

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA

MATEUS ROSSATO SILVA

**Avaliação de redes neurais artificiais como
aproximador da função de entropia amostral para
séries temporais**

Ribeirão Preto–SP

2023

MATEUS ROSSATO SILVA

**Avaliação de redes neurais artificiais como aproximador da
função de entropia amostral para séries temporais**

Versão Corrigida

Versão original encontra-se na FFCLRP/USP.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) da Universidade de São Paulo (USP), como parte das exigências para a obtenção do título de Mestre em Ciências.

Área de Concentração: Computação Aplicada.

Orientador: Prof. Dr. Luiz Otávio Murta Junior

Ribeirão Preto–SP

2023

MATEUS ROSSATO SILVA

**Artificial neural network evaluation as a sample entropy
function approximator for time series**

Corrected Version

The original version is found at FFCLRP/USP.

Dissertation presented to Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) from the Universidade de São Paulo (USP), as part of the requirements to hold the Master of Science degree.

Field of Study: Applied Computing.

Supervisor: Prof. Dr. Luiz Otávio Murta Junior

Ribeirão Preto–SP

2023

Mateus Rossato Silva

Avaliação de redes neurais artificiais como aproximador da função de entropia amostral para séries temporais. Ribeirão Preto–SP, 2023.

112p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da USP, como parte das exigências para a obtenção do título de Mestre em Ciências,
Área: Computação Aplicada.

Orientador: Prof. Dr. Luiz Otávio Murta Junior

1. Redes Neurais Artificiais. 2. Séries Temporais. 3. Dados Substitutos.

Agradecimentos

Agradeço a Deus em primeiro lugar, pela vida e pela oportunidade de caminhar até aqui. Agradeço ao professor Dr. Luiz Otavio Murta Junior, pelo desafio em assumir a minha orientação, pela paciência, dedicação e todo ensinamento.

Aos meus pais Camilo e Marlene, a minha esposa Ana Laura, aos meus irmãos Filipe, Marina e Marcela. E a toda a minha família, pelo apoio, incentivo e compreensão ao meu tempo gasto dedicado ao desenvolvimento deste trabalho.

Aos amigos e também professores Alexandre Gomes e Airton Borin pela dedicação, auxílio e comprometimento nas reuniões de sexta-feira.

Por último, mas não menos importante, agradeço a Universidade de São Paulo – Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, pela oportunidade de cursar o programa de Pós-graduação em Computação Aplicada.

Resumo

A entropia amostral (SampEn) é amplamente utilizada para quantificar irregularidades em séries temporais fisiológicas com aplicações clínicas e de pesquisa. Por outro lado, a rede neural artificial (RNA) podem ser uma aproximadora universal de funções. Além disso, a abordagem da técnica de dados substitutos é amplamente utilizada para lidar com testes de hipóteses de não linearidade. Embora as RNAs sejam utilizadas como aproximadores da função SampEn, nenhuma investigação foi realizada para avaliar o potencial das RNAs em detectar a não linearidade do sistema. Neste estudo, avaliamos o comportamento das RNAs como um aproximador da SampEn em séries temporais fisiológicas não lineares e suas correspondentes séries temporais de dados substitutos lineares. Esta investigação nos permitiu responder a algumas questões, ou seja, até que ponto é possível aproximar SampEn com RNA sob condições de linearidade e não linearidade, e que tipo de RNA pode prever a entropia com precisão em séries temporais não lineares em comparação com seus substitutos correspondentes? É possível usar dados substitutos para aumento de dados? RNAs com funções de ativação linear, sigmoidal e ReLU foram treinadas com números de época fixos e analisadas para os conjuntos de dados originais e substitutos. Os resultados para ritmo sinusal normal mostram erros médios de desvios padrão de $\pm 0,0258 \pm 0,0022$ e $0,0227 \pm 0,0022$ para as séries original e substituta com função de ativação linear, respectivamente, $0,0229 \pm 0,0011$ e $0,0213 \pm 0,0011$ com a função ReLU, e $0,0224 \pm 0,0014$ e $0,0204 \pm 0,0014$ para a função sigmoidal com RNA treinada com diversas condições fisiológicas. Quando treinada com uma única condição, a função sigmoidal responde melhor aos dados originais, obtendo o melhor resultado para o erro. Além disso, observou-se que, para a função ReLU, a série substituta resultou em erros menores que os obtidos para a série original. Observamos erros semelhantes para a função de ativação linear. Os resultados indicam que o processo sigmoidal gera o menor erro com aumento do erro para séries substitutas, sugerindo que a rede com função sigmoidal pode se aproximar do valor real de entropia ao lidar com séries reais. Assim, dados substitutos são bons candidatos para aumento de dados de séries temporais fisiológicas.

Palavras-chave: Redes Neurais Artificiais. Séries Temporais. Entropia Amostral. Dados Substitutos.

Abstract

Sample entropy (SampEn) is widely used to quantify irregularities in physiological time series with clinical and research applications. On the other hand, artificial neural networks (ANN) can be a universal approximation of functions. Furthermore, the surrogate-data technique approach is widely used to deal with nonlinearity hypothesis tests. Although ANNs are used as approximators of the SampEn function, no investigations were carried out to evaluate the potential of ANNs to detect system nonlinearity. In this study, we evaluated the behavior of ANNs as an approximator of SampEn in nonlinear physiological time series and their corresponding linear surrogate data time series. This investigation allowed us to answer a couple of questions, i.e., to what extent is it possible to approximate SampEn with ANN under linearity and nonlinearity conditions, and what kind of ANN can predict entropy accurately in nonlinear time series compared to their corresponding surrogates? Is it possible to use surrogate data for data augmentation? ANNs with linear, sigmoidal, and ReLU activation functions were trained with fixed epoch numbers and analyzed for the original and surrogate data sets. The results for normal sinus rhythm show mean \pm standard deviation errors of 0.0258 ± 0.0022 and 0.0227 ± 0.0022 for the original and surrogate series with linear activation function, respectively, 0.0229 ± 0.0011 and 0.0213 ± 0.0011 with the ReLU function, and 0.0224 ± 0.0014 and 0.0204 ± 0.0014 for the sigmoidal function with ANN trained with various physiological conditions. When trained with a single condition, the sigmoidal function responds better to the original data, obtaining the best result for the error. Furthermore, it was observed that, for the ReLU function, the surrogate series resulted in smaller errors than those obtained for the original series. We observe similar errors for the linear activation function. The results indicate that the sigmoidal process generates the smallest error with an increase in the error for surrogate series, suggesting that the network with a sigmoidal function can get closer to the real entropy value when dealing with real series. Thus, surrogate-data is a good candidates for data augmentation of phisiological time-series.

Keywords: Artificial Neural Network. Time Series. Sample Entropy. Surrogate Data.

Lista de figuras

| | |
|---|----|
| Figura 1 – Arquitetura Rede Neural Convolutacional | 40 |
| Figura 2 – Arquitetura Rede Neural Recorrente | 41 |
| Figura 3 – Arquitetura Rede Neural LSTM | 42 |
| Figura 4 – Função de Ativação Linear | 45 |
| Figura 5 – Função de Ativação ReLU | 46 |
| Figura 6 – Função de Ativação Sigmoides | 47 |
| Figura 7 – Geração de Dados Substitutos | 58 |
| Figura 8 – Amostras do Conjunto de Dados Originais | 58 |
| Figura 9 – Amostras do Conjunto de Dados Substitutos | 59 |
| Figura 10 – Comparação Série Original x Série Substituta | 59 |
| Figura 11 – Rede Neural Artificial - Uma Camada Intermediária | 60 |
| Figura 12 – Rede Neural Artificial - Duas Camadas Intermediárias | 60 |
| Figura 13 – Funcionamento da Rede Neural | 61 |
| Figura 14 – Tipos de Treinamento da Rede Neural | 62 |
| Figura 15 – Representação da Técnica de Validação Cruzada | 63 |
| Figura 16 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 66 |
| Figura 17 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu | 66 |
| Figura 18 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 67 |
| Figura 19 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 68 |

| | |
|---|----|
| Figura 20 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 68 |
| Figura 21 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 69 |
| Figura 22 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 70 |
| Figura 23 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 70 |
| Figura 24 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 71 |
| Figura 25 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 72 |
| Figura 26 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 72 |

| | |
|--|----|
| Figura 27 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 73 |
| Figura 28 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 74 |
| Figura 29 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 74 |
| Figura 30 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 75 |
| Figura 31 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 76 |
| Figura 32 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu | 76 |
| Figura 33 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 77 |

| | |
|--|----|
| Figura 34 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 78 |
| Figura 35 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 78 |
| Figura 36 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 79 |
| Figura 37 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 80 |
| Figura 38 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 80 |
| Figura 39 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 81 |
| Figura 40 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 82 |
| Figura 41 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 82 |

| | |
|---|----|
| Figura 42 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 83 |
| Figura 43 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 84 |
| Figura 44 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 84 |
| Figura 45 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 85 |
| Figura 46 – Representação da Técnica de Conjunto de Arquivos | 92 |
| Figura 47 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 93 |
| Figura 48 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu | 93 |
| Figura 49 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 94 |
| Figura 50 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 95 |

| | |
|---|----|
| Figura 51 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 95 |
| Figura 52 – Comparação entre os resultados originais (LTAFDB - STO) e os resultados substitutos (LTAFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 96 |
| Figura 53 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camadas intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 97 |
| Figura 54 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 97 |
| Figura 55 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 98 |
| Figura 56 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 99 |
| Figura 57 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 99 |

| | |
|---|-----|
| Figura 58 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 100 |
| Figura 59 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear. | 101 |
| Figura 60 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu. | 101 |
| Figura 61 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide. | 102 |
| Figura 62 – Comparação entre os resultados do teste t de Student entre STO e STS, utilizando função de ativação linear. | 105 |
| Figura 63 – Comparação entre os resultados do teste t de Student entre STO e STS, utilizando função de ativação relu. | 106 |
| Figura 64 – Comparação entre os resultados do teste t de Student entre STO e STS, utilizando função de ativação sigmoide. | 106 |

Lista de tabelas

| | |
|--|-----|
| Tabela 1 – Configurações de Camadas da Rede Neural | 86 |
| Tabela 2 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para NSR2DB | 87 |
| Tabela 3 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para LTAFDB | 88 |
| Tabela 4 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para LTSTDB | 89 |
| Tabela 5 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para CHF2DB | 90 |
| Tabela 6 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para CHFDB | 91 |
| Tabela 7 – Comparações de desempenho de acordo com o número de épocas no conjunto NSR2DB. | 104 |

Lista de abreviaturas e siglas

| | |
|--------|---|
| IAAFT | - Transformada de Fourier Ajustada de Amplitude Iterativa |
| IID | - Independente e Identicamente Distribuídos |
| EL | - Expoente de Lyapunov |
| EQM | - Erro Quadrático Médio |
| KS | - Entropia de Kolmogorov Sinai |
| ReLU | - Unidade Linear Retificada |
| REQM | - Raiz do Erro Quadrático Médio |
| RNA | - Redes Neurais Artificiais |
| SampEn | - Entropia de Amostra |
| SD | - Dados Substitutos |
| STO | - Séries Temporais Originais |
| STS | - Séries Temporais Substitutas |

Sumário

| | | |
|----------|---|-----------|
| 1 | FUNDAMENTAÇÃO TEÓRICA | 33 |
| 1.1 | Teoria do Caos | 33 |
| 1.2 | Entropia de Kolmogorov-Sinai (K-S) | 34 |
| 1.3 | Expoente de Lyapunov (EL) | 35 |
| 1.4 | Entropia Amostral - Sample Entropy (SampEn) | 36 |
| 1.5 | Dados Substitutos | 37 |
| 1.6 | Redes Neurais Artificiais (RNA) | 39 |
| 1.6.1 | Redes Neurais Convolucionais (RNC) | 39 |
| 1.6.2 | Redes Neurais Recorrentes (RNN) | 41 |
| 1.6.3 | Redes Neurais de Memória de Longo Prazo | 42 |
| 1.6.4 | Funções de Ativação | 44 |
| 1.6.4.1 | Linear | 44 |
| 1.6.4.2 | ReLU | 45 |
| 1.6.4.3 | Sigmoide | 46 |
| 2 | TRABALHOS RELACIONADOS | 49 |
| 2.1 | Um método para estimar a entropia de séries temporais usando redes neurais artificiais | 49 |
| 2.2 | Deteção de padrões não lineares em sinais fisiológicos | 50 |
| 2.3 | Estimando a entropia do sistema com um modelo de rede neural recorrente profunda | 51 |
| 3 | OBJETIVOS | 53 |
| 4 | MATERIAIS E MÉTODOS | 55 |
| 4.1 | Base de Dados | 55 |
| 4.1.1 | NSR2DB - Normal Sinus Rhythm RR Interval Database | 56 |
| 4.1.2 | LTAFDB - Long Term Atrial Fibrillation Database | 56 |
| 4.1.3 | LTSTDB - Long Term Segment ST Database | 56 |
| 4.1.4 | CHF2DB - Congestive Heart Failure RR Interval Database | 57 |
| 4.1.5 | CHFDB - Severe Congestive Heart Failure Database | 57 |
| 4.2 | Seleção do Conjunto de Dados | 58 |
| 4.3 | Arquitetura da Rede Neural | 60 |
| 4.3.1 | Validação Cruzada | 63 |
| 4.3.2 | EQM e REQM | 63 |

| | | |
|------------|---|------------|
| 5 | RESULTADOS E DISCUSSÃO | 65 |
| 5.1 | Experimentos | 65 |
| 5.1.1 | Teste com uma camada intermediária | 65 |
| 5.1.2 | Teste com duas camadas intermediárias | 75 |
| 5.1.3 | Teste com uma camada intermediária e variação na função de ativação | 86 |
| 5.1.4 | Teste com uma camada intermediária e conjunto de dados | 92 |
| 6 | CONCLUSÕES | 107 |
| | REFERÊNCIAS | 109 |

Introdução

A análise de séries temporais está presente em diversos campos da ciência, desde áreas da computação até áreas da medicina. Diversos métodos matemáticos e estatísticos são utilizados para medir a irregularidade em séries temporais, um exemplo é a entropia, um conceito importante em áreas distintas: física estatística e teoria da informação. Na física estatística, a entropia é uma medida da desordem molecular em um sistema fechado, quanto maior a entropia, maior a desordem. Na teoria da informação a entropia é uma medida utilizada para quantificar o grau de incerteza associada a uma variável aleatória, quanto maior a entropia, maior a incerteza em relação à variável em análise.

De acordo com Yin e Shang (2016), a entropia é uma ferramenta interessante para analisar séries temporais dado que não impõe restrições sobre a distribuição de probabilidade subjacente. Alguns exemplos de entropia são:

- A entropia de Shannon (ShEn);
- A entropia Condicional (ConEn);
- A entropia de Permutação (PerEn);
- A entropia Aproximada (ApEn); e
- A entropia Amostral (SaEn).

Para Azami e Escudero (2018) a entropia de Shannon (ShEn) e a entropia Condicional (ConEn) são medidas básicas para avaliar a entropia, dado que ShEn e ConEn medem a quantidade de informação e a taxa de geração de informação respectivamente.

Com base nessas medidas, outras métricas de entropia foram introduzidas para avaliar a irregularidade em séries temporais, como no trabalho de Letellier (2006), onde os autores utilizaram *recurrence plots* (gráficos recorrentes), para estimar a entropia de Shannon (ShEn). A entropia de permutação proposta no trabalho de and (2019), também é uma medida de entropia popular utilizada para identificar o padrão de permutação em séries temporais. Outra entropia comumente utilizada na literatura é a entropia aproximada (ApEn), proposta por Pincus (1991).

Para o desenvolvimento deste trabalho consideramos a entropia amostral (SaEn), introduzida por Richman e Moorman (2000), que também é uma métrica estatística utilizada para medir a irregularidade/complexidade em séries temporais. As medidas ApEn e SaEn são fundamentadas na entropia ConEn. Todos esses métodos são baseados na distribuição de probabilidade e possuem deficiências como, sensibilidade em séries temporais de curta duração no trabalho de Keum e Coulibaly (2017), ou a questão da igualdade em séries temporais discutida no trabalho de Zunino et al. (2017). Igualdade refere-se a valores repetidos na série temporal que podem influenciar negativamente as estimativas da entropia.

Com objetivo de superar estas dificuldades, muitos pesquisadores tem modificado seus métodos, como o trabalho de Molavipour et al. (2021), onde os autores utilizaram redes neurais para aproximar as probabilidades em equações de informação mútua, que são baseadas na entropia de Shannon (ShEn). Outro exemplo é o proposto em Martinez-Garcia et al. (2019), os autores utilizaram redes neurais recorrentes profundas para aproximar a distribuição de probabilidade das saídas do sistema. Ainda é possível citar os trabalhos de Deng (2016), Zhang e Deng (2021), onde foi introduzida a entropia de Deng, uma generalização da entropia de Shannon (ShEn).

Neste trabalho analisamos um novo método para avaliar a complexidade/irregularidade de uma série temporal, que tem uma estrutura completamente diferente em comparação com os outros métodos com redes neurais artificiais (RNA), no contexto de séries temporais originais e séries substitutas geradas a partir do conjunto de séries temporais originais por meio de um algoritmo iAAFT (*Iterated Amplitude Adjusted Fourier Transform*). Dados Substitutos (Surrogate Data), são transformações de uma determinada série temporal, que mantém parte das características desta série, permitindo o teste de hipóteses sobre a estrutura desta série temporal.

Comparamos os resultados obtidos a partir do treinamento e teste das redes neurais com conjuntos de dados originais e o conjunto de dados substitutos. Com isso, nosso principal objetivo é identificar se redes neurais artificiais são capazes de prever as taxas de produção de informação pelos sistemas, medidas pelas entropias colocadas da mesma forma em séries originais e em séries substitutas. Também é válido ressaltar que as séries consideradas possuem características caóticas e/ou não lineares.

Este trabalho encontra-se dividido em 7 capítulos. No capítulo 1, abordamos a fundamentação teórica, apresentando conceitos importantes para o desenvolvimento deste trabalho. No capítulo 2, exploramos pesquisas relacionadas que compartilham propósitos correlatos e que contribuíram para o desenvolvimento deste projeto. Os objetivos são detalhados no capítulo 3, enquanto o capítulo 4 descreve os materiais, métodos e o processo de seleção e geração de dados, além de apresentar a arquitetura da rede neural utilizada. O capítulo 5 apresenta os resultados obtidos e a discussão decorrente dos testes realizados.

No capítulo 6, encerramos o trabalho com as conclusões do estudo e perspectivas futuras.

Fundamentação Teórica

Neste capítulo são apresentados alguns temas de suma importância para o desenvolvimento deste trabalho. Os conceitos de Teoria do Caos, Entropia de Kolmogorov-Sinai, Expoente de Lyapunov, Entropia Amostral, Dados Substitutos e Redes Neurais Artificiais.

1.1 Teoria do Caos

A teoria do caos é um campo da matemática e da física que estuda sistemas dinâmicos complexos e sensíveis às condições iniciais, permitindo identificar como pequenas mudanças nas condições iniciais de um sistema, podem levar a grandes e imprevisíveis variações em seu comportamento.

De acordo com Su (2021), a teoria do caos teve sua origem na segunda metade do século XX e sua origem está diretamente relacionada com o clima e a meteorologia. Segundo Cardoso e Pereira (2005), Edward Lorenz, um dos pioneiros nesse campo, desenvolveu um modelo computacional simples com o objetivo de prever as condições climáticas futuras e verificar se sua suposição sobre a repetição de padrões climáticos estava correta. A partir disso, Lorenz descobriu que, embora existisse um padrão, ele sempre apresentava pequenas alterações.

Para Biswas, Hasan e Bala (2018), a palavra Caos origina-se do grego *Khaos*, cujo significado é vazio, além disso, a teoria do caos é o estudo de sistemas dinâmicos complexos e não lineares. Para Zeng, Pielke e Eykholt (1993), podemos classificar sistema dinâmico como qualquer sistema que possa ser descrito matematicamente por equações diferenciais ou mapeamentos iterativos.

Conforme Marietto, Sanches e Meireles (2011), a teoria do caos surge com a ideia fundamental de que, em determinados sistemas, pequenas variações nas condições iniciais podem gerar grandes variações nos resultados. Além disso, a teoria do caos ainda é uma disciplina científica em desenvolvimento, focada no estudo dos sistemas não-lineares

complexos. Ainda sobre Marietto, Sanches e Meireles (2011), para o entendimento do caos, se faz necessário o entendimento de três termos básicos:

- Sistema;
- Não linearidade; e
- Complexidade.

É possível definir sistema como uma relação de interdependência e inter-relacionamento entre partes, um exemplo seria uma pilha de pedras, quando se tira uma, as outras caem.

O conceito de não linearidade está associado a estrutura matemática necessária para representar o comportamento do sistema real. Em sistemas não lineares é possível identificar a falta de proporcionalidade entre variáveis, diferente de sistemas lineares. Para Su (2021), consideramos um sistema como não linear quando a saída não é proporcional à entrada e a frequência pode mudar, diferentemente de sistemas lineares, onde a saída é proporcional à entrada e a frequência permanece imutável.

O terceiro termo, complexidade, é definido por Marietto, Sanches e Meireles (2011) como a dificuldade de construir um modelo capaz de prever o comportamento de um sistema real. Na visão da complexidade, o termo caos faz referência ao comportamento de sistemas dinâmicos, derivados da dependência sensível às condições iniciais e as interações complexas entre variáveis do sistema.

Biswas, Hasan e Bala (2018) afirma que áreas como, matemática, ciência da computação e robótica, economia e finanças, meteorologia e fisiologia, se beneficiam da teoria do caos para o desenvolvimento de aplicações de uso diário.

1.2 Entropia de Kolmogorov-Sinai (K-S)

A entropia de Kolmogorov-Sinai (K-S) é uma medida utilizada na teoria da informação, para compreender sistemas complexos e caóticos. Por meio desta medida é possível quantificar a complexidade e a imprevisibilidade do comportamento de um sistema dinâmico.

Para Scarciglia et al. (2021), a entropia K-S é um quantificador não linear de sistemas dinâmicos e representa a quantidade máxima de informação necessária para descrever o comportamento desses sistemas.

Segundo Badino (2014), a entropia K-S compartilha características com a entropia de Shannon. Os autores definem a entropia de kolmogorov-Sinai como uma forma de medir a complexidade algorítmica em sistemas dinâmicos.

De acordo com Scarciglia et al. (2022), os sistemas estocásticos estão frequentemente associados a valores infinitos de entropia K-S, ao contrário dos sistemas determinísticos, que são definidos por valores finitos da entropia K-S. A maneira correta de calcular a entropia K-S de uma série temporal consiste nas seguintes etapas:

- Identificar o intervalo de amplitude (I) da série temporal;
- Dividir (I) em um conjunto finito (Z) de subconjuntos disjuntos; e
- Calcular a entropia K-S em relação a uma partição finita (Z).

De forma complementar, as séries temporais originais são divididas em séries com granularidade grosseira utilizando um fator de escala τ . A série original é dividida em janelas não sobrepostas ao comprimento τ , e a média é calculada dentro de cada janela.

Ainda segundo Scarciglia et al. (2022), quanto maior a previsibilidade de uma série, menor é o seu valor de entropia, pois é necessário menos informações para descrever padrões repetitivos.

1.3 Expoente de Lyapunov (EL)

Por meio dos Expoentes de Lyapunov (EL), é possível quantificar a taxa de divergência ou convergência durante as diversas evoluções que o sistema pode seguir ao longo do tempo.

Para Silva et al. (2012), o EL mede a taxa de divergência ou convergência entre trajetórias próximas, proporcionando uma estimativa do nível do caos em um sistema dinâmico.

Segundo Simon e Kumar (2013), um sistema caótico é caracterizado pela sensibilidade nas condições iniciais e pela rápida divergência de órbitas próximas no espaço de fase. Por intermédio do EL é possível medir a taxa de divergência ou convergência da órbita. Caso o EL for positivo, indica divergência da órbita, caso seja negativo, indica convergência da órbita.

De acordo com Hasan et al. (2023), uma característica definidora de um sistema caótico é a dependência sensível da condição inicial. A métrica mais utilizada para quantificar essa dependência das condições iniciais é o EL, representada pela equação abaixo em um sistema caótico de tempo discreto.

$$LE = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \quad (1.1)$$

As variáveis que compõem a equação 1.1, são descritas a seguir.

n : Refere-se ao número de iterações ou passos no tempo.

$\frac{1}{n}$: Representa a média das iterações ao longo do tempo. Quanto maior o número de iterações, mais precisa será a avaliação do comportamento do sistema.

$\sum_{i=0}^{n-1}$: Soma ao longo das iterações. Cada termo da soma colabora para o cálculo final do expoente.

\ln : Logaritmo natural.

$|f'(x_i)|$: Valor absoluto da derivada de f em relação a x , avaliado em (x_i) .

1.4 Entropia Amostral - Sample Entropy (SampEn)

De acordo com Lv et al. (2021), a Sample Entropy (SampEn) ou entropia amostral é uma ferramenta valiosa para compreender padrões complexos em dados temporais, sendo útil em diversas aplicações, especialmente em áreas da medicina. Por meio desta medida, é possível quantificar a irregularidade de padrões em dados temporais, quanto mais irregular o padrão, maior é o valor da SampEn, por outro lado, se o conjunto de dados é caracterizado por auto-similaridade, indicando padrões mais simples e regulares, a SampEn tenderá a apresentar valores mais baixos.

Para a realização do cálculo da *Sample Entropy*, são necessárias algumas etapas, são elas:

- Construção de Sequências;
- Medição de Distância;
- Probabilidade de Similaridade;
- Média da Probabilidade;
- Ajuste e Repetição; e
- Cálculo final.

Na primeira etapa, serão criadas sequências de dados menores a partir da série de dados original, posteriormente é realizada a comparação entre as sequências com o objetivo de identificar a distância entre elas. O próximo passo consiste no cálculo de duas sequências de dados serem semelhantes, baseado em um limiar específico. Posteriormente calcula-se a média das probabilidades, aumenta o tamanho das sequências de dados e repete os passos anteriores para obtenção da média da nova probabilidade.

A partir dos valores obtidos, calcula-se a SampEn utilizando a formula ?? descrita abaixo.

$$\text{SampEn}(m, r, N) = -\ln\left(\frac{A_{m+1}}{B_m}\right) \quad (1.2)$$

Fórmula SampEn.

O parâmetro m indica a dimensão da incorporação. O r é o limiar efetivo, responsável por definir a distância máxima permitida entre duas subsequências para serem consideradas semelhantes, e N é o comprimento da sequência. Já A_{m+1} representa o número de pares de sequência de comprimento $m + 1$, cuja distância não excede o valor de r . B_m representa o número de pares de sequência de comprimento m , cuja distância também não excede o valor de r . O \ln representa o logaritmo natural, também conhecido como neperiano.

1.5 Dados Substitutos

Dados Substitutos (Surrogate Data), são transformações de uma determinada série temporal, que mantém parte das características desta série, permitindo o teste de hipóteses sobre a estrutura desta série temporal. De acordo com Lee et al. (2019) os dados substitutos foram projetados para identificar a presença de não linearidade em uma série temporal, além disso os testes utilizando dados substitutos são baseados em estatísticas, onde a hipótese nula é que a série temporal em análise resulte um processo linear estocástico e estacionário com entradas aleatórias.

Para Theiler et al. (1992) a presença ou não, da dinâmica não linear em uma série temporal pode ser testada utilizando a técnica de dados substitutos (*surrogate data*). O teste consiste nas seguintes etapas: A definição da hipótese nula, o conjunto de séries substitutas e o cálculo do discriminante estatístico que irá dizer se a hipótese nula será rejeitada ou não.

Na definição de hipótese nula, considera-se que a série temporal é originada de um processo linear, esta é precisamente a hipótese que se pretende rejeitar. Um conjunto de séries substitutas é gerado, compartilhando propriedades e características idênticas ou similares à série original, dependendo da hipótese definida, porém tendo como origem um processo estocástico. Um discriminante estatístico é então calculado para a série original e para todas as séries substitutas. Caso exista diferença significativa entre os valores discriminantes para a série original e as séries substitutas, a hipótese nula é rejeitada.

Uma das hipóteses nulas mais simples é considerar os dados gerados por processos ou variáveis aleatórias independente e identicamente distribuídos (IID). Neste caso, as séries substitutas podem ser obtidas pelo simples embaralhamento da série original, o que faz com que as séries substitutas tenham a mesma distribuição da série original, porém qualquer correlação existente entre os instantes temporais é destruída. Consequentemente, se não for observada diferença significativa entre as medidas discriminantes para séries originais e substitutas, decorre que a série original foi decorrente de uma IID.

Outras duas hipóteses são também frequentes. A primeira considera a série temporal como resultante de um processo linear gaussiano autocorrelacionado, neste caso, as séries substitutas devem ter o mesmo espectro de potências da série original. Na prática, pode ser obtido de diferentes maneiras, uma delas é calcular a transformada de fourier da série original, embaralhar as componentes de fase mantendo os módulos, e então aplicar a transformada inversa. O resultado será uma série com distribuição distinta, porém com as mesmas propriedades de autocorrelação da série original.

Em contexto mais geral, um terceiro tipo de dados substitutos considera a possibilidade da série ter sido mensurada por uma medida não linear estática e inversível. Neste caso, séries substitutas devem emular tanto o espectro de potências como a distribuição da série original. O método mais utilizado para a obtenção de séries substitutas deste tipo é o chamado IAAFT (*Iterated Amplitude Adjusted Fourier Transform*), utilizando um processo iterativo. O algoritmo IAAFT é descrito abaixo:

1. Dada uma série temporal $u(n)$, armazenar seu espectro de potências (Pu).
2. Gerar uma cópia de $u(n)$, com seus valores pontuais embaralhados aleatoriamente, resultando em $u_a(n)$.
3. Aplicar a Transformada de Fourier em $u_a(n)$, substituir seu espectro de potências pelo espectro original P_u , e então aplicar a Transformada Inversa de Fourier, resultando em $\bar{u}_a(n)$.
4. Ordenar os valores em $u_a(n)$ de acordo com $\bar{u}_a(n)$, de modo que $u_a(n)$ contenha os pontos da série original, mas dispostos de forma que a posição do maior valor corresponda à mesma posição do maior valor em $\bar{u}_a(n)$, e assim sucessivamente.
5. Repetir os passos 3 e 4 iterativamente até que haja convergência.

A última técnica de geração de dados substitutos é a mais adequada neste projeto por ser a mais geral e considerar tanto o aspecto das IID como as autocorrelações. A hipótese nula aqui se refere a afirmação que a rede neural artificial (RNA) é capaz de prever o SampEn da mesma forma em dados originais e dados substitutos. Em caso de

confirmação ou rejeição da hipótese nula tem consequências que podem levar a discussões muito interessantes.

1.6 Redes Neurais Artificiais (RNA)

Com base no funcionamento do cérebro humano, uma rede neural artificial (RNA), é um modelo computacional composto por neurônios conectados, capaz de processar informações a fim de possibilitar o reconhecimento de padrões e tomadas de decisão.

Para Duran et al. (2021), uma RNA é composta por unidades de processamento, agrupadas em conjunto e interconectadas, denominadas neurônios artificiais. Os sinais são recebidos pela rede e posteriormente transmitidos e processados, resultando em uma resposta de saída. A ativação dos neurônios artificiais ocorre com base na intensidade dos sinais de entrada que recebem.

De acordo com Polepally et al. (2021), embora o conceito de redes neurais já existisse algum tempo, foi no ano de 1980 que as redes neurais ganharam destaque, posteriormente no ano de 1990, começaram a ser amplamente aplicadas para fins de previsão. O uso de RNA é frequentemente utilizado para o desenvolvimento de soluções em diversos setores, e.g: Negócios, saúde, economia e tecnologia. As redes neurais imitam o sistema nervoso humano para realizar cálculos, este, por sua vez, consiste em neurônios que são ativados por impulsos elétricos específicos para a realização de determinada função. Não sendo diferente para as redes neurais, das quais também são compostas por neurônios associados a pesos, sendo assim, quando um peso específico é fornecido como entrada o neurônio é ativado.

É válido ressaltar a existência de alguns tipos de redes neurais, são elas:

- Redes Neurais Convolucionais;
- Redes Neurais Recorrentes; e
- Redes Neurais de Memória de Longo Prazo

1.6.1 Redes Neurais Convolucionais (RNC)

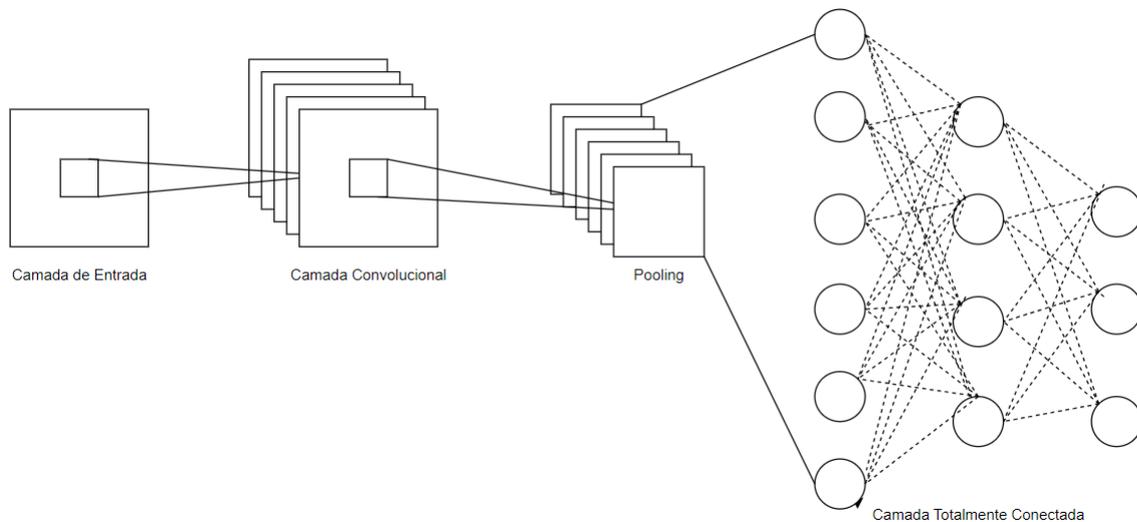
Para Duran et al. (2021), as redes neurais convolucionais (RNC), constituem um tipo específico de rede neural artificial, composta de várias camadas, projetadas para realizar tarefas de reconhecimento de padrões e características de imagens.

De acordo com Avila et al. (2023), a RNC possui diversas aplicações, como classificação de imagens e detecção de objetos. Este tipo de rede neural possui uma elevada

capacidade de processamento, permitindo que grandes quantidades relevantes de dados sejam processados com rapidez e precisão. Para otimizar o desempenho deste tipo de rede neural, podem ser aplicadas técnicas de *pooling* e a utilização de camadas totalmente conectadas. Além disso, é possível considerar o aumento de dados para refinar a capacidade de generalização da rede neural e conseqüentemente evitar *overfitting*.

A figura 1 abaixo, apresenta a arquitetura de uma rede neural convolucional.

Figura 1 – Arquitetura Rede Neural Convolucional



Por meio da imagem acima, podemos visualizar a representação de uma rede neural convolucional, composta pelas seguintes camadas:

- **Camada de Entrada:** Responsável por receber os dados de treinamento da rede neural.
- **Camada Convolucional:** Realiza a aplicação de filtros ou *kernels* à entrada, com objetivo de extrair características específicas dos dados.
- *Pooling*: Também conhecida como *subsampling*, reduz a dimensionalidade das características extraídas pela camada convolucional.
- **Camada totalmente conectada:** Cada neurônio está conectado à todos os neurônios da camada anterior. Essa camada é utilizada para a interpretação final das características extraídas pelas camadas convolucional e de *pooling*.

De modo geral, o fluxo de dados inicia na camada de entrada, passa pelas camadas convolucionais e de *pooling* para extração de características, e finalmente passa pela camada totalmente conectada para que seja realizada a classificação ou regressão.

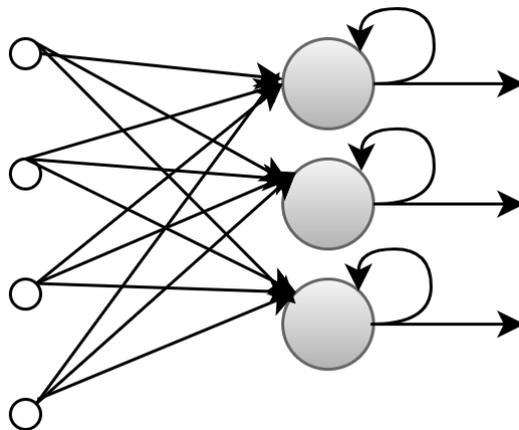
1.6.2 Redes Neurais Recorrentes (RNN)

De acordo com Polepally et al. (2021), as redes neurais recorrentes são constituídas de uma única camada ou um conjunto de camadas que se repetem ao longo das séries completas de dados. Este tipo de rede neural divide os conjuntos de dados de entrada com base em intervalos de tempo, os quais são determinados com base na definição do problema e nas informações fornecidas à camada a partir de estados anteriores. Os pesos desta rede são atualizados por meio da retropropagação, onde o erro gerado entre a saída prevista e a saída real são enviados de volta para a rede.

Para Liu, Li e Ye (2023), uma rede neural recorrente é uma rede com arquitetura flexível que realiza o compartilhamento de pesos. Essa rede possui a capacidade de aprender continuamente, realizando previsões sobre o estado futuro de desenvolvimento, com base na passagem do tempo.

A figura 2 abaixo, apresenta a arquitetura de uma rede neural recorrente.

Figura 2 – Arquitetura Rede Neural Recorrente



Observando a imagem anterior, é possível identificar a camada de entrada da rede, sendo representada pelos círculos menores à esquerda. Cada nó de entrada corresponde à um valor de entrada no vetor de dados. Os círculos na direita, dispostos na cor cinza representam as camadas ocultas da rede, sendo esta a camada responsável por armazenar informações ao decorrer do tempo para que a rede possa “lembrar” de estados anteriores. As setas de retorno, representam as conexões de *feedback*, por meio destas conexões é permissível que a saída de um nó na camada oculta em um passo temporal de $t - 1$, seja utilizada como entrada para o mesmo nó no passo temporal t . As setas pretas entre os nós representam os pesos das conexões entre os nós de entrada e os nós da camada oculta, esses pesos são ajustados durante o treinamento da rede para minimizar a função de perda.

1.6.3 Redes Neurais de Memória de Longo Prazo

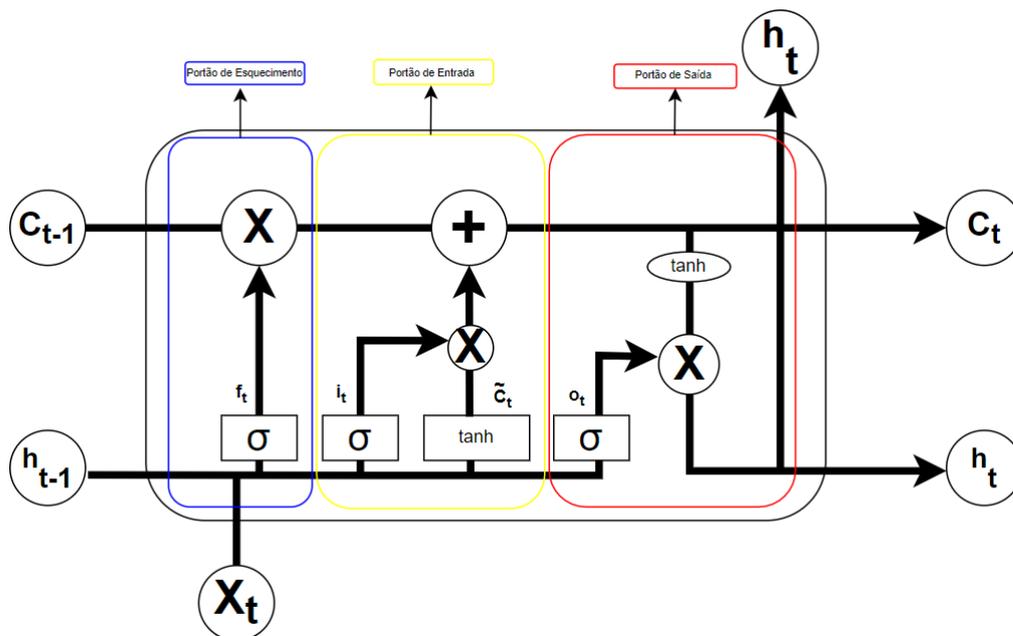
Derivado termo *Long Short-Term Memory* (LSTM), a rede neural de memória de longo prazo, é um tipo de rede neural recorrente muito utilizada no processamento de informações e tarefas que envolvam séries temporais. De acordo com Polepally et al. (2021), o modelo LSTM foi proposto por Schmidhuber e Hochreiter, inicialmente com a finalidade de superar as limitações das redes neurais recorrentes.

Segundo Bkassiny (2022) as redes neurais do tipo LSTM são uma categoria especial de redes neurais recorrentes, projetadas para identificar a correlação temporal entre as amostras de entrada. Este modelo de rede neural assegura uma eficácia superior na prevenção do problema do desaparecimento do gradiente, utilizando uma estrutura celular específica. Este tipo de problema é bem comum no treinamento de redes neurais recorrentes em sequências de dados maiores.

Para Mesquita, Oliveira e Pereira (2020), as redes neurais do tipo LSTM utilizam portas que permitem o ajuste de pesos através da capacidade de modificar (adicionar, descartar ou atualizar) as informações no tempo, permitindo prever melhor estados futuros. Essa característica de utilização de portas garante uma vantagem em relação as redes neurais recorrentes comuns, além de torná-las ideais para o processamento de séries temporais.

A figura 3 abaixo, apresenta a arquitetura de uma rede neural LSTM.

Figura 3 – Arquitetura Rede Neural LSTM



Em um primeiro momento, as células de memória, responsáveis pelo armazenamento de informações são inicializadas. A equação 1.3 apresenta a inicialização, onde \tilde{c}_t representa

a célula de memória inicializada.

$$c_t = \tilde{c}_t \quad (1.3)$$

Posteriormente, as informações passam pelo portão de esquecimento, dado que é nesta etapa que a rede LSTM decide a quantidade de informações das células anteriores deverá ser esquecida. Este processo é realizado por meio de uma função de ativação, que será responsável por determinar o esquecimento. Na equação 1.4, o portão de esquecimento é representado por f_t e os valores de W_{xf} , W_{hf} , e b_f são os pesos e bias associados ao portão de esquecimento.

$$f_t = \sigma(W_{xf}x[t] + W_{hf}h_{t-1} + b_f) \quad (1.4)$$

A próxima etapa consiste na passagem das informações pelo portão de entrada. Nesta fase, a rede determina quais novas informações devem ser adicionadas à célula de memória. A função de ativação nesta etapa é responsável por controlar a quantidade de nova informação que será adicionada. O portão de entrada é representada por i_t na equação 1.5. Os valores contidos em W_{xi} , W_{hi} , e b_i são os pesos e bias associados ao portão de entrada.

$$i_t = \sigma(W_{xi}x[t] + W_{hi}h_{t-1} + b_i) \quad (1.5)$$

Durante o processo de execução da rede neural, as células de memória são atualizadas com base na informação fornecida pelo portão de esquecimento e a nova informação proposta pelo portão de entrada. Essa etapa é representada pela equação 1.6.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (1.6)$$

Por último, o cálculo de saída é realizado através do portão de saída, com base na informação atualizada na célula de memória. O portão de saída irá determinar também o quanto da informação atualizada na célula deve ser a saída final. Na equação 1.7 o portão de saída é representado por o_t e os valores de W_{xo} , W_{ho} , e b_o são os pesos e bias associados ao portão de saída.

$$o_t = \sigma(W_{xo}x[t] + W_{ho}h_{t-1} + b_o) \quad (1.7)$$

A saída da célula LSTM é determinada pela multiplicação do resultado do portão de saída pelo valor da tangente hiperbólica da célula de memória atualizada, podendo ser definida como:

$$h_t = o_t \odot \tanh(c_t) \tag{1.8}$$

O ciclo é repetido para cada etapa de tempo, permitindo com que a rede LSTM aprenda a lembrar ou esquecer informações ao longo do tempo.

1.6.4 Funções de Ativação

As funções de ativação em uma rede neural são funções matemáticas capazes de serem aplicadas em um conjunto de dados na saída de uma camada da rede neural, permitindo assim introduzir a não linearidade na rede e conseqüentemente torná-la capaz de aprender e executar tarefas mais complexas. Alguns tipos de funções de ativação são:

- Linear;
- ReLU; e
- Sigmoid.

1.6.4.1 Linear

Segundo Sharma, Sharma e Athaiya (2017), a função de ativação linear é diretamente proporcional a sua entrada, e são ideais em casos que a interpretabilidade é necessária para tarefas simples. A função de ativação linear, pode ser representada pela equação 1.9 abaixo, onde o valor da variável x pode ser qualquer valor constante determinado pelo usuário.

$$f(x) = x \tag{1.9}$$

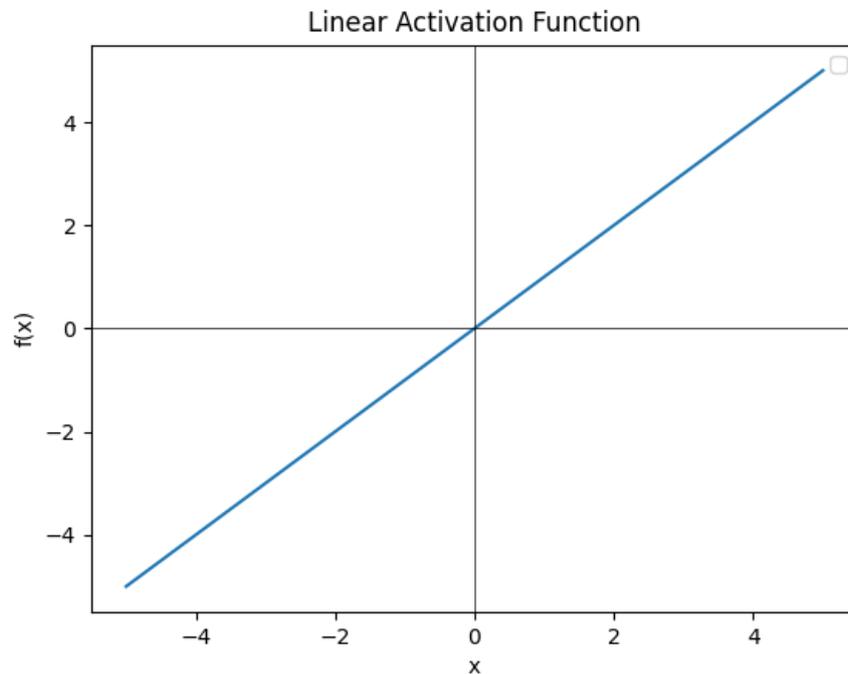
A imagem 4 a seguir representa a saída de uma função de ativação linear.

Sendo:

$f(x)$: O valor resultante da aplicação da função de ativação linear para um valor de entrada x .

x : O valor de entrada para a função linear, podendo ser qualquer número real.

Figura 4 – Função de Ativação Linear.



1.6.4.2 ReLU

De acordo com Agarap (2018), a função de ativação Rectified Linear Unit (ReLU) é uma função bastante utilizada em trabalhos de áreas biológicas e matemáticas. A função de ativação relu, pode ser representada pela equação 1.10 abaixo.

$$f(x) = \max(0, x) \quad (1.10)$$

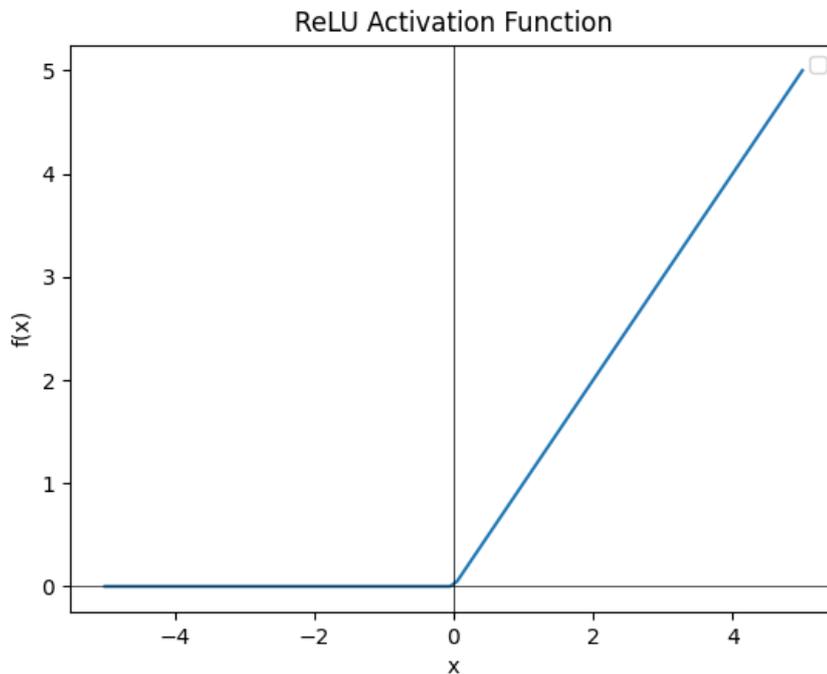
Sendo:

$f(x)$: O resultado da aplicação da função de ativação ReLU para um determinado valor de entrada x .

x : O valor de entrada para a função ReLU, podendo ser qualquer número real.

Este tipo de função de ativação executa uma operação de limite para cada valor de entrada, definindo como valor de saída 0 (zero) quando o valor é menor que 0 (zero), e em seguida produz uma inclinação linear de 1 (um) quando o valor é maior que 0 (zero). A imagem 5 a seguir representa a saída de uma função de ativação linear.

Figura 5 – Função de Ativação ReLU.



1.6.4.3 Sigmoide

Também conhecida como função logística, a função de ativação sigmoide é uma função matemática que realiza a transformação de qualquer valor real em um intervalo entre 0 e 1. Segundo Ray e Ray (2021), para valores relativamente menores, a função sigmoide retorna um valor próximo a 0 e para valores mais altos, retorna valores próximos a 1. Essa função de ativação pode ser representada pela equação 1.11 abaixo:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1.11)$$

Sendo:

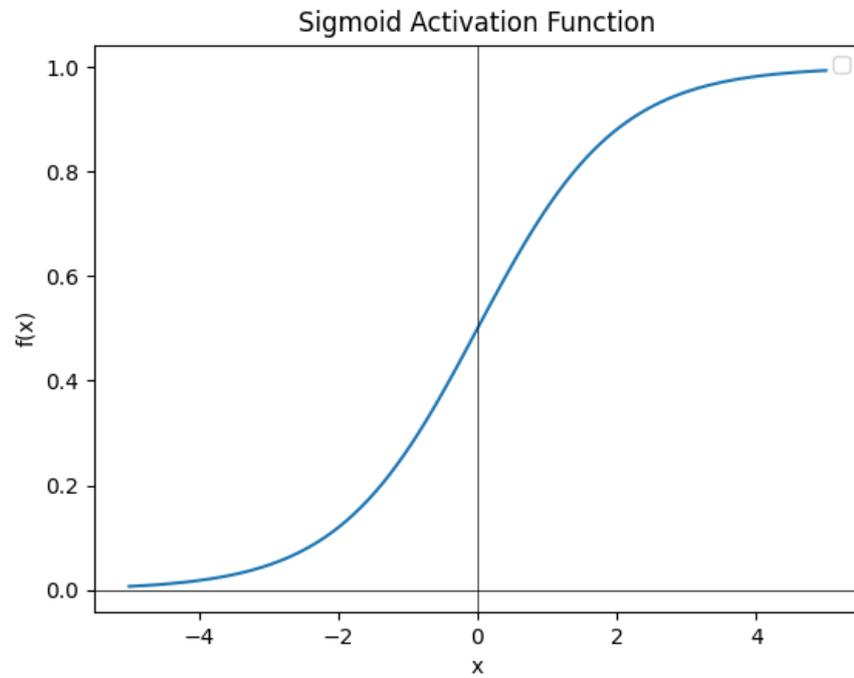
$\sigma(x)$: O resultado da aplicação da função sigmoide para um valor de entrada x .

x : O valor de entrada para a função sigmoide, podendo ser qualquer número real. Posteriormente a função sigmoide irá transformar este valor de entrada em um valor no intervalo $(0,1)$.

e : A base do logaritmo natural.

A imagem 6 seguir representa a saída de uma função de ativação sigmoide.

Figura 6 – Função de Ativação Sigmoide.



A função de ativação sigmoide é bastante utilizada em redes neurais para introduzir não linearidade.

Trabalhos Relacionados

Neste capítulo são apresentados alguns trabalhos que contribuíram com o desenvolvimento deste, dado que possuem propósitos semelhantes aos deste projeto.

2.1 Um método para estimar a entropia de séries temporais usando redes neurais artificiais

No trabalho de Velichko e Heidari (2021) é proposto um novo método para avaliação de complexidade de uma série temporal, baseado em cálculos diretos de entropia sem considerar ou realizar a aproximação das distribuições de probabilidade. O modelo de rede neural utilizado é denominado como LogNNet, desenvolvido pelos autores deste trabalho. O modelo foi projetado para reconhecer dígitos manuscritos retirados de um conjunto de dados públicos conhecido como MNIST-10, o qual possui um conjunto de treinamento de 60.000 imagens e um conjunto de teste de 10.000 imagens.

O modelo LogNNet é dividido em duas partes: uma atua como reservatório, garantindo que os dados de entrada sejam processados de maneira eficiente antes de serem enviados para a rede neural e transformando o vetor de entrada, enquanto a segunda parte é uma rede neural feedforward de camada única, convertendo os vetores em dígitos na camada de saída. O processo de funcionamento compreende desde a recepção dos dados da série temporal até o cálculo da entropia NNetEn.

Para realizar a validação do modelo, os autores empregaram mapas caóticos, mapas logísticos e séries temporais aleatórias, binárias e constantes. Após o processo de treinamento e teste, é calculada a precisão da classificação. A entropia NNetEn, é uma nova medida proposta, determinada pela divisão da precisão da classificação por 100%.

$$\text{NNetEn} = \frac{\text{Precisão da Classificação}}{100\%}$$

Cálculo da entropia NNetEn.

Ao contrário da precisão da classificação, que varia de 0 a 100%, a NNetEn é normalizada para um intervalo entre 0 e 1. Considerando assim a NNetEn como a nova medida de entropia, a qual possui valores entre 0 e 1, diferentemente da precisão da classificação que varia de 0 a 100%. Os testes foram realizados com 20, 100 e 400 épocas, e foi possível observar que a entropia NNetEn aumenta gradualmente com um número crescente de épocas, permitindo concluir que há uma melhora na capacidade de generalização e na representação dos dados conforme o modelo é treinado por mais épocas.

2.2 Detecção de padrões não lineares em sinais fisiológicos

No estudo conduzido por Radhakrishnan, Wilson e Hawk (2000), os autores propõem um método inovador capaz de detectar estruturas não lineares em sinais obtidos de sistemas dinâmicos, incluindo sistemas fisiológicos. O processo de aplicação do método inicia-se com o mapeamento das amostras de séries temporais discretas para um espaço de fase, utilizando a técnica de reconstrução por atrasos. O espaço de fase é uma representação multidimensional que identifica as relações dinâmicas entre as variáveis do sistema ao longo do tempo.

Posteriormente, as amostras selecionadas são segmentadas em *clusters* por meio da técnica *k-means* (k-médias). O objetivo principal é identificar diferenças significativas no volume de dispersão hiperesferoidal entre os dados originais e suas substituições simuladas. Para gerar conjuntos de dados simulados, emprega-se o algoritmo *Iterated Amplitude Adjusted Fourier Transform* (IAAFT), que realiza ajustes iterativos na transformada de Fourier para preservar propriedades estatísticas essenciais. Essa abordagem detalhada do método proporciona uma compreensão aprofundada das estruturas não lineares presentes nos dados, destacando as nuances entre as realizações nos conjuntos de séries temporais originais e as substitutas.

Os resultados obtidos neste estudo, são originários de diversas técnicas aplicadas durante os testes, como por exemplo: *Bleached Data*, *Uterine EMG Data*, *Laser Data* e *Cluster Number Effect*.

Para a técnica de *bleached data*, os pesquisadores analisaram dados gerados por sistemas caóticos e desenvolveram um método capaz de reconhecer padrões contidos

no conjunto de dados, mesmo após o processo de filtragem que normalmente causa distorções. No caso dos dados *uterine emg data*, que refletem as contrações uterinas durante o parto, a análise indicou que essas são melhor modeladas como processos não determinísticos, confirmando estudos anteriores que propõem a natureza não linear desse fenômeno fisiológico. Foram utilizados registros de eletromiografia (EMG) uterina de pacientes em trabalho de parto, com eletrodos posicionados na superfície abdominal.

A avaliação do método utilizando *laser data*, que representam séries temporais caóticas, apresentou uma clara capacidade do método proposto em distinguir entre dados originais e suas substituições simuladas. Os valores de sigma (S) obtidos, como $S = 6.426$ para os dados de laser, indicam a rejeição clara da hipótese nula, confirmando a eficácia do método. Além disso, ao explorar a sensibilidade ao número de *clusters*, os pesquisadores observaram que, ao aumentar o número de *clusters*, detalhes mais intrincados dos dados caóticos foram identificados.

2.3 Estimando a entropia do sistema com um modelo de rede neural recorrente profunda

Conduzido por Martinez-Garcia et al. (2019), este trabalho propõe uma abordagem capaz de aproximar a distribuição de probabilidade apresentada como resposta de um sistema, utilizando redes neurais profundas.

Os testes foram conduzidos em dois modelos distintos de Redes Neurais Artificiais (RNAs), um classificador e outro regressor. O modelo de regressão é composto por duas camadas recorrentes e uma camada de saída totalmente conectada. A mesma arquitetura foi aplicada ao modelo classificador, com exceção da camada de saída.

Para treinar esses modelos, foram utilizados registros de 10 motores movidos a gás industrial, distribuídos em diferentes regiões geográficas e sujeitos a diversas condições ambientais. Foram utilizados dados de 35 sensores, como potência do combustível e temperatura e pressão do ar comprimido. O conjunto de dados também incluiu registros de motores em condições comprometidas de funcionamento, com o propósito de validar o modelo proposto. A análise demonstrou que o modelo de rede neural profunda foi capaz de distinguir com precisão entre dados normais e anômalos, mesmo sob diferentes condições ambientais e de funcionamento. Em linhas gerais, o modelo demonstrou sua eficácia na detecção de comportamentos anormais em sistemas complexos, sugerindo sua robustez e aplicabilidade em diversos cenários industriais.

Objetivos

Este trabalho tem como objetivo principal, avaliar uma nova abordagem para estimar a entropia de séries temporais por meio de redes neurais. Sendo assim, buscamos responder as seguintes perguntas:

- As Redes Neurais Artificiais (RNA) são capazes de prever as taxas de produção de informação pelos sistemas, medidas pelas entropias, de maneira equivalente em séries temporais originais e em séries substitutas?
- Uma RNA treinada com séries substitutas, responde bem quando apresentadas séries fisiológicas ou caóticas?
- Existem variações no comportamento das RNAs quando se trata de séries temporais fisiológicas e séries caóticas simuladas?

Materiais e Métodos

Neste trabalho, propomos a utilização de um modelo de rede neural para realizar a aproximação da função de entropia amostral (*sample entropy*), tanto em dados de séries temporais originais, como em séries temporais substitutas. Descrevemos a base de dados empregada nesta pesquisa na seção 4.1, enquanto o processo de obtenção dos dados substitutos, é explicado na seção 4.2. Na seção 4.3, apresentamos a estrutura da rede neural, bem como seu funcionamento para realizar a aproximação desejada. Introduzimos o conceito de validação cruzada na seção 4.3.1, destacando sua importância na avaliação do desempenho do modelo proposto. Como métricas de avaliação dos resultados, foram utilizados o Erro Quadrático Médio (EQM) e a Raiz do Erro Quadrático Médio (REQM), descritos na seção 4.3.2.

4.1 Base de Dados

Os dados utilizados neste trabalho estão disponíveis na plataforma *Physionet.org*. De acordo com Goldberger et al. (2000). PhysioNet - Recurso de Pesquisa para Sinais Fisiológicos Complexos, foi criado em 1999 e apoiado pelos Institutos Nacionais de Saúde (NIH). As missões originais e contínuas do PhysioNet Resource eram conduzir e catalisar a pesquisa e educação biomédica, em parte oferecendo acesso gratuito a grandes coleções de dados fisiológicos e clínicos e software de código aberto.

Em cooperação com a conferência anual *Computing in Cardiology*, a PhysioNet também organiza uma série anual de desafios, concentrando a pesquisa em problemas não resolvidos na ciência clínica e básica. Os membros da equipe da PhysioNet estão ativamente envolvidos em trabalhos inovadores na análise de sinais fisiológicos, tanto da perspectiva básica quanto da translacional.

Para o desenvolvimento deste trabalho, consideramos utilizar as seguintes bases de dados:

- Normal Sinus Rhythm RR Interval Database (NSR2DB);
- Long Term Atrial Fibrillation Database (LTAfDB);
- Long Term Segment ST Database (LTSTDB);
- Congestive Heart Failure RR Interval Database (CHF2DB); e
- Severe Congestive Heart Failure Database (CHFDB).

4.1.1 NSR2DB - Normal Sinus Rhythm RR Interval Database

Inclui arquivos de batimentos cardíacos de 54 gravações de eletrocardiograma (ECG), de longo prazo de indivíduos em ritmo sinusal normal. Sendo 30 homens com idades entre 28 e 76 anos, e 24 mulheres com idades entre 58 e 73 anos. As gravações originais de ECG (não disponíveis) foram digitalizadas em 128 amostras por segundo, e as anotações dos batimentos foram obtidas por análise automatizada com revisão e correção manuais.

4.1.2 LTAfDB - Long Term Atrial Fibrillation Database

Possui 84 registros de eletrocardiograma (ECG) de longo prazo de indivíduos com fibrilação atrial (FA) paroxística ou sustentada. Cada registro contém dois sinais de ECG gravados simultaneamente digitalizados a 128 Hz com resolução de 12 bits em uma faixa de 20 mV; as durações dos registros variam, mas geralmente são de 24 a 25 horas.

4.1.3 LTSTDB - Long Term Segment ST Database

Contém gravações de eletrocardiograma (ECG) de 86 indivíduos, escolhidos para exibir uma variedade de eventos de alterações do segmento ST, incluindo episódios de ST isquêmicos, episódios de ST não isquêmicos relacionados ao eixo, episódios de desvio lento do nível de ST e episódios contendo misturas desses fenômenos. As gravações contidas nesta base possuem um prazo entre 21 e 24 horas de duração e contêm dois ou três sinais de ECG.

4.1.4 CHF2DB - Congestive Heart Failure RR Interval Database

Contém arquivos de anotação de batimentos para 29 gravações de eletrocardiograma (ECG) de longo prazo de indivíduos com idade entre 34 e 79 anos, com insuficiência cardíaca congestiva (classes I, II e III da NYHA). Foram incluídos dados de 8 homens e 2 mulheres, o restante dos dados não possui identificação de gênero. Os registros originais de ECG (não disponíveis) foram digitalizados a 128 amostras por segundo e as anotações dos batimentos foram obtidas por análise automatizada com revisão e correção manual.

4.1.5 CHFDB - Severe Congestive Heart Failure Database

É composto por registros de eletrocardiograma (ECG) de longo prazo de 15 indivíduos, sendo 11 homens, com idades entre 22 e 71 anos e 4 mulheres, com idades entre 54 e 63 anos com insuficiência cardíaca congestiva grave (classes III e IV da NYHA). A gravação individual dos dados possui em média 20 horas de duração contêm dois sinais de ECG. As gravações analógicas originais foram feitas no Hospital Beth Israel de Boston (agora o Beth Israel Deaconess Medical Center).

4.2 Seleção do Conjunto de Dados

Para o desenvolvimento deste trabalho, foram selecionadas 13 colunas e 15.839 linhas de dados de cada um dos conjuntos mencionados na seção 4.1, o qual denominamos conjunto de dados originais. Os dados foram selecionados aleatoriamente e incluem o cálculo do intervalo RR conforme o seu tamanho e o tempo de coleta.

Posteriormente, foram gerados os conjuntos de dados substitutos para cada um dos cinco conjuntos descritos anteriormente. Esses dados foram produzidos por meio de um algoritmo iAAFT desenvolvido em Python, capaz de gerar séries temporais mantendo o espectro e as características estatísticas da série original de entrada.

A imagem 7 a seguir, apresenta o funcionamento do algoritmo iAAFT.

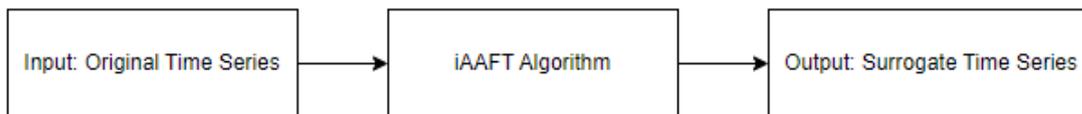


Figura 7 – Processo de Geração de Dados Substitutos

O algoritmo recebe como entrada um arquivo no formato CSV, contendo os conjuntos de séries originais, posteriormente executa e aplica as etapas descritas anteriormente para geração de dados substitutos. Como saída, o algoritmo devolve um arquivo no formato CSV, contendo o conjunto de dados substitutos.

A imagem 8 abaixo, apresenta cinco amostras do conjunto de dados original NSR2DB (4.1.1) e a imagem 9, apresenta cinco amostras do conjunto de dados substituto, gerado a partir do algoritmo iAAFT.

Figura 8 – Amostras do Conjunto de Dados Originais

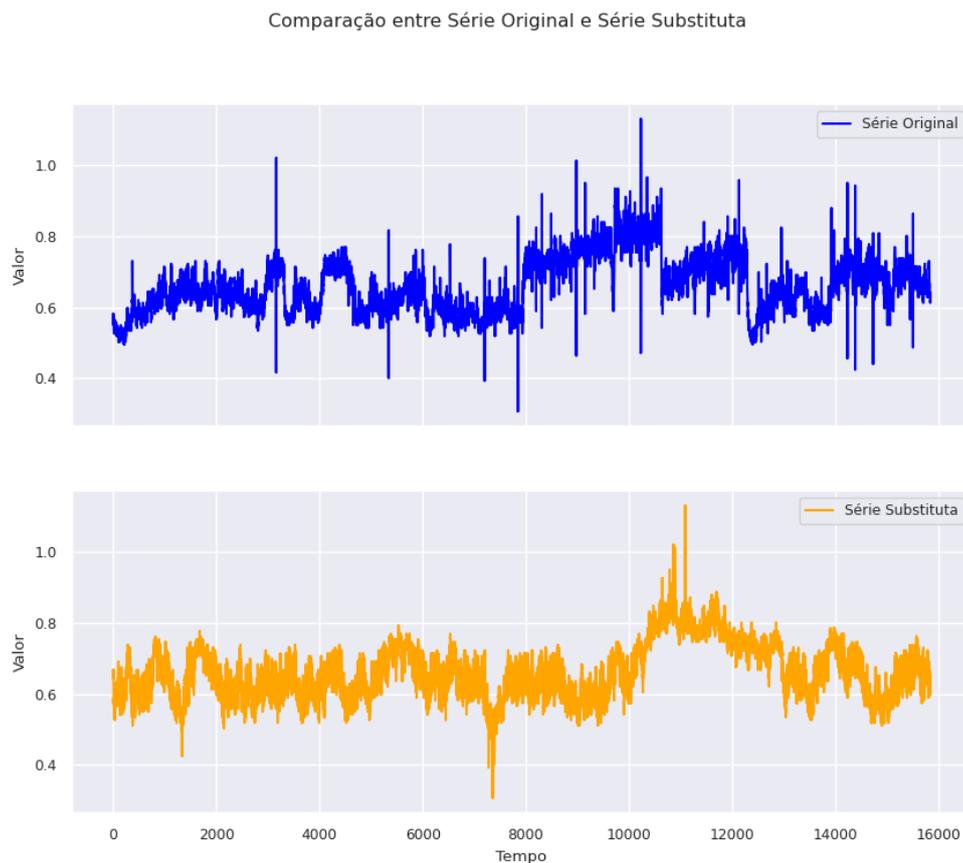
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0 | 0.581548 | 0.785936 | 0.943814 | 0.924614 | 0.503412 | 0.940017 | 0.723222 | 0.897483 | 0.806894 | 0.738723 | 0.759771 | 0.519133 | 0.456473 |
| 1 | 0.565830 | 0.778077 | 0.967607 | 0.955957 | 0.495546 | 0.924218 | 0.715274 | 0.944719 | 0.830396 | 0.730864 | 0.751857 | 0.519133 | 0.432862 |
| 2 | 0.581548 | 0.770218 | 0.983470 | 1.002971 | 0.495546 | 0.916319 | 0.707327 | 0.991955 | 0.830396 | 0.738723 | 0.767685 | 0.519133 | 0.448602 |
| 3 | 0.565830 | 0.754499 | 0.951745 | 0.924614 | 0.503412 | 0.876822 | 0.707327 | 0.991955 | 0.853898 | 0.746582 | 0.751857 | 0.526999 | 0.448602 |
| 4 | 0.581548 | 0.746639 | 0.943814 | 0.877600 | 0.503412 | 0.829427 | 0.715274 | 1.015573 | 0.838230 | 0.723006 | 0.759771 | 0.526999 | 0.448602 |

Figura 9 – Amostras do Conjunto de Dados Substitutos

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0 | 0.667994 | 0.589452 | 0.927951 | 0.846257 | 0.448351 | 0.742534 | 0.731169 | 0.669176 | 0.806894 | 0.778017 | 0.728114 | 0.652850 | 0.550915 |
| 1 | 0.660135 | 0.597312 | 0.912089 | 0.846257 | 0.448351 | 0.710937 | 0.739117 | 0.684921 | 0.806894 | 0.809452 | 0.807257 | 0.652850 | 0.550915 |
| 2 | 0.660135 | 0.597312 | 0.880364 | 0.877600 | 0.432619 | 0.718836 | 0.770907 | 0.700667 | 0.900901 | 0.880181 | 0.759771 | 0.660715 | 0.550915 |
| 3 | 0.660135 | 0.605171 | 0.888295 | 0.854093 | 0.432619 | 0.687239 | 0.731169 | 0.692794 | 0.877400 | 0.872322 | 0.751857 | 0.637118 | 0.543045 |
| 4 | 0.667994 | 0.589452 | 0.872433 | 0.924614 | 0.456217 | 0.710937 | 0.739117 | 0.692794 | 0.916569 | 0.895898 | 0.775600 | 0.660715 | 0.558785 |

Ao observarmos as imagens acima, é notável a diferença de valores em questões de ordenação, no entanto, para exemplificar melhor, a imagem 10 a seguir, apresenta uma comparação entre a primeira coluna de dados de cada um dos conjuntos (originais e substitutos).

Figura 10 – Comparação Série Original x Série Substituta



Analisando a imagem anterior, fica nítido o entendimento e a compreensão dos conjuntos de dados, bem como uma visualização facilitada de uma série substituta gerada, que possui propriedades estatísticas semelhantes as da série original. É válido ressaltar que o conceito de dados substitutos, bem como o algoritmo iAAFT são descritos na seção 1.5

4.3 Arquitetura da Rede Neural

No decorrer deste estudo, conduzimos experimentos utilizando duas arquiteturas distintas de redes neurais. A primeira configuração envolve uma rede neural com uma única camada intermediária/oculta, enquanto a segunda conta com uma arquitetura de duas camadas intermediárias/ocultas, ambas do tipo LSTM. As imagens 11 e 12 a seguir apresentam a topologia destas redes neurais.

Figura 11 – Rede Neural Artificial - Uma Camada Intermediária

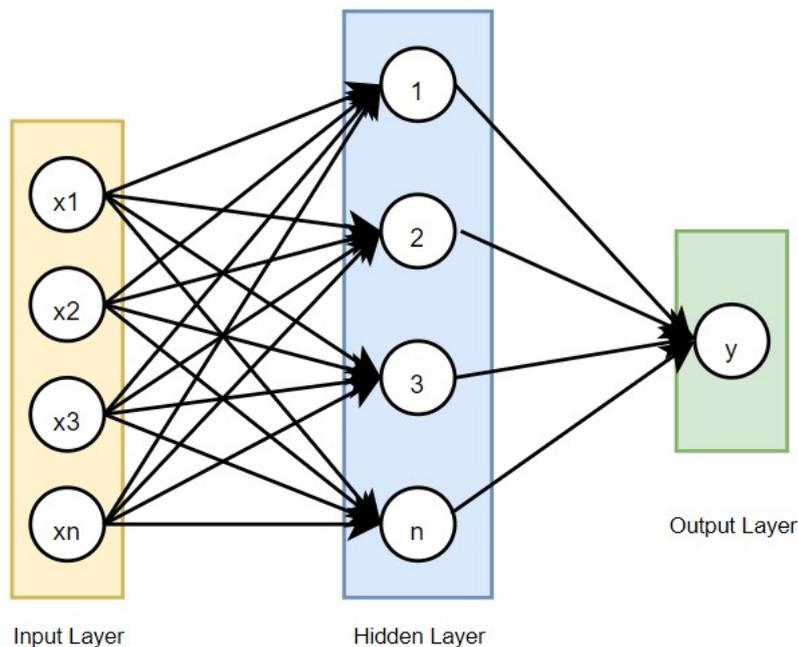
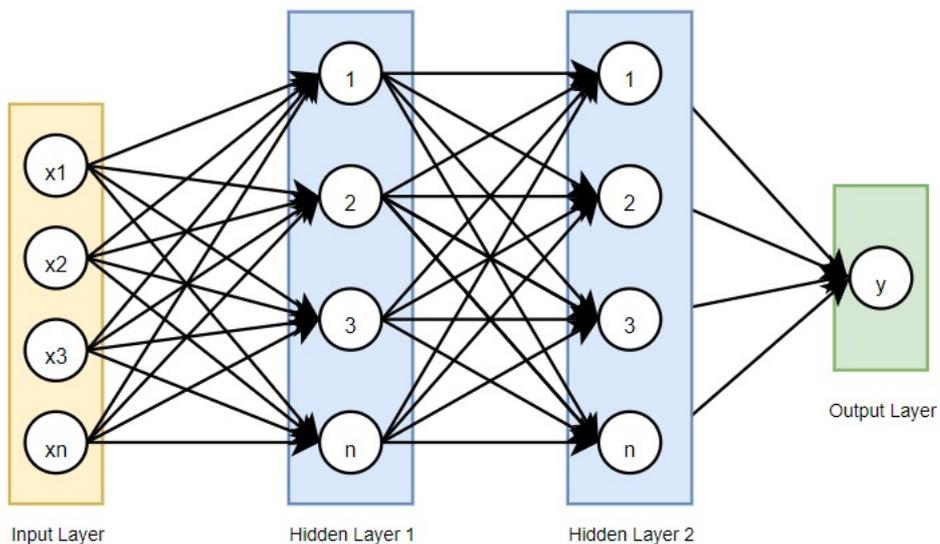


Figura 12 – Rede Neural Artificial - Duas Camadas Intermediárias



A estrutura da primeira rede trata-se de um modelo sequencial do tipo LSTM (*Long Short-Term Memory*), implementado usando TensorFlow e Keras. O modelo é composto por uma camada de entrada com um número de nós é dependente do número de características do conjunto de dados, uma camada intermediária/oculta com 50 unidades e uma camada de saída com apenas 1 unidade. A rede utiliza o otimizador Adam e a função de perda *Mean Squared Error* (MSE). O número de nós na camada de entrada é dependente do tamanho do sinal e do número de colunas do arquivo de dados. Durante os experimentos, ocorreu a variação da quantidade de unidades por camadas, com valores entre 30 e 60 unidades, onde as redes com 50 unidades performaram melhor se comparadas as outras.

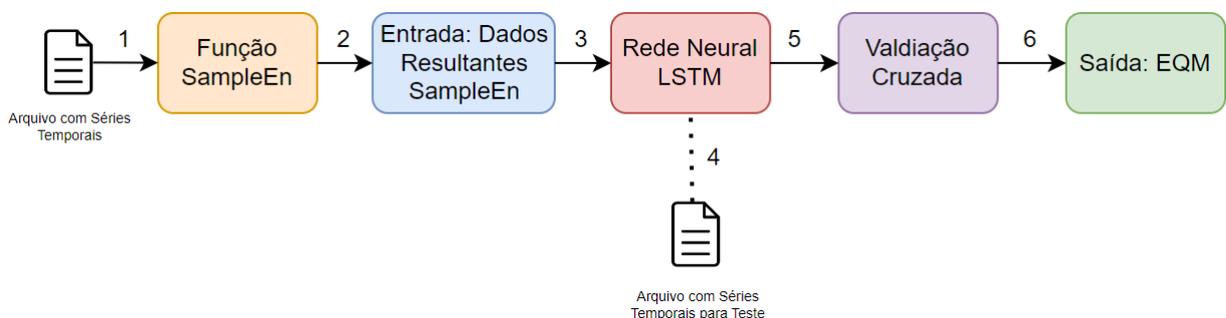
Com o intuito de investigar a possibilidade de obtenção de melhores resultados, estruturamos o segundo modelo de rede neural, o qual consideramos a mesma arquitetura anterior, no entanto com o acréscimo de uma segunda camada intermediária/oculta, com 50 unidades. Consideramos o *adam* como otimizador da rede neural e o erro quadrático médio para função de perda, frequentemente utilizado em redes neurais com abordagem em regressão.

Para garantir a validação eficaz do desempenho da rede neural, adotamos o uso da validação cruzada, na qual definimos o valor de $k = 5$. Uma explicação mais aprofundada será apresentada na seção 4.3.1.

Ao longo das fases de treinamento e teste da rede neural, alternamos entre diferentes funções de ativação, incluindo linear, relu e sigmoide. Além disso, realizamos testes com diferentes configurações de épocas de treinamento, abrangendo 100, 500 e 1000 épocas.

A imagem 13 a seguir, apresenta o funcionamento dessa rede neural.

Figura 13 – Funcionamento da Rede Neural



No primeiro passo, ocorre o carregamento do arquivo contendo as séries temporais para treinamento da rede neural, em sequência, esses são enviados para a função *sample entropy*, que por sua vez calcula a entropia amostral para cada série temporal de forma individual. Os parâmetros *data*, *m* e *r* da função, receberam os seguintes valores

respectivamente:

- *data*: As séries temporais do conjunto de dados;
- *m*: O valor 2 para o parâmetro da entropia; e
- *r*: O desvio padrão da série multiplicado por 0.2.

Os valores dos parâmetros *m* e *r*, foram mantidos conforme a documentação da função *sample entropy* para a linguagem de programação *python*.

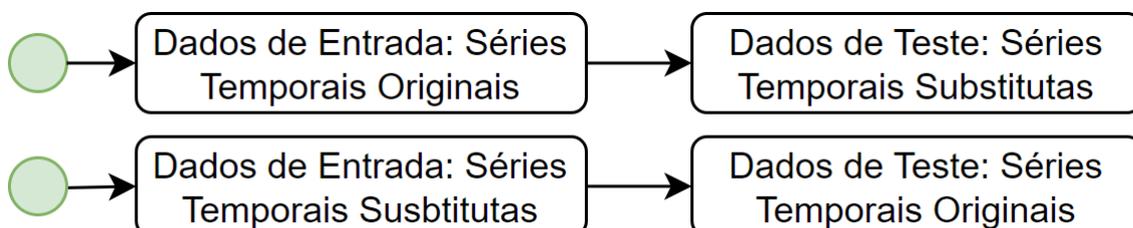
Após a execução da função *sample entropy*, os resultados são concatenados aos dados de entrada, dando origem a um novo conjunto de dados, contendo as medidas de complexidade obtidas pela função de entropia amostral.

O próximo passo, consiste na recepção desse conjunto de dados como entrada para a rede neural, onde esta receberá um outro conjunto de dados de séries temporais para realização dos testes. Tendo então o arquivo de entrada e o arquivo de testes a rede é executada e realiza o processo de treinamento e teste, juntamente com o processo de validação cruzada.

Finalmente, como resultado do processo, calculamos o erro quadrático médio (EQM) e, em seguida, extraímos a raiz quadrada desse valor, utilizando a métrica de avaliação conhecida como Raiz do Erro Quadrático Médio (REQM), descrita na seção 4.3.2.

Neste estudo consideramos dois tipos distintos de treinamento da rede neural. O primeiro, utilizando como dados de entrada as séries temporais originais e para realização dos testes, utilizamos os dados das séries temporais substitutas geradas por meio do algoritmo iAAFT. No segundo tipo de treinamento, seguimos a abordagem inversa, onde utilizamos as séries temporais substitutas como dados de entrada e as séries temporais originais como dados de teste. Os tipos de testes descritos, estão representados na imagem 14 abaixo.

Figura 14 – Tipos de Treinamento da Rede Neural



4.3.1 Validação Cruzada

De acordo com Berrar (2019) a cross-validation ou validação cruzada em uma tradução livre, é um método de reamostragem de dados para avaliar a capacidade de generalização de modelos preditivos. Segundo Browne (2000) a validação cruzada envolve técnicas que particionam um conjunto de dados e geram modelos de forma repetida, a fim de realizar testes preditivos. Para Koul, Becchio e Cavallo (2018) a divisão de dados em subconjuntos de treinamento e teste podem ser realizadas utilizando diferentes métodos, e.g.: *holdout cross-validation* e *k-fold cross-validation*.

O *holdout* é o tipo de validação cruzada mais simples, este método divide o conjunto de dados fornecidos apenas uma vez em um conjunto de dados de treinamento e um conjunto de teste. O conjunto de dados é selecionado de forma aleatória e a parte restante dos dados é atribuída ao conjunto de testes.

O *k-fold* realiza a validação dividindo o conjunto de dados em **k** subconjuntos (ou dobras). Durante cada iteração, uma dessas divisões é selecionada como conjunto de teste, enquanto os outros **k-1** subconjuntos são utilizados para o treinamento do modelo. Este procedimento é realizado por **k** vezes, de acordo com o valor definido pelo usuário. Neste caso, a oscilação da estimativa é reduzida conforme o valor de **k** é aumentado. A imagem 15 a seguir apresenta um exemplo do modelo de validação cruzada *k-fold*.

Figura 15 – Representação da Técnica de Validação Cruzada

| | | | | | | |
|-------------------------------|-------------|-----------|-----------|-----------|-----------|-----------|
| Número de Iterações (K-Folds) | 1º Iteração | Validação | Treino | Treino | Treino | Treino |
| | 2º Iteração | Treino | Validação | Treino | Treino | Treino |
| | 3º Iteração | Treino | Treino | Validação | Treino | Treino |
| | 4º Iteração | Treino | Treino | Treino | Validação | Treino |
| | 5º Iteração | Treino | Treino | Treino | Treino | Validação |

4.3.2 EQM e REQM

De acordo com Wang e Bovik (2009) podemos considerar o *Mean Squared Error* (MSE) ou Erro Quadrático Médio (EQM), como uma medida de fidelidade de sinal, a qual possui

como finalidade comparar dois sinais fornecendo uma pontuação capaz de descrever o grau de semelhança/fidelidade ou o nível de erro/distorção entre eles.

O erro quadrático médio é calculado como a média dos quadrados das diferenças entre os valores do sinal estimado e os valores reais do sinal de referência, sendo bastante utilizado na verificação de acurácia de modelos, podendo ser representado pela equação 4.1 abaixo.

$$EQM = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.1)$$

Equação EQM - Erro Quadrático Médio

As variáveis que compõem a equação 4.1, estão descritas a seguir:

N : Representa o número total de observações no conjunto de dados.

$\sum_{i=1}^N$: Soma onde o valor de i varia de 1 a N , abrangendo todas as observações.

y_i : Valores reais do sinal de referência.

\hat{y}_i : Valores estimados pelo modelo.

$(y_i - \hat{y}_i)^2$: Quadrado da diferença entre o valor real e o valor estimado para cada observação.

$\frac{1}{N} \sum_{i=1}^N$: Média dos quadrados da diferença, dividindo a soma pelo número total de observações.

A Root Mean Squared Error (RMSE) ou Raiz Quadrada do Erro Médio (REQM), é uma medida padrão para estimar o erro de um modelo na previsão de dados quantitativos, sendo bastante utilizada em séries temporais por ser mais sensível a erros maiores. A REQM pode ser representada pela equação 4.2 abaixo.

$$REQM = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (4.2)$$

As variáveis que compõem a equação 4.2, são iguais as descritas na equação 4.1, no entanto, com o acréscimo da raiz quadrada do erro médio, representada por: $RMSE = \sqrt{MSE}$.

Resultados e Discussão

Neste capítulo, é apresentado o processo completo de obtenção dos resultados deste trabalho, onde realizamos diversas abordagens visando atingir um desempenho otimizado. Os resultados aqui apresentados foram obtidos através de testes utilizando a rede neural descrita em 4.3, nos quais variamos funções de ativação, números de épocas e números de camadas. Além disso, utilizamos um conjunto de dados agrupados para treinamento da rede neural. Essas abordagens possibilitaram a exploração de diferentes configurações e a identificação daquelas que produzem os melhores resultados.

5.1 Experimentos

Para obtenção dos resultados, realizamos os seguintes testes:

- Teste com uma camada intermediária;
- Teste com duas camadas intermediárias;
- Teste com uma camada intermediária e variação na função de ativação; e
- Teste com uma camada intermediária e dados agrupados.

5.1.1 Teste com uma camada intermediária

Como experimento inicial, após o processo de obtenção do conjunto de dados substitutos (4.2), a etapa de treinamento da rede neural foi conduzida da seguinte maneira: utilizamos a rede neural com apenas uma camada intermediária/oculta, descrita em 4.3, testes para as três funções de ativação, linear, relu e sigmoide. Além disso realizamos o treinamento da rede neural com os dados originais e o teste com os dados substitutos e posteriormente o inverso.

Os resultados para cada conjunto e as respectivas funções de ativação linear, relu e sigmoide, são apresentados nas figuras a seguir. Os valores indicados com **STO**, são as séries temporais originais, enquanto os valores indicados com **STS**, são as séries temporais substitutas.

Figura 16 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

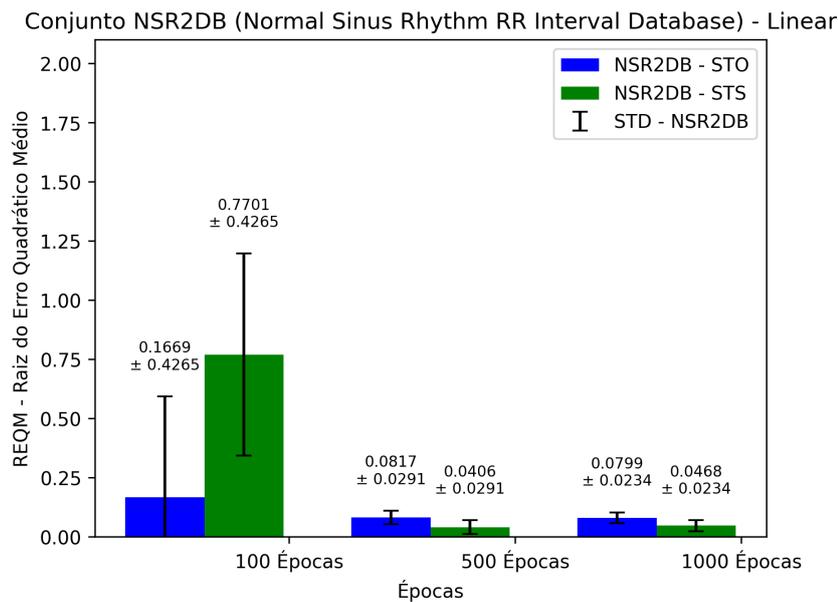


Figura 17 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu

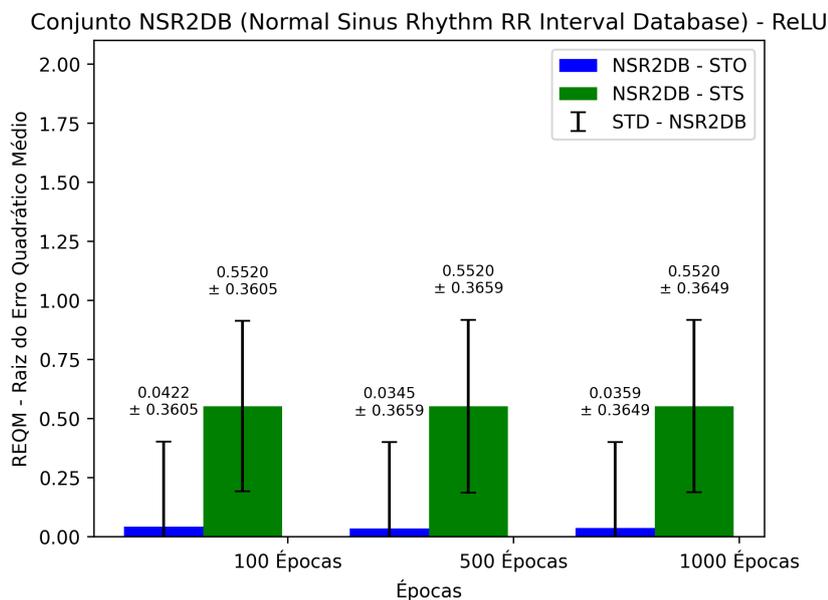
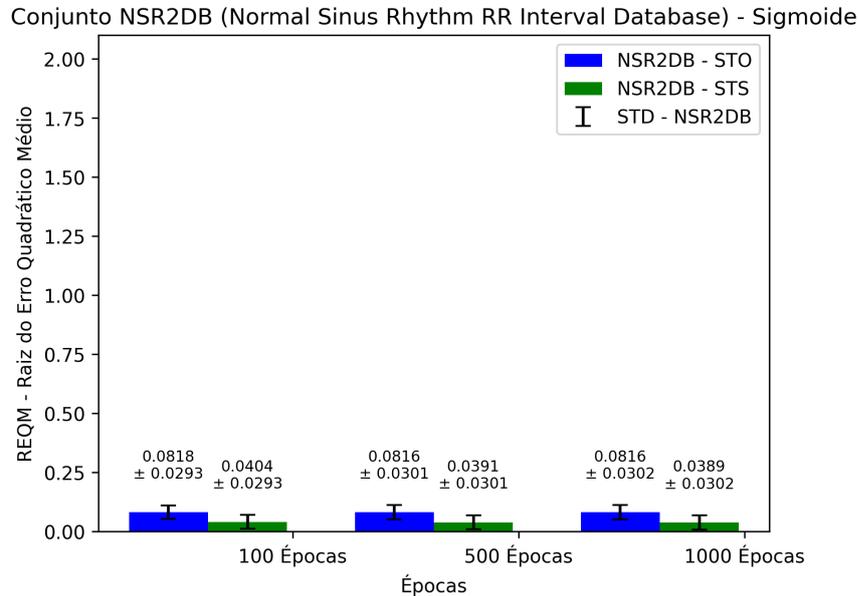


Figura 18 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Ao analisarmos os resultados do primeiro teste para o conjunto NSR2DB, para a função de ativação linear, é possível identificar uma queda significativa no valor da raiz do erro quadrático médio (REQM) ao longo do aumento do número de épocas, partindo de 0.1669 para 0.0799. Esse padrão foi consistente para ambas as séries temporais, tanto as originais (STO) quanto as substitutas (STS).

Na função de ativação RELU, ocorre uma redução no valor REQM para as séries temporais originais nas 1000 épocas, enquanto, para as substitutas, o REQM permanece constante.

Já para a função de ativação sigmoide, os valores permanecem constantes, com pequenas variações para as séries temporais substitutas.

O próximo conjunto de dados a ser analisado é o LTAFDB, os resultados são apresentados nas imagens a seguir.

Figura 19 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

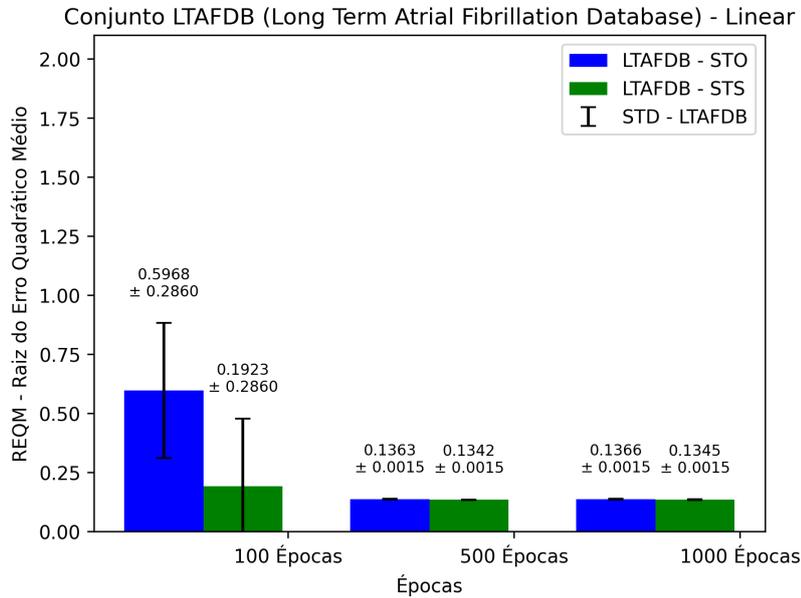


Figura 20 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

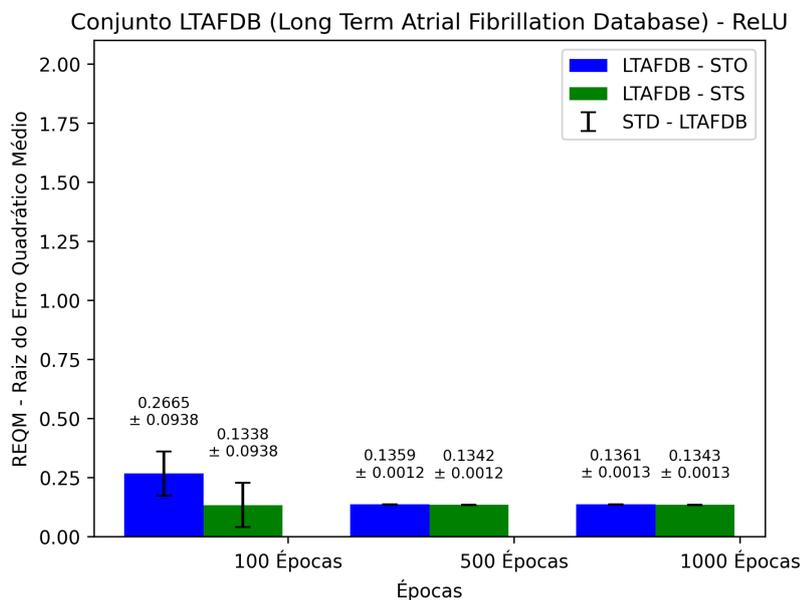
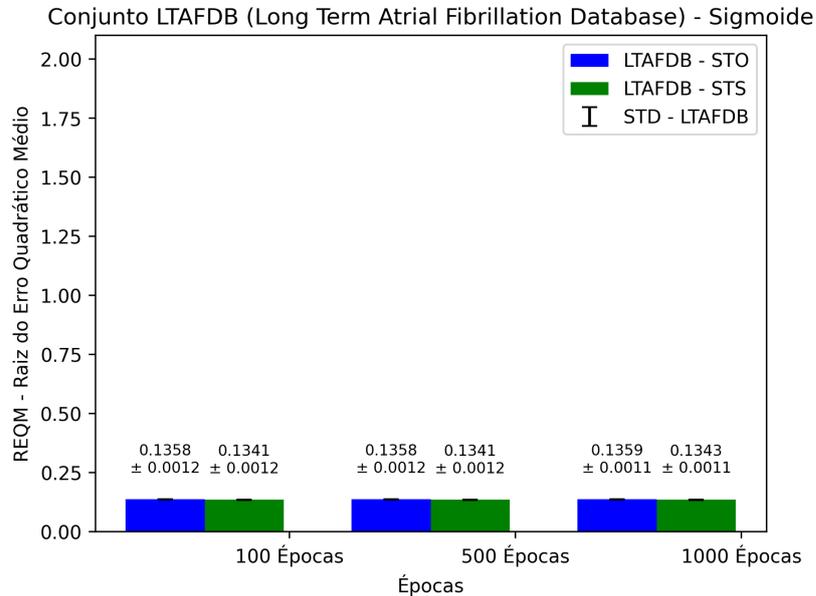


Figura 21 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Observando os resultados deste conjunto, destaca-se uma melhora progressiva nos resultados com o aumento do número de épocas para a função de ativação linear. Isso é observado tanto para os conjuntos originais quanto para os substitutos. A partir das 1000 épocas, é notável um aumento mínimo no REQM para ambos os conjuntos.

Analisando os resultados da função de ativação relu, verifica-se uma melhora nos resultados a partir das 500 épocas para as séries originais, com um leve aumento nas 1000 épocas. Para as séries substitutas, o melhor valor é alcançado nas 100 épocas, com um aumento gradual nos resultados para 500 e 1000 épocas.

A função de ativação sigmoide mantém-se constante nos valores entre 100 e 500 épocas, com um aumento sutil nos resultados para ambos os conjuntos de dados nas 1000 épocas de treinamento.

Os próximos resultados a serem apresentados, são do conjunto LTSTDB.

Figura 22 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

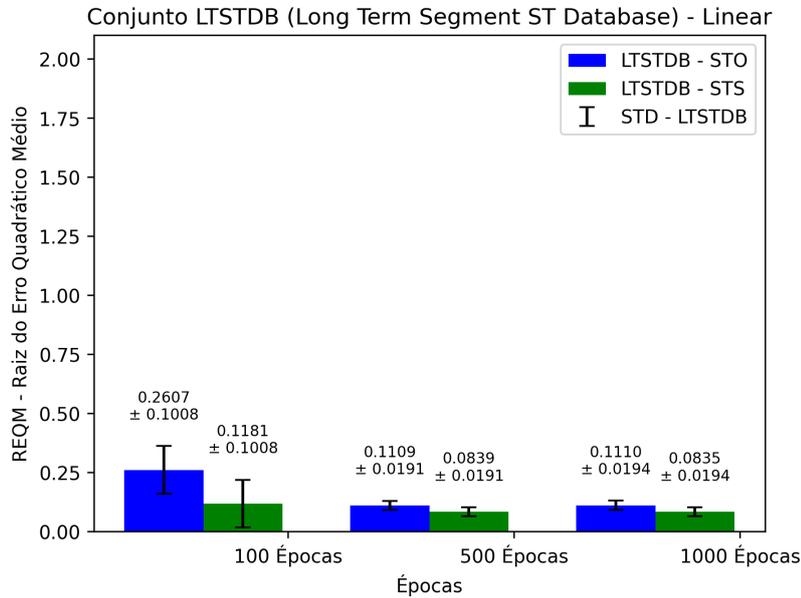


Figura 23 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

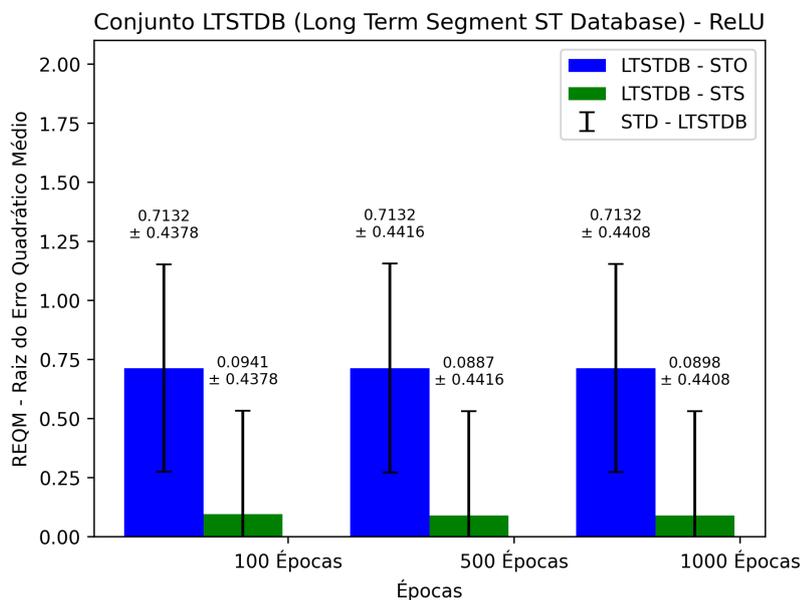
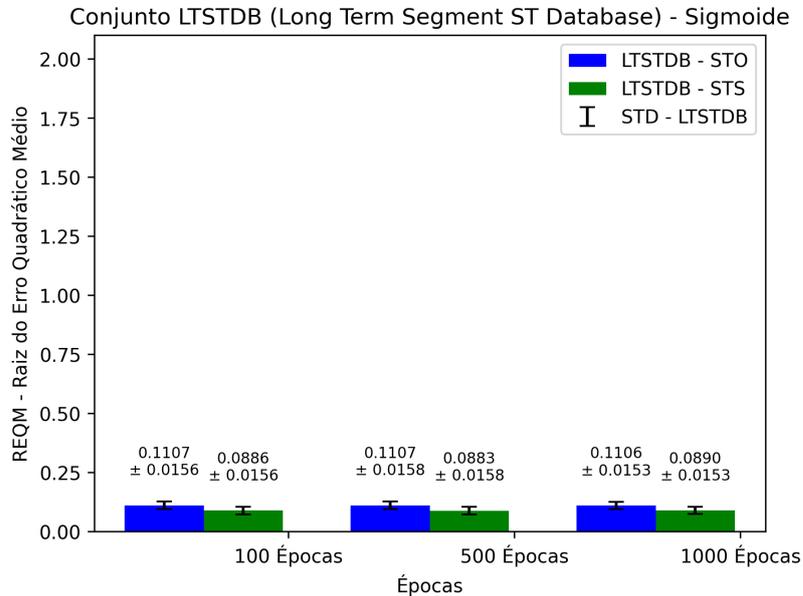


Figura 24 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Para o conjunto LTSTDB, a análise dos resultados indica que a função de ativação linear atingiu o melhor REQM para o conjunto de séries temporais originais nas 500 épocas de treinamento. Entretanto, para o conjunto de séries substitutas, o desempenho máximo foi observado nas 1000 épocas de treinamento. Comparando as séries originais, houve um leve aumento do REQM de 500 para 1000 épocas, passando de 0.1109 para 0.1110.

No caso da função de ativação relu, os resultados mantiveram-se constantes para as séries originais, enquanto para as séries substitutas houve uma redução do valor nas 500 épocas e um aumento do valor nas 1000 épocas.

A função de ativação sigmoide apresentou resultados constantes até as 500 épocas, com uma leve redução de 0.1107 para 0.1106 nas 1000 épocas de treinamento, para o conjunto de séries temporais originais. Já para as séries substitutas, a melhor REQM foi obtida nas 500 épocas, com uma leve alteração do valor nas 1000 épocas, passando de 0.0883 para 0.0890.

A próxima análise, refere-se ao conjunto CHF2DB.

Figura 25 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

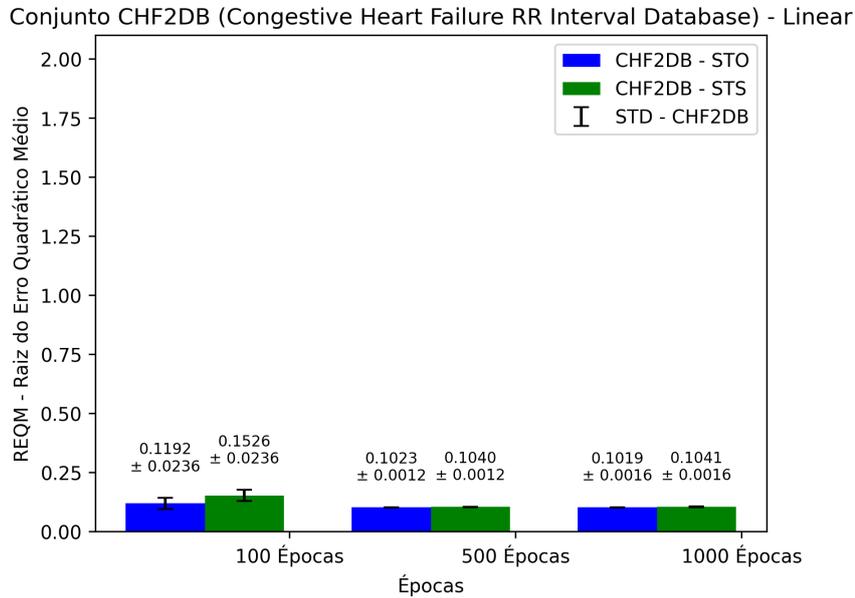


Figura 26 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

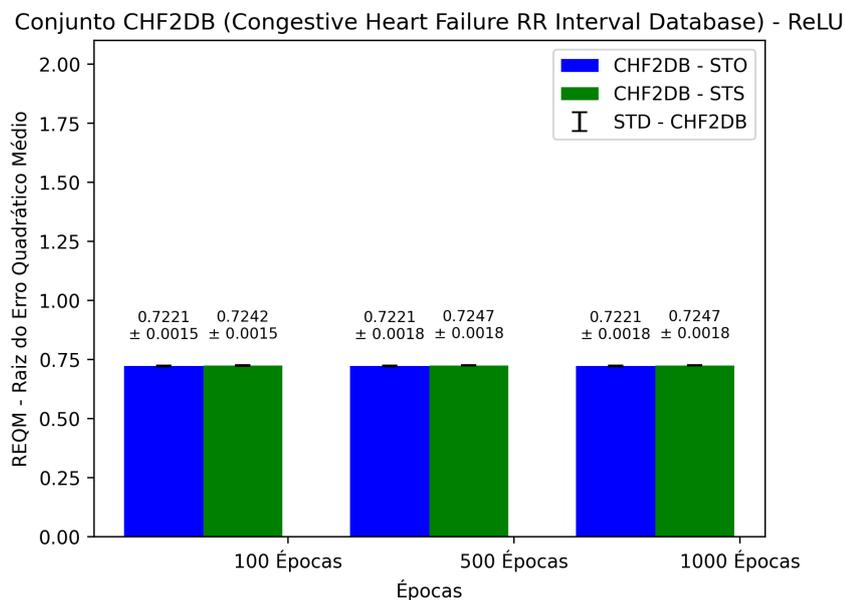
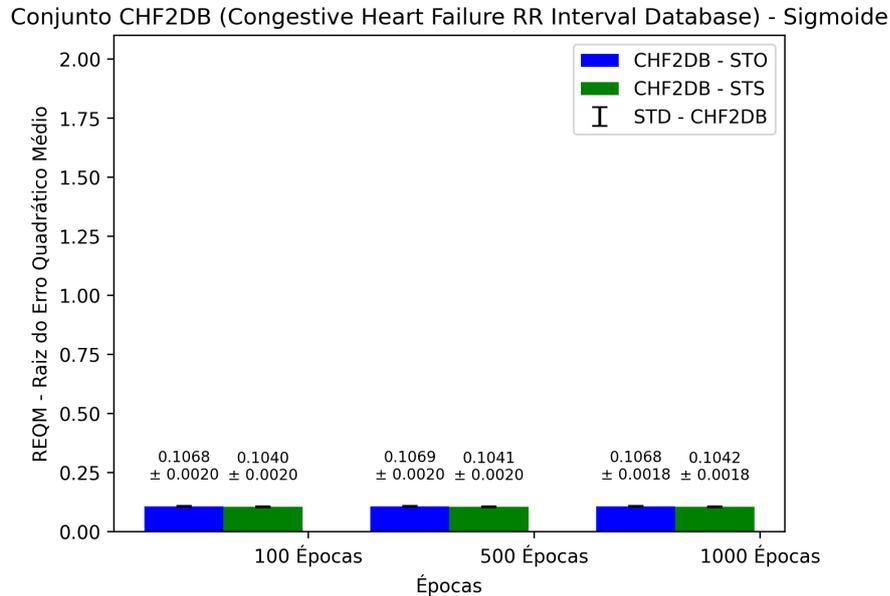


Figura 27 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



No conjunto CHF2DB, a análise dos resultados mostra que a função de ativação linear proporcionou uma redução significativa da REQM nas 1000 épocas de treinamento para o conjunto de séries temporais originais, passando de 0.1192 para 0.1019. Entretanto, para as séries substitutas, o desempenho máximo foi observado com 500 épocas de treinamento, com um aumento muito sutil de 0.1040 para 0.1041 nas 1000 épocas subsequentes.

A função de ativação relu manteve-se constante em seu resultado para as séries originais e também para as séries substitutas, com um leve aumento do valor nas 500 épocas de treinamento, passando de 0.7242 para 0.7247.

A função de ativação sigmoide apresentou resultados consistentes tanto para as séries temporais originais quanto para as substitutas, com um leve aumento nas 500 épocas de treinamento para as séries originais e uma redução nas 1000 épocas. Para as séries substitutas, houve um leve aumento nas 500 épocas.

Os próximos resultados a serem apresentados, são do conjunto CHFDB.

Figura 28 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

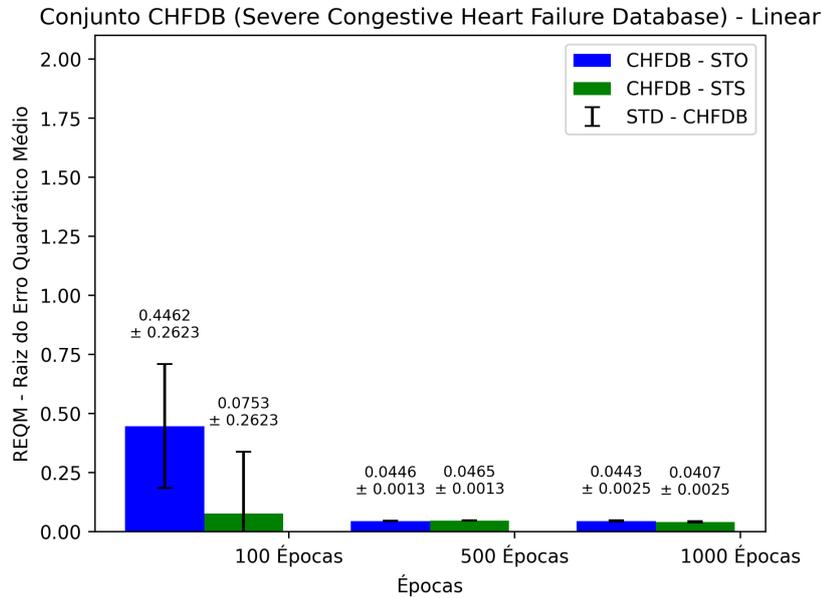


Figura 29 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

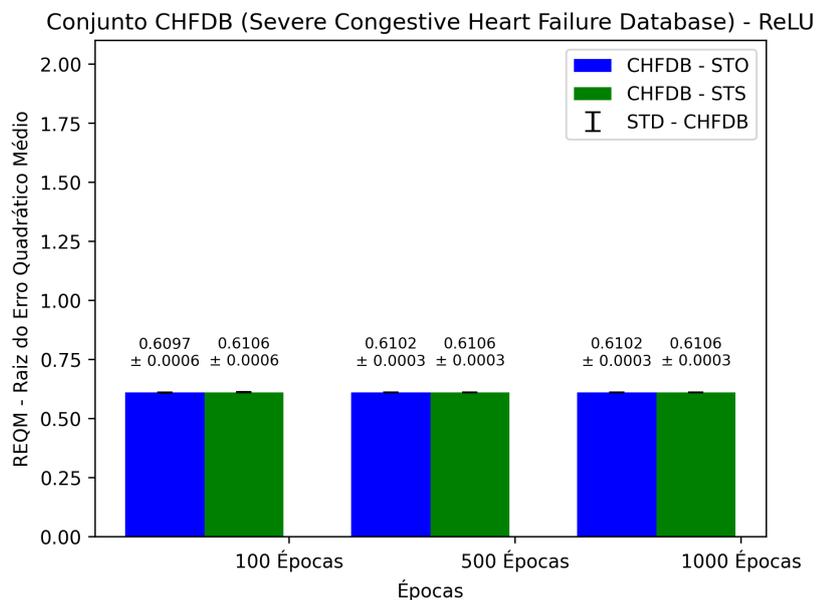
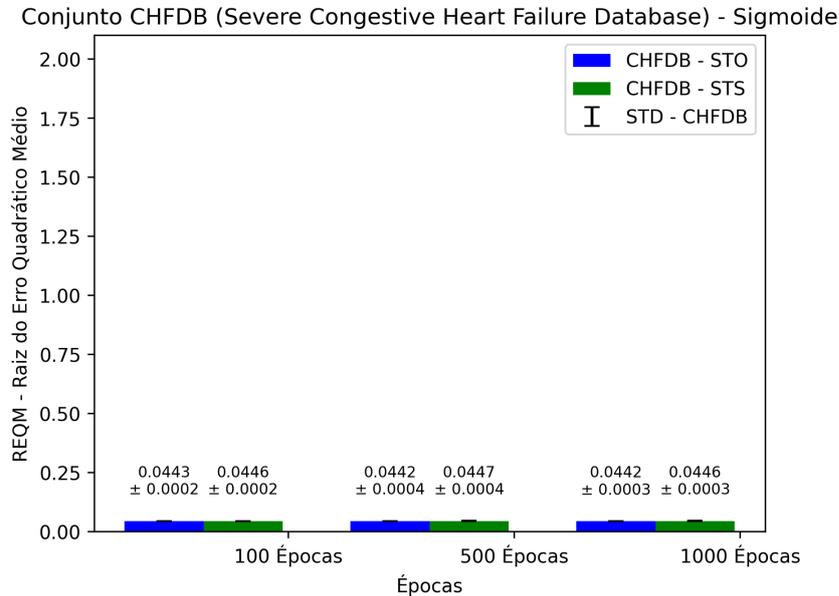


Figura 30 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



No contexto deste conjunto, a função de ativação linear demonstrou desempenho positivo para ambos os conjuntos de dados (originais e substitutos) a partir das 1000 épocas de treinamento.

Embora a função de ativação relu tenha apresentado um resultado discrepante em comparação com a função de ativação linear, demonstrou consistência nos resultados para ambos os conjuntos de séries temporais ao longo das diferentes épocas de treinamento.

Os resultados da função de ativação sigmoide também permaneceram relativamente estáveis para ambos os conjuntos, sem variações significativas ao longo do treinamento.

5.1.2 Teste com duas camadas intermediárias

Ao revisar os resultados do teste inicial apresentado anteriormente, observa-se que os desempenhos não foram completamente satisfatórios para alguns conjuntos de dados, havendo variações notáveis tanto entre séries originais quanto substitutas. Em busca de obter resultados melhores, foi realizado o mesmo procedimento do teste anterior, no entanto, utilizando duas camadas intermediárias na rede.

Os resultados deste segundo teste para cada conjunto de dados serão apresentados a seguir, iniciando com o conjunto NSR2DB.

Figura 31 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

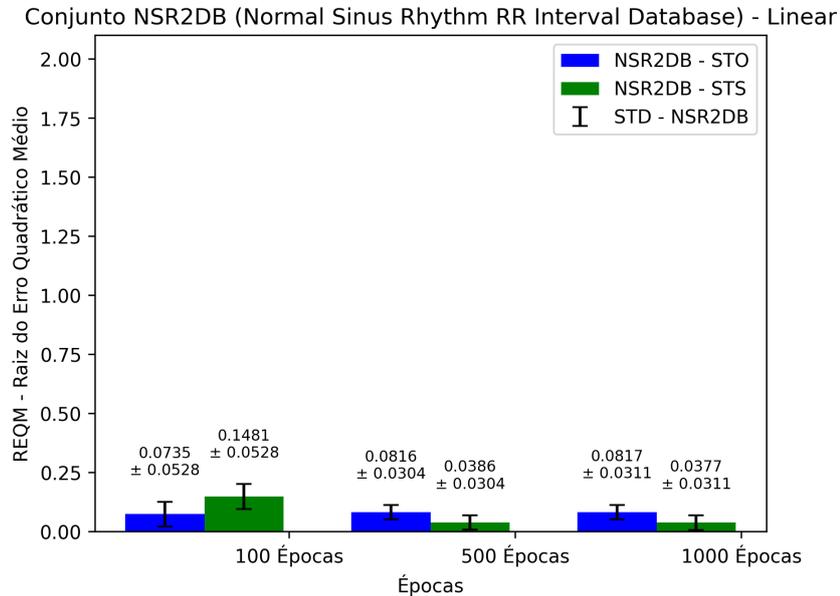


Figura 32 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu

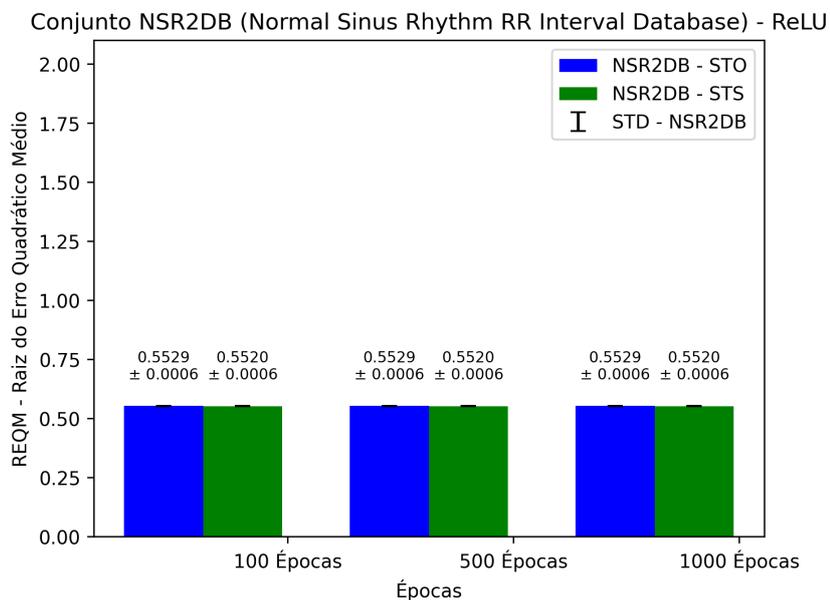
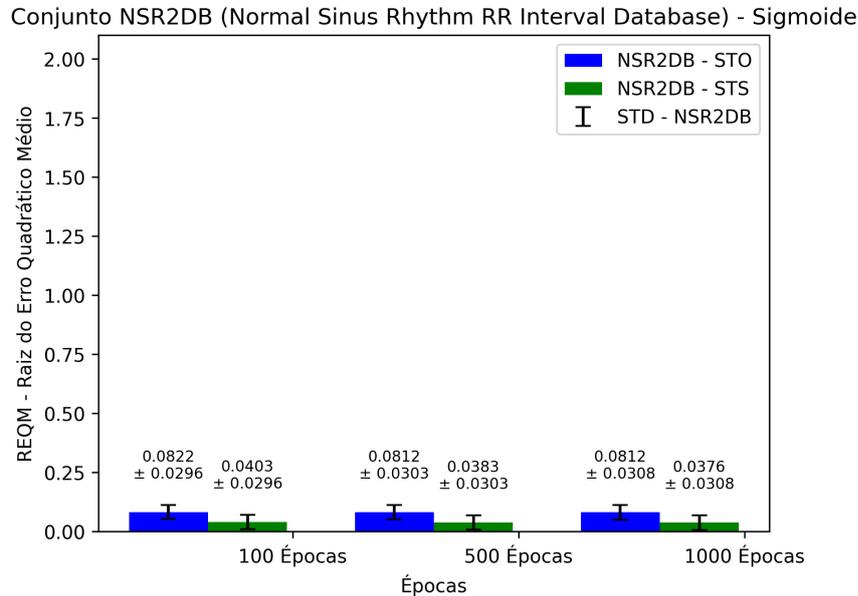


Figura 33 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Ao analisar os resultados do conjunto NSR2DB, observamos que a função de ativação linear apresentou uma convergência mais eficaz em 100 épocas para as séries temporais originais, resultando em um aumento gradual no valor da REQM à medida que o número de épocas aumentava, passando de 0.0735 com 100 épocas para 0.0817 com 1000 épocas. Em contraste, para as séries substitutas, os valores diminuíram à medida que o número de épocas aumentava, partindo de 0.1481 com 100 épocas e atingindo 0.0377 nas 1000 épocas.

Por outro lado, a função de ativação relu, gerou valores consideravelmente elevados para este conjunto, mantendo-se constantes em 0.5529 para as séries temporais originais e 0.5520 para as substitutas ao longo das três épocas avaliadas.

Quanto à função sigmoide, observamos uma diminuição gradual nos valores da REQM à medida que o número de épocas aumentava. Enquanto as séries temporais originais mantiveram certa estabilidade nos resultados a partir das 500 épocas, as séries substitutas continuaram a apresentar valores menores, passando de 0.0383 nas 500 épocas para 0.0376 nas 1000 épocas.

Os próximos resultados a serem apresentados, pertencem ao conjunto LTAfDB.

Figura 34 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

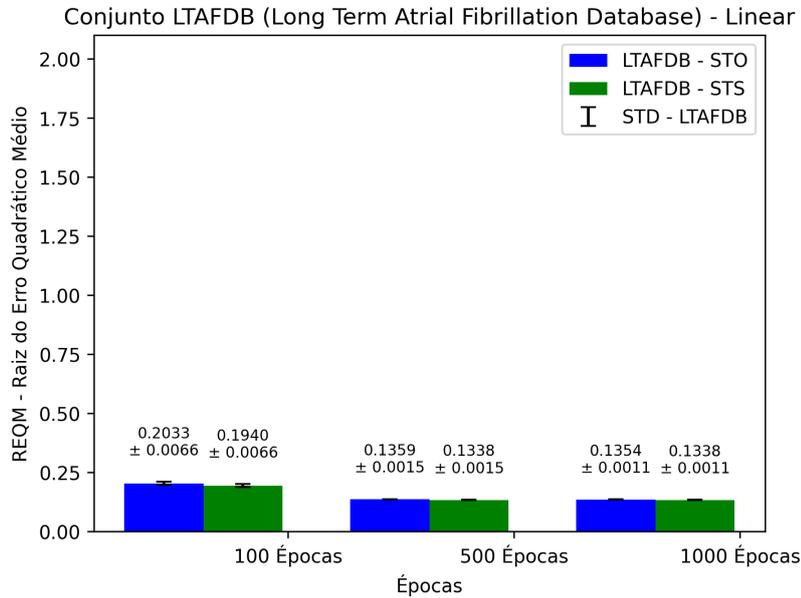


Figura 35 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

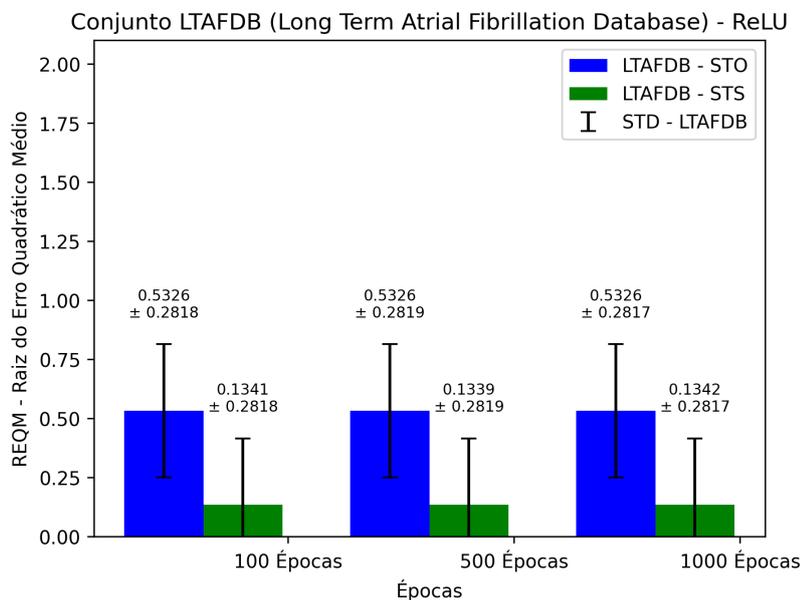
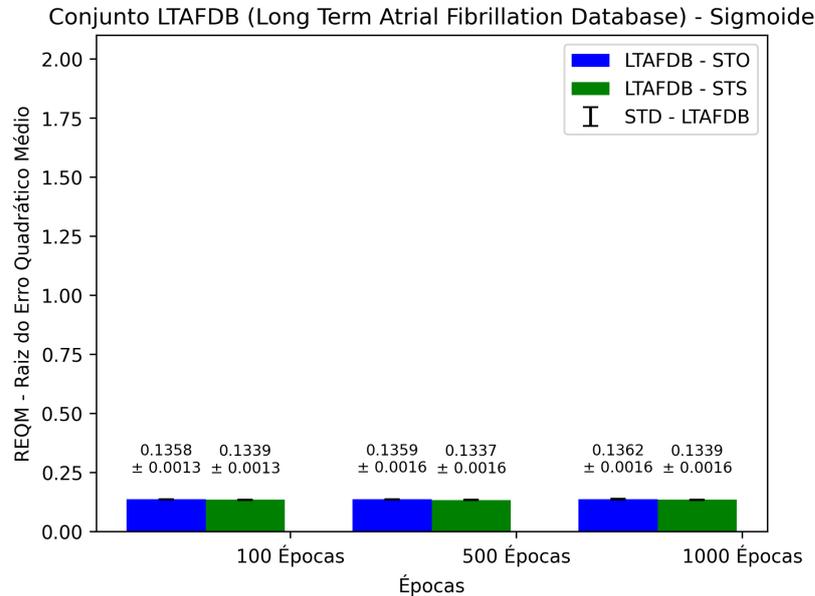


Figura 36 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Analisando os resultados do conjunto LTAfDB, a função de ativação linear demonstrou resultados próximos entre as séries temporais originais (STO) e substitutas (STS). Para as STO, observou-se uma redução contínua na REQM à medida que o número de épocas aumentava, passando de 0.2033 em 100 épocas para 0.1354 em 1000 épocas. Já para as STS, o valor permaneceu constante a partir das 500 épocas, partindo de 0.1940 em 100 épocas e mantendo-se em 0.1338 nas 500 épocas.

A função de ativação relu, por sua vez, gerou resultados elevados para as séries temporais originais, atingindo 0.5326 para as três épocas de treinamento. Para as STS, ocorreu uma redução moderada nas 500 épocas, indo de 0.1341 para 0.1339, seguida por um aumento moderado nas 1000 épocas, chegando a 0.1342.

Já a função sigmoide apresentou seu desempenho mais eficaz nas séries temporais originais durante as 100 épocas iniciais, seguido por um aumento nos valores, de 0.1358 para 0.1362 nas 1000 épocas. Para as séries substitutas, o resultado mais favorável foi alcançado nas 500 épocas de treinamento, partindo de 0.1339 em 100 épocas, para 0.1337 nas 500 épocas, e retornando a 0.1339 nas 1000 épocas.

As imagens a seguir, apresentam os resultados de treinamento para o conjunto LTSTDB.

Figura 37 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

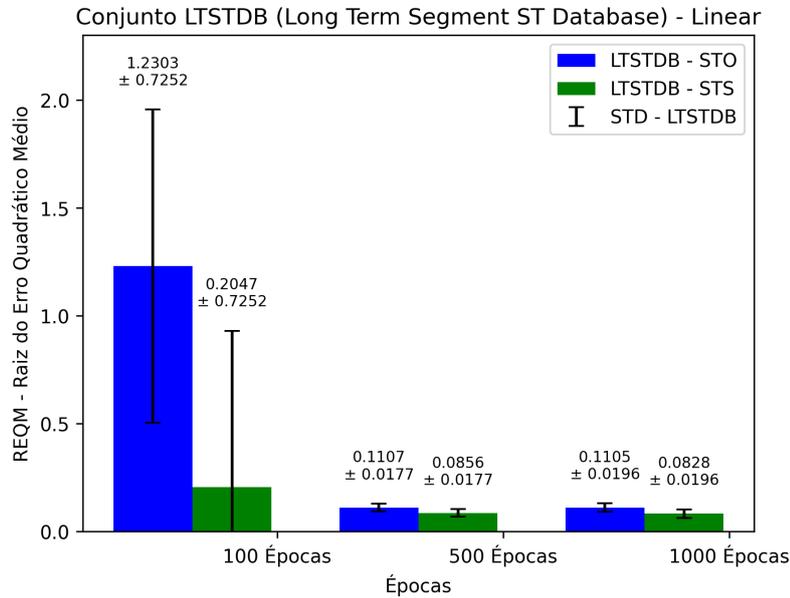


Figura 38 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

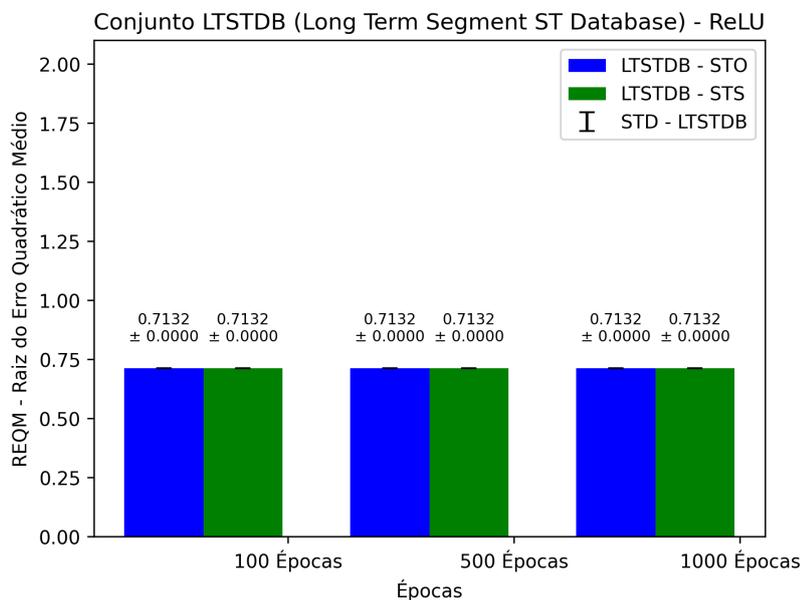
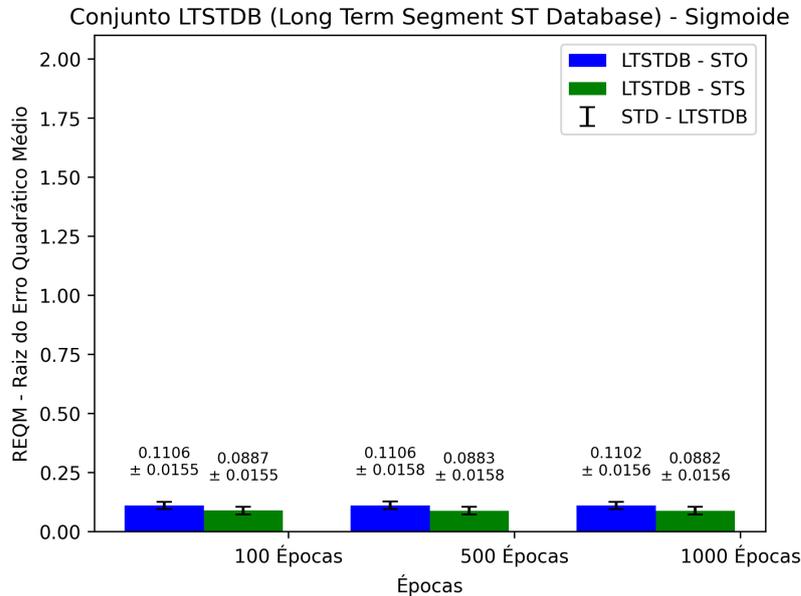


Figura 39 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Observando os resultados anteriores, concluímos que a função de ativação linear resultou em uma diminuição na REQM à medida que o número de épocas aumenta, tanto para as séries temporais originais quanto para as substitutas. Apesar da disparidade nos valores iniciais nas 100 épocas, essas diferenças se reduziram consideravelmente, tornando-se mais próximas a partir das 500 épocas. Para as séries originais, a REQM iniciou em 1.2303 nas 100 épocas, atingindo 0.1105 nas 1000 épocas. Enquanto isso, as séries substitutas começaram em 0.2047 e alcançaram 0.0828 nas 1000 épocas.

Para a função de ativação relu, os resultados foram relativamente elevados e estáveis, mantendo-se em 0.7132 para ambas as séries ao longo das três épocas de treinamento.

A função sigmoide demonstrou desempenho positivo para ambos os conjuntos, tanto originais quanto substitutas. Além disso, apresentou uma redução na REQM à medida que o número de épocas aumentava. Para as séries originais, a REQM foi de 0.1106 nas 100 épocas para 0.1102 nas 1000 épocas. Já para as séries substitutas, os valores foram de 0.0887 nas 100 épocas para 0.0882 nas 1000 épocas.

Os próximos resultados deste teste a serem representados, são do conjunto CHF2DB.

Figura 40 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

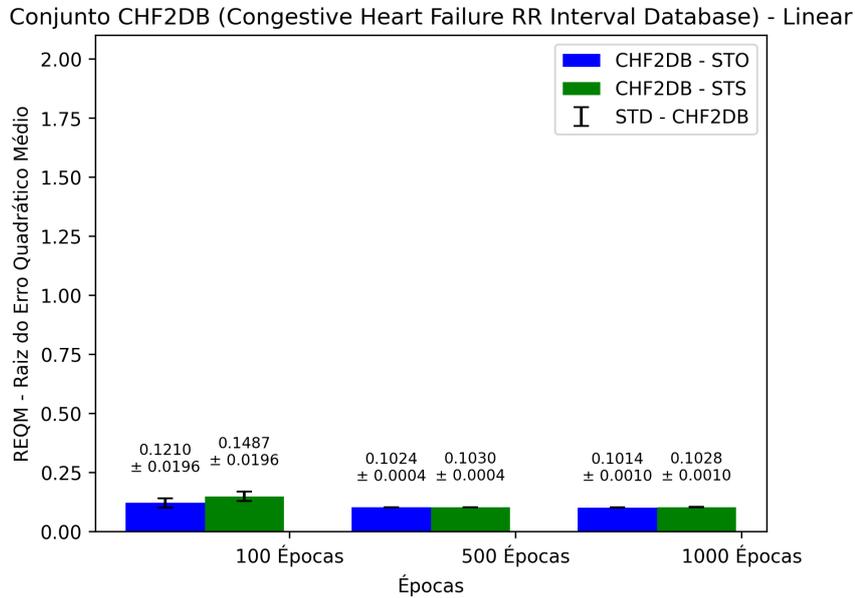


Figura 41 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

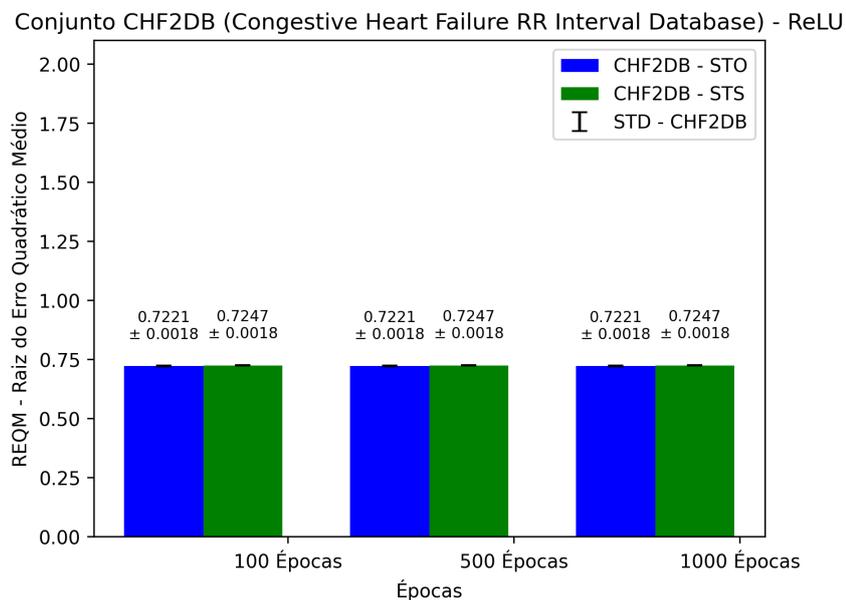
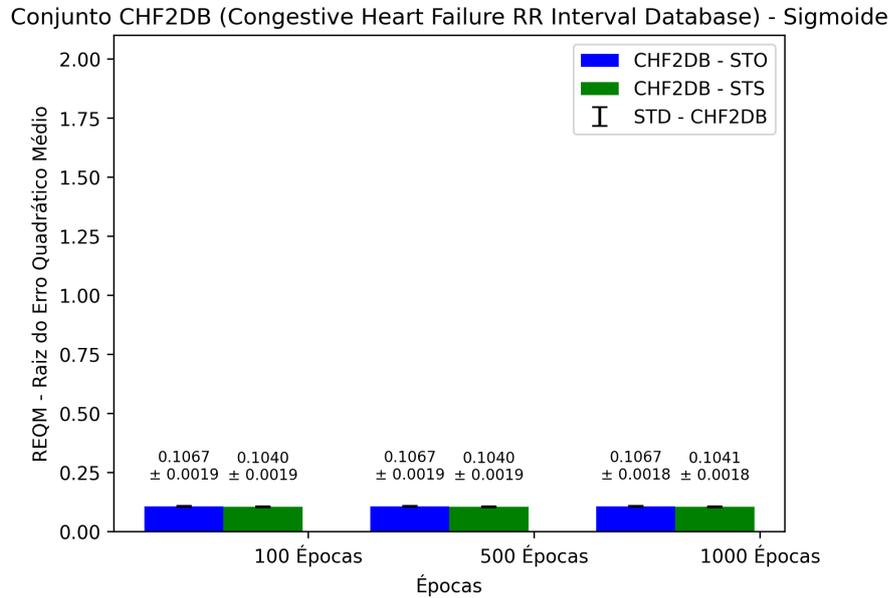


Figura 42 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Fica evidente que a função de ativação linear proporcionou resultados satisfatórios para ambas as séries, partindo de 0.1210 nas 100 épocas e atingindo 0.1014 nas 1000 épocas para o conjunto original. No caso do conjunto substituto, a REQM iniciou em 0.1487 nas 100 épocas e atingiu 0.1028 nas 1000 épocas.

A função relu, manteve valores elevados e estáveis para ambos os conjuntos, independentemente do número de épocas considerado.

Quanto à função sigmoide, observamos resultados estáveis para as séries originais, mantendo-se em 0.1067 em todas as épocas. Para as séries substitutas, houve um pequeno aumento nas 1000 épocas, passando de 0.1040 para 0.1041.

Os próximos dados apresentados, são do conjunto CHFDB.

Figura 43 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

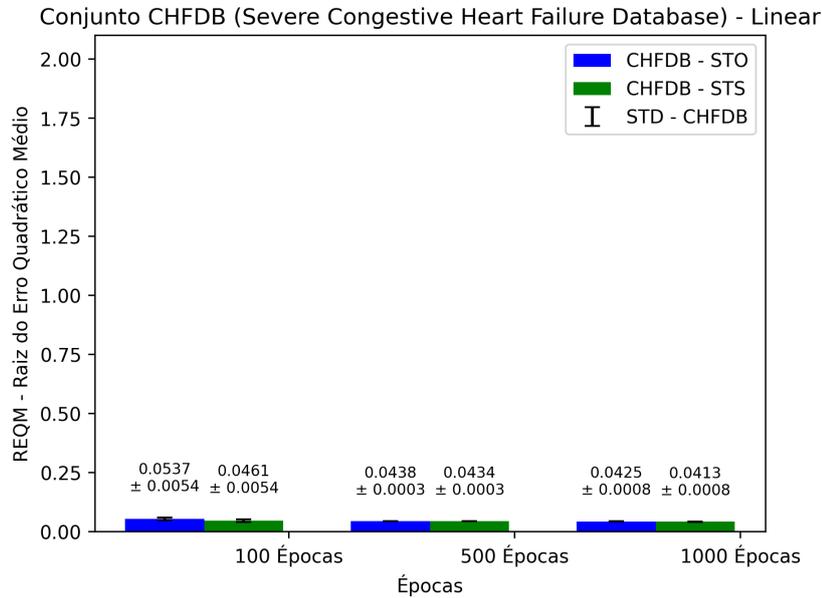


Figura 44 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

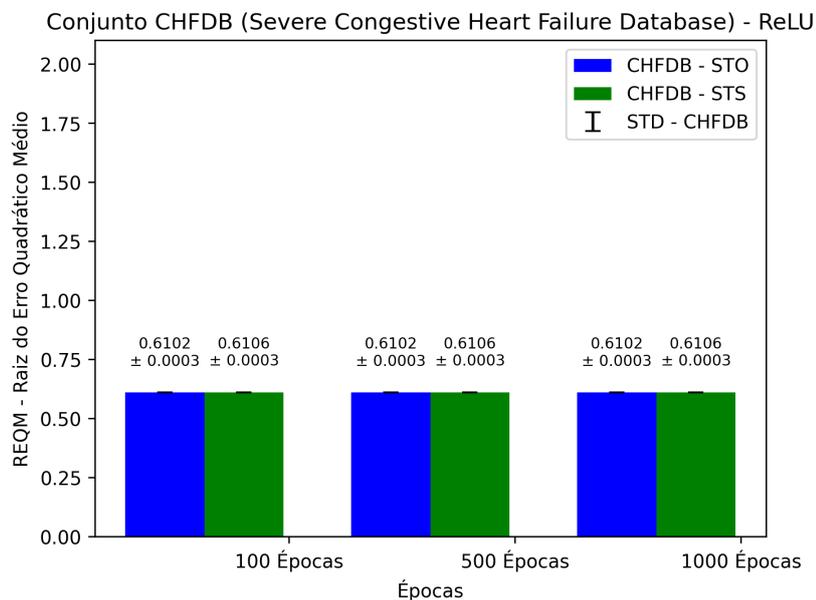
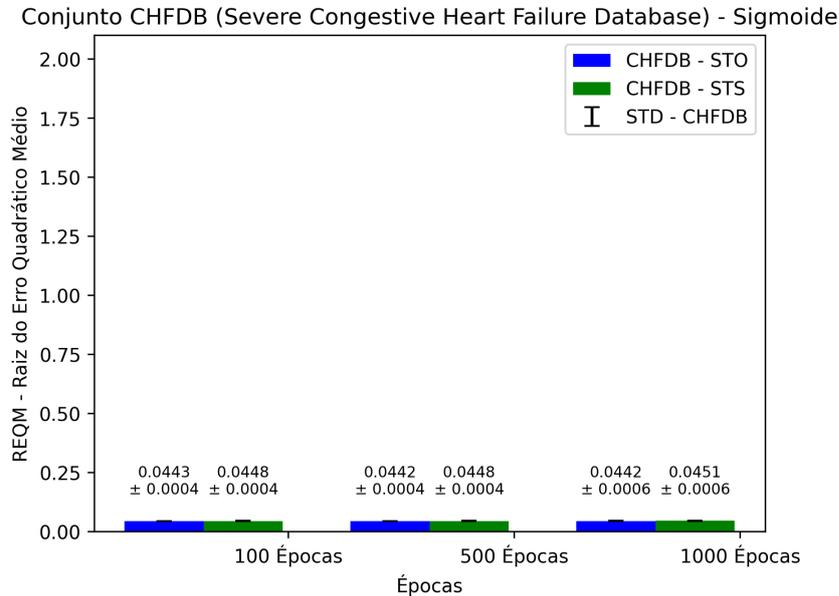


Figura 45 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com duas camadas intermediárias com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Neste conjunto, a função de ativação linear demonstrou eficácia ao longo do treinamento, refletindo em uma redução consistente nos valores da REQM. Inicialmente, o conjunto de séries originais registrou 0.0537 nas 100 épocas, alcançando 0.0425 nas 1000 épocas. Similarmente, para as séries substitutas, houve uma diminuição de 0.0461 para 0.0413 no mesmo intervalo de épocas.

Em relação aos resultados da função relu, esta apresentou valores elevados e estáveis para este conjunto. Tanto para o conjunto de séries originais quanto para o conjunto substituto, a REQM permaneceu constante em 0.6102 e 0.6106, respectivamente.

A função sigmoide, por sua vez, obteve bons resultados para ambos os conjuntos. Para o conjunto de séries originais, a REQM apresentou uma leve redução de 0.0443 para 0.0442 a partir de 500 épocas. No conjunto substituto, houve um pequeno aumento de 0.0448 para 0.0451 nas 1000 épocas.

5.1.3 Teste com uma camada intermediária e variação na função de ativação

Considerando que os resultados do teste anterior não apresentaram valores satisfatórios para todos os conjuntos, foi então realizado um teste adicional. Desta vez, utilizando apenas uma camada intermediária/oculta, variando entre diferentes funções de ativação. As combinações utilizadas estão descritas abaixo.

Tabela 1 – Configurações de Camadas da Rede Neural

| CAMADA DE ENTRADA | CAMADA OCULTA | CAMADA DE SAÍDA |
|-------------------|---------------|-----------------|
| LINEAR | RELU | LINEAR |
| LINEAR | SIGMOIDE | LINEAR |
| RELU | LINEAR | RELU |
| RELU | SIGMOIDE | RELU |
| SIGMOIDE | LINEAR | SIGMOIDE |
| SIGMOIDE | RELU | SIGMOIDE |

Assim como nos testes anteriores, os resultados demonstraram variações entre os conjuntos de dados. Em alguns casos, as séries temporais originais e suas substitutas apresentaram valores bastante próximos, indicando uma boa capacidade do modelo em reproduzir os padrões. No entanto, em outros conjuntos, observou-se uma diferença mais significativa. As tabelas abaixo apresentam os resultados para cada conjunto de dados.

Tabela 2 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para NSR2DB

| DADOS | FUNÇÃO DE ATIVAÇÃO | 100 ÉPOCAS | 500 ÉPOCAS | 1000 ÉPOCAS |
|--------------|------------------------------|------------|------------|-------------|
| NSR2DB - STO | LINEAR - RELU - LINEAR | 2,3115 | 0,0738 | 0,0817 |
| NSR2DB - STS | LINEAR - RELU - LINEAR | 0,0961 | 0,0627 | 0,0827 |
| NSR2DB - STO | LINEAR - SIGMOIDE - LINEAR | 0,0821 | 0,0824 | 0,0829 |
| NSR2DB - STS | LINEAR - SIGMOIDE - LINEAR | 0,0827 | 0,0844 | 0,0843 |
| NSR2DB - STO | RELU - LINEAR - RELU | 0,5529 | 0,5529 | 0,5529 |
| NSR2DB - STS | RELU - LINEAR - RELU | 0,5520 | 0,5520 | 0,5520 |
| NSR2DB - STO | RELU - SIGMOIDE - RELU | 0,0818 | 0,0387 | 0,0592 |
| NSR2DB - STS | RELU - SIGMOIDE - RELU | 0,5520 | 0,5520 | 0,5520 |
| NSR2DB - STO | SIGMOIDE - LINEAR - SIGMOIDE | 0,0817 | 0,0816 | 0,0816 |
| NSR2DB - STS | SIGMOIDE - LINEAR - SIGMOIDE | 0,0403 | 0,0400 | 0,0401 |
| NSR2DB - STO | SIGMOIDE - RELU - SIGMOIDE | 0,0816 | 0,0816 | 0,0816 |
| NSR2DB - STS | SIGMOIDE - RELU - SIGMOIDE | 0,0397 | 0,0411 | 0,0383 |

Analisando os resultados do conjunto NSR2DB, observa-se que as variações nas combinações de funções de ativação e épocas de treinamento exercem influência significativa nos resultados da REQM para séries temporais originais (STO) e séries substitutas (STS).

A combinação linear-relu exibiu um padrão de aumento na REQM para ambos conjuntos de dados, saindo de 2.3115 nas 100 épocas para 0.0817 nas 1000 épocas para STO, e de 0.0961 para 0.0827 nas mesmas épocas para STS. Já a combinação linear-sigmoide, os melhores resultados foram obtidos em 100 épocas em ambas as séries, com um aumento da REQM conforme o número de épocas.

Resultados elevados e constantes foram obtidos nas combinações relu-linear e relu-sigmoide, independentemente do número de épocas, exceto por uma leve variação nos resultados das STO para a função de ativação relu-sigmoide.

As combinações sigmoide-linear e sigmoide-relu apresentaram estabilidade nos resultados. A combinação sigmoide-linear atingiu seu melhor desempenho na REQM em 500 épocas para STO e STS, mantendo-se estável nas 1000 épocas. Sigmoide-relu permaneceu constante para STO, mas obteve sua melhor REQM em 1000 épocas para STS.

Tabela 3 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para LTAfDB

| DADOS | FUNÇÃO DE ATIVAÇÃO | 100 ÉPOCAS | 500 ÉPOCAS | 1000 ÉPOCAS |
|--------------|------------------------------|------------|------------|-------------|
| LTAfDB - STO | LINEAR - RELU - LINEAR | 0,2411 | 0,1360 | 0,1361 |
| LTAfDB - STS | LINEAR - RELU - LINEAR | 1,0735 | 0,1343 | 0,1345 |
| LTAfDB - STO | LINEAR - SIGMOIDE - LINEAR | 0,1359 | 0,1370 | 0,1365 |
| LTAfDB - STS | LINEAR - SIGMOIDE - LINEAR | 0,1344 | 0,1346 | 0,1344 |
| LTAfDB - STO | RELU - LINEAR - RELU | 0,5326 | 0,5326 | 0,5326 |
| LTAfDB - STS | RELU - LINEAR - RELU | 0,5300 | 0,5300 | 0,5300 |
| LTAfDB - STO | RELU - SIGMOIDE - RELU | 0,5326 | 0,5326 | 0,5326 |
| LTAfDB - STS | RELU - SIGMOIDE - RELU | 0,5300 | 0,5300 | 0,5300 |
| LTAfDB - STO | SIGMOIDE - LINEAR - SIGMOIDE | 0,1357 | 0,1358 | 0,1361 |
| LTAfDB - STS | SIGMOIDE - LINEAR - SIGMOIDE | 0,1340 | 0,1341 | 0,1341 |
| LTAfDB - STO | SIGMOIDE - RELU - SIGMOIDE | 0,1358 | 0,1359 | 0,1360 |
| LTAfDB - STS | SIGMOIDE - RELU - SIGMOIDE | 0,1340 | 0,1341 | 0,1341 |

Observando o conjunto LTAfDB, destaca-se que a combinação linear-relu proporcionou o melhor desempenho de REQm durante as 500 épocas de treinamento para ambas as séries temporais (STO e STS), atingindo 0.1360 para séries originais e 0.1343 para séries substitutas. Já a combinação linear-sigmoide alcançou sua melhor REQm após 100 épocas de treinamento, registrando valores de 0.1359 e 0.1344 para STO e STS, respectivamente.

As combinações relu-linear e relu-sigmoide exibiram resultados constantes e elevados para este conjunto, sem oscilação nos valores de acordo com o número de épocas. Enquanto isso, nas combinações sigmoide-linear e sigmoide-relu, os melhores resultados foram obtidos nas 100 épocas de treinamento para ambos os conjuntos de dados, com valores de 0.1357 para STO e 0.1340 para STS na sigmoide-linear, e 0.1358 para STO e 0.1340 para STS na sigmoide-relu.

Tabela 4 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para LTSTDB

| DADOS | FUNÇÃO DE ATIVAÇÃO | 100 ÉPOCAS | 500 ÉPOCAS | 1000 ÉPOCAS |
|--------------|------------------------------|------------|------------|-------------|
| LTSTDB - STO | LINEAR - RELU - LINEAR | 0,1235 | 0,1109 | 0,1107 |
| LTSTDB - STS | LINEAR - RELU - LINEAR | 0,1402 | 0,0841 | 0,1040 |
| LTSTDB - STO | LINEAR - SIGMOIDE - LINEAR | 0,1112 | 0,0956 | 0,0922 |
| LTSTDB - STS | LINEAR - SIGMOIDE - LINEAR | 0,0915 | 0,0818 | 0,0793 |
| LTSTDB - STO | RELU - LINEAR - RELU | 0,7132 | 0,7132 | 0,7132 |
| LTSTDB - STS | RELU - LINEAR - RELU | 0,7132 | 0,7132 | 0,7132 |
| LTSTDB - STO | RELU - SIGMOIDE - RELU | 0,1111 | 0,1115 | 0,1070 |
| LTSTDB - STS | RELU - SIGMOIDE - RELU | 0,0875 | 0,0793 | 0,0845 |
| LTSTDB - STO | SIGMOIDE - LINEAR - SIGMOIDE | 0,1106 | 0,1105 | 0,1104 |
| LTSTDB - STS | SIGMOIDE - LINEAR - SIGMOIDE | 0,0887 | 0,0884 | 0,0882 |
| LTSTDB - STO | SIGMOIDE - RELU - SIGMOIDE | 0,1108 | 0,1106 | 0,1107 |
| LTSTDB - STS | SIGMOIDE - RELU - SIGMOIDE | 0,0886 | 0,0881 | 0,0881 |

Ao analisar os resultados para o conjunto LTSTDB, destaca-se que a função de ativação linear-relu se destacou, resultando na melhor REQM para as séries originais após 1000 épocas de treinamento. Houve uma significativa redução da REQM, partindo de 0.1235 nas 100 épocas iniciais para atingir 0.1107. Por outro lado, para as séries substitutas, o ponto de mínimo foi alcançado nas 500 épocas, com uma leve elevação posterior para as 1000 épocas, passando de 0.0841 para 0.1040.

Na combinação linear-sigmoide, os resultados mais favoráveis foram obtidos nas 1000 épocas de treinamento para ambos os conjuntos de dados. A REQM iniciou em 0.1112 nas 100 épocas e diminuiu para 0.0922 nas 1000 épocas para as séries originais. No caso das séries substitutas, partiu de 0.0915 nas 100 épocas para 0.0793 nas 500 épocas, mantendo-se estável nas 1000 épocas.

A combinação relu-linear manteve-se estável para ambas as séries, com valores elevados, independente do número de épocas. A combinação relu-sigmoide apresentou resultados promissores, atingindo 0.1070 para as 1000 épocas nas séries originais e 0.0793 nas 500 épocas para as séries substitutas, com uma leve elevação para 0.0845 nas 1000 épocas.

A função sigmoide-linear proporcionou resultados consistentes para ambos os conjuntos, alcançando a melhor REQM nas 1000 épocas de treinamento. Por outro lado, a combinação sigmoide-relu teve sua melhor REQM nas 500 épocas para ambos os conjuntos, mantendo-se estável para as séries substitutas.

Tabela 5 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para CHF2DB

| DADOS | FUNÇÃO DE ATIVAÇÃO | 100 ÉPOCAS | 500 ÉPOCAS | 1000 ÉPOCAS |
|--------------|------------------------------|------------|------------|-------------|
| CHF2DB - STO | LINEAR - RELU - LINEAR | 0,1274 | 0,0440 | 0,0444 |
| CHF2DB - STS | LINEAR - RELU - LINEAR | 0,1440 | 0,1040 | 0,1043 |
| CHF2DB - STO | LINEAR - SIGMOIDE - LINEAR | 0,0446 | 0,0447 | 0,0457 |
| CHF2DB - STS | LINEAR - SIGMOIDE - LINEAR | 0,1045 | 0,1052 | 0,1044 |
| CHF2DB - STO | RELU - LINEAR - RELU | 0,6102 | 0,6102 | 0,6102 |
| CHF2DB - STS | RELU - LINEAR - RELU | 0,7247 | 0,7247 | 0,7247 |
| CHF2DB - STO | RELU - SIGMOIDE - RELU | 0,0446 | 0,0447 | 0,0445 |
| CHF2DB - STS | RELU - SIGMOIDE - RELU | 0,1045 | 0,1033 | 0,1045 |
| CHF2DB - STO | SIGMOIDE - LINEAR - SIGMOIDE | 0,0442 | 0,0440 | 0,0433 |
| CHF2DB - STS | SIGMOIDE - LINEAR - SIGMOIDE | 0,1040 | 0,1042 | 0,1043 |
| CHF2DB - STO | SIGMOIDE - RELU - SIGMOIDE | 0,0442 | 0,0442 | 0,0442 |
| CHF2DB - STS | SIGMOIDE - RELU - SIGMOIDE | 0,1042 | 0,1041 | 0,1042 |

Ao analisar o conjunto CHF2DB, é notável que a combinação linear-relu alcançou os melhores resultados após 500 épocas de treinamento, tanto para as séries temporais originais (STO) quanto para as séries substitutas (STS). Ambas as séries apresentaram uma elevação moderada da REQM ao longo das 1000 épocas subsequentes.

Para a combinação linear-sigmoide, os resultados mais favoráveis foram atingidos nas 100 épocas para as STO, registrando 0.0446, e nas 1000 épocas para as STS, com 0.1044. Esse padrão sugere que o modelo atingiu uma estabilidade após 100 épocas para as STO, enquanto nas STS a melhoria foi mais gradual e ocorreu ao longo das 1000 épocas.

A avaliação da combinação relu-linear apresentou resultados consideravelmente elevados para ambas as séries, diferentemente da combinação relu-sigmoide, que proporcionou o melhor desempenho para as STO após 1000 épocas, alcançando 0.0445. Para as STS, o ponto de mínimo foi encontrado nas 500 épocas, registrando 0.1033, com um aumento posterior nas 1000 épocas.

As combinações sigmoide-linear e sigmoide-relu apresentaram resultados próximos entre si, com variações nos pontos de melhor desempenho para STO e STS. Para a sigmoide-linear, a melhor REQM para as STO ocorreu nas 1000 épocas, enquanto para as STS foi nas 100 épocas. Já para a sigmoide-relu, o melhor desempenho foi registrado nas 100 épocas para STO, mantendo-se estável com o aumento de épocas, enquanto para as STS, o ponto de mínimo foi alcançado nas 500 épocas.

Tabela 6 – Resultados para Diferentes Configurações de Funções de Ativação e Épocas para CHFDB

| DADOS | FUNÇÃO DE ATIVAÇÃO | 100 ÉPOCAS | 500 ÉPOCAS | 1000 ÉPOCAS |
|-------------|------------------------------|------------|------------|-------------|
| CHFDB - STO | LINEAR - RELU - LINEAR | 0,2622 | 0,1067 | 0,1050 |
| CHFDB - STS | LINEAR - RELU - LINEAR | 0,0528 | 0,0449 | 0,0449 |
| CHFDB - STO | LINEAR - SIGMOIDE - LINEAR | 0,1071 | 0,1070 | 0,1072 |
| CHFDB - STS | LINEAR - SIGMOIDE - LINEAR | 0,0451 | 0,0450 | 0,0454 |
| CHFDB - STO | RELU - LINEAR - RELU | 0,7221 | 0,7221 | 0,7221 |
| CHFDB - STS | RELU - LINEAR - RELU | 0,6106 | 0,6106 | 0,6106 |
| CHFDB - STO | RELU - SIGMOIDE - RELU | 0,1073 | 0,1033 | 0,1020 |
| CHFDB - STS | RELU - SIGMOIDE - RELU | 0,0450 | 0,0450 | 0,0451 |
| CHFDB - STO | SIGMOIDE - LINEAR - SIGMOIDE | 0,1068 | 0,1068 | 0,1068 |
| CHFDB - STS | SIGMOIDE - LINEAR - SIGMOIDE | 0,0446 | 0,0447 | 0,0447 |
| CHFDB - STO | SIGMOIDE - RELU - SIGMOIDE | 0,1068 | 0,1068 | 0,1067 |
| CHFDB - STS | SIGMOIDE - RELU - SIGMOIDE | 0,0447 | 0,0446 | 0,0446 |

Na análise do conjunto CHFDB, a combinação de funções linear-relu atingiu o menor valor após 1000 épocas de treinamento para o conjunto STO. Para as séries substitutas (STS), esse ponto mínimo foi alcançado em 500 épocas, permanecendo estável nas 1000 épocas de treinamento. Já a combinação linear-sigmoide apresentou seus melhores resultados para ambos os conjuntos de dados em 500 épocas, com um leve aumento nas 1000 épocas.

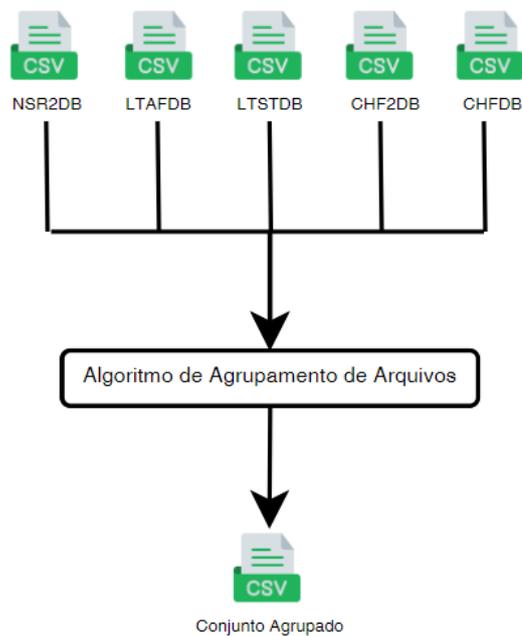
A combinação relu-linear manteve resultados elevados e consistentes para as séries STO e STS, com valores de REQM de 0.7221 e 0.6106, respectivamente, independente do aumento no número de épocas. Por outro lado, a combinação relu-sigmoide obteve o melhor resultado para as STO em 1000 épocas, enquanto para as STS, o ponto mínimo foi alcançado em 100 épocas.

As combinações sigmoide-linear e sigmoide-relu mostraram resultados bastante próximos quando comparadas entre os conjuntos STO e STS, com diferenças mínimas observadas.

5.1.4 Teste com uma camada intermediária e conjunto de dados

Para enfrentar os desafios e aprimorar os resultados em todos os conjuntos, foram conduzidos testes adicionais, dessa vez, criando um único arquivo contendo todos os conjuntos de dados selecionados. Esta etapa é apresentada na imagem 46 a seguir.

Figura 46 – Representação da Técnica de Conjunto de Arquivos



Utilizando a biblioteca *pandas*, um algoritmo escrito em *python*, recebe como entrada os cinco conjuntos de dados e após o processo de conversão, o algoritmo devolve um único arquivo, contendo todas as séries temporais, resultando em uma matriz de dados, com 65 colunas contendo os diferentes conjuntos de dados e 15839 linhas com os respectivos batimentos cardíacos. Esse processo foi realizado tanto para o conjunto de séries originais, quanto para o conjunto de séries substitutas.

Tendo então um único arquivo com todos os conjuntos de dados para treinamento da rede, seguiu-se para a execução, onde mantivemos as características da rede neural descrita anteriormente e para esse teste consideramos o uso de apenas uma camada intermediária. Os testes foram realizados de forma separada para todos os conjuntos de dados com as séries substitutas e posteriormente o inverso, onde o treinamento da rede ocorreu com um conjunto de séries substitutas e o treinamento individual com o conjunto de dados originais. As imagens a seguir apresentam os resultados deste treinamento para cada conjunto de dados.

Figura 47 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

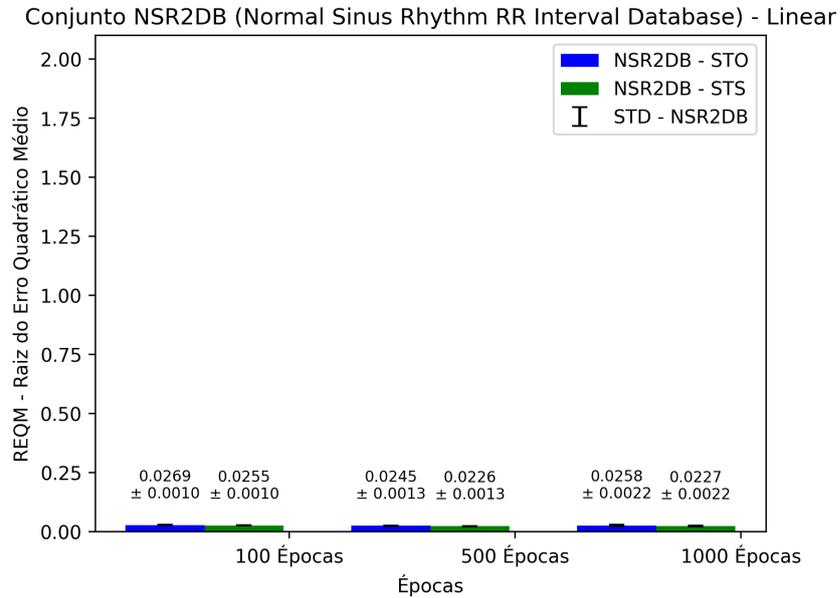


Figura 48 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu

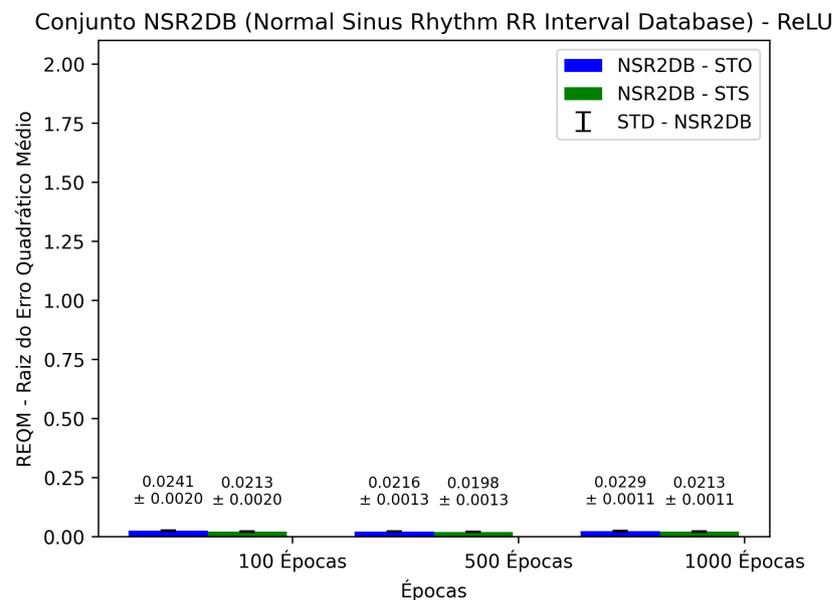
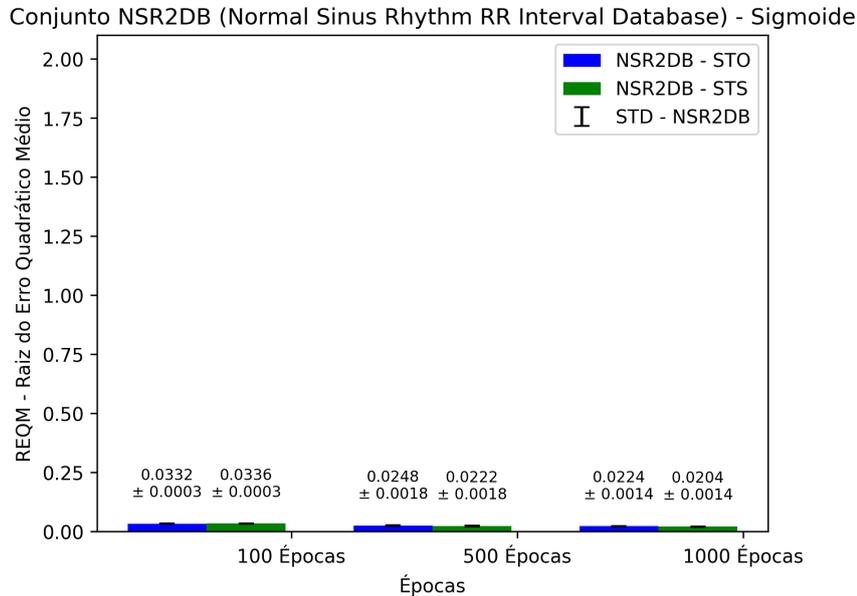


Figura 49 – Comparação entre os resultados originais (NSR2DB - STO) e os resultados substitutos (NSR2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Ao avaliar o conjunto NSR2DB, observa-se que os resultados gerados pela função de ativação linear foram bastante semelhantes para ambos os conjuntos de dados (STO e STS), apresentando diferenças sutis em todas as épocas. A melhor REQM foi alcançada com 500 épocas de treinamento para ambos os conjuntos, com valores de 0.0245 para STO e 0.0226 para STS.

No caso da função de ativação relu, os resultados mais expressivos também foram obtidos nas 500 épocas de treinamento, atingindo 0.0216 para STO e 0.0198 para STS. Nota-se uma certa oscilação nos valores de REQM durante as 1000 épocas de treinamento para ambas as séries.

Por fim, a função de ativação sigmoide destacou-se ao proporcionar os melhores resultados para os conjuntos STO e STS nas 1000 épocas de treinamento, com valores de 0,0224 para STO e 0,0204 para STS.

Figura 50 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

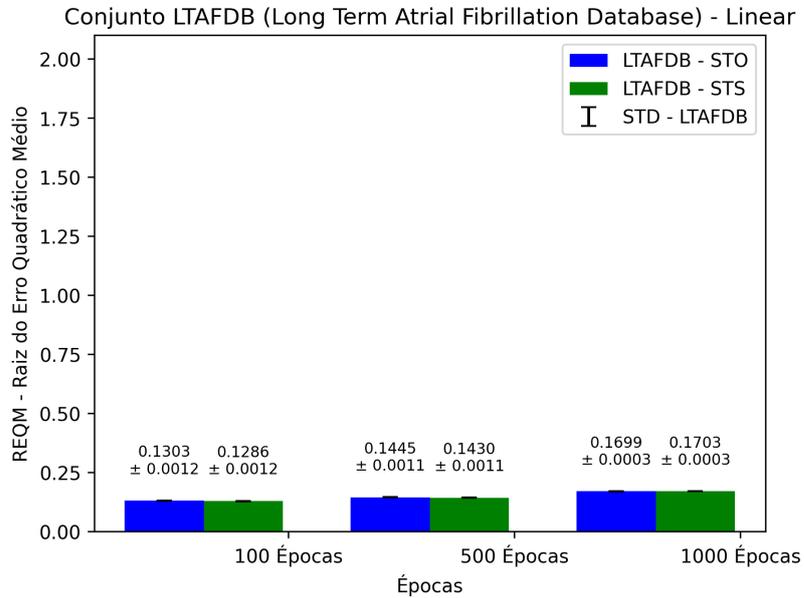


Figura 51 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

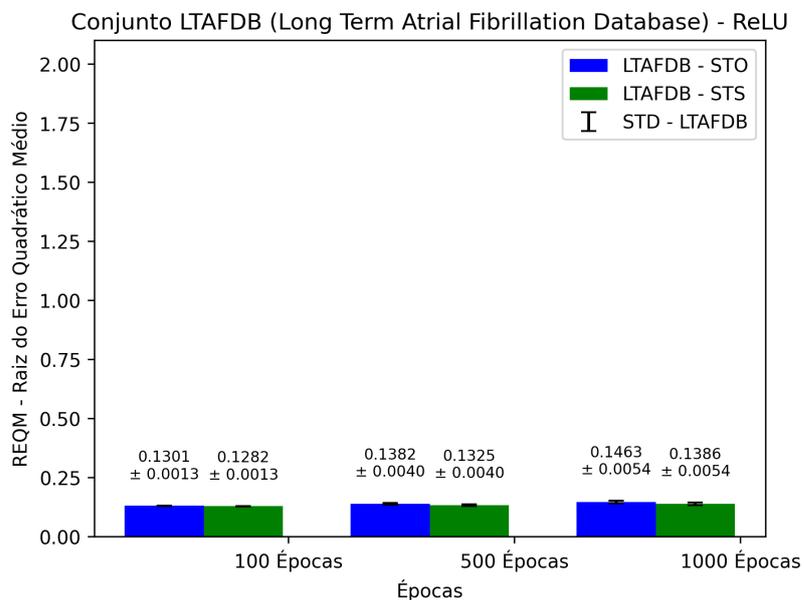
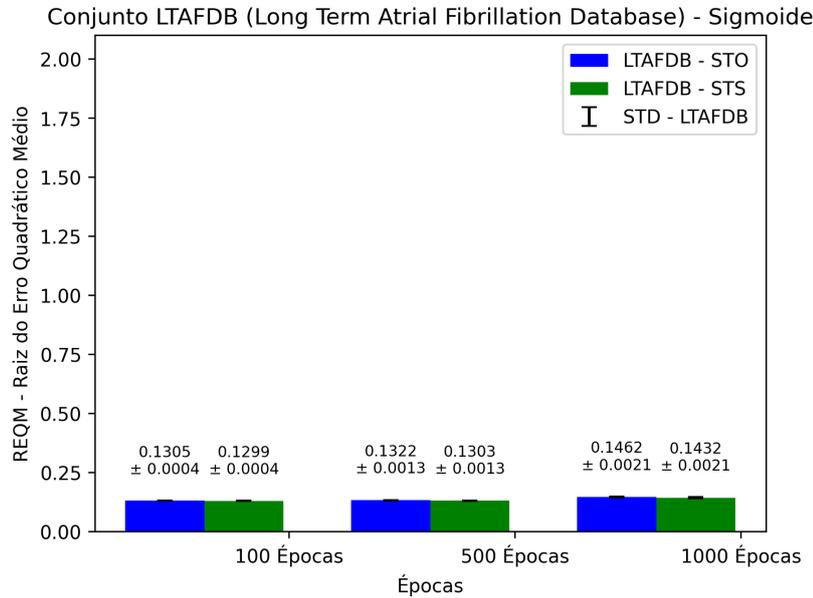


Figura 52 – Comparação entre os resultados originais (LTAfDB - STO) e os resultados substitutos (LTAfDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Para o conjunto LTAfDB, a função de ativação linear obteve o melhor desempenho nas séries temporais originais e substitutas durante as 100 épocas de treinamento, registrando 0.1303 para STO e 0.1286 para STS. No entanto, nota-se um aumento nos valores da REQM a partir das 500 épocas de treinamento.

De maneira semelhante, a função de ativação relu alcançou resultados expressivos, com valores de 0.1301 para STO e 0.1282 para STS nas 100 épocas. Assim como na linear, observou-se um crescimento gradual da REQM à medida que o número de épocas aumentou.

Quanto à função sigmoide, também apresentou bons resultados nas 100 épocas, com valores de 0.1305 para STO e 0.1299 para STS. Contudo, à medida que o número de épocas de treinamento aumenta, observa-se uma elevação na REQM.

Figura 53 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camadas intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

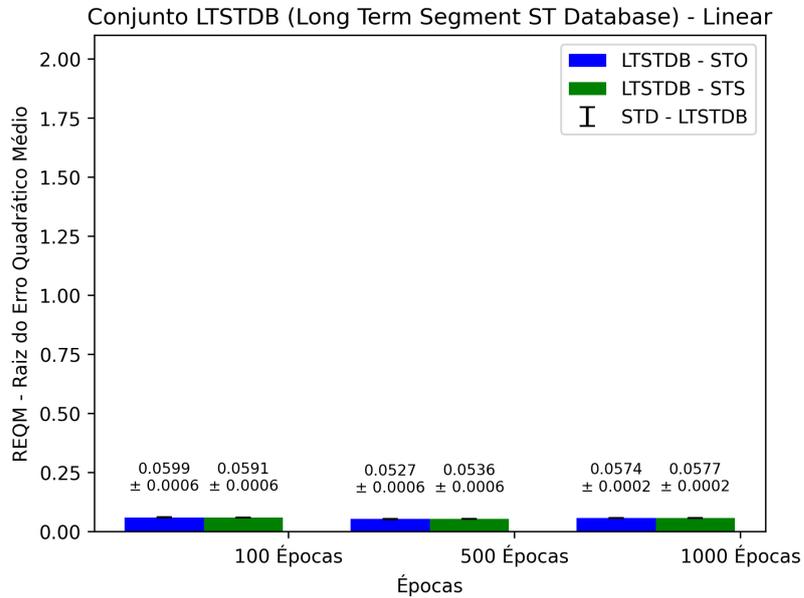


Figura 54 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

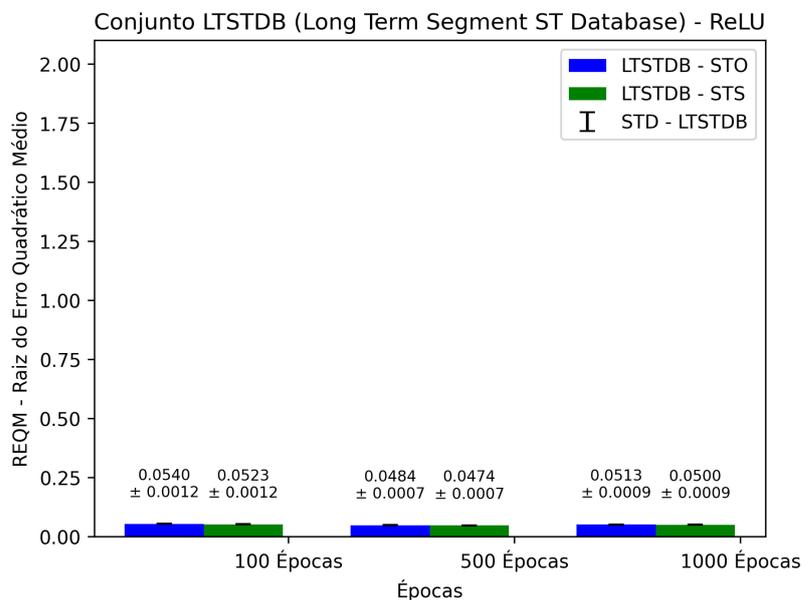
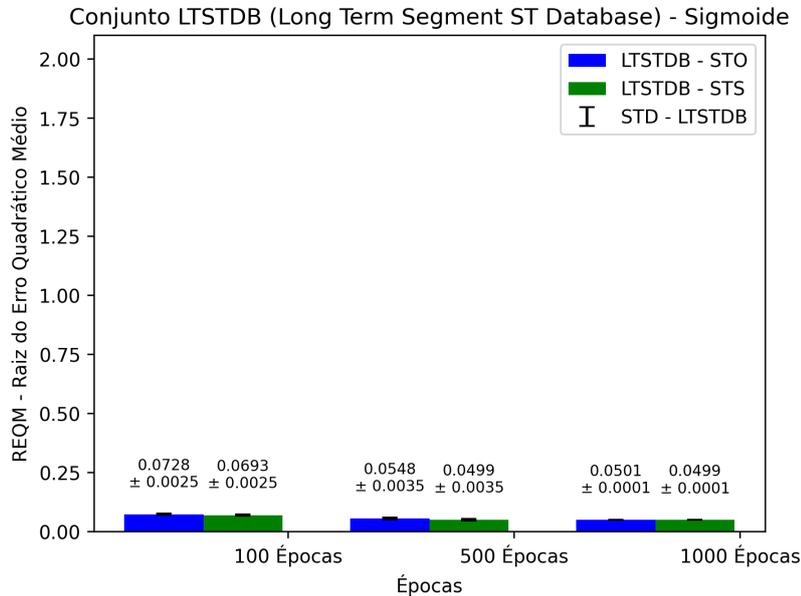


Figura 55 – Comparação entre os resultados originais (LTSTDB - STO) e os resultados substitutos (LTSTDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Neste conjunto, os resultados foram mais favoráveis para a função de ativação linear durante as 500 épocas de treinamento, com valores de 0.0527 para as STO e 0.0536 para as STS. Nota-se uma redução desses valores em relação aos resultados das 100 épocas, seguida por um aumento nas 1000 épocas de treinamento.

De maneira semelhante, a função de ativação relu também alcançou seu melhor desempenho com 500 épocas, registrando 0.0484 para as STO e 0.0474 para as STS, com um aumento na REQM ao longo das 1000 épocas.

Quanto à função de ativação sigmoide, as STO apresentaram a melhor REQM nas 1000 épocas (0.0501), enquanto as STS obtiveram o melhor resultado com 500 épocas (0.0499), mantendo esse valor no treinamento com 1000 épocas.

Figura 56 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

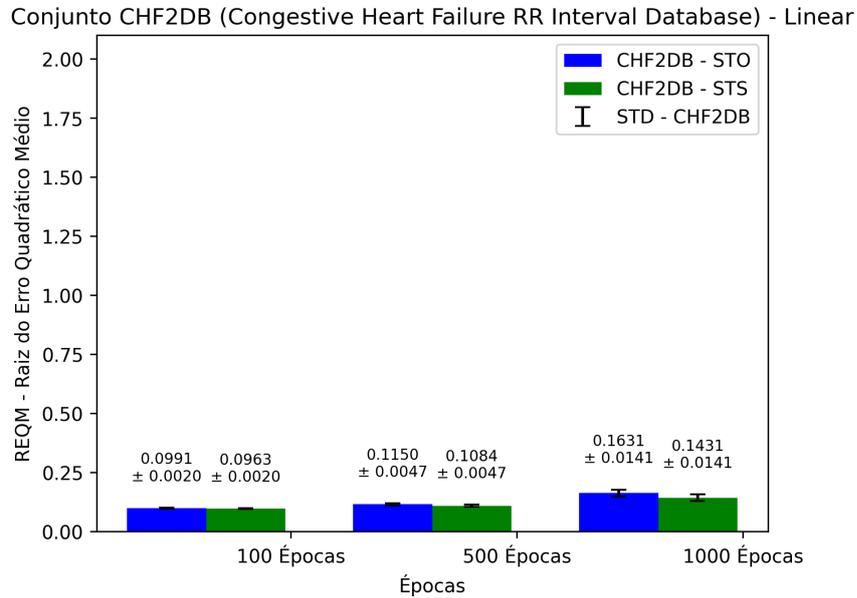


Figura 57 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

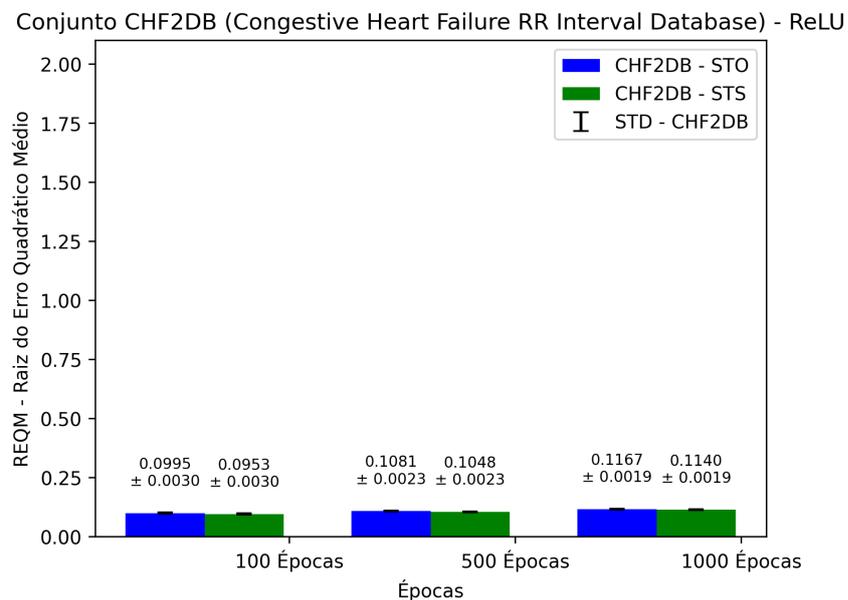
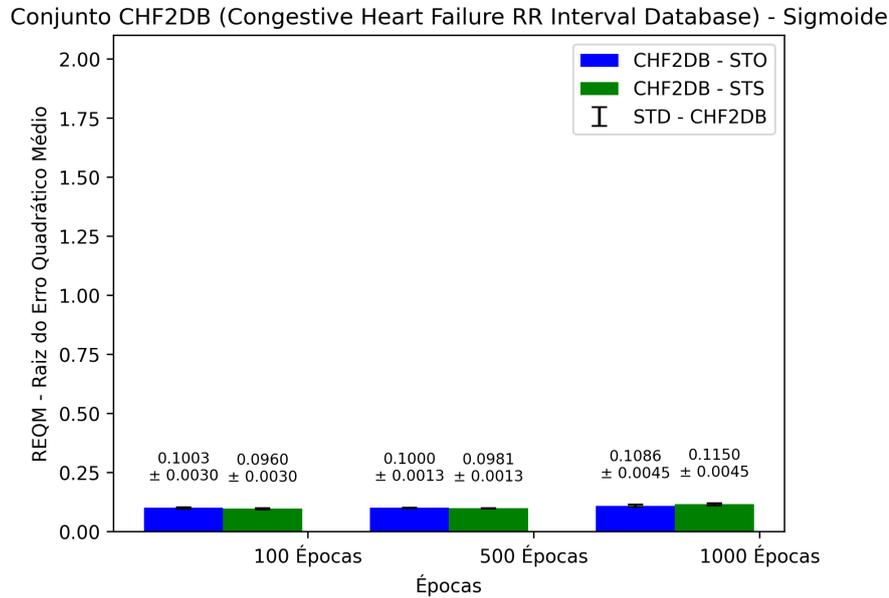


Figura 58 – Comparação entre os resultados originais (CHF2DB - STO) e os resultados substitutos (CHF2DB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



A função de ativação linear gerou os melhores resultados para as STO e STS neste conjunto durante as 100 épocas de treinamento, registrando 0.0991 e 0.0963, respectivamente. Entretanto, ocorre aumento na REQM à medida que o número de épocas cresce.

O mesmo foi observado para a função de ativação relu, que proporcionou uma REQM de 0.0995 para as STO e 0.0953 para as STS nas 100 épocas.

Em relação à função de ativação sigmoide, as STO obtiveram seu melhor resultado nas 500 épocas (0.1000), enquanto as STS alcançaram a melhor REQM nas 100 épocas (0.0960).

Figura 59 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação linear.

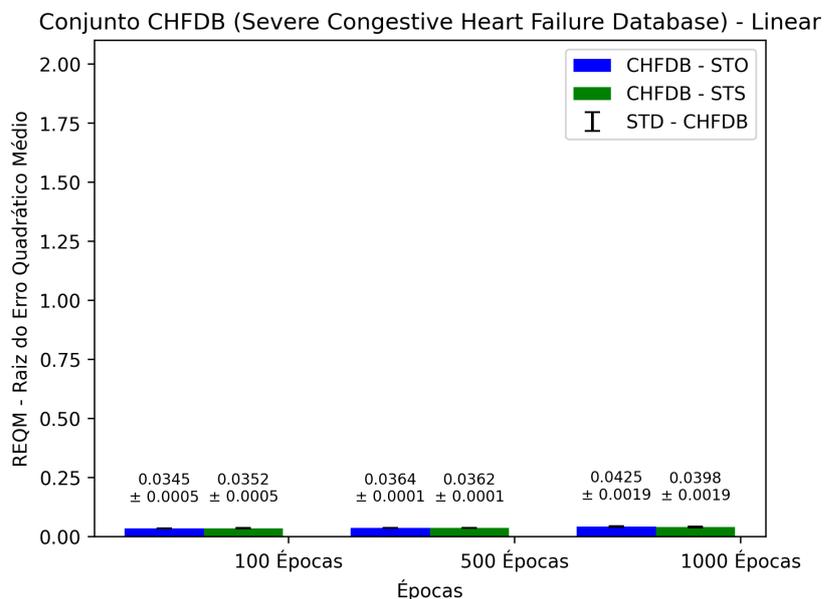


Figura 60 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação relu.

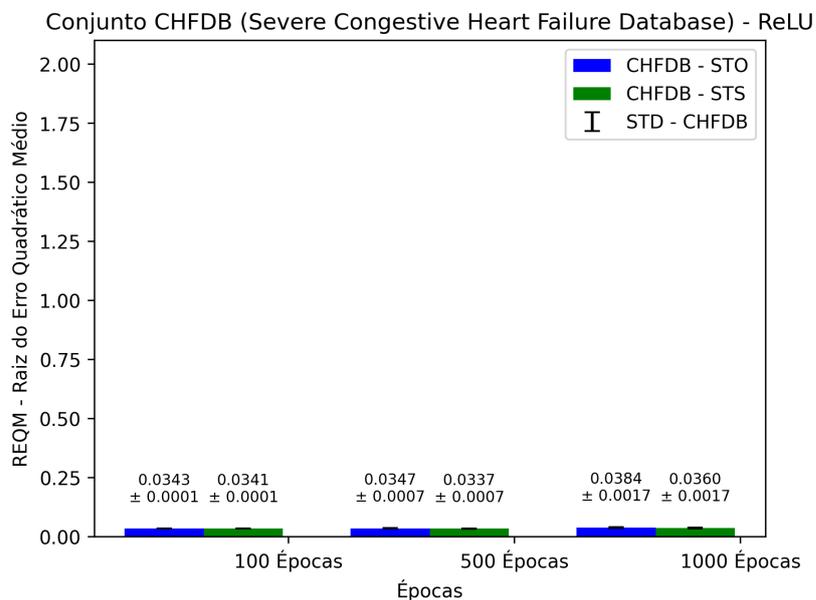
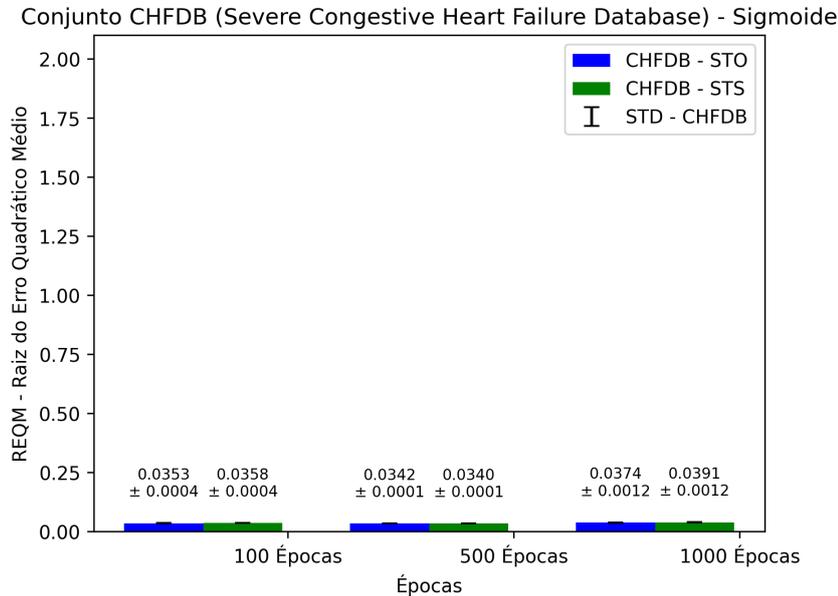


Figura 61 – Comparação entre os resultados originais (CHFDB - STO) e os resultados substitutos (CHFDB - STS) obtidos através do treinamento de redes neurais com uma camada de entrada, uma camada intermediária com 50 unidades e uma camada de saída. Utilizando função de ativação sigmoide.



Ao analisar os resultados do conjunto CHFDB, fica evidente que os melhores desempenhos para as séries STO e STS, utilizando a função de ativação linear, foram alcançados nas 100 épocas, atingindo valores de 0.0345 e 0.0352, respectivamente. Contudo, houve um aumento na REQM nas épocas subsequentes.

Para a função de ativação relu, o ponto de menor REQM para as STO foi observado nas 100 épocas (0.0343), enquanto para as STS ocorreu nas 500 épocas (0.0337).

No caso da função sigmoide, os resultados mais favoráveis foram obtidos com 500 épocas de treinamento para ambas as séries, registrando 0.0342 para STO e 0.0340 para STS, seguido por um aumento do valor nas 1000 épocas de treinamento.

No primeiro teste realizado 5.1.1, observamos que, apesar dos valores baixos de REQM para alguns conjuntos de dados em determinadas funções de ativação ou números de épocas, é evidente a disparidade de valores ao compararmos as séries temporais originais (STO) com as séries temporais substitutas (STS). Por exemplo, no conjunto de dados NSR2DB, notamos que a função de ativação linear produziu o melhor resultado (0.0799) para as STO durante as 1000 épocas de treinamento do modelo, enquanto para as STS ocorreu em 500 épocas (0.0406), com um aumento do valor nas 1000 épocas.

Para a função de ativação relu, no conjunto STO, houve uma redução da REQM nas 500 épocas; no entanto, para as STS, o valor manteve-se estável. No caso da função sigmoide, as STO apresentaram uma leve queda no valor durante as 500 épocas, mantendo-se estáveis com 0.0816, enquanto as STS atingiram o melhor valor de 0.0389 nas 1000 épocas.

Em alguns conjuntos e com determinadas funções de ativação e números de épocas, é possível identificar resultados próximos entre as STO e as STS, como no caso dos conjuntos CHF2DB e CHFDB nas funções de ativação linear, relu e sigmoide. Para a função de ativação linear, essa proximidade pode ser identificada a partir das 1000 épocas em ambos os conjuntos. A função de ativação relu forneceu resultados um pouco mais elevados quando comparados com outras funções de ativação; no entanto, é possível notar a proximidade dos valores REQM entre STO e STS, ocorrendo com sutis diferenças na função sigmoide para esses mesmos conjuntos.

Como resultado da busca por melhorias, conduzimos um segundo teste 5.1.2, que envolveu a adição de uma camada intermediária à rede neural. Em comparação com o primeiro teste, observamos que alguns conjuntos de dados apresentaram resultados mais próximos entre séries temporais originais (STO) e séries temporais substitutas (STS) para determinadas funções de ativação.

Por exemplo, nos conjuntos LTAFDB, a função de ativação linear produziu resultados de 0.1354 para STO e 0.1338 para STS, enquanto no conjunto CHFDB, também para a função linear, obtivemos REQM de 0.0425 para STO e 0.0413 para STS.

Entretanto, nem todos os conjuntos se beneficiaram dessa adição. No conjunto LTSTDB, apesar dos valores próximos na função de ativação relu, esses valores são consideravelmente elevados para a REQM. Além disso, nas funções de ativação linear e sigmoide, ainda há uma diferença perceptível.

Outro aspecto a destacar é que, em alguns casos, o melhor resultado para as STO foi alcançado em um número específico de épocas, enquanto para as STS ocorreu em um número diferente de épocas.

A combinação de diferentes funções de ativação em redes neurais podem ser uma abordagem eficaz, sendo assim, foi realizado o teste descrito em 5.1.3. Apesar da melhoria

na aproximação dos resultados de REQM entre séries temporais originais (STO) e séries temporais substitutas (STS), essa estratégia ainda não beneficiou todos os conjuntos de dados.

Em alguns casos, a alternância entre funções de ativação resultou em uma notável aproximação dos valores de REQM para STO e STS. Um exemplo positivo é o conjunto LTFADB, onde todas as combinações de funções de ativação apresentaram efeitos significativos, aproximando consideravelmente os valores de REQM.

No entanto, é importante observar que algumas combinações geraram disparidades mais expressivas, como a combinação linear-relu no conjunto CHF2DB. Nesse caso, os melhores valores foram alcançados nas 500 épocas de treinamento, resultando em 0.0440 para STO e 0.1040 para STS.

A busca por resultados mais benéficos para todos os conjuntos levou à realização do teste detalhado em 5.1.4. Uma análise dos testes anteriores destaca que esse último, demonstrou resultados mais satisfatórios ao comparar conjuntos de dados originais e substitutos, assim como diferentes funções de ativação.

A técnica de agrupar os conjuntos de dados resultou em um aumento significativo no tamanho do conjunto de treinamento, conferindo à rede neural a capacidade de aprender relações mais complexas.

É importante destacar também, que o aumento no número de épocas de treinamento não implica necessariamente uma melhora nos resultados. Embora alguns conjuntos de dados tenham apresentado melhor desempenho com um número maior de épocas, tal observação não é universal, como apresentado na tabela 7 abaixo, referente ao conjunto NSR2DB.

Tabela 7 – Comparações de desempenho de acordo com o número de épocas no conjunto NSR2DB.

| DADOS | FUNÇÃO DE ATIVAÇÃO | 100 ÉPOCAS | 500 ÉPOCAS | 1000 ÉPOCAS |
|--------------|--------------------|------------|---------------|---------------|
| NSR2DB - STO | LINEAR | 0,0269 | 0,0245 | 0,0258 |
| NSR2DB - STS | LINEAR | 0,0255 | 0,0226 | 0,0227 |
| NSR2DB - STO | RELU | 0,0241 | 0,0216 | 0,0229 |
| NSR2DB - STS | RELU | 0,0213 | 0,0198 | 0,0213 |
| NSR2DB - STO | SIGMOIDE | 0,0332 | 0,0248 | 0,0224 |
| NSR2DB - STS | SIGMOIDE | 0,0336 | 0,0222 | 0,0204 |

Para as funções de ativação linear e relu, o desempenho ótimo foi alcançado com 500 épocas tanto para as STO quanto para as STS, enquanto para a função de ativação sigmoide, o melhor resultado foi obtido com 1000 épocas para os dois conjuntos de dados (STO e STS), um dos fatores está relacionado as características dos dados e das funções de ativação.

Apesar da existência de casos isolados nos quais uma função de ativação superou a

outra, a função relu destacou-se, ajustando-se de maneira notável aos conjuntos de dados do último teste, produzindo ótimos resultados para ambas as séries temporais originais (STO) e substitutas (STS).

No último teste realizado, conduzimos a aplicação do teste t de Student, utilizando a biblioteca *scipy.stats* em *Python*. Os testes foram realizados para os cinco conjuntos de dados, incluindo as funções de ativação linear, relu e sigmoide, ao longo de 100, 500 e 1000 épocas de treinamento.

Por meio desse teste, foi possível identificar e avaliar a existência ou não de diferenças estatisticamente significativas nos conjuntos de dados STO e STS.

Os resultados obtidos para todos os conjuntos de dados, independentemente do número de épocas e da função de ativação, indicaram valores de p-valor superiores ao nível estipulado em 0.01, conforme apresentado nas imagens abaixo. Portanto, pode-se aceitar a hipótese nula de que as séries STO e STS produzem resultados indiferentes.

Figura 62 – Comparação entre os resultados do teste t de Student entre STO e STS, utilizando função de ativação linear.

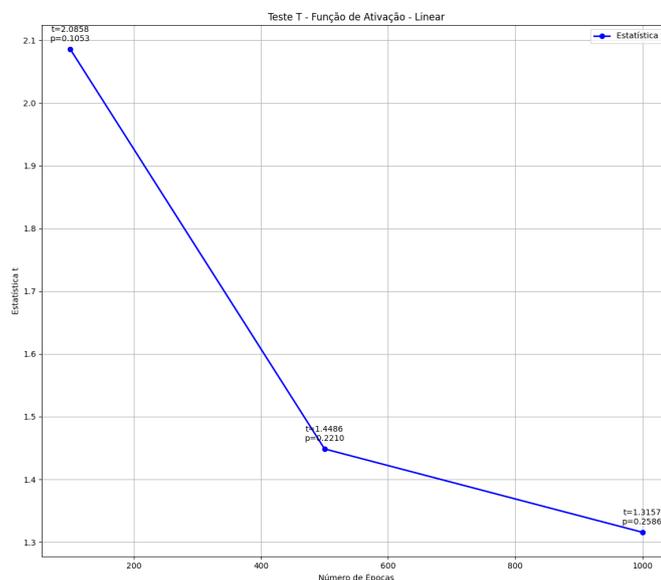


Figura 63 – Comparação entre os resultados do teste t de Student entre STO e STS, utilizando função de ativação relu.

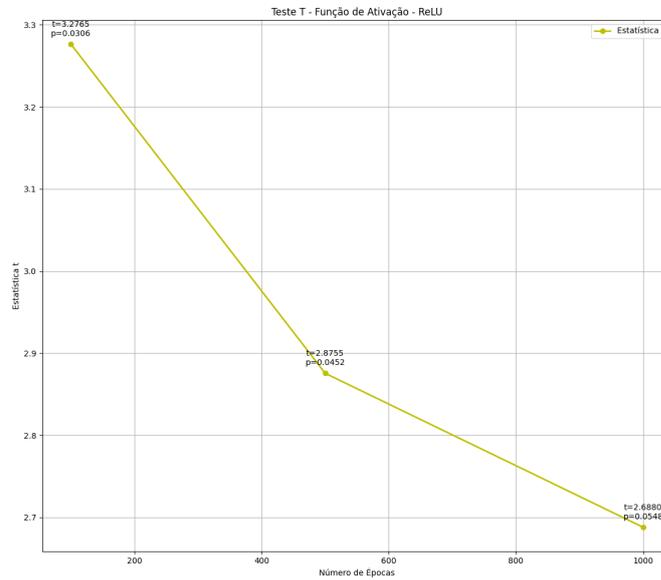
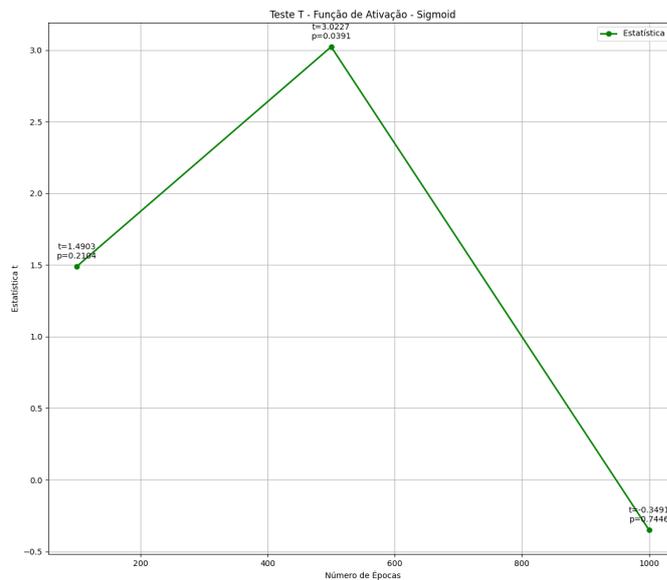


Figura 64 – Comparação entre os resultados do teste t de Student entre STO e STS, utilizando função de ativação sigmoide.



Conclusões

Neste trabalho, propõe-se a utilização de um modelo de redes neurais LSTM para realizar a aproximação da função de entropia amostral em séries temporais. Os dados de entrada consistem em séries temporais provenientes do site PhysioNet, juntamente com séries substitutas, geradas a partir das séries temporais originais.

Foram conduzidos diversos testes com o objetivo de alcançar resultados satisfatórios. A escolha desses testes é justificada pela inexistência de uma fórmula fixa para obtenção direta de resultados em redes neurais. Portanto, a exploração de diferentes testes e a manipulação de hiperparâmetros, como funções de ativação, número de camadas intermediárias/ocultas e épocas, tornam-se essenciais para ajustar o modelo de maneira eficaz. A utilização da validação cruzada durante o treinamento da rede neural desempenha um papel importante na avaliação da capacidade de generalização do modelo.

O último teste conduzido, que envolveu o agrupamento dos conjuntos de dados para o treinamento da rede neural, resultou em melhorias significativas. Não apenas foram obtidos bons resultados de REQM para as funções de ativação, mas também foi observada uma paridade entre as séries temporais originais e substitutas. Esse resultado sugere que a rede foi capaz de aprender e reproduzir padrões semelhantes em ambas as séries, consolidando a eficácia do modelo.

Aplicamos aos resultados do último teste, o teste t de Student, e ao visualizar a variabilidade entre os valores, é perceptível que as medidas REQM geradas pela rede não apresentaram diferenças significativas entre STO e STS. Portanto, as séries temporais substitutas (STS) podem ser empregadas para aumento do conjunto de dados (*data augmentation*).

Para pesquisas futuras, outras perspectivas poderão ser exploradas com objetivo de expandir a análise. Além disso, considera-se relevante a aplicação do modelo em séries temporais curtas, proporcionando uma visão abrangente do seu desempenho em diferentes contextos.

Referências

AGARAP, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

AND, D. C. Permutation entropy: Influence of amplitude information on time series classification performance. *Mathematical Biosciences and Engineering*, American Institute of Mathematical Sciences (AIMS), v. 16, n. 6, p. 6842–6857, 2019. Disponível em: <<https://doi.org/10.3934/mbe.2019342>>.

AVILA, R. et al. Revisão sistemática de visão computacional para veículos autônomos agrícolas. In: *2023 15th IEEE International Conference on Industry Applications (INDUSCON)*. [S.l.: s.n.], 2023. p. 485–492.

AZAMI, H.; ESCUDERO, J. Amplitude- and fluctuation-based dispersion entropy. *Entropy*, MDPI AG, v. 20, n. 3, p. 210, mar. 2018. Disponível em: <<https://doi.org/10.3390/e20030210>>.

BADINO, M. Bridging conceptual gaps: the kolmogorov-sinai entropy. *Isonomia*, v. 2037, p. 4348, 2014.

BERRAR, D. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, v. 1, n. April, p. 542–545, 2019.

BISWAS, H.; HASAN, M.; BALA, S. K. Chaos theory and its applications in our real life. v. 5, p. 123–140, 01 2018.

BKASSINY, M. A deep learning-based signal classification approach for spectrum sensing using long short-term memory (lstm) networks. In: *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. [S.l.: s.n.], 2022. p. 667–672.

BROWNE, M. W. Cross-validation methods. *Journal of Mathematical Psychology*, v. 44, n. 1, p. 108–132, 2000. ISSN 0022-2496. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0022249699912798>>.

CARDOSO, L. H.; PEREIRA, E. C. Teoria do caos e gestão da informação: uma integração na complexidade dos negócios e dos sistemas de informação. *Transinformação*, Pontifícia Universidade Católica de Campinas, v. 17, n. 3, p. 221–233, Sep 2005. ISSN 0103-3786. Disponível em: <<https://www.scielo.br/j/tinf/a/gc5xfJYKz3y3K7fG95Y876m/>>.

DENG, Y. Deng entropy. *Chaos, Solitons & Fractals*, Elsevier BV, v. 91, p. 549–553, out. 2016. Disponível em: <<https://doi.org/10.1016/j.chaos.2016.07.014>>.

- DURAN, G. C. et al. Reconstrução do problema da tomografia por impedância elétrica 3d com o uso de redes neurais artificiais [not available in english]. In: *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*. [S.l.: s.n.], 2021. p. 994–998.
- GOLDBERGER, A. et al. Physiobank, physiokit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, v. 101, n. 23, p. E215–20, June 2000. ISSN 0009-7322. Disponível em: <<https://doi.org/10.1161/01.cir.101.23.e215>>.
- HASAN, M. S. et al. Normalized linearly-combined chaotic system: Design, analysis, implementation, and application. *IEEE Open Journal of the Industrial Electronics Society*, v. 4, p. 486–505, 2023.
- KEUM, J.; COULIBALY, P. Sensitivity of entropy method to time series length in hydrometric network design. *Journal of Hydrologic Engineering*, American Society of Civil Engineers (ASCE), v. 22, n. 7, jul. 2017. Disponível em: <[https://doi.org/10.1061/\(asce\)he.1943-5584.0001508](https://doi.org/10.1061/(asce)he.1943-5584.0001508)>.
- KOUL, A.; BECCHIO, C.; CAVALLO, A. Cross-validation approaches for replicability in psychology. *Frontiers in Psychology*, Frontiers Media SA, v. 9, p. 1117, 2018.
- LEE, T. E. K. et al. Surrogate rehabilitative time series data for image-based deep learning. In: *2019 27th European Signal Processing Conference (EUSIPCO)*. [S.l.: s.n.], 2019. p. 1–5.
- LETELLIER, C. Estimating the shannon entropy: Recurrence plots versus symbolic dynamics. *Physical Review Letters*, American Physical Society (APS), v. 96, n. 25, jun. 2006. Disponível em: <<https://doi.org/10.1103/physrevlett.96.254102>>.
- LIU, A.; LI, J.; YE, H. A prediction model combining convolutional neural network and lstm neural network. In: *2023 2nd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*. [S.l.: s.n.], 2023. p. 318–321.
- LV, X. et al. Human gait analysis method based on sample entropy fusion alphapose algorithm. In: *2021 33rd Chinese Control and Decision Conference (CCDC)*. [S.l.: s.n.], 2021. p. 1543–1547.
- MARIETTO, M. L.; SANCHES, C.; MEIRELES, M. Teoria do caos: uma contribuição para a formação de estratégias. *Revista Ibero Americana de Estratégia*, Universidade Nove de Julho, v. 10, n. 3, p. 66–93, 2011.
- MARTINEZ-GARCIA, M. et al. Measuring system entropy with a deep recurrent neural network model. In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. IEEE, 2019. Disponível em: <<https://doi.org/10.1109/indin41052.2019.8972068>>.
- MESQUITA, C. M.; OLIVEIRA, R.; PEREIRA, A. C. M. Utilização de uma rede neural lstm e teste da razão de variância para previsões em séries de ativos da bovespa. *Acesso em*, v. 11, 2020.
- MOLAVIPOUR, S. et al. Neural estimator of information for time-series data with dependency. *Entropy*, MDPI AG, v. 23, n. 6, p. 641, maio 2021. Disponível em: <<https://doi.org/10.3390/e23060641>>.

- PINCUS, S. M. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, Proceedings of the National Academy of Sciences, v. 88, n. 6, p. 2297–2301, mar. 1991. Disponível em: <<https://doi.org/10.1073/pnas.88.6.2297>>.
- POLEPALLY, V. et al. A deep learning approach for prediction of stock price based on neural network models: Lstm and gru. In: *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2021. p. 1–4.
- RADHAKRISHNAN, N.; WILSON, J.; HAWK, R. Detecting nonlinear patterns in physiological signals. In: *Proceedings 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine. ITAB-ITIS 2000. Joint Meeting Third IEEE EMBS International Conference on Information Technol.* [S.l.: s.n.], 2000. p. 365–370.
- RAY, A.; RAY, H. Study of overfitting through activation functions as a hyper-parameter for image clothing classification using neural network. In: *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2021. p. 1–5.
- RICHMAN, J. S.; MOORMAN, J. R. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, American Physiological Society, v. 278, n. 6, p. H2039–H2049, jun. 2000. Disponível em: <<https://doi.org/10.1152/ajpheart.2000.278.6.h2039>>.
- SCARCIGLIA, A. et al. Quantifying partition-based kolmogorov-sinai entropy on heart rate variability: a young vs. elderly study. In: IEEE. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. [S.l.], 2021. p. 5469–5472.
- SCARCIGLIA, A. et al. Multiscale partition-based kolmogorov-sinai entropy: a preliminary hrv study on heart failure vs. atrial fibrillation. In: IEEE. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. [S.l.], 2022. p. 131–134.
- SHARMA, S.; SHARMA, S.; ATHAIYA, A. Activation functions in neural networks. *Towards Data Sci*, v. 6, n. 12, p. 310–316, 2017.
- SILVA, F. et al. Applying lyapunov exponents in heart rate time series to identify the anaerobic threshold in healthy men. In: *2012 Computing in Cardiology*. [S.l.: s.n.], 2012. p. 629–632.
- SIMON, R. A.; KUMAR, P. B. V. A nonlinear sea clutter analysis using chaotic system. In: *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2013. p. 1–5.
- SU, F. The chaos theory and its application. *Journal of Physics: Conference Series*, IOP Publishing, v. 2012, n. 1, p. 012118, sep 2021. Disponível em: <<https://dx.doi.org/10.1088/1742-6596/2012/1/012118>>.
- THEILER, J. et al. Testing for nonlinearity in time series: the method of surrogate data. *Physica D: Nonlinear Phenomena*, Elsevier BV, v. 58, n. 1-4, p. 77–94, set. 1992. Disponível em: <[https://doi.org/10.1016/0167-2789\(92\)90102-s](https://doi.org/10.1016/0167-2789(92)90102-s)>.

VELICHKO, A.; HEIDARI, H. A method for estimating the entropy of time series using artificial neural networks. *Entropy*, v. 23, n. 11, 2021. ISSN 1099-4300. Disponível em: <<https://www.mdpi.com/1099-4300/23/11/1432>>.

WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, v. 26, n. 1, p. 98–117, 2009.

YIN, Y.; SHANG, P. Weighted permutation entropy based on different symbolic approaches for financial time series. *Physica A: Statistical Mechanics and its Applications*, Elsevier BV, v. 443, p. 137–148, fev. 2016. Disponível em: <<https://doi.org/10.1016/j.physa.2015.09.067>>.

ZENG, X.; PIELKE, R. A.; EYKHOLT, R. Chaos theory and its applications to the atmosphere. *Bulletin of the American Meteorological Society*, American Meteorological Society, v. 74, n. 4, p. 631–644, 1993. ISSN 00030007, 15200477. Disponível em: <<http://www.jstor.org/stable/26230484>>.

ZHANG, H.; DENG, Y. Entropy measure for orderable sets. *Information Sciences*, Elsevier BV, v. 561, p. 141–151, jun. 2021. Disponível em: <<https://doi.org/10.1016/j.ins.2021.01.073>>.

ZUNINO, L. et al. Permutation entropy based time series analysis: Equalities in the input signal can lead to false conclusions. *Physics Letters A*, Elsevier BV, v. 381, n. 22, p. 1883–1892, jun. 2017. Disponível em: <<https://doi.org/10.1016/j.physleta.2017.03.052>>.