

Estimação de parâmetros de modelo do sistema cardiovascular humano: SVR, CNN e CNN-LSTM

DOUGLAS SILVA,¹ JORGE LEANDRO,² AND LUÍS FELIPE³

¹dhmeis@ic.ufal.br, ²jsl2@ic.ufal.br, ³felipemmha@ic.ufal.br

Abstract: Este trabalho compara o desempenho de três técnicas aprendizado de máquina empregadas na estimação de parâmetros de um modelo do sistema cardiovascular humano. Para melhor comparar os resultados dos experimentos, foram utilizados os mesmos dados e o mesmo métodos de validação para os três algoritmos: *Support Vector Regression* (SVR); *Convolutional Neural Networks* (CNN) e; uma combinação de *Convolutional Neural Networks* com Long Short-Term Memory (CNN-LSTM). Para comparação dos modelos, foram calculadas as médias dos Erros Médios Quadráticos, onde obtivemos os resultados: **SVR:** 0,05954; **CNN:** 0,0572 e **CNN-LSTM:** 0,0621. Classificando assim, o modelo CNN como o melhor desempenho obtido nos experimentos investigados.

1. Introdução

Modelos do sistema cardiovascular humano são de extrema relevância no estudo do comportamento hemodinâmico de pacientes em estado grave, uma vez que tais modelos permitem, ao alterar os valores dos parâmetros, simular o comportamento do coração desses pacientes. Dentre as variáveis geradas pelo modelo utilizado neste experimento, a única que não está diretamente no coração é a pressão arterial sistêmica. Dessa forma, torna-se evidente que quanto mais "afastada" a variável está do coração, menos agressivo é o procedimento para obtê-la. Logo, considerando que as formas de aquisição dessa variável apresentam caráter não invasivo, o estudo de sua correlação com os parâmetros do modelo mostra-se essencial para permitir aos médicos um maior conhecimento do sistema cardiovascular daquele paciente específico. Assim, os procedimentos de transplante de coração ou implantes de dispositivos de assistência (bombas mecânicas) ficam, sem dúvida, muito mais seguros.

2. Métodos

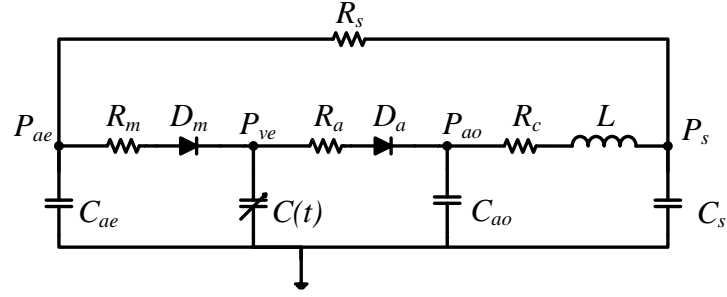
Para os fins deste trabalho, foram realizados três experimentos utilizando a base de dados descrita na seção 2.1. Cada experimento consiste na aplicação de treino e validação com um dos algoritmos escolhidos: SVR (seção 2.2); CNN (seção 2.3) e uma combinação CNN-LSTM (seção 2.4).

A métrica escolhida para avaliação dos modelos foi a Raiz do Erro Médio Quadrático (equação 1) - por se tratar de uma tarefa de regressão, essa métrica proporciona uma boa avaliação do desempenho do modelo. Os dados são divididos em 70% para treino e 30% para validação. O modelo é treinado com a base de treino e estima valores de saída para a base de teste. Os valores estimados e os valores reais são utilizados para o cálculo da métrica de avaliação, que determina o desempenho do modelo para o experimento ao qual foi submetido.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{d_i - f_i}{\sigma_i} \right)^2} \quad (1)$$

2.1. Base de dados

O modelo do sistema cardiovascular utilizado neste trabalho foi proposto em [1], e é representado pelo circuito elétrico apresentado na Figura 1. Trata-se de um modelo 0D de quinta ordem capaz de simular o comportamento de variáveis hemodinâmicas como pressão, fluxo e volume



h

Figura 1. Circuito elétrico utilizado para representar o modelo 0D do sistema cardiovascular (adaptado de [1]).

sanguíneo. As variáveis de tensão e corrente elétrica deste circuito são equivalentes a: pressão na aorta (P_{ao}); fluxo na aorta (Q_{ao}); pressão do ventrículo esquerdo (P_{ve}); pressão sistêmica (P_s); pressão no átrio esquerdo (P_{ae}). As válvulas mitral e aórtica são representadas de modo ideal, com relação aos movimentos de abertura e fechamento, pelo conjunto resistor-diodo $R_m - D_m$ e $R_a - D_a$, respectivamente. A complacência aórtica é representada pelo capacitor C_{ao} e a complacência do ventrículo esquerdo é modelada pelo capacitor $C(t)$ variante no tempo.

Para o experimento em questão, o conjunto de parâmetros $R_s, R_m, R_a, R_c, C_{ae}, C_s, C_{ao}, L, E_{max}, E_{min}$ será referenciado como o vetor θ . Com o objetivo de manter um padrão nos dados, cada um dos parâmetros será gerado aleatoriamente com valores entre 50 à 200% de seu valor ideal, conforme Tabela 1 disponível em [1]. A base completa será composta por 1000 (mil) vetores θ

Tabela 1. Valores de referência utilizados em [1]

Parâmetros	Valor	Mecanismo fisiológico
Resistências ($mmHg.s/ml$)		
R_s	1,0000	Vascular Sistêmica (RVS)
R_m	0,0050	Mitral
R_a	0,0010	Aórtica
R_c	0,0398	Característica
Complacências ($ml/mmHg$)		
$C(t)$	Variante no tempo	Ventrícular esquerda
C_{ae}	4,4000	Átrial esquerda
C_s	1,3300	Sistêmica
C_{ao}	0,0800	Aórtica
Inertância ($mmHg.s^2/ml$)		
L	0,0005	Sanguinea na aorta

Para cada vetor θ será gerado um ciclo da pressão sistêmica P_s . Considerando que o objetivo do trabalho é estimar θ baseado somente na curva P_s , esta será inserida como entrada nos métodos de estimação e o vetor θ como saída.

2.2. Máquinas de Vetores de Suporte para Regressão - SVR

Para problemas com duas classes separáveis, separadores lineares são muito utilizados em aprendizagem de máquina. Entretanto, quando uma linha não é suficiente para separar um conjunto de dados em duas classes, métodos mais sofisticados como Máquinas de Vetores de Suporte (SVM) são utilizadas. As SVMs, como o nome já diz, utiliza vetores de suporte que referenciam os dados e são mais próximos da superfície de decisão, o hiperplano.

Em geral, dado um conjunto de entradas e saídas de treinamento, a SVM retorna um conjunto de pesos, um para cada característica, cuja combinação linear prediz o valor da saída. Entretanto, diferentemente de redes neurais, SVMs utilizam otimização para maximizar a margem para reduzir o número de pesos nulos e assim, os pesos não nulos correspondem aos vetores de suporte.

SVMs também podem ser utilizadas para regressão, mas para encontrar uma função que mapeie uma entrada em uma saída, a essas abordagens, dá-se o nome de *Support Vector Regression* (SVR). Na base de dados utilizada neste trabalho, não há classes, foi preciso realizar regressões lineares que mapeassem os parâmetros de entrada nos dez parâmetros de saída. Para fazer isso foi utilizado uma SVR para cada parâmetro de saída, com isso, o modelo final resulta em uma combinação linear dos pesos obtidos em cada SVR.

A SVR utilizada foi configurada com a biblioteca de aprendizado de máquina *scikit-learn* na linguagem de programação Python. O núcleo utilizado por todas as SVRs foi a função de base radial (RBF). A base de treinamento foi feita com 70% das amostras, enquanto a base de teste foi feita utilizando 30% das amostras.

2.3. Convolutional Neural Networks - CNN

Redes neurais têm se destacado muito nos últimos anos, e esse crescimento de popularidade pode ser explicado por esta ser uma técnica resistente ao mal da dimensionalidade, o que se faz bastante útil na sociedade guiada a dados na qual vivemos hoje em dia. Além disso, redes neurais têm uma vasta abrangência de problemas onde podem ser aplicadas. Parte dessa abrangência se deve as diferentes categorias de redes neurais, dentre estas estão as chamadas Redes Neurais Convolucionais (do inglês Convolutional Neural Networks - CNN) que se destacam pela capacidade de extração de características importantes dos dados de entrada. Características essas que podem fugir aos olhos do ser humano, mesmo que este seja especialista no domínio de aplicação.

A proposta para esse experimento é, principalmente, investigar o desempenho que um modelo CNN proporciona para tarefa de regressão proposta neste trabalho. Para uma melhor compreensão, é necessário separar as responsabilidades das diferentes camadas da arquitetura da rede neural, onde, no modelo proposto por esse sessão, as camadas de início são convolucionais e agem para extrair características importantes sobre os dados, com o intuito de reduzir dimensionalidade ao passo que destaca informações representativas dos dados de entrada. Os resultados das camadas convolucionais são então fornecidos como entrada para as camadas seguintes (densas) que utilizarão essas características para realização da tarefa de regressão.

Para realizar uma otimização empírica de hiper-parametros do modelo, foram projetadas três arquiteturas de CNN: para uma, duas ou três camadas convolucionais (Tabela 2, Tabela 3 e Tabela 4, respectivamente). O modelo que obteve o melhor desempenho será utilizado para os fins de comparação com as outras estratégias descritas neste trabalho.

Tabela 2. CNN - 1

Camada	Tipo	Função de ativação	Formato de saída	<i>Kernel</i>	Filtros
0	Entrada	-	8000x1		
1	Convolução 1D	ReLU	20x32	8	32
1.5	<i>Max Pooling 1D</i>	-	10x32		
2	<i>Flatten</i>	-	320	-	-
3	<i>Dropout (0.4)</i>	-	320	-	-
4	Densa	ReLU	50	-	-
5	Densa	Linear	10	-	-

Tabela 3. CNN - 2

Camada	Tipo	Função de ativação	Formato de saída	<i>Kernel</i>	Filtros
0	Entrada	-	8000x1		
1	Convolução 1D	ReLU	20x32	8	32
1.5	<i>Max Pooling 1D</i>	-	10x32		
2	Convolução 1D	ReLU	8x64	3	64
2.5	<i>Max Pooling 1D</i>	-	4x64		
3	<i>Flatten</i>	-	256	-	-
4	<i>Dropout (0.4)</i>	-	256	-	-
5	Densa	ReLU	50	-	-
6	Densa	Linear	10	-	-

Tabela 4. CNN - 3

Camada	Tipo	Função de ativação	Formato de saída	<i>Kernel</i>	Filtros
0	Entrada	-	8000x1		
1	Convolução 1D	ReLU	20x32	8	32
1.5	<i>Max Pooling 1D</i>	-	10x32		
2	Convolução 1D	ReLU	8x64	3	64
2.5	<i>Max Pooling 1D</i>	-	4x64		
3	Convolução 1D	ReLU	2x96	3	96
3.5	<i>Max Pooling 1D</i>	-	1x96		
4	<i>Flatten</i>	-	96	-	-
5	<i>Dropout (0.4)</i>	-	96	-	-
6	Densa	ReLU	50	-	-
7	Densa	Linear	10	-	-

2.4. CNN-LSTM

Esta etapa tem como objetivo unir a capacidade de destaque de características da CNN com um tipo de rede neural capaz de obter informações temporais. Redes LSTM (Long Short-Term Memory) são classificadas como redes neurais recorrentes, também conhecidas pelo termo em inglês *RNN - Recurrent Neural Networks*, que são modelos sequenciais adequados para o estudo do comportamento temporal, uma vez que permitem a correlação entre informações passadas e informações atuais [2]. Destaca-se o uso neste trabalho pois o sinal avaliado varia no tempo. A estrutura da rede neural utilizada pode está disponível na tabela 5

Tabela 5. Estrutura da rede neural CNN-LSTM

Camada	Tipo	Função de ativação	Formato de saída	Kernel	Filtros	Conectada à
0	Entrada	-	8000x1			-
1	Conv 1D	ReLU	10x32	8	32	0
1.1	<i>Max Pooling 1D</i>	-	10x32			1
2	Convolução 1D	ReLU	8x64	3	64	1.1
2.1	<i>Max Pooling 1D</i>	-	8x64	-	-	2
3	LSTM	-	20	-	-	2.1
4	<i>Flatten</i>	-	256	-	-	2.1
5	<i>Flatten</i>	-	20	-	-	3
6	Concatenar	-	276	-	-	4 e 5
7	Densa	ReLU	50	-	-	6
8	Densa	ReLU	50	-	-	7
9	Densa	Linear	10	-	-	8

3. Resultados

3.1. Resultados utilizando Máquinas de Vetores de Suporte para Regressão

As figuras nas Tabelas 7 e 8 mostram a distribuição dos pontos que foram preditos pelas SVRs em cada parâmetro (em vermelho) e diferencia dos pontos da base original (em azul), com este dado foi calculado os valores de erro médio quadrado mostrados na tabela 6, pelas figuras, é possível ver nitidamente que o parâmetro L não conseguiu ter valores preditos pela SVR utilizada, entretanto alguns parâmetros conseguiram obter uma boa predição do modelo obtido, como o caso dos parâmetros R_s , R_a , R_c e a elastância máxima.

Tabela 6. Tabela de erro médio quadrático e coeficiente de determinação obtidos com SVR para cada parâmetro.

Parâmetros	MSE
Resistências (<i>mmHg.s/ml</i>)	
R_s	0,0129
R_m	0,0824
R_a	0,0844
R_c	0,0732
C_{ae}	0,0636
C_s	0,0144
C_{ao}	0,0860
L	0,0837
E_{max}	0,0089
E_{min}	0,0859

Obtendo, assim, uma média 0,05954 de Erros Médios Quadráticos.

Tabela 7. E_{max} e E_{min} preditos e simulados

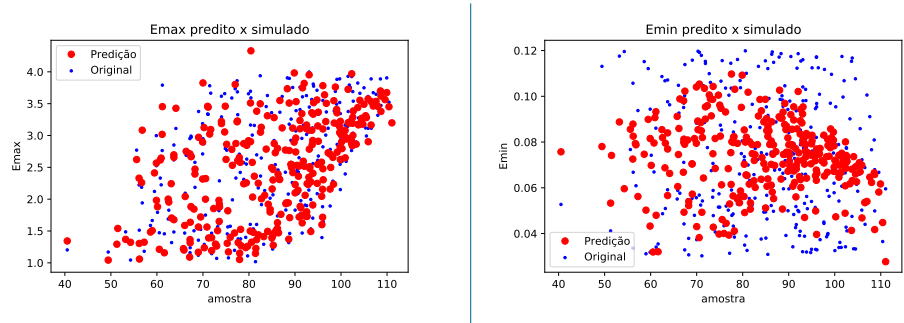
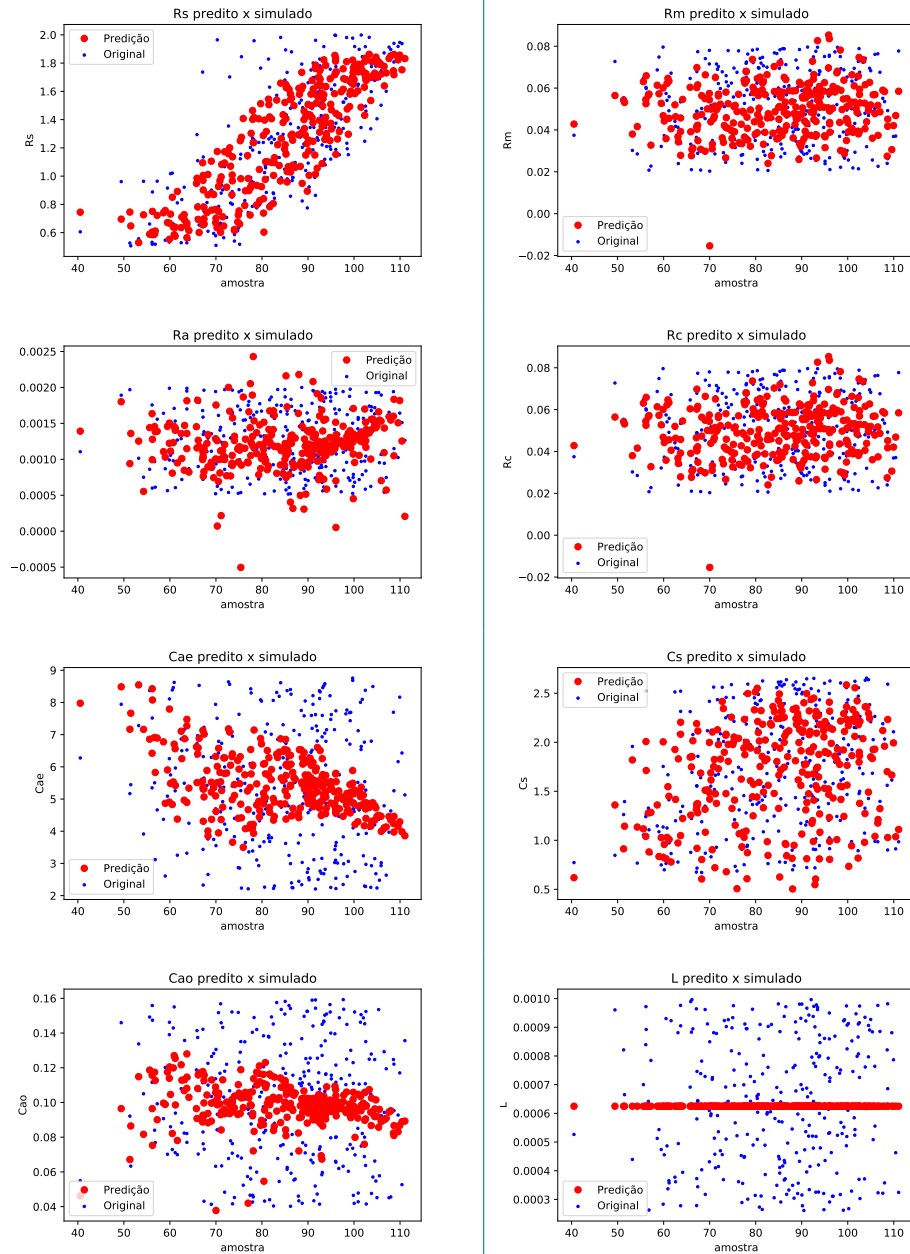


Tabela 8. Gráficos de pontos para os parâmetros simulados em comparação com os pontos preditos pela SVR.



3.2. Resultados utilizando CNN-1D

As curvas de treinamento e validação dos modelos avaliados estão presente na Figura 2. Os erros médios quadráticos presentes em Tabela 9, Tabela tal2 e Tabela tal3, foram estimados com os modelos de uma, duas e três camadas convolucionais, respectivamente.

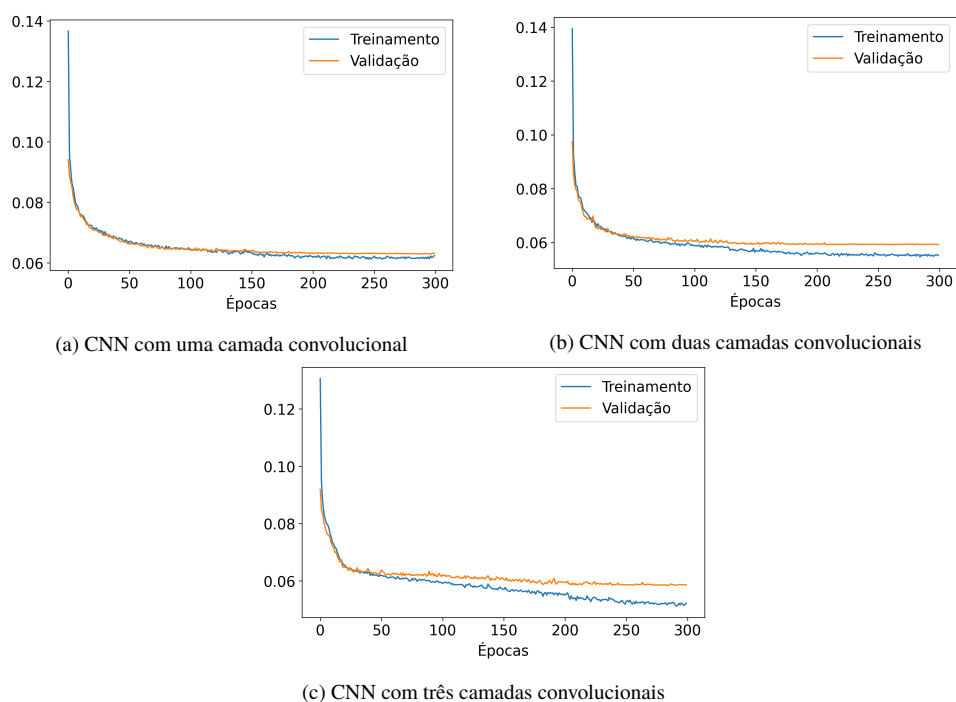


Figura 2. Curvas de perda para os modelos de CNN com uma, duas ou três camadas convolucionais

Tabela 9. Erro médio quadrático - CNN com uma, duas e três camadas convolucionais

Parâmetros	CNN-1	CNN-2	CNN-3
R_s	0,0163	0,0159	0,0135
R_m	0,0764	0,0743	0,0731
R_a	0,0814	0,0795	0,0814
R_c	0,0889	0,0774	0,0630
C_{ae}	0,0666	0,0611	0,0659
C_s	0,0218	0,0159	0,0175
C_{ao}	0,0837	0,0833	0,0842
L	0,0920	0,0900	0,0827
E_{max}	0,0166	0,0125	0,0122
E_{min}	0,0730	0,0739	0,0782

Para escolher o modelo com melhor desempenho, foi calculada a média simples dos Erros Médios Quadráticos obtidos. Os resultados foram: **CNN-1**: 0.06172; **CNN-2**: 0.05843 e **CNN-3**: 0.05724.

Desse modo, o modelo com três camadas convolucionais, por ter obtido a menor média de erros médios quadráticos foi eleito o melhor modelo de CNN e seus resultados serão utilizados

para comparação com as demais estratégias empregadas neste trabalho.

3.3. Resultados utilizando CNN-LSTM

Como mencionado anteriormente o método de comparação desta pesquisa consiste no erro quadrático médio e o mesmo foi usado como função de perda na etapa de treinamento da rede. Essa etapa, onde a função é aplicada nas bases de dados de treinamento e validação, pode ser vista na Figura 3.

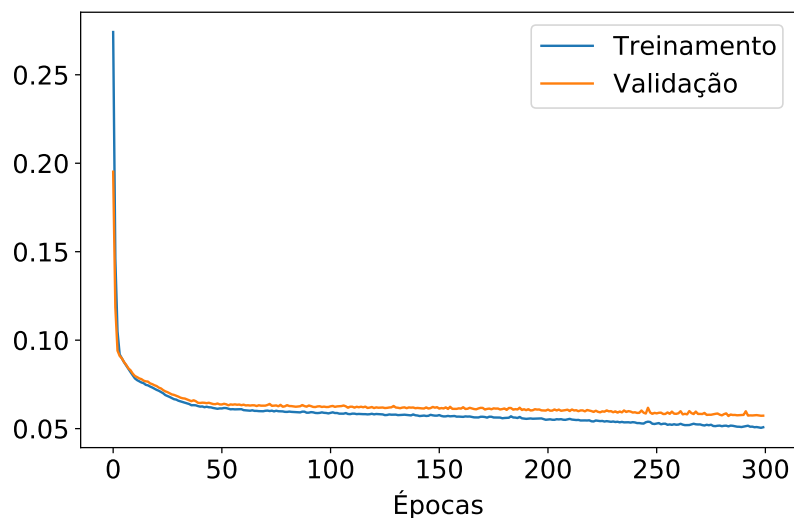


Figura 3. Gráfico da função de perda da etapa de treinamento da estrutura CNN-LSTM, aplicada as bases de dados de treinamento e validação).

Em seguida a rede neural treinada foi usada para fazer as previsões de toda a base de teste e as previsões foram comparadas com os valores reais através do erro quadrático médio, conforme Tabela 10

Tabela 10. Erro quadrático médio da base de dados de teste utilizando a estrutura CNN-LSTM

Parâmetros	MSE
R_s	0,0113
R_m	0,0787
R_a	0,0905
R_c	0,0575
C_{ae}	0,0538
C_s	0,0147
C_{ao}	0,1106
L	0,0958
E_{max}	0,0112
E_{min}	0,0969

Dessa maneira, a média de Erros Quadráticos Médios foi 0,0621 para a abordagem com CNN-LSTM.

4. Discussão

Um resumo dos resultados obtidos pelos modelos a serem comparados está presente na Tabela 11. Na Tabela 12, estão presentes as médias dos Erros Médios Quadráticos obtidos pelos modelos em comparação.

Tabela 11. Erro médio quadrático - SVR, CNN e CNN-LSTM

Parâmetros	SVR	CNN	CNN-LSTM
R_s	0,0129	0,0135	0,0113
R_m	0,0824	0,0731	0,0787
R_a	0,0844	0,0814	0,0905
R_c	0,0732	0,0630	0,0575
C_{ae}	0,0636	0,0659	0,0538
C_s	0,0144	0,0175	0,0147
C_{ao}	0,0860	0,0842	0,1106
L	0,0837	0,0827	0,0958
E_{max}	0,0089	0,0122	0,0112
E_{min}	0,0859	0,0782	0,0969

Tabela 12. Média de Erros médios quadráticos - SVR, CNN e CNN-LSTM

	SVR	CNN	CNN-LSTM
Média	0,05954	0.0572	0,0621

Os valores obtidos foram bem semelhantes, no entanto, a diferença apresentada elege o modelo com apenas convolucionais (CNN) como o melhor modelo para esse problema de regressão, quando comparado aos outros dois modelos. Uma outra informação importante que os resultados nos fornecem é de que, o modelo com arquitetura mais complexa (CNN-LSTM) foi o que obteve o pior desempenho, o que pode ser considerado contra-intuitivo, pois se conjectura que modelos com uma maior arquitetura teriam um melhor desempenho. Esse tipo de problema é um dos fatores mais importantes que motivam a existência de trabalhos como este.

Uma discussão pertinente é que, dentre os modelos apresentados, o que possui o menor custo computacional é o modelo SVR, e seu desempenho foi dentro da média entre o primeiro colocado e o segundo colocado, se pusermos em ordem os desempenhos modelos avaliados. Dessa maneira, um cenário onde os recursos computacionais são muito limitados ou o tempo de resposta deve ser mínimo, o trade-off de desempenho vs. complexidade age em favor do modelo com SVR.

Referências

1. M. A. Simaan, A. Ferreira, S. Chen, J. F. Antaki, and D. G. Galati, "A dynamical state space representation and performance analysis of a feedback-controlled rotary left ventricular assist device," *IEEE Transactions on Control. Syst. Technol.* **17**, 15–28 (2008).
2. W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid* **10**, 841–851 (2017).