a partir de 2010, especialmente em tarefas de classificação de imagens. Esse avanço se deve, em parte, ao surgimento de grandes bases de dados e ao aumento da capacidade computacional, mas também à criação de uma nova arquitetura de rede neural, inspirada no funcionamento da percepção visual humana.

As CNNs constroem uma hierarquia de *features* que se tornam progressivamente mais complexas. Na figura 8, é apresentado um esquema ilustrativo de como uma CNN reconheceria um tigre: nas primeiras camadas são extraídos padrões simples, como bordas e formas básicas; então, elas são combinadas para formar os olhos, orelhas, até que, enfim, a rede reconhece o tigre.

Figura 8 – Esquema de como uma CNN identificaria um tigre



Fonte: Adaptado de James et al. (2023)

De modo geral, uma CNN é composta por duas estruturas principais: as *camadas convolucionais*, responsáveis por extrair características relevantes dos dados, e as *camadas de pooling*, que reduzem a dimensionalidade e mantêm apenas as informações mais significativas.

Para compreender o funcionamento de uma convolução, considere a seguinte matriz, representando uma imagem:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

e um filtro convolucional  $2 \times 2$ :

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

A operação de convolução consiste em multiplicar o filtro elemento a elemento com cada submatriz  $2 \times 2$  da imagem, somar os resultados e, em seguida, deslizar o filtro sobre a matriz. O resultado dessa operação é:

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

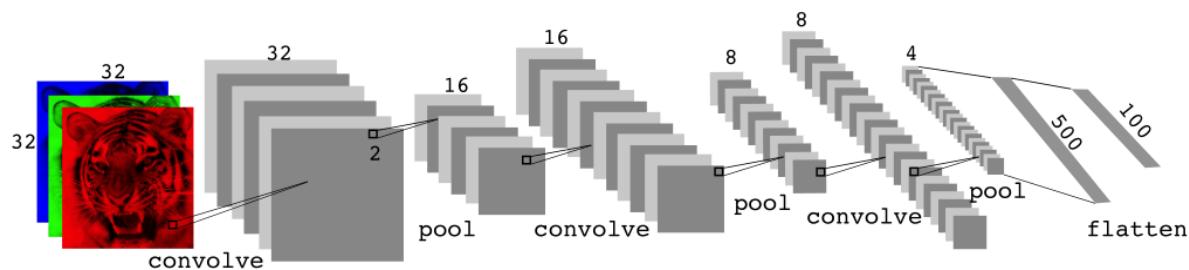
O filtro atua como um detector de padrões locais: se uma submatriz se assemelha ao filtro, ela será destacada na saída. É importante notar que a convolução **não** é uma multiplicação matricial tradicional.

Após a convolução, aplica-se normalmente uma camada de *pooling*, que resume as *features* extraídas. O *max pooling*, por exemplo, seleciona o maior valor em cada submatriz  $2 \times 2$  não sobreposta:

$$\text{Max pool } \begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 5 \\ 2 & 4 \end{bmatrix}$$

Na figura 9, é ilustrada uma arquitetura típica de CNN.

Figura 9 – Arquitetura típica de uma CNN



Fonte: Adaptado de James et al. (2023)

A entrada consiste em uma imagem tridimensional, na qual a terceira dimensão — denominada *canal* — corresponde às cores no modelo RGB. A primeira camada convolucional transforma a imagem inicial em uma nova representação, por exemplo, de dimensão (32, 32, 6), onde o número 6 corresponde à quantidade de filtros aplicados. Em seguida, uma camada de *pooling* reduz essa dimensão para (16, 16, 6). Novas camadas convolucionais e de *pooling* podem ser adicionadas até que, ao final, os dados sejam achatados (*flattened*) e passados para camadas totalmente conectadas (*FC*) para a classificação.

De modo geral, cada camada convolucional aumenta a profundidade do canal (número de filtros), enquanto as camadas de *pooling* reduzem as dimensões espaciais da imagem.

Por fim, conforme Géron (2022), CNNs não se limitam a imagens: elas também podem ser aplicadas a sinais unidimensionais. Nesses casos, um filtro unidimensional é deslizado sobre o sinal, permitindo que a rede aprenda padrões locais (com tamanho máximo igual ao do filtro). Com múltiplos filtros, obtém-se uma sequência de *features* que podem, em seguida, ser processadas por uma RNN, capturando dependências temporais mais longas.

## 2.7 Trabalhos correlatos

A aplicação de métodos para a detecção de arritmias não é uma ideia nova. Conforme Moody e Mark (2001), desde a década de 60, havia estudos que buscavam desenvolver algoritmos para a análise de arritmias. Safdar, Nowak e Pałka (2024), aponta algumas técnicas clássicas utilizadas como *Support Vector Machine (SVM)*, *K-nearest neighbor (KNN)* e *Random Forest*.

Os autores mostram que nos últimos anos técnicas de *deep learning*, redes neurais, vem ganhando tração. Muito por dispensar ou depender menos intensamente de *features*

manuais, podendo aprendê-las automaticamente. Dentre as primeiras técnicas usadas está as redes *feed-forward*, seguido pelas CNN e LSTM, além de abordagens híbridas.

A presente seção apresenta uma revisão de alguns trabalhos voltados a aplicação de modelos de aprendizado de máquina para a detecção de arritmias cardíacas.

Chazal, O'Dwyer e Reilly (2004) propuseram um método automático para classificação de arritmias nas cinco classes definidas pela AAMI (descritas na seção 2.3.2), utilizando o banco de dados MIT-BIH. O classificador empregado foi o discriminante linear, um método estatístico paramétrico baseado na suposição de distribuição Gaussiana dos dados.

A metodologia incluiu etapas de pré-processamento, extração de features e classificação. Foram avaliadas diferentes configurações e estratégias de particionamento, incluindo a tradicional (intrapaciente) e a proposta pelos autores (interpaciente, com os conjuntos Ds1 e Ds2 descritos em 3.1). O estudo demonstrou que a participação intrapaciente gera resultados otimistas e não representa adequadamente a capacidade de generalização do modelo.

O pré-processamento aplicava filtros de mediana e passa-baixa para remoção do baseline wander. Em seguida, as features foram extraídas a partir de duas abordagens principais: (i) janelas fixas centradas no pico R e (ii) janelas adaptadas à duração do batimento, considerando o complexo QRS e a onda T. Também foram utilizadas medidas temporais, como duração do QRS, intervalos RR e presença da onda P.

A melhor configuração encontrada combinava features segmentadas com janela temporal flexível, intervalos RR, presença da onda P e durações do QRS e da onda T — todas extraídas sem escalar o sinal —, processadas por dois discriminantes lineares independentes, um para cada derivação do MIT-BIH. Essa abordagem obteve desempenho superior às demais e se tornou referência para estudos posteriores.

Apesar dos bons resultados, o método depende fortemente da qualidade das features e do pré-processamento, sendo sensível à variação morfológica entre pacientes — um ponto que trabalhos mais recentes, baseados em aprendizado profundo, buscam superar.

Mousavi e Afghah (2018), 2019, propuseram um modelo baseado em *encoder-decoder* para a classificação de cada batimento em uma sequência (*sequence-to-sequence*, ou seq2seq). O pré-processamento consiste na normalização do sinal, segmentação em batimentos individuais e reamostragem para 280 amostras, sem o uso de filtros para remoção de ruídos. O modelo foi avaliado nas partições interpaciente e intrapaciente, utilizando os conjuntos Ds1 e Ds2.

A arquitetura é composta por uma sub-rede convolucional com três camadas contendo 32, 64 e 128 filtros, responsável pela extração inicial das *features*. As representações obtidas alimentam o *encoder*, que codifica a sequência em uma nova representação latente, utilizada pelo *decoder* para gerar uma sequência de classificações. Tanto o *encoder* quanto o *decoder* são implementados com LSTM bidirecionais, de modo que a rede processa a

sequência de entrada em ambas as direções — direta e reversa.

Para lidar com o desbalanceamento das classes, os autores aplicaram a técnica *SMOTE (Synthetic Minority Oversampling Technique)* para gerar amostras sintéticas apenas no conjunto de treinamento, aumentando a representatividade das classes minoritárias sem afetar os dados de validação e teste.

Em Zhang et al. (2021), é proposta uma rede neural convolucional profunda para a classificação de arritmias nas cinco classes definidas pela AAMI. Os autores enfatizam a importância de o modelo aprender características invariantes entre pacientes, de modo a melhorar sua capacidade de generalização. Enquanto outros trabalhos — como este TCC — buscam que o modelo aprenda essas características diretamente a partir dos dados, Zhang et al. propuseram uma estratégia adversarial para induzir esse comportamento.

A arquitetura é composta por um encoder, responsável por extrair as características mais relevantes da entrada, seguido por dois ramos: um classificador, que prevê o tipo de batimento, e uma rede adversária, que tenta identificar a qual paciente o batimento pertence. A função de perda é baseada na cross-entropy (uma generalização da BCE para múltiplas classes) e é modificada de forma a maximizar a perda da rede adversária enquanto minimiza a do classificador.

Esses dois objetivos são conflitantes: quanto melhor a rede adversária for em identificar o paciente, menor será a invariância das features aprendidas. Assim, um hiperparâmetro  $k$  é introduzido para controlar o trade-off entre a generalização (obtida ao maximizar a perda adversária) e o desempenho de classificação (que tende a cair quando o modelo foca apenas em padrões gerais).

A entrada do modelo consiste em batimentos individuais e em razões entre intervalos RR: a razão entre o pré-RR e a média dos pré-RR da gravação, e a razão entre o pré-RR e o pré-RR mais próximo. Os autores destacam que o intervalo pré-RR tem alta capacidade discriminativa para distinguir batimentos ventriculares e supraventriculares de batimentos normais — ainda que, isoladamente, não seja suficiente para separar esses dois tipos de arritmia entre si.

Em Kiranyaz, Ince e Gabbouj (2016), foi proposto um leve modelo de classificação de arritmias baseado em CNN (ver Seção 2.6). Os autores seguiram o particionamento intra-paciente com cinco minutos de treinamento específico — oriundo do conjunto ds2 — e treinamento global a partir de amostras aleatórias do conjunto ds1. Os dados de treino totais somam menos de 1% dos batimentos totais.

A AAMI define que dados de pacientes específicos não podem ultrapassar cinco minutos; regra seguida pelos autores.

Não foi utilizado engenharia de features, empregando apenas o sinal crú em duas entradas: uma com o batimento segmentado e uma outra com três sequências de batimentos; para fornecer contexto temporal. Em ambos os casos os autores utilizaram 64 ou 128 amostras, sendo que no primeiro caso, a perda de desempenho foi insignificante. O

modelo é formado por três camadas convolucionais modificadas, isto é, elas fazem tanto a convolução quanto o *downsampling*, e duas camadas de perceptrons.

No trabalho, os autores destacam a leveza do modelo, justificando que a partição usada dispensa o uso de redes profundas e o uso de engenharia de features.

Em Saadatnejad, Oveisi e Hashemi (2020), foi proposto um modelo baseado em redes LSTM (ver Seção 2.5) para a classificação de arritmias voltado a dispositivos móveis. Com o objetivo de atender às restrições computacionais de sistemas embarcados, os autores desenvolveram uma arquitetura composta por dois modelos simples que realizam classificações independentes, seguidos por uma rede neural artificial (ANN) responsável pela decisão final.

A entrada do modelo consiste em batimentos cardíacos segmentados utilizando uma janela temporal fixa, características extraídas via transformada *wavelet* de Daubechies de ordem dois, e intervalos RR, conforme sugerido por Chazal, O'Dwyer e Reilly (2004). O sistema é dividido em dois submodelos, denominados  $\alpha$  e  $\beta$ . O primeiro ( $\alpha$ ) possui dois ramos de LSTM: um recebe o sinal reamostrado (*downsampled*) concatenado com os intervalos RR, enquanto o outro recebe o sinal combinado com as características *wavelet*. Já o segundo submodelo ( $\beta$ ) é composto por duas camadas LSTM, recebendo como entrada os componentes principais do sinal, concatenados com as características *wavelet* e os intervalos RR. Em ambos os casos, a saída das LSTM é conectada a uma camada totalmente conectada (FC, *Fully Connected*) para a classificação.

O treinamento foi conduzido sob uma divisão *intrapaciente*, na qual foi desenvolvido um modelo especialista para cada indivíduo. De acordo com as recomendações da AAMI, os autores separaram os dados em conjuntos locais e globais: os dados locais corresponderam aos cinco primeiros minutos dos registros dos pacientes do conjunto Ds2, enquanto o conjunto global foi composto pelo conjunto Ds1 completo. Para teste, utilizou-se o restante do conjunto Ds2, excluindo os minutos empregados no treinamento.

Por se tratar de um modelo voltado a dispositivos vestíveis — e, portanto, de uso pessoal —, essa divisão não constitui necessariamente um vazamento de dados, embora o viés de sobreposição entre indivíduos permaneça. Assim, é importante considerar esse aspecto ao comparar os resultados com trabalhos que adotam a divisão *interpaciente*.

De modo geral, observa-se que os trabalhos diferem amplamente em suas escolhas metodológicas — desde o tipo de particionamento (intrapaciente ou interpaciente), passando pela inclusão ou não de *features* manuais, até o uso de arquiteturas simples, híbridas ou profundas. Essas diferenças dificultam comparações diretas de desempenho, mas permitem identificar tendências metodológicas importantes que orientam o desenvolvimento de modelos mais robustos e generalizáveis, como os abordados neste TCC.

Na tabela 1 é resumido a abordagem dos trabalhos apresentados incluído a utilizada neste TCC.

Tabela 1 – Metodologias de Classificação de Batimentos Cardíacos: Modelos, Particionamento, Features, Segmentação e Limpeza

Trabalho (Ano)	Tipo de Modelo (Arquitetura)	Particionamento	Features Incluídas	Segmentação do Batimento	Limpeza / Pré-processamento
de Chazal et al. (2004)	<b>Classificador Estatístico / Discriminantes Lineares (LDs)</b> (Configuração IX: 2 LDs combinados).	<b>Inter-Patient</b> (Baseado em Registros – DD1). Treino em DS1, Teste em DS2 (Independente).	<ul style="list-style-type: none"> <li>Morfologia ECG (amostragem segmentada e de intervalo fixo).</li> <li>Intervalos de Batimento (Duração QRS/T, Presença P-wave).</li> <li>Intervalos RR (Pré/Pós-RR, RR Médio, RR Local).</li> </ul>	Batimentos <b>manualmente detectados</b> (pontos fiduciais). Segmentação QRS/T estimada por programa externo.	<ul style="list-style-type: none"> <li><b>Filtros de Mediana</b> (200ms e 600ms de largura) para remover <i>baseline wander</i>.</li> <li><b>Filtro passa-baixa de 12 taps</b> (3-dB em 35 Hz) para ruído de alta frequência.</li> <li>Ponderação de exemplos para evitar domínio de classes grandes.</li> </ul>
Kiranyaz et al. (2016)	<b>CNN 1-D Adaptativa</b> (Funde extração de features e classificação).	<b>Patient-Specific</b> . Treino: Dados Globais + Dados Locais (primeiros 5 min do paciente).	<ul style="list-style-type: none"> <li><b>Amostras brutas de ECG</b> (64 ou 128 amostras no pico R).</li> <li>Beat Trio (captura características temporais dos vizinhos).</li> <li>Representação FFT (Opcional).</li> </ul>	Segmentação em <b>64 ou 128 amostras centradas no pico R</b> . Uso de <i>Beat Trio</i> para contexto temporal.	Dados crús.
Mousavi et al. (2019)	<b>Sequence-to-Sequence Deep Learning</b> (CNN-Bi-RNN Encoder-Decoder com LSTM).	Testado nos paradigmas <b>Inter-patient</b> e <b>Intra-patient</b> .	<ul style="list-style-type: none"> <li><b>Amostras brutas de ECG</b>.</li> <li>CNN automaticamente extrai 128 características.</li> </ul>	Sinal de ECG contínuo dividido em <b>sequências de 280 amostras</b> .	<ul style="list-style-type: none"> <li><b>Filtro passa-banda</b> (0.5 – 40 Hz).</li> <li>Subtração da onda T/P.</li> <li>Aumento de dados (<b>SMOTE</b>) para reequilibrar classes minoritárias.</li> </ul>
Li et al. (2019)	<b>Rede Neural Convolucionial Adversarial</b> (Adversarial CNN) (Encoder-Classifier-Adversary).	<b>Inter-patient</b> (DS1 Treino, DS2 Teste), garantindo a separação de pacientes.	<ul style="list-style-type: none"> <li><b>Amostras brutas de ECG</b> (2 canais).</li> <li>Features RR Interval (pré-RR ratio, near-pre-RR ratio, RR médio).</li> </ul>	<b>150 pontos</b> centrados no pico R (50 antes do pico R e 100 depois).	<ul style="list-style-type: none"> <li><b>Remoção do baseline wander</b> por escalonamento da média dos segmentos subtraídos.</li> </ul>

Continua na próxima página...

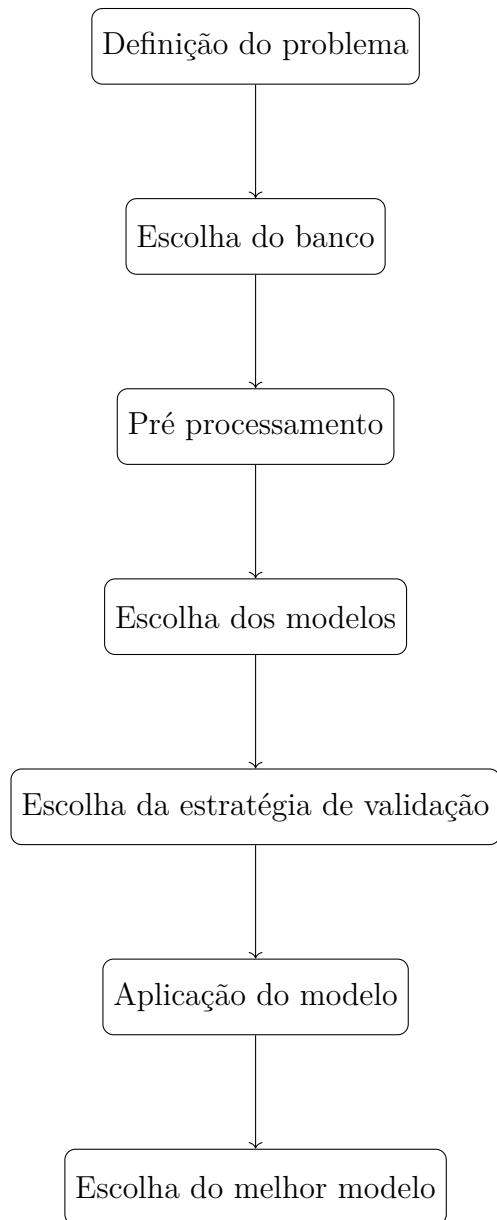
Tabela 1 – Metodologias de Classificação de Batimentos Cardíacos: Modelos, Particionamento, Features, Segmentação e Limpeza

Trabalho (Ano)	Tipo de Modelo (Arquitetura)	Particionamento	Features Incluídas	Segmentação do Batimento	Limpeza / Pré-processamento
Saadatnejad et al. (2020)	<b>LSTM-Based RNN</b> (Múltiplas LSTMs combinadas: Modelos $\alpha$ e $\beta$ ). Arquitetura Híbrida Wavelet + LSTM.	<b>Patient-Specific.</b> Treino: Local (primeiros 5 min) + Global (batimentos rep.).	<ul style="list-style-type: none"> <li>• Amostras brutas de ECG (<math>X_{ecg}</math>).</li> <li>• Features de Wavelet (db2, 4 níveis) (<math>X_w</math>).</li> <li>• Features de Intervalo RR (<math>X_{rr}</math>: RR anterior/seguinte, RR médio local/geral).</li> </ul>	Segmento de comprimento fixo baseado no pico R (Pan-Tompkin). 0.25s antes e 0.45s depois do pico R.	<ul style="list-style-type: none"> <li>• Down sampling por um fator de 2 antes da Transformada Wavelet (DWT).</li> <li>• Uso da DWT para capturar info. de domínio de tempo e frequência.</li> </ul>
Este TCC	<b>GRU e GRU com CNN</b> (Comparação entre dois modelos distintos).	<b>Interpaciente.</b> Cross-validation no DS1.	<ul style="list-style-type: none"> <li>• Sequência de 16 batimentos</li> <li>• Intervalo RR anterior/posterior.</li> </ul>	Janela baseada no tamanho do paciente seguido por reamostragem.	Remoção de <i>baseline wander</i> e <i>powerline</i> .

### 3 Metodologia

A metodologia é esquematizada na figura 10 que ilustra o fluxo de trabalho. Esse processo envolve a definição do problema, a escolha do banco, o pré-processamento, escolha das estratégias de validação, e aplicação do modelo, culminando na escolha do melhor modelo.

Figura 10 – Esquema a metodologia adotada.



Fonte: Elaborado pelo autor.

Cada um desses passos introduz considerações, e as decisões tomadas influenciam nos passos seguintes. Por exemplo, a escolha do banco impacta diretamente em quais tipos

de pré-processamento necessário, como a limpeza. Contanto, antes mesmo da escolha do banco, é necessário definir qual o problema, visto que as anotações presentes podem limitar o escopo dos problemas resolvidos.

Nas seções subsequentes, detalha-se as decisões adotadas em cada etapa da metodologia, bem como os critérios considerados para tais escolha.

### 3.1 O banco de dados

Optou-se pelo *MIT-BIH Arrhythmia Database* (MOODY; MARK, 2005). Segundo Moody (1997), o banco é composto por 58 registros de eletrocardiograma (ECG), cada um com 30 minutos de duração. Os 23 primeiros registros, de 100 a 124, foram selecionados aleatoriamente a partir de um conjunto de 4000 gravações de 24 horas realizadas em pacientes ambulatoriais do Beth Israel Deaconess Medical Center. Os 25 registros, 200 até 234, restantes foram escolhidos de modo a incluir arritmias raras e com formato complexo, mas clinicamente significativas. Cada uma das anotações foram feitas com por três cardiologistas independentes. Os sinais foram coletados com duas derivações; uma superior e outra inferior. A superior é majoritariamente utilizando a derivação MLII (modified limb II) que é feita com o eletrodo no peito. Em alguns casos, foi utilizado as derivações V1 (ou mais raramente, V2, V3 e V4); que também são obtidas com os eletrodos no peito.

Neste trabalho, foi utilizado somente a derivação superior, pois permite uma melhor visão do complexo QRS (MOODY, 1997).

Na tabela 2, é detalhado o mapeamento entre as classes originais de batimentos para as cinco definidas pela AAMI.

Tabela 2 – Mapeamento das anotações originais do MIT-BIH para as classes AAMI.

Anotação Original	Classe AAMI
N, e, j, L, R	N (Normal)
A, a, J, S	S (Supraventricular)
V, E	V (Ventricular)
F, f	F (Fusão)
Q, ?, /	Q (Desconhecida)

Fonte: Adaptado de Chazal, O'Dwyer e Reilly (2004)

O objetivo foi a detecção de batimentos da classe V que compreende: contração prematura ventricular (ou PVC, classe V) e batimento ventricular de escape, classe E, (MOODY, 1997). Conforme discutido na seção 2.3.2, apesar de ocorrerem em indivíduos saudáveis, esses tipos arrítmicos possuem relevância clínica pois estão associados a tipos mais graves.

Essas anotações são anotações de batimento, isto é, elas são feitas em cada pico R no ECG. Além delas, existem as anotações de ritmo dentre as quais, podemos destacar: o ritmo normal identificado por (N, e a taquicardia ventricular; identificado por (VFL. Dentro de um contexto rítmico, podem haver batimentos normais ou arrítmicos.

Por exemplo, na figura 11, é mostrado o trecho de um ECG. Note a anotação de ritmo, (N, indicando que o mesmo é normal. Note, também, que dentro desse contexto rítmico, existem batimentos normais, sinalizados por um ponto em cada pico R, uma arritmia superventricular, mais precisamente, o batimento atrial prematuro, classe A.

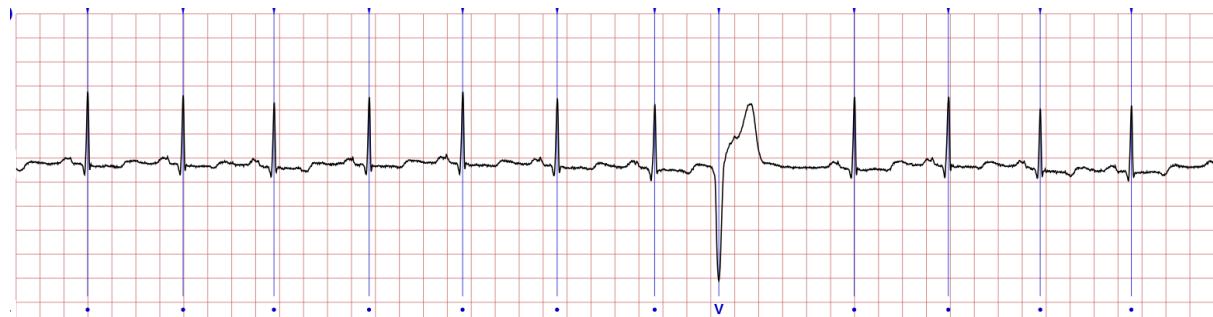
Figura 11 – Trecho ECG com ritmo normal do paciente 100 com arritmia classe A



Fonte: Adaptado de Physionet

Na figura 12 é mostrado um outro trecho do mesmo paciente, o ritmo também é normal

Figura 12 – Trecho ECG com ritmo normal do paciente 100 com arritmia classe V



Fonte: Adaptado de Physionet

Porém, nota-se um PVC, identificado pela anotação V. Na figura 13, um trecho do paciente 106 é mostrado. o ECG é de outro paciente.

Figura 13 – Trecho ECG com ritmo normal do paciente 106 com arritmia classe V



Fonte: Adaptado de Physionet

Aqui temos a ocorrência de uma taquicardia ventricular, identificado por (VT). Nela, ocorrem três PVCs em sequência. Note a diferença morfológica entre eles. Após esse evento, o ritmo é normal. Nesse segundo momento, ocorre um outro PVC.

As classes de ritmo não foram utilizadas explicitamente, visto que o objetivo era classificar batimentos. Ou seja, há sequencias com ritmo normal ou com taquicardia ventricular, mas o algoritmo não as classifica.

O MIT-BIH é um banco aberto e muito utilizado para a classificação de arritmias, permitindo uma comparação com demais trabalhos. Além de ser recomendado pela AAMI.

### 3.2 Pré-processamento

Antes de utilizar o sinal de ECG como entrada dos modelos, foi necessária uma etapa de pré-processamento composta por limpeza de ruídos, segmentação e padronização dos batimentos. Essa etapa é importante porque o ECG está sujeito a diversos artefatos que podem prejudicar o aprendizado das redes neurais. Como o exame registra a atividade elétrica do coração, correntes elétricas externas ou internas ao organismo podem modificar o sinal. Entre os ruídos mais comuns estão o ruído muscular (proveniente da contração de outros músculos), o baseline wander (variação lenta associada à respiração) e a interferência de 60 Hz da rede elétrica, como discutido na seção 2.7.

Apesar disso, alguns trabalhos utilizam o sinal praticamente cru, delegando ao próprio modelo o papel de identificar o que é ou não relevante. Essa abordagem simplifica o pré-processamento, mas aumenta a complexidade do problema de aprendizado e pode demandar bases maiores ou arquiteturas mais robustas.

Outro desafio importante é a segmentação do ECG em batimentos individuais. Conforme observado nos trabalhos correlatos, existem duas abordagens principais: o uso de janelas de tempo fixas ou janelas adaptadas ao tamanho do batimento. Janelas fixas são simples, mas podem cortar partes importantes do complexo QRS ou incluir trechos de

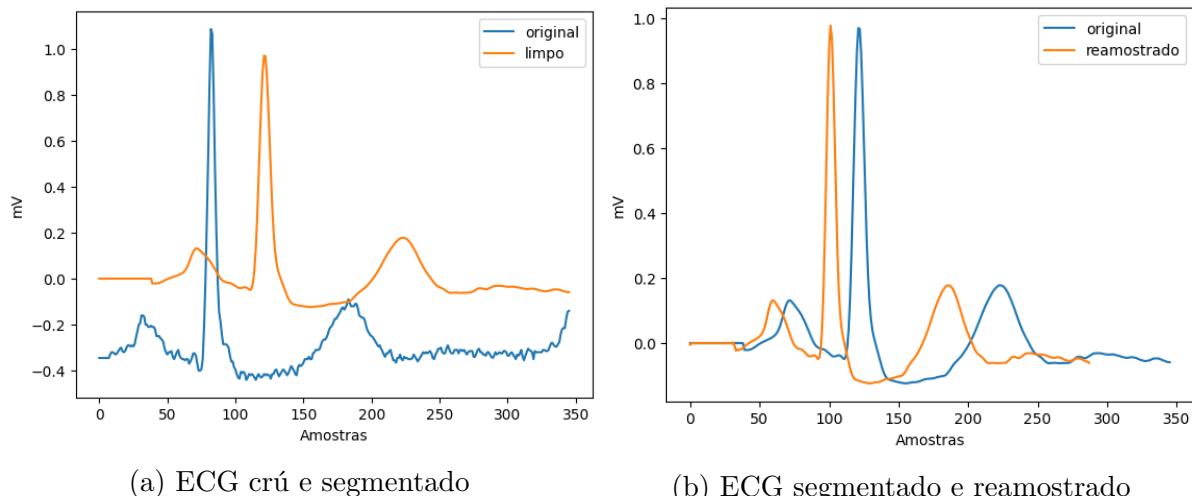
batimentos vizinhos. Em ambas as estratégias, o pico R é normalmente utilizado como referência. Uma vantagem do MIT-BIH é que esses picos já estão anotados; quando não estão, podem ser identificados por algoritmos como o de Pan e Tompkins (1985).

Neste trabalho, optou-se por trabalhar com o sinal o mais próximo possível do original, preservando suas características fisiológicas e facilitando análises posteriores. Por isso, adotou-se a segmentação flexível. Inicialmente, o sinal foi limpo com um filtro passa-alta de 0,5 Hz (ordem 5), seguido de filtragem da linha de energia a 60 Hz. Em seguida, foi executada a segmentação. Ambas as etapas utilizaram a biblioteca NeuroKit2 (MAKOWSKI et al., 2021).

Por fim, foi necessário uniformizar o tamanho dos batimentos antes de alimentá-los nos modelos. A média das amostras por batimento foi de aproximadamente 284; portanto, adotou-se uma reamostragem para 288 amostras, correspondendo a 800 ms de duração. Essa etapa foi realizada com a função resample da biblioteca SciPy (VIRTANEN et al., 2020).

Na figura abaixo é ilustrado um batimento segmentado e limpo e o seu trecho correspondente crú.

Figura 14 – Comparaçao entre o ECG original, segmentado e reamostrado



Fonte: Elaborado pelo autor.

### 3.2.1 Features

Após o pré-processamento, é necessário avaliar quais features devem ser utilizadas. Em problemas de ECG, essas features precisam ser relevantes para o domínio clínico. Por outro lado, redes neurais apresentam a vantagem de aprender representações diretamente dos dados brutos, o que reduz a necessidade de engenharia manual de atributos. Ainda as-

sim, algumas features simples podem complementar o aprendizado, fornecendo informação explícita que ajude o modelo a distinguir padrões.

Neste trabalho, optou-se por utilizar apenas o intervalo RR como feature adicional, permitindo que a própria rede aprenda as demais características relevantes a partir do sinal segmentado. O intervalo RR é uma informação particularmente útil porque descreve o tempo entre batimentos consecutivos. Essa informação ajuda na identificação de arritmias cuja principal manifestação é temporal, como no caso dos batimentos ventriculares prematuros (PVCs), em que ocorre um encurtamento característico desse intervalo.

Assim, para um batimento  $i$ , seu intervalo RR é calculado da seguinte forma.

$$\text{pré RR intervalo} = R_{i-1} - R_i \quad (3.1)$$

$$\text{pós RR intervalo} = R_i - R_{i+1} \quad (3.2)$$

Note que é necessário saber sobre o próximo batimento, isto é o futuro. Em um contexto de classificação em tempo real, por exemplo, isso poderia ser um vazamento, o que não é o caso deste projeto.

### 3.3 Arquiteturas

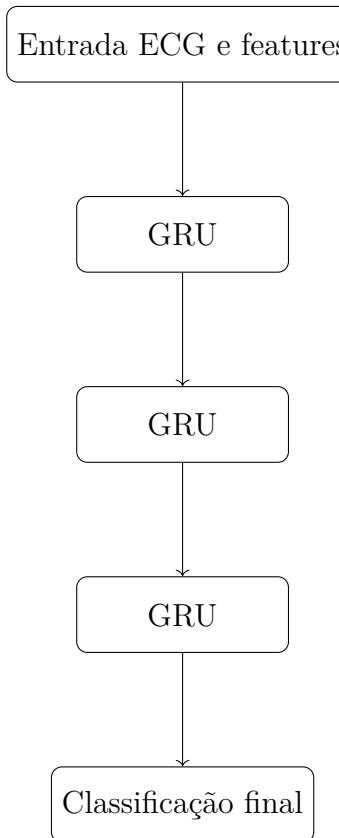
Após o pré-processamento e a definição das features, é necessário escolher as arquiteturas de rede neural que serão avaliadas. Como discutido por Géron (2022), a busca por hiperparâmetros ideais e por configurações de modelo costuma ser um processo altamente experimental, podendo incluir técnicas automáticas — como grid search ou random search —, mas que, no contexto de redes neurais profundas, frequentemente se torna inviável devido ao custo computacional.

Uma alternativa prática consiste em partir de arquiteturas já propostas na literatura e adaptá-las ao problema estudado. Seguindo essa estratégia, para o modelo recorrente puro foi adotada como referência a arquitetura apresentada em Narotamo et al. (2024). No trabalho original, os autores utilizam três camadas de GRUs, cada uma com 256 unidades ocultas, explorando a capacidade das redes recurrentes de modelar dependências temporais no sinal.

Neste projeto, a arquitetura base foi mantida, mas além de receber o sinal do eletrocardiograma, o modelo recebeu também os intervalos RR.

A Figura 15 apresenta um esquema da arquitetura utilizada.

Figura 15 – Arquitetura GRU pura.



Fonte: Elaborado pelo autor.

A segunda arquitetura avaliada é um modelo híbrido composto por camadas convolucionais seguidas de uma camada recorrente do tipo GRU. Nessa abordagem, o bloco convolucional é aplicado individualmente a cada batimento da sequência, gerando para cada um deles um mapa de features que, em seguida, compõe a nova sequência processada pelo bloco recorrente.

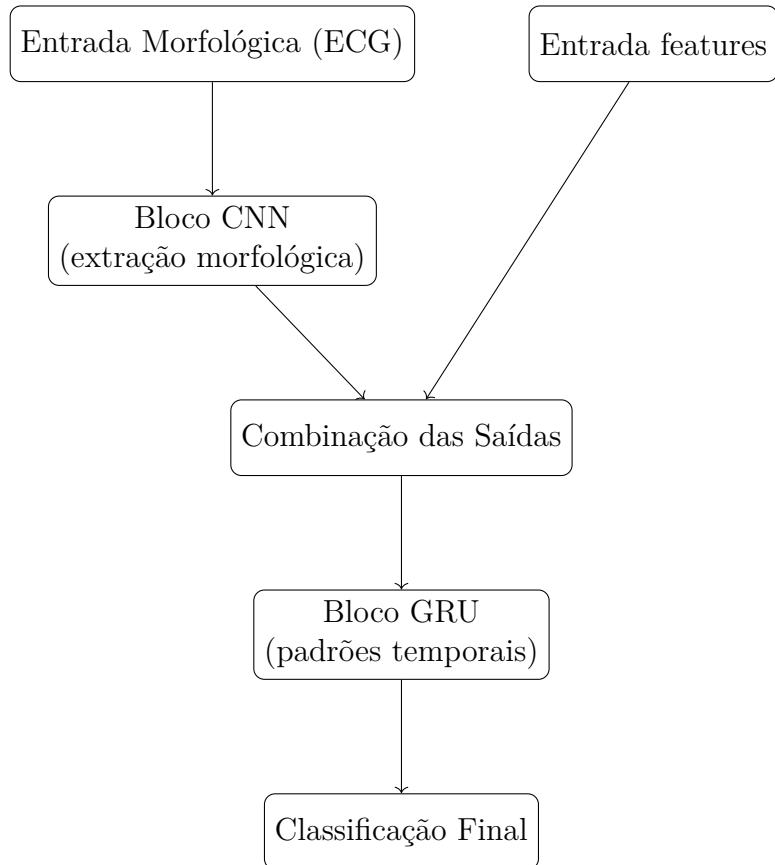
O bloco convolucional é formado por duas camadas de CNN: a primeira com 32 filtros e kernel de tamanho sete, e a segunda com 64 filtros e kernel de tamanho cinco, ambas utilizando padding adequado para preservar o comprimento da entrada. A escolha do tamanho do filtro, ou kernel, envolve a escolha entre padrões globais, com um kernel maior, ou padrões mais locais, com um kernel menor. Arquiteturas profundas costumam usar kernels pequenos. Como esta tem apenas duas camadas, foi utilizado um de tamanho sete, e outro de tamanho 5.

Cada camada convolucional é seguida por batch normalization — para estabilizar o treinamento — e por global max pooling, que reduz a dimensionalidade do mapa de features e contribui para mitigar sobreajuste ao reter apenas as ativações mais relevantes.

A etapa recorrente é composta por uma camada GRU com 256 unidades, responsável por modelar a dependência temporal entre os batimentos por meio das representações produzidas pelo bloco convolucional.

Essa arquitetura representa uma versão simplificada do modelo proposto por Narotamo et al. (2024).

Figura 16 – Arquitetura híbrida CNN e GRU.



Fonte: Elaborado pelo autor.

Enquanto que a rede da figura 15 recebeu o ECG concatenado com as *features*, a rede híbrida as recebeu separadas, sendo conectadas após o processamento das CNNs já que o objetivo era que esta extraísse *features* morfológicas.

### 3.3.1 Tamanho da sequência

Para otimizar o processo de treinamento, foram empregados os mecanismos de early stopping e reduce on plateau, responsáveis por limitar o número de épocas e ajustar dinamicamente a taxa de aprendizagem, respectivamente. Ambos monitoraram o f1-score, de forma que a rede buscasse um equilíbrio entre precisão e recall. Assim, mesmo após o treinamento, era possível ajustar manualmente esse compromisso com auxílio da curva PR.

Em ambos os modelos, utilizou-se uma sequência composta por 16 batimentos, sendo a classificação realizada apenas no último elemento da sequência. Dessa forma, o

problema caracteriza-se como uma tarefa de sequência para vetor, conforme discutido na Seção 2.5.

A escolha do tamanho da sequência foi feita empiricamente. Inicialmente, avaliou-se uma arquitetura simples — uma única camada de LSTM com 100 unidades — utilizando uma validação cruzada one hold out. Embora validações com mais folds reduzam o viés na estimativa de generalização, conforme James et al. (2023), o custo computacional cresce proporcionalmente, o que motivou a escolha desse arranjo mais enxuto para experimentação inicial.

Foram testadas sequências de 10, 16 e 20 batimentos. Houve melhora de desempenho ao passar de 10 para 16 batimentos, porém o aumento para 20 resultou em um consumo de memória excessivamente elevado, sem ganho proporcional. Por esse motivo, definiu-se o tamanho final da sequência como 16 batimentos.

As redes foram treinadas por até 50 épocas.

### 3.4 Estratégia de avaliação

Em seguida, é preciso escolher uma estratégia de avaliação. Conforme apresentado na seção 2.7, existem duas estratégias principais: a interpaciente e a intra-paciente. Na primeira, o modelo é exposto a um cenário mais realista, ele precisa classificar ECG inéditos. O desafio reside justamente na variabilidade dos tipos arrítmicos. Assim, Chazal, O'Dwyer e Reilly (2004) propôs dois conjuntos de dados para o MIT-BIH; o Ds1 que é formado por: 101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223 e 230 e o Ds2, formado por: 100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233 e 234.

Note que o conjunto Ds1 inclui 12 registros que são resultados da seleção aleatória e 10 dos registros com as morfologias complexas. Já no Ds2 possui oito dessa primeira seleção e 14 da segunda; sendo mais desafiador e feito para testar a capacidade de generalização do modelo.

Na tabela 3, é mostrado a distribuição das cinco classes nos dois conjuntos.

Tabela 3 – Distribuição das classes nos conjuntos Ds1 e Ds2

Conjunto	N	SVEB	VEB	F	Q	Total
DS1	45 866	944	3 788	415	8	51 021
DS2	44 259	1 837	3 221	388	7	49 712
Total	90 125	2 781	7 009	803	15	100 733

Fonte: o autor.

Como pode ser observado, as classes são desbalanceadas dentro do mesmo conjunto, porém, balanceada entre eles.

A avaliação foi feita utilizando uma avaliação cruzada no conjunto Ds1 inicialmente com duas partições (dois *folds*) e, posteriormente, com cinco partições (cinco *folds*) nos modelos finais, utilizando o particionados inter-paciente. Esta estratégia permite avaliar o modelo com ECGs de pacientes para o qual ele não foi treinado; sendo mais próximo de um contexto clínico e expõe o modelo a mais variação. Além disso, conforme a AAMI, batimentos com marcapasso foram excluídos por serem não representativos.

Na tabela 4, abaixo, é mostrado os pacientes em cada *fold*.

Tabela 4 – Distribuição dos IDs dos Pacientes por Fold para Conjuntos de Treinamento e Validação

Fold	Conjunto de Treinamento	Conjunto de Validação
1	101, 106, 108, 109, 112, 114, 116, 119, 122, 124, 203, 205, 207, 208, 209, 215, 220, 223	115, 118, 201, 230
2	101, 108, 109, 112, 114, 115, 116, 118, 119, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230	106, 122, 124
3	101, 106, 108, 109, 112, 114, 115, 116, 118, 122, 124, 201, 207, 208, 215, 220, 223, 230	119, 203, 205, 209
4	106, 112, 114, 115, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 230	101, 108, 109, 116, 220, 223
5	101, 106, 108, 109, 115, 116, 118, 119, 122, 124, 201, 203, 205, 209, 220, 223, 230	112, 114, 207, 208, 215

Fonte: o autor.

### 3.4.1 Métricas

As métricas utilizadas para avaliar o desempenho dos modelos foram: sensibilidade, precisão, acurácia, *F1-score*, AUC (*Area Under the Curve*) e AP (textit{Average Precision}). Esses dois últimos são exibidos juntos ao gráficos *ROC* e *PR*, respectivamente. Além da curva de calibração para avaliar o pior cenário.

A sensibilidade representa a capacidade do modelo em identificar corretamente as classes positivas, isto é, os batimentos arrítmicos. Sua equação é dada por:

$$\text{Sensibilidade} = \frac{TP}{TP + FN} \quad (3.3)$$

em que *TP* são os verdadeiros positivos e *FN* os falsos negativos.

A precisão, por sua vez, indica a proporção de batimentos classificados como arrítmicos que realmente pertencem a essa classe:

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (3.4)$$

onde  $FP$  representa os falsos positivos. Precisão e sensibilidade estão relacionadas por um *trade-off*. No contexto médico, prioriza-se elevada sensibilidade, ainda que à custa de menor precisão, uma vez que falsos negativos são mais prejudiciais que falsos positivos.

O *F1-score* é a média harmônica entre precisão e sensibilidade, buscando um equilíbrio entre ambas:

$$F1\text{-score} = \frac{2 \cdot \text{Precisão} \cdot \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \quad (3.5)$$

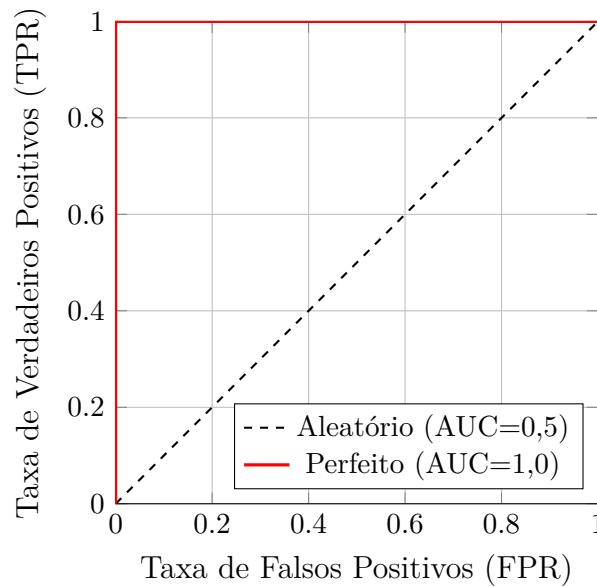
A acurácia corresponde ao acerto global do modelo, considerando tanto as classes positivas quanto as negativas:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.6)$$

A AUC mede a capacidade do modelo em separar as classes positivas das negativas, variando entre 0 e 1. Valores próximos de 1 indicam separação perfeita, enquanto 0,5 corresponde a um modelo com desempenho equivalente ao acaso, o *baseline*.

Essa métrica é calculada a partir da área sob a curva ROC. Na Figura 17, é ilustrado a curva ROC de um classificador perfeito.

Figura 17 – Curva ROC de um Classificador Perfeito: Comparação com Modelo Aleatório.



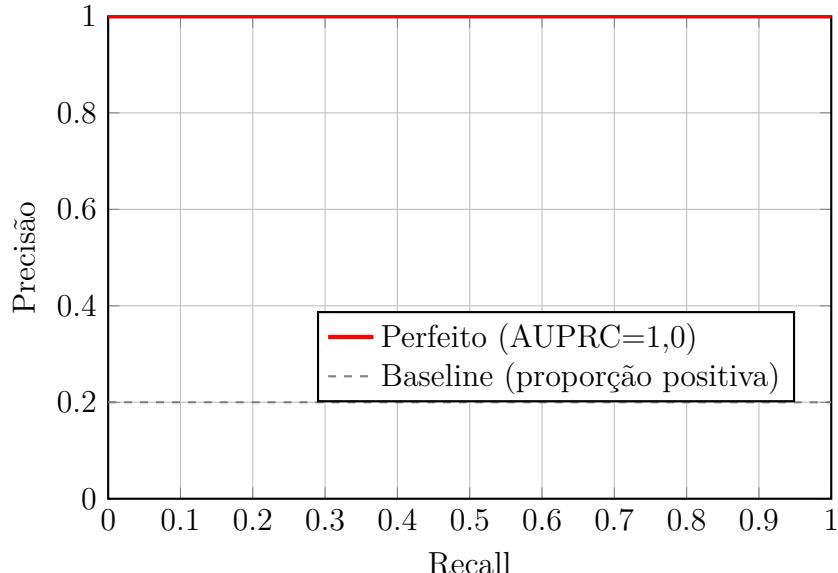
Fonte: Elaborado pelo autor.

A curva vermelha representa um classificador perfeito com TPR, sinônimo de *recall*, sempre igual a um. O raciocínio é que para aumentar o *recall*, a quantidade de classes negativas que são classificadas como positivas, calculada por FPR, aumenta. Entretanto, para um classificador perfeito, esse *trade-off* não existe.

A linha tracejada representa um classificador aleatório, o *baseline*. Neste caso, por exemplo, para achar 60% das classes positivas, cerca de 0,6 de *recall*, o modelo classificaria 60% das classes negativas como positivas.

Já a curva PR, Precisão vs *Recall* representa a precisão em função do *recall*. Na figura Figure 17, é ilustrada a curva PR de um classificador perfeito.

Figura 18 – Curva Precisão–Recall de um Classificador Perfeito: Comparação com Baseline.



Fonte: Elaborado pelo autor.

Como há uma relação de *trade-off* entre a precisão e o *recall*, conforme ajusta-se o limiar de decisão para aumentar o *recall*, a precisão tende a cair. Entretanto, assim como ocorre na ROC, esse *trade-off* não existe para um classificador perfeito; ou seja, a precisão é sempre 100% independente do valor do *recall*.

Já a linha tracejada, marca o desempenho de um classificador aleatório; o *baseline*. A linha corresponde a frequência da classe positiva, isto é, o classificador aleatório sempre tem uma precisão igual a frequência da classe positiva. AP é o análogo da AUC para esta curva.

Na tabela 5 é ilustrada a matriz de confusão.

Tabela 5 – Exemplo de matriz de confusão binária

Classe Verdadeira	Classe Preditiva	
	Positiva	Negativa
Positiva	TP	FN
Negativa	FP	TN

Fonte: Elaborado pelo autor.

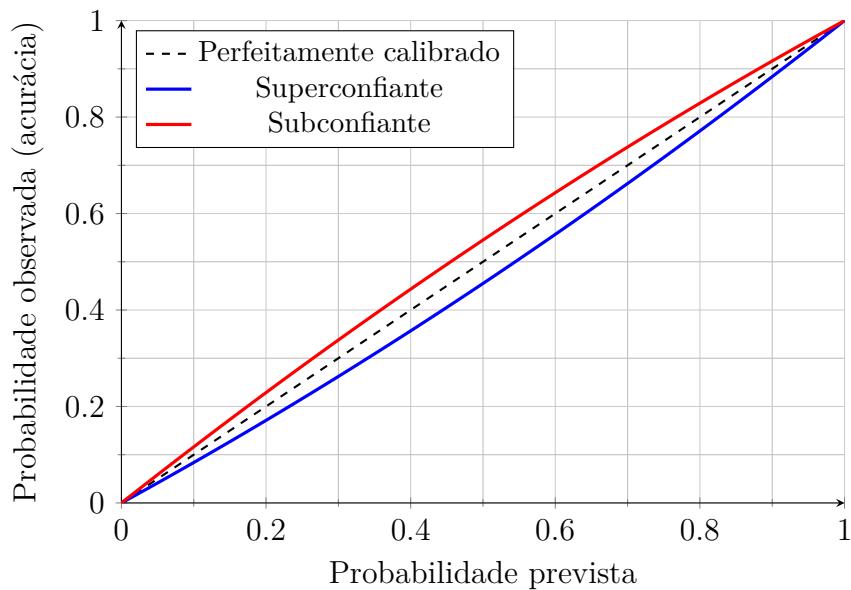
Em sentido anti-horário, a partir do canto superior esquerdo temos:

1. TP *true positive*, quantos casos positivos foram corretamente classificados;
2. FN *false negative*, quantos casos positivos foram incorretamente classificados;
3. FP *false positive*, quantos casos negativos foram incorretamente classificados;
4. TN *true negative*, quantos casos negativos foram incorretamente classificados.

Niculescu-Mizil e Caruana (2005) apresentam uma definição visual de um modelo perfeitamente calibrado. Em um diagrama de confiabilidade, o modelo é considerado bem calibrado quando, para cada bin de probabilidade, a média das probabilidades previstas corresponde à frequência observada da classe positiva nesse mesmo bin.

Na figura 19 é ilustrada um exemplo desse diagrama.

Figura 19 – Diagrama de calibração mostrando curvas de modelos perfeitamente calibrado, superconfiante e subconfiante.



Um modelo perfeitamente calibrado terá uma curva na diagonal. Logo, caso a probabilidade média prevista seja de 40%, então a ocorrência da classe positiva será de 40%. Um modelo superconfiante terá sua curva abaixo da diagonal, assim, nesse mesmo caso, a ocorrência da classe positiva seria abaixo de 40% e para um modelo subconfiante, a ocorrência da classe positiva é maior que a média da probabilidade prevista.

Os autores explicam que calibração é um aspecto aparte do desempenho, medido pelas demais métricas supracitadas. Muitos contextos, apenas ter um bom ROC, por exemplo, não é suficiente. Logo, é possível ter um alto desempenho, porém, ter uma calibração ruim.

Essas métricas mostram o desempenho do modelo em perspectivas diferentes, precisão, *recall*, *f1 score* e acurácia, mostram o desempenho do modelo para um determinado limiar. Neste trabalho, foi escolhido como 50%. Já as curvas PR e ROC mostram o

impacto no desempenho do modelo para diferentes limiares e a matriz de confusão permite visualizar os tipos de erros e acertos individualmente. A calibração, por outro lado, está relacionada com o quanto as probabilidades previstas refletem a ocorrência real dos casos positivos.

## 4 Resultados e discussões

Nesta seção, serão apresentados os resultados alcançados pelos modelos. Primeiro, a média das métricas com o objetivo de ter uma visão geral do desempenho. Em seguida, será feita uma comparação entre as médias na validação com as do treino, para identificar o *overfit*. Então, a partir das métricas individuais em cada *fold*, será identificado o pior e o melhor caso de cada modelo utilizando como critério o *f1-score*. Apesar do recall ser mais importante, foi adotado o *f1-score* como critério, pois o ajuste no *trade-off* pode ser feito manualmente e a critério de um tomador de decisão, usando a curva ROC ou *precisão-recall*. Em caso de empate será usado o *recall*.

Para estes casos, será analisada a matriz de confusão, onde poderá ser feito a análise dos tipos de erros e acertos e as curvas de *precisão-recall*, para averiguar o impacto da precisão e recall em diferentes *folds* e como os modelos se comparam com o *baseline*.

Então, será eleito o melhor modelo considerando o maior *f1-score* médio. Nesta etapa, é importante analisar também o desvio padrão; é desejável que o melhor modelo não tenha apenas um *f1-score* médio alto, mas um desvio padrão baixo, indicando maior estabilidade.

Por fim, será feita a comparação dos modelos com os apresentados na revisão da literatura.

### 4.1 Resultados do modelo GRU

Na Tabela 6 é mostrado o resultado médio do modelo GRU puro na validação junto com os respectivos desvio padrão.

Tabela 6 – Média das métricas do GRU para a classificação normal vs. ventricular na validação

Métrica	Média	Desvio Padrão
Precisão	0,8515	0,1825
Recall	0,8039	0,0795
F1-Score	0,8060	0,0760
Acurácia	0,9640	0,0278

Fonte: Elaborado pelo autor.

Os resultados indicam que o modelo achou aproximadamente 80% dos casos positivos, com um desvio padrão relativamente baixo, indicando boa estabilidade. Além disso, a precisão do modelo foi maior que seu *recall*, indicando um perfil mais conservador na classificação.

A seguir os resultados no treino:

Tabela 7 – Média das métricas do GRU para a classificação normal vs. ventricular no treino

Métrica	Média	Desvio Padrão
Precisão	0,9872	0,0121
<i>Recall</i>	0,9782	0,0150
F1-Score	0,9827	0,0134
Acurácia	0,9969	0,0024

Fonte: Elaborado pelo autor.

Comparando os resultados do treino na tabela 9 com os resultados da validação na tabela 8, observa-se um diferença substancial; evidenciando sobreajuste, isto é, o modelo apresentou uma baixa capacidade de generalização para pacientes não vistos.

Esse fenômeno ocorre pois modelos com alta flexibilidade, como redes neurais, conseguem se ajustar intimamente com os dados de treino. Caso eles não sejam representativos da população, tais modelos podem aprender ruído e particularidades dessa amostra ao invés de padrões generalizáveis.

No contexto do MIT-BIH, o desbalanceamento das classes pode ter causado isso. Como há poucos exemplos da classe positiva, é fácil para o modelo memorizar padrões morfológicos e rítmicos das arritmias do conjunto de treino, falhando ao encontrar variações dessas instâncias em pacientes diferentes. Conforme descrito nas seções 2.3.2 e também em 3.1, existe uma grande variação entre as arritmias, tanto de paciente para paciente quanto intra-paciente.

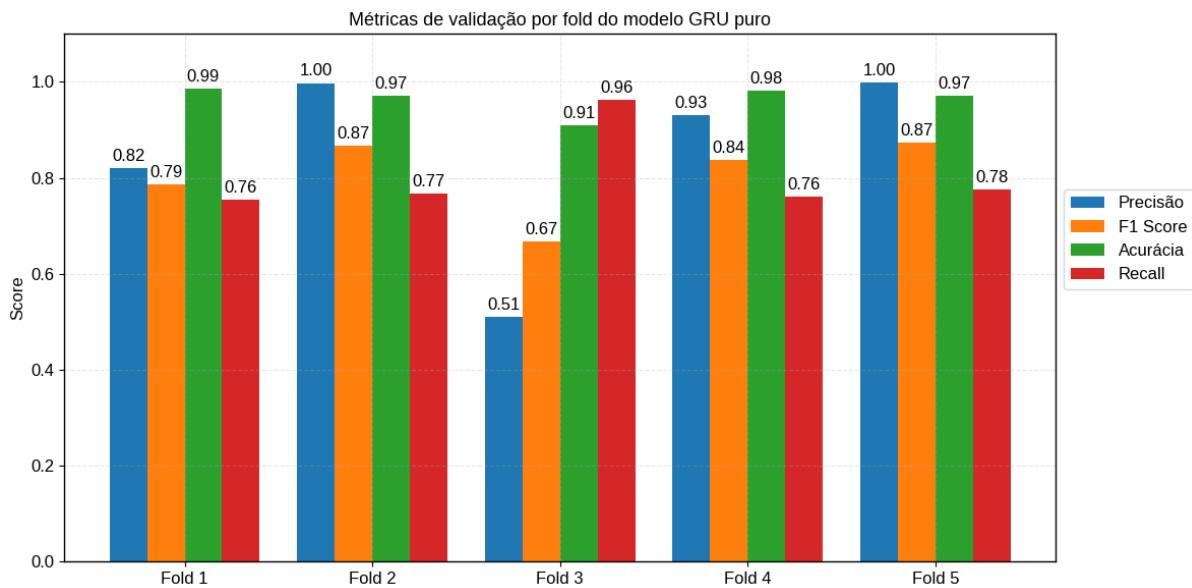
Outra evidência é a diferença entre a acurácia média do conjunto de treino em relação ao conjunto de validação. Observa-se uma diferença significantemente menor. Esta métrica é dominada pela classe negativa, indicando que o modelo conseguiu aprender padrões mais generalizáveis ao ser exposto a mais exemplos dessa classe.

Diferente da acurácia, as demais métricas são muito mais sensíveis ao desempenho na classe positiva.

O particionamento usado torna a tarefa de generalização mais desafiadora, pois o modelo é avaliado com ECGs não vistos durante o treino.

#### 4.1.0.1 Resultados no pior e melhor caso (Modelo GRU)

Na figura 20, está os resultados alcançado pelo modelo em cada *fold* na validação:

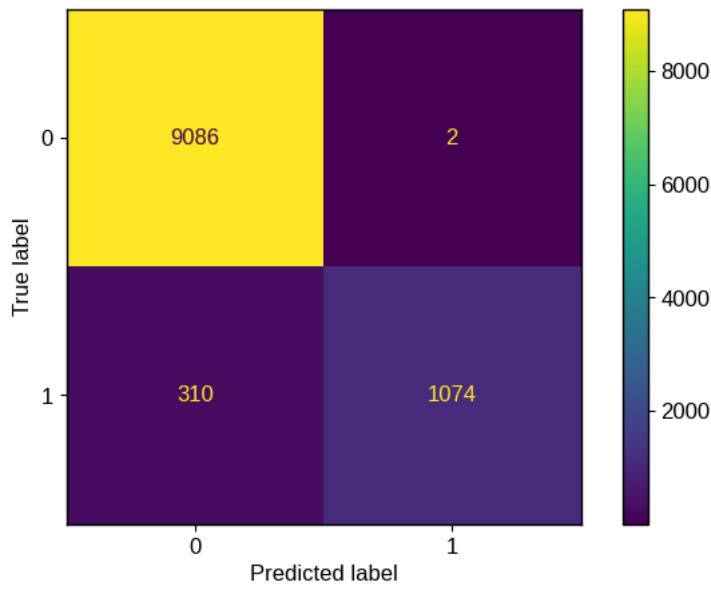
Figura 20 – Métricas do modelo GRU por *fold*

Fonte: Elaborado pelo autor.

No *fold* três, o modelo obteve sua menor precisão, aproximadamente 0,51 porém obteve um alto *recall*, aproximadamente 0,96. Essa discrepância sugere que neste *fold*, havia batimentos normais que fugiam do padrão aprendido no treino, fazendo com que o modelo confundisse eles com batimentos da classe ventricular. Nos demais *folds*, a precisão foi maior que o *recall*, sugerindo a presença de arritmias com características mais sutis, que fizeram com que o modelo as confundisse com batimentos normais.

Considerando o *f1-score*, o terceiro *fold* foi eleito o pior. Como o *fold* cinco empatou com o segundo por esse mesmo critério, como desempate, aquele com o maior *recall*, o quinto, foi considerado o melhor.

Na figura 21, está a matriz de confusão do modelo em seu melhor *fold*:

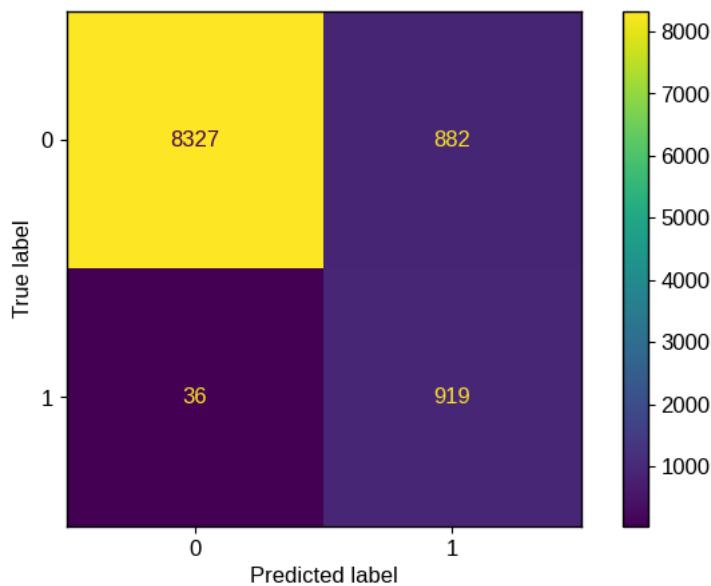
Figura 21 – Matriz de confusão do modelo GRU em seu melhor *fold*

Fonte: Elaborado pelo autor.

Na matriz, é possível ver o desbalanceamento das classes. Neste *fold*, o número de sequencias pertencentes a classe negativa é 9.088, enquanto que 1.384 pertencem a positiva; ou seja, aproximadamente, 13,21% de todas as sequencias são da classe positiva.

A maioria dos erros cometidos são de falsos negativos; o modelo classificou 310 sequencias arrítmicas como normais e apenas duas normais como arrítmicas. Algo que já era evidenciado no gráfico 20, pois sua precisão foi maior que seu *recall*.

Na figura 22, é dada a matriz de confusão no pior *fold*

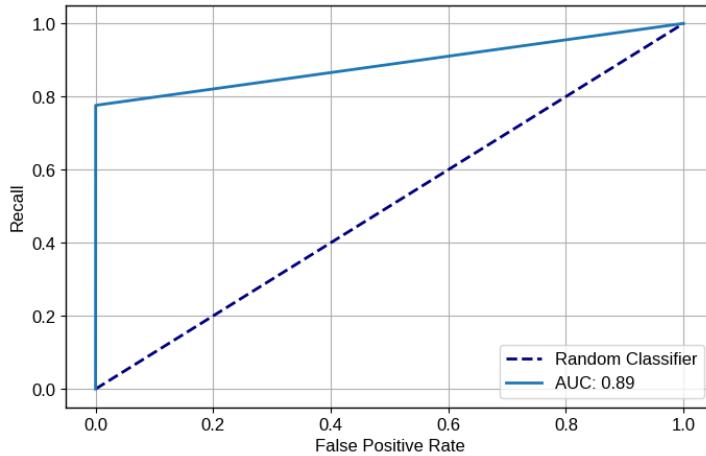
Figura 22 – Matriz de confusão do modelo GRU em seu pior *fold*

Fonte: Elaborado pelo autor.

Aqui o desbalanceamento foi mais severo; havia 9.209 classes negativas e 955 classes positivas; 9,39% aproximadamente. Neste *fold*, a situação se inverte: a maioria dos erros foram de falsos positivos, confirmando o que foi visto no gráfico 20.

A seguir, na figura 30, a curva ROC no melhor *fold*. Conforme discutido na seção 3.4.1, esta curva mostra os diferentes valores do *recall* e do FPR para todos os possíveis limiares de decisão.

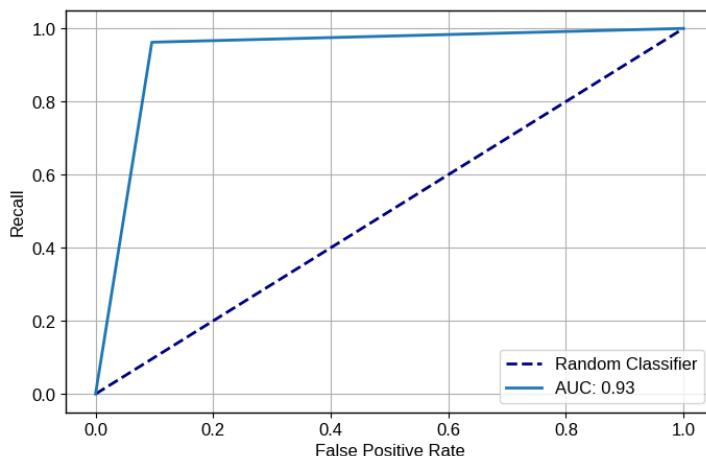
Figura 23 – Curva *ROC* modelo GRU em seu melhor *fold*



Fonte: Elaborado pelo autor.

Considerando que o *baseline*, um classificador aleatório, tem uma *AUC* de 0,5, o melhor foi significantemente melhor. O gráfico mostra ainda que o modelo consegue ter um *recall* de quase 80% ao custo de um FPR de 0%; ou seja, ele não comete erros de falso positivo. Para além desse valor, o modelo começaria a cometer tais erros, aumentando o FPR.

Figura 24 – Curva *ROC* do modelo GRU em seu pior *fold*

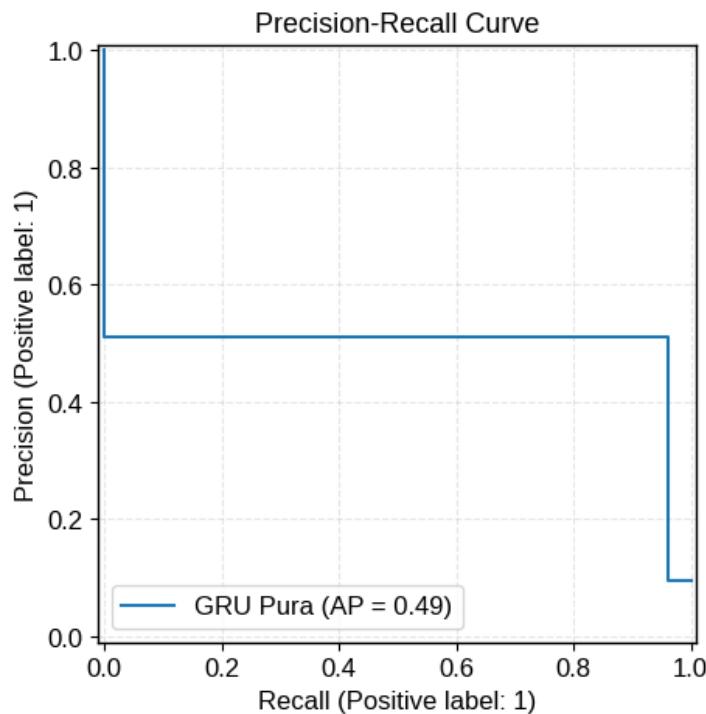


Fonte: Elaborado pelo autor.

No pior fold, 24, o modelo ainda conseguiu manter uma performance satisfatória, com um *AUC* de 0,93. Neste caso, o modelo tem um *recall* de quase 100% enquanto mantém um FPR de menos de 20%. Além desse valor, o *recall* cresce devagar conforme o erro aumenta, comportamento diferente do observado no pior caso.

Entretanto, devido ao desbalanceamento dos conjuntos, o desempenho pode ser melhor analisado com a curva PR. Conforme a seção 3.4.1, esse gráfico mostra os diferentes valores do *recall* e da precisão para todos os possíveis limiares de decisão.

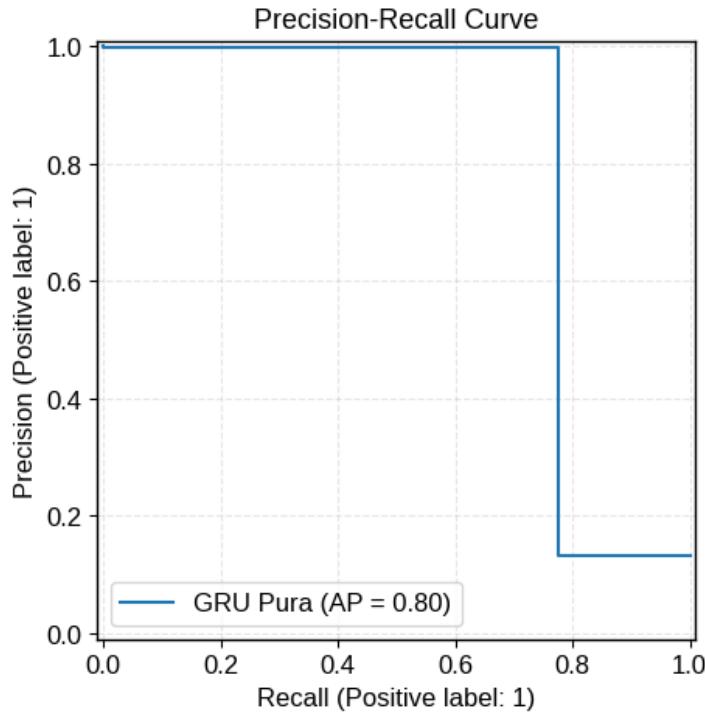
Figura 25 – Curva precisão vs *recall* do modelo GRU em seu pior *fold*



Fonte: Elaborado pelo autor.

Nesse gráfico, o *baseline* não é fixo, mas igual a prevalência da classe positivos. No pior *fold*, a proporção foi de aproximadamente 9,39%, contrastando com o 49% alcançado pelo modelo. Entretanto, a precisão foi baixa. Pelo gráfico, é possível ver que, por exemplo, seria possível ter um *recall* de 80% porém com uma precisão menor que 60%.

No melhor caso:

Figura 26 – Curva precisão vs *recall* do modelo GRU em seu melhor *fold*

Fonte: Elaborado pelo autor.

Nesse *fold*, o modelo alcançou um *AP* de 80%, enquanto que a proporção de casos positivos foi de 13,21%. No melhor caso, entretanto, o modelo para ter 80% de *recall*, teria que baixar sua precisão para menos de 20%.

Portanto, apesar do desbalanceamento, o modelo alcançou resultados satisfatórios, considerando o extremo desbalanceamento do conjunto.

## 4.2 Resultados do modelo híbrido GRU e CNN

O modelo híbrido apresentou resultado superior em relação ao modelo anterior. Na tabela 8 abaixo, é possível ver que o modelo obteve média maior em todas as métricas. Apesar disso, o modelo obteve um desvio padrão maior no *recall* e *F1-score*, mas a diferença foi de apenas 0.0106 e 0.0062.

Tabela 8 – Média das métricas do modelo híbrido CNN e GRU para a classificação normal vs. ventricular na validação

Métrica	Média	Desvio Padrão
Precisão	0,8800	0,1684
<i>Recall</i>	0,8726	0,0857
<i>F1-Score</i>	0,8593	0,0866
Acurácia	0,9730	0,0258

Fonte: Elaborado pelo autor.

Na tabela 9, é possível observar que ainda há *overfit* porém a diferença entre os resultados do treino e validação do modelo híbrido é menor quando comparado com o modelo GRU puro. Por exemplo, a diferença relative entre o *f1-score* de treino e validação para o modelo híbrido foi de aproximadamente 9,56% enquanto que para o modelo recorrente puro, foi de, aproximadamente, 17,98%.

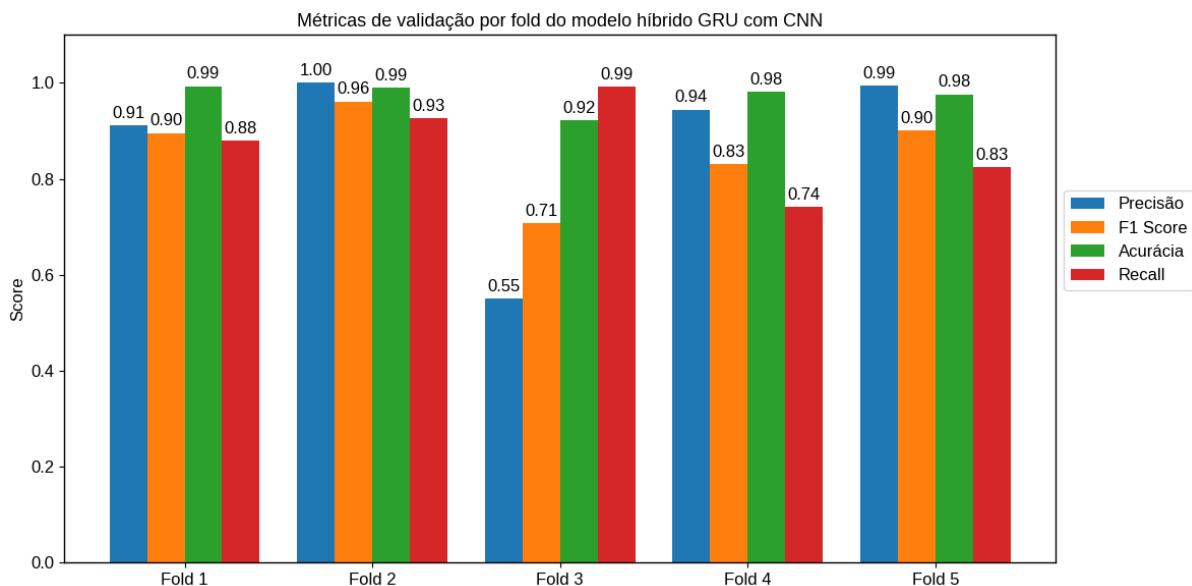
Tabela 9 – Média das métricas do modelo híbrido CNN e GRU para a classificação normal vs. ventricular no treino

Métrica	Média	Desvio Padrão
Precisão	0,9698	0,0180
Recall	0,9313	0,0268
F1-Score	0,9502	0,0222
Acurácia	0,9915	0,0035

Fonte: Elaborado pelo autor.

Na figura 27, estão os resultados obtidos pelo modelo híbrido em cada *fold*.

Figura 27 – Métricas do modelo híbrido CNN e GRU por *fold*



Fonte: Elaborado pelo autor.

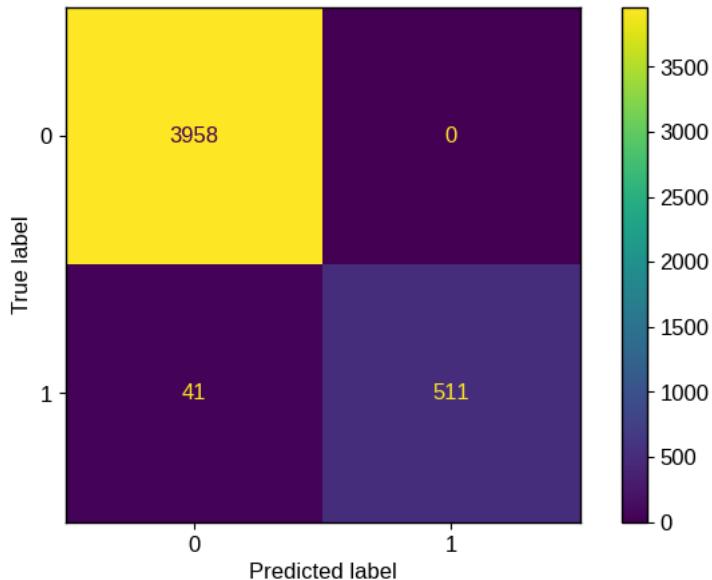
O modelo manteve a tendência de ter uma precisão acima da acurácia na maioria dos *folds*. É possível observar também que o modelo obteve um *recall* maior que o modelo GRU puro em todos os *folds* e uma precisão, no geral, maior ou igual. Sendo as exceções, os *folds* quatro e cinco, porém a diferença foi de, aproximadamente, 0,01 pontos percentuais.

A seguir, o desempenho do modelo em seu melhor e pior *fold*. Repetindo os critérios descritos na seção 4.1, o melhor *fold* foi o segundo e o pior foi o terceiro.

#### 4.2.0.1 Resultados no pior e melhor caso (Modelo Híbrido)

Na figura 28, é ilustrada a matriz de confusão do modelo em seu melhor *fold*. Neste cenário, o modelo não cometeu nenhum erro de falso positivo, mas 41 de falso positivo. Além de ter acertado 511 arritmias e 3.968 batimentos normais.

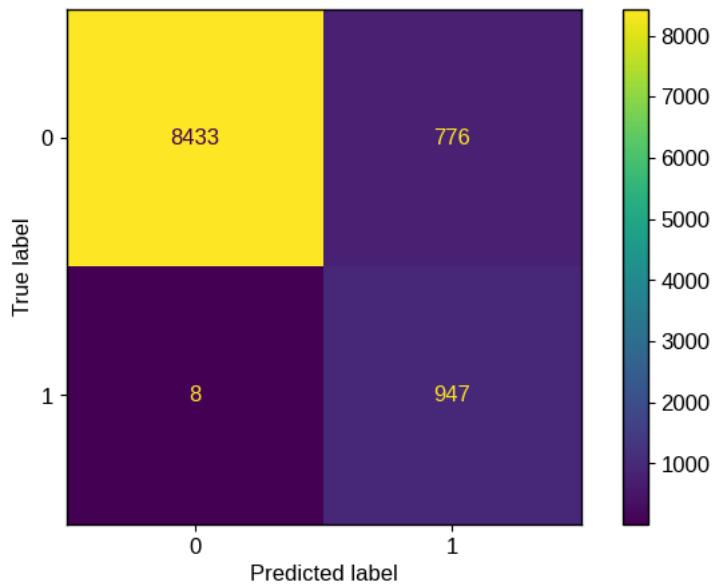
Figura 28 – Matriz de confusão modelo híbrido CNN e GRU em seu melhor *fold*



Fonte: Elaborado pelo autor.

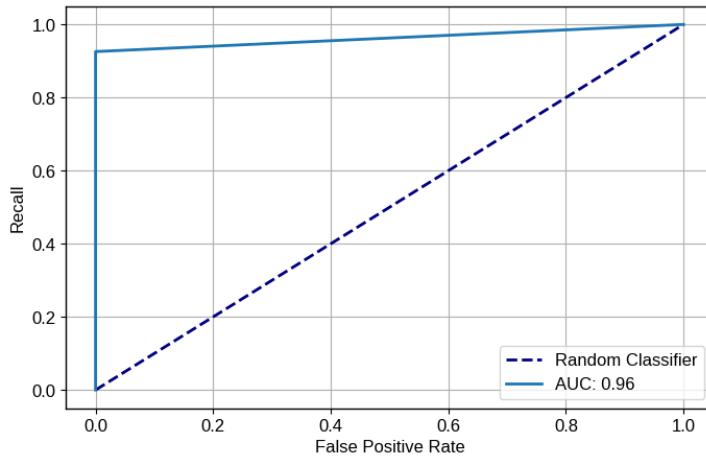
Como pode ser visto no gráfico; o modelo não cometeu nenhum erro de falso positivo e errou 41 arritmias, classificando-as como batimentos normais. Neste *fold*, a classe positiva compõe, aproximadamente. 12,23% do total. Assim, sendo levemente menos balanceado que o melhor *fold* do modelo GRU puro, que tinha 13,21% de classes positivas, e com um desempenho superior: com um *f1-score* de 0,97 contra 0,87.

Na matriz de confusão do pior *fold*, ilustrado na figura 29, novamente, a situação se inverte; a quantidade de erros de falsos negativos, oito, foi menor que as de falsos positivos, 776, refletindo em um *recall* maior que a precisão; como pode ser observado no gráfico 27.

Figura 29 – Matriz de confusão modelo híbrido CNN e GRU em seu pior *fold*

Fonte: Elaborado pelo autor.

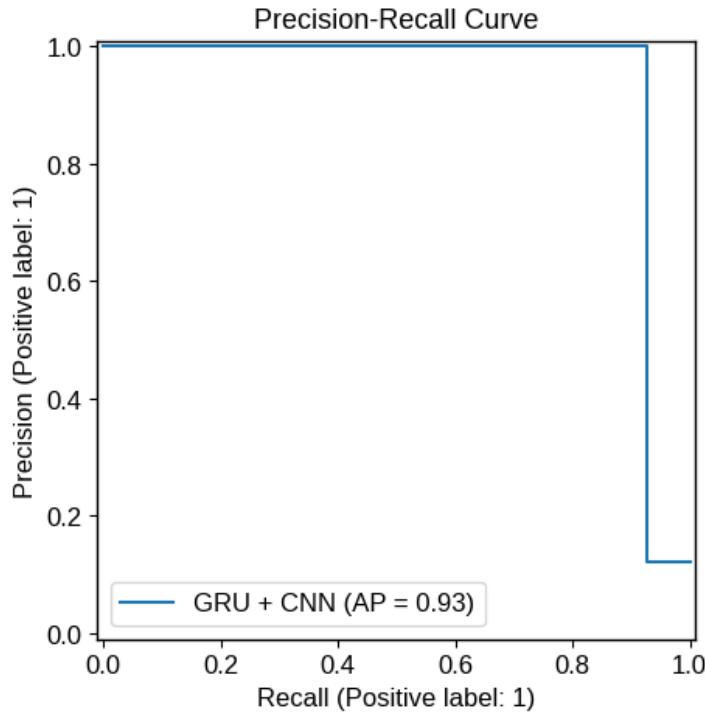
No melhor, o modelo híbrido obteve um AP de 0,96, como pode ser visto na figura 30:

Figura 30 – Curva *ROC* modelo híbrido CNN e GRU em seu melhor *fold*

Fonte: Elaborado pelo autor.

Que é maior que o AP do modelo GRU puro, como pode ser observado na figura 23. No seu melhor caso, o modelo híbrido consegue ter um *recall* maior que 80% sem cometer erros de falso positivo enquanto que o modelo puro teria conseguido apenas com um *recall* menor que 80%.

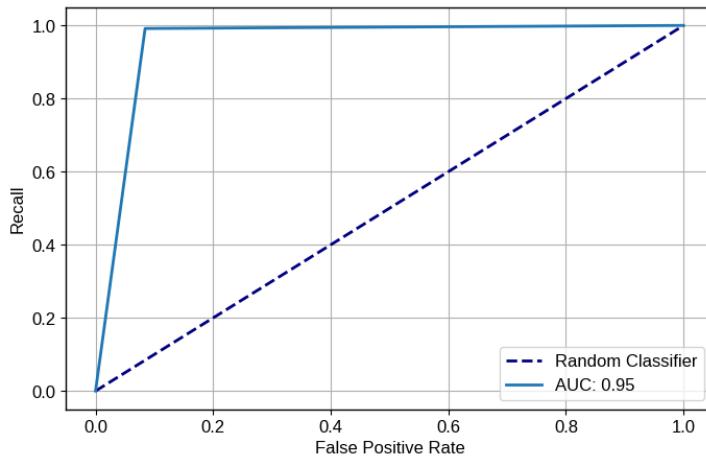
Considerando a curva PR, o AP também foi maior:

Figura 31 – Curva precisão vs *recall* do modelo híbrido CNN e GRU em seu melhor *fold*

Fonte: Elaborado pelo autor.

Pelo gráfico 31. O AP do modelo foi de 0,96 enquanto que o *baseline* seria de aproximadamente 12,23% e, além disso, o AP foi maior que o obtido pelo GRU puro. Entretanto, assim como ocorre com esse modelo, a precisão cai bastante quando o *recall* passa de um determinado valor. Entretanto, no modelo híbrido, isso ocorre quando o *recall* se aproxima de pouco mais de 90% enquanto que outro modelo, quando ele se aproxima de 80%.

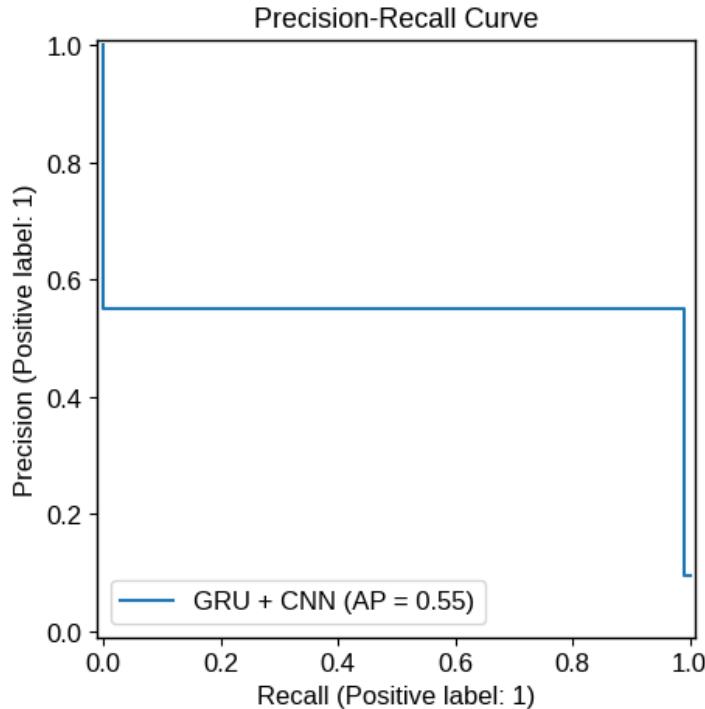
A curva *ROC* desse modelo no pior *fold* é descrita a seguir:

Figura 32 – Curva *ROC* modelo híbrido CNN e GRU em seu pior *fold*

Fonte: Elaborado pelo autor.

O AP foi de 0,95. Levemente maior que a do GRU puro que foi de 0,93. No pior caso, o modelo consegue ter um *recall* de quase 100% ao custo de um FPR de aproximadamente 20%.

Figura 33 – Curva precisão vs *recall* do modelo híbrido CNN e GRU em seu melhor *fold*



Fonte: Elaborado pelo autor.

Nesse cenário, o modelo conseguia manter um *recall* de até 80% com um pouco menos de 60% de precisão. O AP foi de 55% enquanto que o do GRU puro foi e 49% e o *baseline* foi de 9,39%.

De modo geral, os modelos exibiram um perfil semelhante em seu pior e melhor caso. No pior, a sensibilidade foi maior, resultados em maiores erros de falso positivo, como resultado, o *recall* foi alto e a precisão foi baixa. No melhor caso, houve um equilíbrio maior e, apesar do *recall* mais baixo, a alta precisão aumentou o *f1-score*.

Esse ganho de desempenho sugere que o uso da CNN para extração de *features* morfológicas pode ter sido uma vantagem. A camada recorrente do modelo híbrido não precisou aprendê-las do sinal, apenas precisou se concentrar em como elas se encaixam no contexto da sequência.

#### 4.2.1 Conclusão dos resultados

Nesta seção, foi apresentado os resultados obtidos pelos modelos GRU puro e híbrido. Ambos os modelos apresentaram um caso de *overfit*, evidenciado pela grande diferença entre o desempenho no treino com o da validação. A partir das métricas

individuais em cada *fold*, foi identificado o pior e melhor caso. Neste primeiro, ambos os modelos tiveram uma precisão bem menor que seu *recall*, invertendo a tendência apresentada nos outros *folds*. O que pode ter sido devido a batimentos normais com características distoantes das apresentadas no treino, fazendo com que os modelos os confundissem com batimentos arrítmicos.

Em contextos médicos, é preferível um *recall* maior, pois falsos negativos são mais danosos que um falso positivo; isto é, é melhor dizer que um batimento normal é arrítmico do que o contrário. Entretanto, uma precisão muito baixa pode indicar que o modelo é tão bom quanto um modelo aleatório; o que o tornaria inútil.

Entretanto, os APs dos modelos no pior caso evidenciou que eles superaram esse *baseline*, indicando que foi aprendido padrões que lhes permitem classificar as arritmias, entretanto, esses padrões ainda não são perfeitamente generalizáveis.

Por fim, na tabela a seguir é resumido os resultados alcançados pelos modelos e os apresentados na seção 2.7. Devido as diferenças metodológicas, a comparação direta entre os resultados não é possível. Além do mais, devido ao sobreajuste observado em ambos os modelos, optou-se por não realizar o teste final no Ds2, pois este é um teste único feito quando se tem um modelo com um desempenho satisfatório.

Tabela 10 – Resumo dos desempenho dos modelos para a classificação de arritmia (ventricular)

Trabalho (Ano)	Dataset	Sensibilidade (%)	Precisão (%)
de Chazal et al. (2004)	DS2.	77,7.	81,9.
Mousavi et al. (2019)	DS2.	98,98.	97,40.
Li et al. (2019)	DS2.	92,5.	97,40.
Saadatnejad et al. (2020)	Intrapaciente.	98,22.	92,97.
Kiranyaz et al. (2016)	Intrapaciente.	95,0.	89,5.
TCC (GRU)	Cross-validação 5 folds (ds1).	80,39.	85,15.
TCC (Híbrido)	Cross-validação 5 folds (ds1).	87,26.	88,0.

## 5 Análise de erros no pior *fold*

Nesta seção, será apresentada uma breve análise de erros do modelo híbrido em seu pior *fold*. Conforme os critérios adotados, o eleito foi o terceiro.

Como o recall foi superior à precisão, foi suposto que a causa esteja relacionada à presença de batimentos normais com características morfológicas atípicas, o que pode ter confundido os modelos. Assim o objetivo dessa seção é fazer uma breve exploração desse *fold* e verificar como os erros estão distribuídos em relação aos pacientes, e quais características os dados desses pacientes têm; como a morfologia do ECG e o intervalo RR. Por fim, será apresentado também a curva de calibração do modelo.

Devido ao fato de redes neurais serem modelos caixa-preta, não é possível afirmar quais características contribuíram para os erros, logo, a análise serve apenas para construir uma intuição inicial para os resultados achados.

### 5.1 Análise de erros do modelo híbrido CNN com GRU

Na tabela 11, a seguir, é possível ver que a maioria dos erros foi oriunda de um paciente, o 203.

Tabela 11 – Total dos erros e acertos por paciente no *fold* de validação

Pacientes	Erros	Acertos
119	0	1972
203	772	2186
205	11	2616
209	1	2606

Fonte: Elaborado pelo autor.

Aproximadamente, 98,46% de todos os erros foram desse paciente. Conforme visto na figura 29, a maioria desses erros são de falsos positivos. Segundo as anotações do MIT-BIH, disponíveis em (MOODY, 1997), o paciente 203 é considerado como muito difícil. As anotações ainda citam a presença de mudança de morfologia no complexo QRS e contrações ventriculares prematuras (PVC) de múltiplas formas.

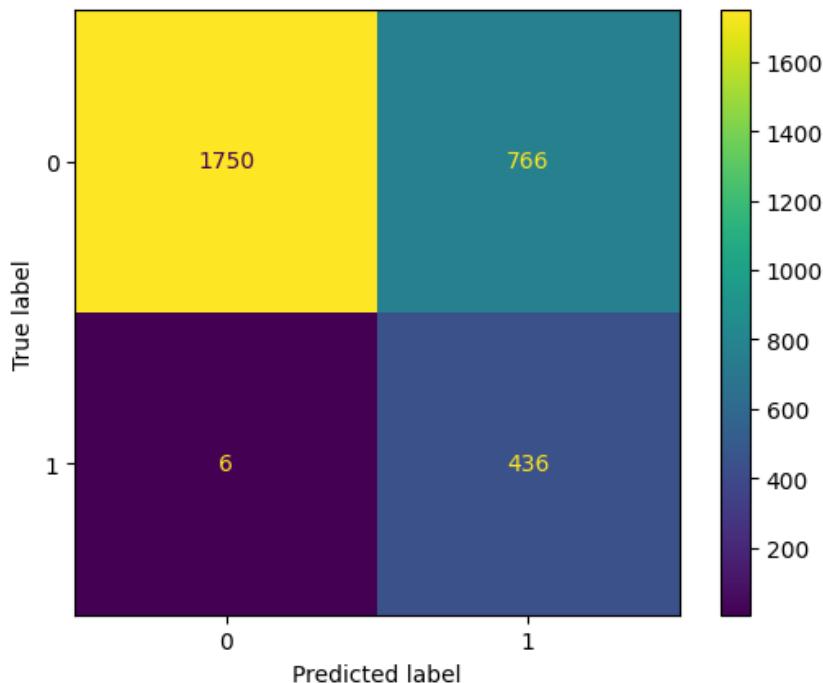
Já no paciente 209, o modelo acertou a única classe positiva que existia. O único erro cometido foi um falso positivo. No paciente 205, o modelo acertou 69 das 71 classes positivas e errou 9 classes negativas, das 2.556. Nesses dois pacientes, a classe positiva era extremamente rara, mas em números absolutos, a maioria dos erros foram de falsos positivos.

Como o paciente 203 dominou os erros neste *fold* e para o paciente 119, não houve erros, na próxima seção, a análise se concentrar neste dois casos.

#### 5.1.0.1 Comparação morfológica entre o paciente 203 e 119

Na figura 34, é mostrada a matriz de confusão desse paciente.

Figura 34 – Matriz de confusão do paciente 203

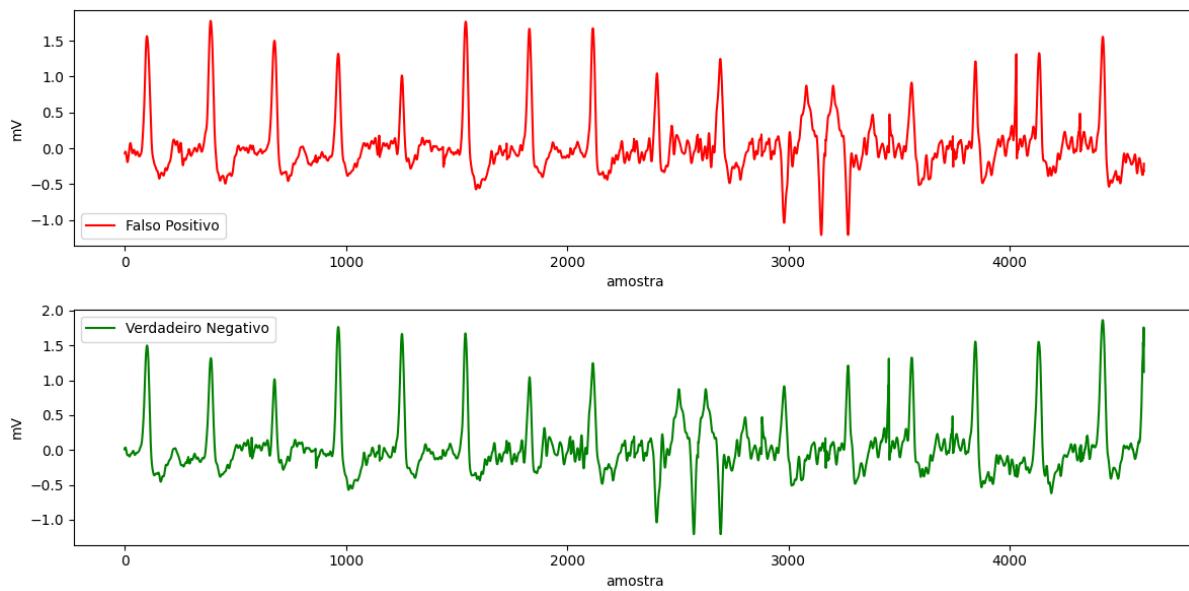


Fonte: Elaborado pelo autor.

Das 2.516 sequências normais, o modelo classificou 766 como arrítmicas e acertou 1.750 e das 472 sequências arrítmicas, o modelo acertou 436 e errou 6. Assim, ao todo, o modelo errou 35,31% dos batimentos do paciente 209;

Na figura 35, é ilustrado duas sequencias desse paciente, na primeira um falso positivo e na segunda um verdadeiro negativo.

Figura 35 – ECG normal do paciente 203



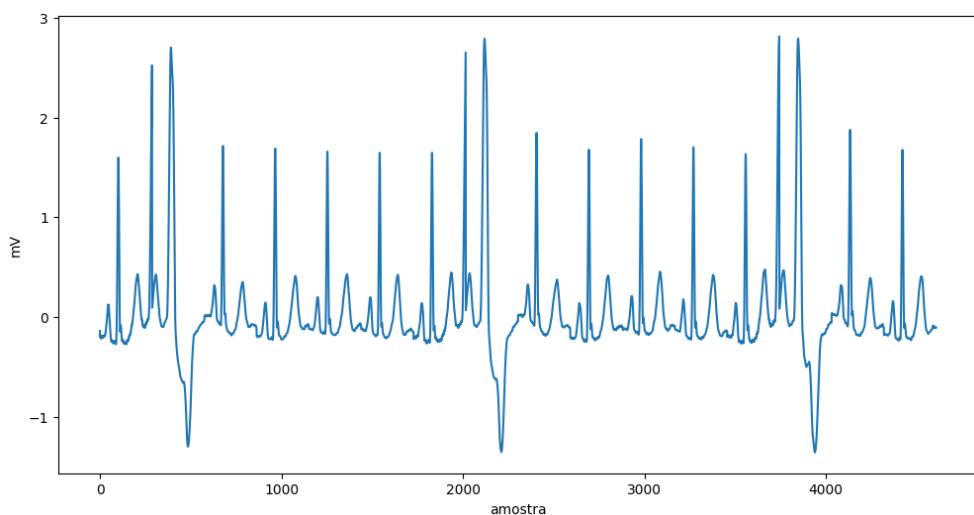
Fonte: Elaborado pelo autor.

É possível observar a forte presença de ruído em ambos os casos. E a presença de batimentos com a morfologia bem deformada; após a amostra 2000 no primeiro gráfico e após a amostra 1000 no segundo.

O modelo tinha 97% de confiança que o primeiro ECG era da classe positiva, logo foi um erro com muita confiança. No segundo caso, o modelo tinha 12% de confiança que a sequência pertencia a classe positiva, ou seja, 88% de chance de ser da classe negativa; um acerto com confiança.

Na figura 36, é ilustrada a sequência normal do paciente mais fácil;

Figura 36 – ECG normal do paciente 119.



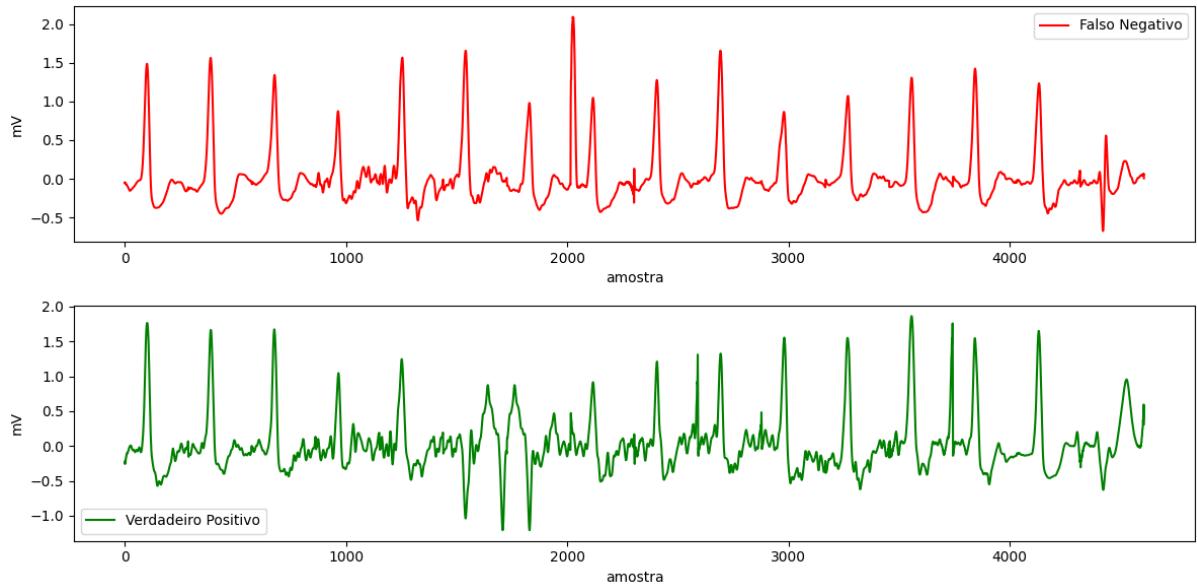
Fonte: Elaborado pelo autor.

É possível notar uma sequencia mais limpa e com o complexo QRS com morfologia usual. Note em torno da amostra 2000 uma contração prematura ventricular.

Na figura 37 é ilustrado duas sequências arrítmicas do paciente 203, a primeiro o modelo errou e a segunda ele acertou:

Em ambos os casos, é observável o ruído presente na figura 35. O último batimento da sequência também apresenta uma morfologia diferente da usual.

Figura 37 – ECG arrítmico do paciente 203: acerto e erro



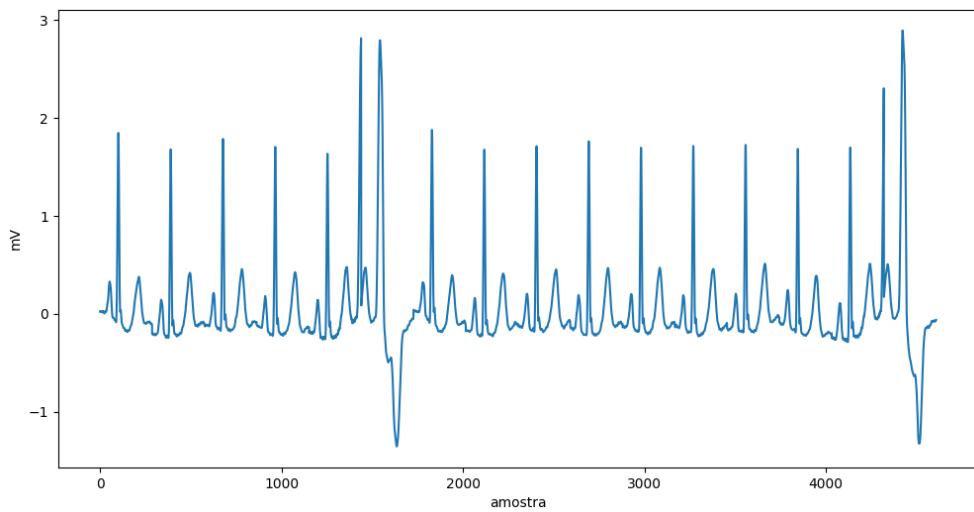
Fonte: Elaborado pelo autor.

O modelo tinha 16% de confiança que era um caso da classe positiva, logo, 84% que era da classe negativa; um erro confiante. É possível notar uma morfologia relativamente mais limpa e uniforme, principalmente quando comparada às demais sequências apresentadas deste paciente, levando o modelo a confundi-la com uma sequência normal.

Já no segundo caso, o modelo tinha 97% de confiança que era um caso da classe positiva; sendo um acerto com confiança. Nota-se a presença de arritmias ventriculares e um sinal com mais ruído.

Na figura 38, é ilustrada uma sequencia arrítmica do paciente 119. Observe no último batimento, uma arritmia ventricular.

Figura 38 – ECG arrítmico do paciente 119.



Fonte: Elaborado pelo autor.

#### 5.1.0.2 Comparação temporal entre o paciente 203 e 119

Na figura 39a, é ilustrado o gráfico de dispersão de um falso positivo do paciente 209. Note o último batimento da sequência (o 16º), em relação aos demais, ele ocorreu de forma mais prematura uma vez que se encontra na região inferior esquerda do gráfico, em comparação a outros batimentos — como o quarto, o oitavo e o décimo. Além disso, o intervalo entre ele e seu sucessor é menor do que o intervalo em relação ao batimento anterior, o que poderia sugerir um padrão compatível com um batimento ventricular prematuro (PVC). No entanto, essa interpretação dependeria da frequência cardíaca do paciente, e como discutido na seção 2.3.2, não existe um padrão universal de ECG normal.

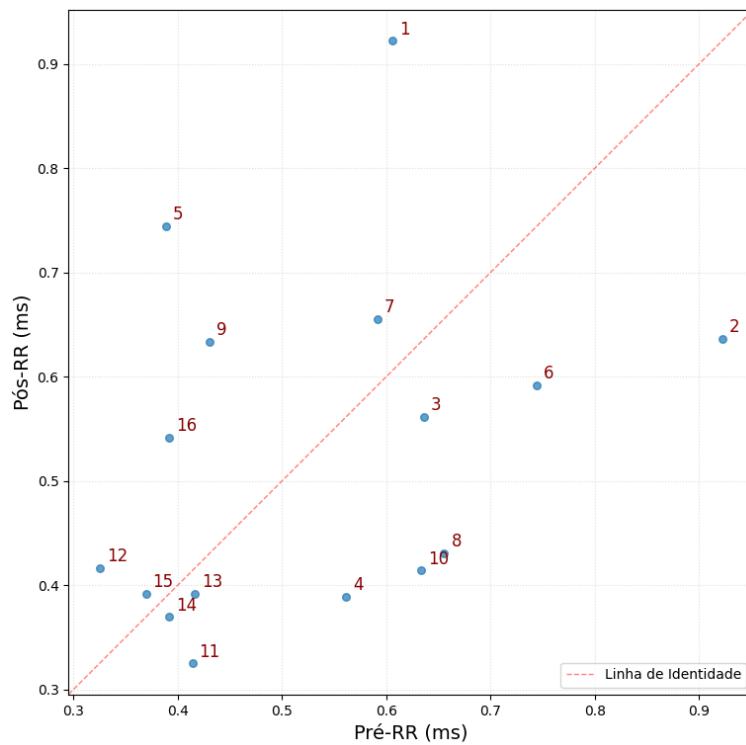
A figura 39b mostra um caso de verdadeiro positivo do mesmo paciente. O último batimento também ocorre de forma antecipada, porém sua posição mais próxima à linha de identidade indica que o intervalo com o sucessor é aproximadamente igual ao intervalo com o anterior.

Já a figura 40a apresenta um verdadeiro negativo. Embora os pontos estejam igualmente dispersos, o último batimento aparece mais deslocado para o canto inferior direito quando comparado aos dois casos anteriores. Essa diferença pode ter contribuído para que o modelo confundisse o caso negativo anterior como um batimento prematuro, resultando em um falso positivo.

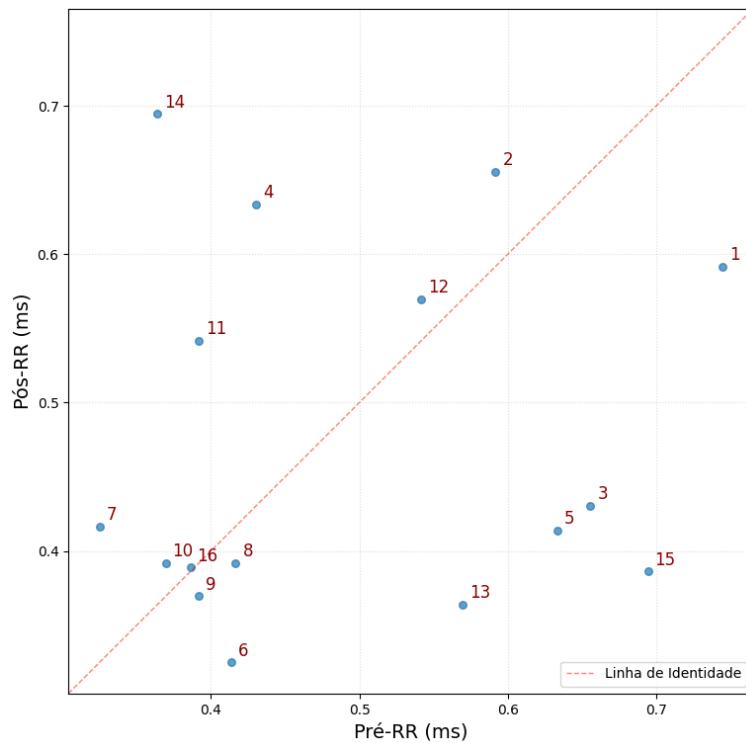
Por fim, a figura 40b mostra uma sequência normal do paciente 119. Diferentemente das anteriores, os batimentos formam agrupamentos mais concentrados, com o último batimento situado próximo ao centro e à linha de identidade.

Para visualizar o quanto confiante o modelo foi em seus erros, foi feito a curva de calibração, na figura 41, do modelo para o paciente 203 e o paciente 119.

Figura 39 – Scatter plot do paciente 203 de um falso positivo e um verdadeiro positivo.



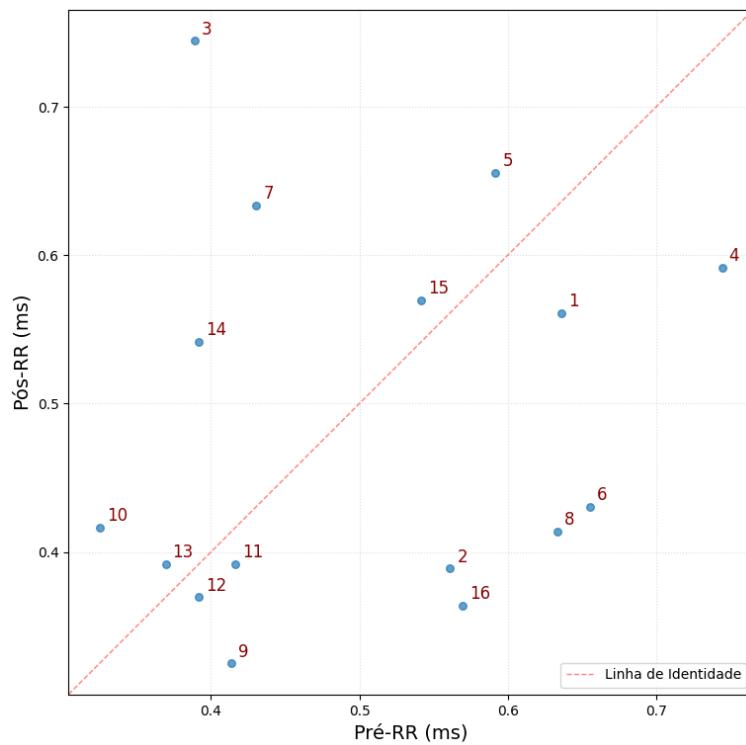
(a) Falso positivo do paciente 203



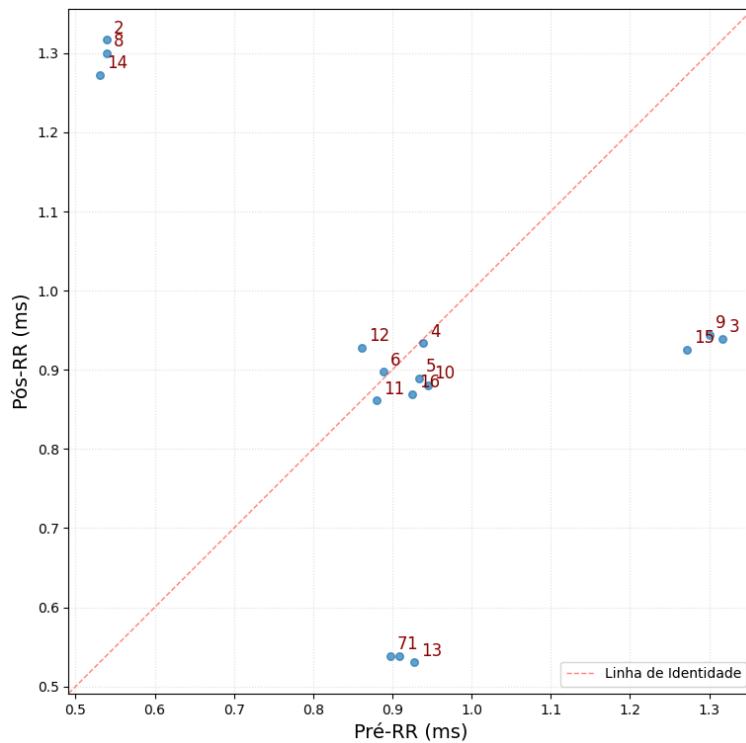
(b) Verdadeiro positivo do paciente 203

Fonte: Elaborado pelo autor.

Figura 40 – Scatter plot da sequência normal do paciente 203 e 119



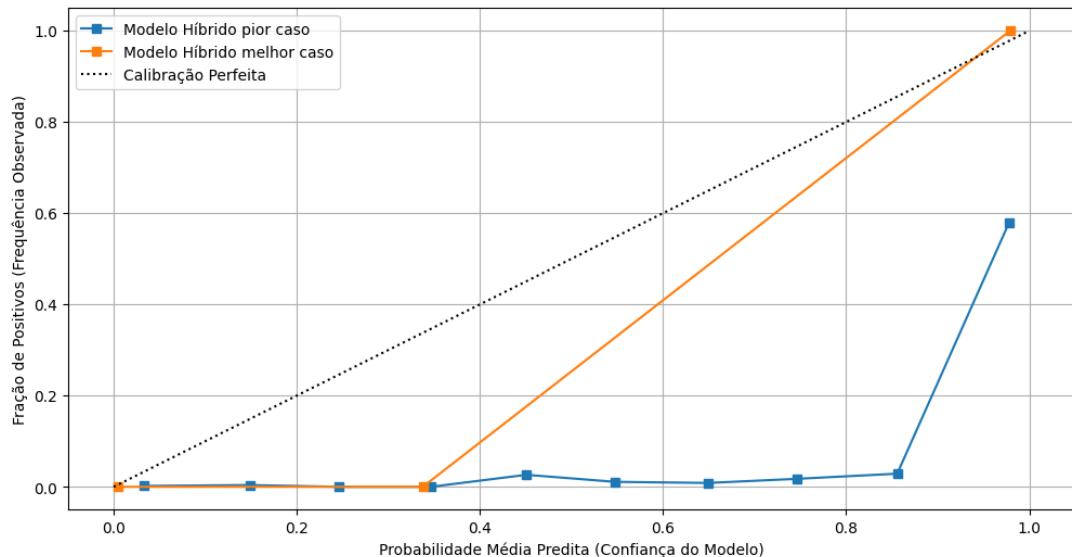
(a) Verdadeiro negativo do paciente 203



(b) Verdadeiro negativo do paciente 119

Fonte: Elaborado pelo autor.

Figura 41 – Curva de calibração para o paciente 203 e 119.



Fonte: Elaborado pelo autor.

A análise de calibração indica que o modelo está superconfiante em ambos os pacientes. Este cenário é caracterizado pelas curvas de calibração que se mantêm abaixo da diagonal principal.

No caso do paciente 119, a falta de calibração é menos severa, pois sua curva se aproxima mais da diagonal principal. Já para o paciente 203, a superconfiança é mais intensa, especialmente em *bins* abaixo de 80%. Um exemplo é quando o modelo prevê que 90% de um grupo será arrítmico, a realidade é que menos de 60% deles o são.

Conforme discutido na seção 3.4.1, a calibração não está diretamente relacionada ao desempenho, tanto que em ambos os pacientes, a rede foi superconfiante. Seria possível que o modelo tivesse um desempenho baixo e fosse bem calibrado; neste cenário, o tomador de decisão poderia confiar que as probabilidades previstas seriam realistas, mesmo que o modelo não discernisse tão bem.

#### 5.1.0.3 Conclusão da análise de erros

O objetivo desta seção foi analisar as sequências de batimentos do modelo híbrido em seu pior *fold* para ganhar uma intuição adicional às razões da falha do mesmo. Foi identificado que a grande maioria dos erros veio de um único paciente, o 203, que as anotações do MIT-BIH o classifica como muito difícil, devido a presença de ruído e arritmias atípicas.

Comparando o paciente 203 com o 119 — paciente para o qual o modelo não cometeu erros — foi identificado que esse possui um sinal mais limpo e sem grandes morfologias atípicas. Além disso, no paciente 203, num caso de falso positivo, o intervalo do último batimento da em relação ao seu antecessor é mais curto, o que poderia fazer

com que o modelo o confundisse com uma arritmia.

Além disso, a curva de calibração mostrou um cenário de superconfiança para ambos os pacientes, reforçando a ideia de que ela não tem relação direta com o desempenho.

Conforme visto na seção 4, este *fold* também foi o pior caso do modelo GRU e o mesmo ainda se saiu pior que o híbrido, com menor *recall* e precisão. O que pode ser outra evidência da vantagem da CNN. Como a CNN atua como extrator de *features* da morfologia, ela pode ter mitigado o impacto do ruído e da forma menos usual do ECG deste paciente, resultando em um desempenho melhor. Além disso, conforme visto, é possível que os intervalos RR tenha contribuído menos para a separação. Como ambos os modelos receberam as mesmas *features* RR, então a vantagem da extração feita pela CNN, pode ter ajudado o híbrido a se destacar.

# 6 Conclusão

O presente trabalho buscou compreender o processo de aplicação de redes neurais na classificação de arritmias ventriculares. Para isso, foram avaliados dois modelos — um baseado apenas em GRUs e outro híbrido, combinando CNN com GRUs — de modo a observar como diferentes arquiteturas lidam com a extração de padrões locais e dependências temporais do ECG.

A análise das métricas evidenciou que o modelo híbrido apresentou melhor desempenho, embora sujeito a certo grau de overfit. Para aprofundar a compreensão sobre esse resultado, foi realizada uma investigação do pior caso do modelo, revelando que a falha ocorreu em um paciente com sinais atípicos, porém clinicamente plausíveis. Além disso, identificou-se o problema de superconfiança, um desafio importante para aplicações em cenários críticos, embora não exclusivo deste trabalho.

De modo geral, o objetivo de explorar e compreender o processo de aplicação de modelos de deep learning para o ECG foi alcançado, permitindo identificar limitações práticas, desafios do domínio e potenciais caminhos de melhoria.

Durante a análise foi identificado outros problemas como superconfiança da rede que é um empecilho para uma adoção em cenário real, embora tais problemas não sejam restritos a esse modelo.

## 6.1 Trabalhos futuros

Como trabalhos futuros, sugere-se aprofundar a análise do pior cenário, incluindo a avaliação dos demais sinais disponíveis e o uso de métodos de explicabilidade. Existem métodos *post-hoc* como o LIME (RIBEIRO; SINGH; GUESTRIN, 2016); que fornece explicabilidade local e que poderia auxiliar a entender quais *features* mais contribuíram para o erro do modelo. Outro método, o SHAP (LUNDBERG; LEE, 2017) fornece explicabilidade global. Além de auxiliar na etapa de debug, tais métodos podem contribuir para a confiança de tomadores de decisão nos modelos; pois os mesmos poderiam entender o "raciocínio" dos modelos.

Para a calibração, Niculescu-Mizil e Caruana (2005) apontou métodos como a calibração de Platt, regressão isotônica que são apropriados para a classificação binária.

A etapa de pré-processamento poderia ser reavaliada, para poder remover os ruídos observados no pior caso. Outra alternativa é diminuir o tamanho da sequência e amostras de um batimento. Por mais que uma sequência maior, por exemplo, possa fornecer mais informação aos modelos, ela pode acabar incluindo mais ruído também. O mesmo para a quantidade de amostras.

Considerando que a abordagem híbrida obteve os melhores resultados, recomenda-se investigar novas arquiteturas, especialmente variantes bidirecionais das redes recorrentes. Outro ponto importante é o tratamento do desbalanceamento, que pode ser abordado por meio de geração de dados sintéticos ou estratégias baseadas em redes concorrentes.

Após a superação dessas limitações e a obtenção de resultados mais robustos, propõe-se realizar a avaliação final no conjunto Ds2 e, posteriormente, aplicar técnicas de calibração para produzir probabilidades confiáveis, bem como métodos adicionais de explicabilidade. Essas etapas visam não apenas melhorar o desempenho do modelo, mas também aumentar sua confiabilidade para aplicação em um cenário clínico e assim, contribuir para o uso de inteligência artificial em uma área tão crítica de maneira segura e responsável.

# Referências

- Agência Brasil. *Doenças cardiovasculares matam 400 mil brasileiros por ano.* 2024. <<https://agenciabrasil.ebc.com.br/saude/noticia/2024-09/doencas-cardiovasculares-matam-400-mil-brasileiros-por-ano>>. Acessado em: 15 de novembro de 2025.
- CASCINO, T.; SHEA, M. J. *Eletrocardiograma (ECG)*. 2025. MSD Manual de Medicina. Revisado por Jonathan G. Howlett. Disponível em: <<https://www.msdmanuals.com/pt/casa/dist%C3%BArbios-do-cora%C3%A7%C3%A3o-e-dos-vasos-sangu%C3%ADneos/diagn%C3%B3stico-de-dist%C3%BArbios-do-cora%C3%A7%C3%A3o-e-dos-vasos-sangu%C3%ADneos/eletrocardiograma>>.
- CHAZAL, P. de; O'DWYER, M.; REILLY, R. Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 7, p. 1196–1206, 2004.
- GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 3. ed. Sebastopol, CA: O'Reilly Media, 2022. ISBN 9781098125974.
- JAMES, G. et al. *An Introduction to Statistical Learning: with applications in Python*. Cham: Springer, 2023. Disponível em: <<https://www.statlearning.com/>>. Acesso em: 28 ago. 2025.
- KIRANYAZ, S.; INCE, T.; GABBOUJ, M. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, IEEE, Piscataway, NJ, EUA, v. 63, n. 3, p. 664–674, mar 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7222471>>.
- LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 4768–4777. ISBN 9781510860964.
- MAKOWSKI, D. et al. NeuroKit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, Springer Science and Business Media LLC, v. 53, n. 4, p. 1689–1696, feb 2021. Disponível em: <<https://doi.org/10.3758%2Fs13428-020-01516-y>>.
- MITCHELL, L. B. Web Page, *Bundle Branch Block*. MSD Manual, 2024. Disponível em: <<https://www.msdmanuals.com/home/heart-and-blood-vessel-disorders/abnormal-heart-rhythms/bundle-branch-block>>.
- MITCHELL, L. B. Web Page, *Ventricular Fibrillation*. MSD Manual, 2024. Disponível em: <<https://www.msdmanuals.com/home/heart-and-blood-vessel-disorders/abnormal-heart-rhythms/ventricular-fibrillation>>.

- MITCHELL, L. B. *Ventricular Premature Beats*. 2024. Disponível na versão para leigos. Disponível em: <<https://www.msdmanuals.com/home/heart-and-blood-vessel-disorders/abnormal-heart-rhythms/ventricular-premature-beats>>.
- MITCHELL, L. B. Web Page, *Ventricular Tachycardia*. MSD Manual, 2024. Disponível em: <<https://www.msdmanuals.com/home/heart-and-blood-vessel-disorders/abnormal-heart-rhythms/ventricular-tachycardia>>.
- MITCHELL, L. B. *Visão Geral das Arritmias*. 2025. Disponível em: <<https://www.msdmanuals.com/pt/profissional/doen%C3%A7as-cardiovasculares/vis%C3%A3o-geral-de-arritmias-e-doen%C3%A7as-de-condu%C3%A7%C3%A3o/vis%C3%A3o-geral-das-arritmias?query=arritmia>>.
- MOODY, G.; MARK, R. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, v. 20, n. 3, p. 45–50, 2001.
- MOODY, G. B. *MIT-BIH Arrhythmia Database Directory*. 1997. Recurso online. Revisado em 24 de junho de 2010. Disponível em: <<https://archive.physionet.org/physiobank/database/html/mitdbdir/intro.htm>>.
- MOODY, G. B.; MARK, R. G. *MIT-BIH Arrhythmia Database*. 2005. Disponível em: <<https://www.physionet.org/physiobank/database/mitdb>>. Acesso em: 28 set. 2025.
- MOUSAVI, S.; AFGHAH, F. Inter- and intra- patient ecg heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1308–1312, 2018. Disponível em: <<https://api.semanticscholar.org/CorpusID:56173441>>.
- NAROTAMO, H. et al. Deep learning for ecg classification: A comparative study of 1d and 2d representations and multimodal fusion approaches. *Biomedical Signal Processing and Control*, Elsevier, 2024.
- NICULESCU-MIZIL, A.; CARUANA, R. Predicting good probabilities with supervised learning. In: *Proceedings of the 22nd International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2005. (ICML '05), p. 625–632. ISBN 1595931805. Disponível em: <<https://doi.org/10.1145/1102351.1102430>>.
- PAN, J.; TOMPKINS, W. J. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32, n. 3, p. 230–236, 1985.
- PATTI, L.; HORENSTEIN, M. S.; ASHURST, J. V. *Supraventricular Tachycardia*. Treasure Island, FL: StatPearls Publishing, 2025. Disponível em: NCBI Bookshelf. Acesso em: 25 out. 2025. Disponível em: <<https://www.ncbi.nlm.nih.gov/books/NBK441972/>>.
- RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "why should i trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 1135–1144. ISBN 9781450342322. Disponível em: <<https://doi.org/10.1145/2939672.2939778>>.
- SAADATNEJAD, S.; OVEISI, M.; HASHEMI, M. Lstm-based ecg classification for continuous monitoring on personal wearable devices. *IEE Journal of Biomedical and Health Informatics*, v. 24, n. 2, 2020.

- SAFDAR, M. F.; NOWAK, R. M.; PAŁKA, P. Pre-processing techniques and artificial intelligence algorithms for electrocardiogram (ecg) signals analysis: A comprehensive review. *Computers in Biology and Medicine*, v. 170, p. 107908, 2024. ISSN 0010-4825. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0010482523013732>>.
- SATTAR, Y.; HASHMI, M. F. Premature ventricular complex. *StatPearls [Internet]*, StatPearls Publishing, Treasure Island, FL, 2025. Última atualização: 16 de fevereiro de 2025. Disponível em: <<https://www.ncbi.nlm.nih.gov/books/NBK547713/>>.
- SILVA, G. A. L. et al. *A Systematic Review of ECG Arrhythmia Classification: Adherence to Standards, Fair Evaluation, and Embedded Feasibility*. 2025. ArXiv:2503.07276v1. Acesso em: 28 set. 2025.
- Texas Heart Institute. *Categories of Arrhythmias*. s.d. Acesso em: 25 out. 2025. Disponível em: <<https://www.texasheart.org/heart-health/heart-information-center/topics/categories-of-arrhythmias/>>.
- U.S. National Library of Medicine. *Ventricular Flutter - MeSH Descriptor Data*. 2025. <<https://www.ncbi.nlm.nih.gov/mesh?Db=mesh&Cmd=DetailsSearch&Term=C23.550.073.961>>. Acesso em: 25 out. 2025.
- VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020.
- World Health Organization. *Cardiovascular diseases (CVDs)*. 2025. <[https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))>. Acessado em: 15 de novembro de 2025.
- ZHANG, J. et al. Interpatient ecg heartbeat classification with an adversarial convolutional neural network. *Journal of Healthcare Engineering*, v. 2021, 2021. Disponível em: <<https://api.semanticscholar.org/CorpusID:235683987>>.