Rede neural com conexões densas para previsão de séries temporais de longo prazo

André Quintiliano Bezerra Silva

Rede neural com conexões densas para previsão de séries temporais de longo prazo

André Quintiliano Bezerra Silva

Orientador: *Prof. Dr. Edson Takashi Matsubara*

Tese apresentada à Faculdade de Computação - FACOM-UFMS como parte dos requisitos necessários à obtenção do título de Doutor em Ciência da Computação.

UFMS - Campo Grande
Fevereiro/2024

Dedicatória

Ao meu filho,
Theo Araújo Quintiliano,

À minha família.

Agradecimentos

Dedico este trabalho a várias pessoas especiais que tornaram esta jornada possível e significativa.

À minha querida esposa, Janusa Soares, você é mais do que minha companheira, é a força que sustenta meus sonhos. Sua paciência, amor e apoio incondicional não apenas me guiaram, mas também foram a luz nas jornadas mais desafiadoras. Nos momentos de estresse e nas rotinas diárias, você esteve sempre ao meu lado, compartilhando sabedoria e serenidade. Cada pequeno gesto seu, desde um sorriso acolhedor até um conselho sábio, teceu a rede de apoio que me permitiu alcançar este feito. Você não é apenas a luz dos meus dias; você é a essência das minhas maiores conquistas e alegrias. Obrigado por ser você, por tudo que faz e por tudo que somos juntos.

Aos meus pais, José Quintiliano e Hilda Bezerra, que me deram a vida e a educação que formaram a base de tudo o que sou hoje. Vocês me ensinaram o valor do trabalho duro, da integridade e da humildade. Este diploma é tanto meu quanto de vocês. Aos meus irmãos, Artur e Andrea, que sempre estiveram lá para me apoiar, mesmo nos momentos mais difíceis. A amizade e o amor de vocês são inestimáveis e me deram a força para continuar, mesmo quando as coisas pareciam impossíveis.

Para meu filho amado, que neste momento ainda aguarda seu primeiro olhar para o mundo, mas que, espero, estará presente no dia em que eu celebrar a conquista do título de doutor. Mesmo que ainda em seus primeiros dias de vida, e incapaz de entender as complexidades deste momento, sua presença já é uma inspiração imensa para mim. Este marco não é apenas um reflexo do meu esforço, mas também um símbolo do futuro brilhante que desejo construir para você. Embora ainda não compreenda as palavras e significados, saiba que cada passo que dei foi também pensando em você e no mundo que quero ajudar a moldar para sua jornada. É uma alegria imensa saber que você, meu filho, será parte desse dia especial e de todos os outros que virão.

Aos meus colegas do laboratório LIA, Lucas, Pedro, Mauro, e Kenzo. Agradeço pela ajuda direta e inderetamente para finalizar o meu doutorado. Embora tenhamos cursados poucas disicplinas, foi ótimo ajudar e poder ser ajudado. Aos professores da UFMS que contribuíram para este trabalho através das disciplinas e à Faculdade de Computação (FACOM) pela disponibilização da infraestrutura.

Aos amigos do IFMS, que me proporcionaram uma rede de apoio em várias substiu;icoes que fizeram no meu lugar para eu poder cursas as disciplinas, participar de reuniões, entre outras necessidades. Vocês tornaram a jornada mais leve.

E, finalmente, ao meu orientador, Professor Dr. Edson Takashi, cuja orientação meticulosa e insights valiosos foram indispensáveis. Sua paciência e dedicação não apenas tornaram este trabalho possível, mas também me transformaram em um pesquisador mais competente e um ser humano melhor.

A todos vocês, meu mais sincero agradecimento.

Sumário

Sumário x
Lista de Figuras xiv
Lista de Tabelas xv
Lista de Abreviaturas xvii

1 Introdução **3**
1.1 Justificativa 6
1.2 Hipótese da Pesquisa 7
1.3 Objetivos 7
1.4 Organização da Tese e Contribuições 8

2 Materiais e Métodos **11**
2.1 Séries Temporais 11
2.1.1 Componentes das Séries Temporais 16
2.1.2 Estacionaridade e Linearidade 20
2.1.3 Séries Temporais Longas 21
2.2 Abordagens para previsão de Séries Temporais 23
2.2.1 Múltiplos Passos Diretos (DMS) 24
2.2.2 Múltiplos Passos Iterados (IMS) 25
2.3 Aprendizado Profundo 27
2.3.1 Redes Neurais Recorrentes 29
2.3.2 Redes Convolucionais 33
2.3.3 *Transformers* 35
2.3.4 Comparativo Entre Arquiteturas 39
2.4 Métricas de Erro de previsão 42
2.5 Considerações Finais 43

3 Revisão Literária **45**
3.1 Evolução das Séries Temporais Longas 46
3.1.1 Redes Recorrentes 46

3.1.2 Redes Convolucionais 48
3.1.3 *Transformers* 50
3.1.4 Compostos 51
3.1.5 Diversos 52
3.2 Considerações Finais e *Gaps* das Literaturas 54

4 Estudo de conexões residuais **57**
4.1 Conexões Residuais 58
4.2 Conexões Densas 61
4.3 Prevenção de Singularidades 63
4.4 Análise Matemática das Conexões Residuais 65
4.4.1 Sem conexões residuais 65
4.4.2 Com duas conexões residuais 66
4.5 Considerações Finais 68

5 DESCINet **69**
5.1 Modelo 69
5.2 Base de dados 72
5.2.1 Configurações do Modelo 75
5.3 Estudo em Séries Univariadas 79
5.4 Estudo em Séries Multivariadas 85
5.4.1 *Exchange Rate*, *Electricity* e *Traffic* 90
5.4.2 *Weather* e Illness 96
5.5 Estudo da *Loss landscape* 101
5.6 Estudo Tamanho da Sequência 104
5.7 Impacto das Conexões Residuais Densas 106
5.8 Considerações Finais 110

6 Conclusões **111**
6.1 Conclusões 111
6.2 Publicações e Colaborações 113
6.3 Trabalhos Futuros 113

Referências **125**

Lista de Figuras

2.1 Série temporal fictícia do consumo de energia de uma casa. . . . 12
2.2 Série temporal irregular. 13
2.3 Série temporal apresentada em dois formatos distintos: contínuo e discreto, respectivamente 14
2.4 Série temporal com única entrada e realizando a previsão futura desta variável. 15
2.5 Série temporal multivariável tendo como entrada a temperatura, chuva e pressão e realizando a previsão futura da temperatura. . 16
2.6 Série temporal tendo como entrada a temperatura, chuva e pressão e realizando a previsão futura da temperatura, chuva e pressão.. 16
2.7 Componentes extraídos da série temporal de uma série gerada sinteticamente. 18
2.8 Exemplo comparativo de variação de ciclo e sazonalidade 19
2.9 a) Série temporal estacionária. b) Série temporal não-estacionária com tendência linear. c) Série temporal não-estacionária com crescimento exponencial. 21
2.10 Aplicações em séries temporais. 22
2.11 A capacidade de previsão de uma LSTM na previsão a longo prazo (STL). A partir de um comprimento de 48, o MSE aumenta e a velocidade de inferência cai rapidamente. 23
2.12 Ilustração do funcionamento do método DMS 25
2.13 Ilustração do funcionamento do método IMS 26
2.14 Diagrama comparativo entre aprendizado de máquina tradicional e aprendizado profundo, destacando a automação na extração de características por redes neurais profundas no aprendizado profundo. 28

2.15 Uma representação simplificada de uma Rede Neural Recorrente mostrando como é calculado o estado oculto H_t, que é então retroalimentado na rede e combinado com a próxima entrada da sequência. Esse mecanismo permite que as RNN retenham informações de elementos anteriores da sequência e as utilizem para processar o elemento subsequente da série. 30
2.16 A arquitetura de um neurônio LSTM. O estado celular é representado como C, enquanto a entrada é x_t e o estado oculto é H_t. . 31
2.17 Arquitetura da Gated Recurrent Unit. 32
2.18 A ilustração de uma amostra de CNN com 2 camadas de convolução e uma camada totalmente conectada. 34
2.19 Ilustração de funcionamento de uma CNN 1D com entrada multivariada. 35
2.20 Representação esquemática da arquitetura do *Transformer* mostrando o componente de codificação (*encoders*) e o componente de decodificação (*decoders*). A transformação da entrada é ilustrada, com as setas indicando o fluxo de dados do modelo até a saída. 37
2.21 Representação da arquitetura *Multi-Head Attention* em um *Transformer*, ilustrando como o módulo de Atenção divide suas consultas (Query), chaves (Key) e valores (Value) em múltiplas cabeças de atenção. Cada cabeça processa esses elementos de forma independente, permitindo ao *Transformer* capturar múltiplas relações e nuances para cada palavra, e combinando-as para formar um score de atenção final. 38
2.22 Comparativo entre RNN e *Transformers* 40
2.23 Ilustração das arquiteturas RNN, CNN e *Transformers* 41

3.1 Desempenho do modelo no ETTm2 em várias extensões de saída. 51
3.2 Evolução dos trabalhos em LSTF. 55
3.3 Continuação Evolução dos trabalhos em LSTF. 56

4.1 Evolução da quantidade de camadas em modelos neurais. 58
4.2 Ilustração das diferentes arquiteturas de conexões em redes neurais: (a) apresenta uma arquitetura de conexão simples, (b) exibe conexões residuais, e (c) mostra conexões densas. 59
4.3 Representação da conexão residual: (a) Unidade Residual Convencional (He et al., 2016), (b) Unidade Residual em *Transformers* (LN = Normalização de Camadas) (Vaswani et al., 2017a). Adaptado de Liu et al. (2020) 60

4.4 Singularidades em uma camada totalmente conectada e como pular conexões evita isso. Adaptado de Orhan (2017). 64
4.5 Ilustração de rede sem conexões residuais. 65
4.6 Ilustração de rede sem conexões residuais. 66

5.1 Arquitetura geral do DESBlock. 70
5.2 Passo a passo do fluxo de informações através do modelo DESCINet. 71
5.3 Arquitetura geral do DESCINet. 72
5.4 Gráfico dos conjuntos de dados utilizados para previsão. 76
5.5 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados univariado ETTh1. 81
5.6 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados univariado ETTh2. 84
5.7 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariados ETTh1. 86
5.8 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariados ETTh2. 88
5.9 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariados ETTm1. 89
5.10 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Exchange Rate*. 92
5.11 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Electricity*. 95
5.12 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Traffic*. 96
5.13 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Weather*. 99
5.14 Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *National Illness*. 100
5.15 Comparativo de *loss landscape* entre SCINet e DESCINet. 102

5.16 Comparativo de *loss* de treino e validação entre SCINet e DESCINet utilizando o conjunto de dados ETTh1. 104
5.17 Desempenho de previsão (MSE) com janelas de observação variáveis nos 3 maiores conjuntos de dados: *Traffic*, *Electricity* e *Weather*. As janelas de observação são selecionadas para serem L = 24, 48, 96, 192, 336, 720, e os horizontes de previsão são T = 96, 720. 106
5.18 Boxplots comparando as estimativas médias do erro quadrático médio (MSE) para diferentes configurações do modelo DESCINet. 107

Lista de Tabelas

3.1 Visão geral de trabalhos publicados em predição de séries temporais utilizando como base em modelos RNN. 48
3.2 Visão geral de trabalhos publicados em predição de séries temporais utilizando como base em modelos CNN. 50
3.3 Visão geral de trabalhos publicados em predição de séries temporais utilizando como base modelos Transformers. 52
3.4 Visão resumida dos trabalhos publicados em predição de séries temporais utilizando como base modelos *Transformers*. 53

5.1 Descrição dos campos do conjunto ETT. 74
5.2 Detalhes resumidos das características dos oito conjuntos de dados. 75
5.3 Hiperparâmetros do modelo para cada conjunto de dados 77
5.4 Resultados de previsão de série temporal univariada nos conjuntos de dados ETT. 80
5.5 MSE / MAE para diferentes horizontes de previsão em séries multivariadas no conjunto de dados ETT. 85
5.6 Comparação do desempenho da previsão de longo prazo com modelos nos conjuntos de dados *Exchange Rate*, *ECL*, *Traffic*. 91
5.7 Comparação do desempenho da previsão de longo prazo nos conjuntos de dados multivariado *Weather* e *ILI*. 97
5.8 Simulação dos modelos DESCNet com diferentes números de conexões de salto em séries temporais (multivariadas) para diferentes horizontes. 108
5.9 Statistic e P-Value. 109

Lista de Abreviaturas

AM Aprendizado de Máquina

AP Aprendizado Profundo

AR AutoRegressive

ARIMA Autoregressive Integrated Moving Average

ARMA AutoRegressive Moving Average

BiLSTMs LSTMs Bidirecionais

BP BackPropagation

CNN Convolucional Neural Networks

FCN Fully Convolutional Network

GRU Gated Recurrent Unit

IA Inteligência Artificial

LSTM Long Short-Term Memory

LSTF Long-term Time Series Forecasting

MA Moving Average

MAE Mean Absolute Error

MSE Mean Squared Error

NLP Natural Language Processing

RNA Redes Neurais Artificiais

RNN Recurrent Neural Networks

STL Séries Temporais Longas

SVM Support Vector Machines

Abstract

Time series forecasts are essential for understanding and anticipating patterns in data that vary over time. These predictions apply across a variety of fields, from meteorology, where they are used to forecast future weather conditions, to the financial market, to anticipate movements in stocks and currencies. This thesis details the innovation brought about by the integration of dense networks, aimed at improving both the modeling and the accuracy of predictions. Surpassing the SCINet model, which was already recognized for its good results in univariate and multivariate time series, the study introduces DESCINet. This new model resolves issues identified in SCINet, particularly those arising from the use of downsampling, a technique that, despite being useful, could lead to the loss of critical information and heavily depended on fine-tuning of hyperparameters. Furthermore, the thesis addresses the difficulty SCINet had in maintaining accuracy in long-term forecasts due to its limited ability to capture complex patterns across various temporal scales. DESCINet, with its approach of dense residual connections, promises to overcome these barriers, preserving detailed information and enhancing the ability to model complex temporal dependencies. This innovative approach allows the model to maintain consistent performance across extended forecasting horizons. The practical application of DESCINet was tested on a wide range of data sets, such as ETT, Weather, Electricity, Illness, Traffic, and Exchange Rate. In all these cases, DESCINet demonstrated superiority over SCINet, validating its efficacy in varied and complex contexts. The selection of these datasets illustrates the diversity of challenges inherent in time series forecasting and highlights the adaptability and robustness of DESCINet. This study contributes to the field of time series by exploring the yet untapped potential of dense networks. Integrating these networks into time series forecasting models paves the way for significant advancements, both academically and in practical applications. The proposition of DESCINet indicates a promising direction for future research, suggesting that overcoming current

limitations in time series forecasting is within reach. In conclusion, this thesis offers a better understanding of the impact of dense connections on time series forecasting, encouraging the scientific community to investigate DESCINet more deeply. The work is expected to stimulate ongoing research in this area, laying the groundwork for new innovations and practices that enhance time series modeling, making it more effective and precise.

Keywords: long sequence time series, time series forecasting, dense connections, convolutional neural networks.

Resumo

Previsões de séries temporais são essenciais para compreender e antecipar padrões em dados que variam ao longo do tempo. Essas previsões aplicam-se em uma variedade de campos, desde a meteorologia, onde são usadas para prever condições climáticas futuras, até o mercado financeiro, para antecipar movimentos de ações e moedas. Esta tese detalha a inovação trazida pela integração de redes densas, que visa melhorar tanto a modelagem quanto a precisão das previsões. Superando o modelo SCINet, que já era reconhecido por seus bons resultados em séries temporais univariadas e multivariadas, o estudo introduz o DESCINet. Este novo modelo resolve problemas identificados no SCINet, particularmente aqueles decorrentes do uso de *downsampling*, uma técnica que, apesar de útil, podia levar à perda de informações críticas e dependia fortemente do ajuste fino de hiperparâmetros. Além disso, a tese aborda a dificuldade do SCINet em manter a precisão em previsões de longo prazo devido à sua limitada capacidade de capturar padrões complexos em várias escalas temporais. O DESCINet, com sua abordagem de conexões residuais densas, promete superar essas barreiras, preservando informações detalhadas e aprimorando a capacidade de modelar dependências temporais complexas. Essa abordagem inovadora permite ao modelo manter um desempenho consistente ao longo de horizontes de previsão estendidos. A aplicação prática do DESCINet foi testada em uma ampla gama de conjuntos de dados, como *ETT*, *Weather*, *Electricity*, *Illness*, *Traffic*, and *Exchange Rate*. Em todos esses casos, o DESCINet demonstrou superioridade em relação ao SCINet, validando sua eficácia em contextos variados e complexos. A seleção desses datasets ilustra a diversidade dos desafios inerentes à previsão de séries temporais e destaca a adaptabilidade e robustez do DESCINet. Este estudo adiciona uma contribuição para o campo de séries temporais ao explorar o potencial ainda pouco aproveitado das redes densas. A integração dessas redes em modelos de previsão de séries temporais abre caminho para avanços importantes, tanto em termos acadêmicos quanto em aplicações práticas. A proposta do DESCINet indica uma direção promissora para futuras pesquisas,

sugerindo que a superação de limitações atuais na previsão de séries temporais está ao alcance. Ao concluir, esta tese oferece uma melhor compreensão sobre o impacto das conexões densas na previsão de séries temporais, encorajando a comunidade científica a investigar mais profundamente o DESCINet. Espera-se que o trabalho estimule pesquisas contínuas nesta área, pavimentando o caminho para novas inovações e práticas que aprimorem a modelagem de séries temporais, tornando-as mais eficazes e precisas.

Palavras-chave: séries temporais longas, previsão de séries temporais, conexões densas, redes neurais convolucionais.

CAPÍTULO

1

Introdução

A previsão Zhang et al. (2018) de séries temporais é uma tarefa que tem como motivação a previsão de valores futuros e tem sido utilizada em diversas aplicações do mundo real, como preço de ações do mercado financeiro (Khan et al., 2020), sensores industriais (Wang et al., 2022b), cidades inteligentes (Chen et al., 2022), mercado financeiro (Cheng et al., 2022) e monitoramento ambiental (Ali et al., 2022). Esta ampla aplicabilidade reflete a versatilidade e a importância das séries temporais em vários setores, cada um com suas características e necessidades. No mundo do mercado financeiro, por exemplo, as séries temporais são importantes para analisar as tendências dos preços das ações, permitindo que investidores tomem decisões estratégicas. Da mesma forma, em sensores industriais, a previsão de séries temporais possibilita a detecção antecipada de falhas em equipamentos para prever quando uma máquina pode precisar de manutenção, evitando assim paradas não planejadas.

As séries temporais são um tipo de representação de dados, organizadas de forma cronológica, que permitem analisar mudanças nas variáveis com o objetivo de reduzir incertezas futuras (Zhu et al., 2019). Para realizar essas análises em séries temporais, diversos métodos têm sido estudados.

Os métodos estatísticos tradicionais, como AutoRegressive (AR) (Triebe et al., 2019), Moving Average (MA) (Mehdizadeh, 2020) e AutoRegressive Moving Average (ARMA) (Moon et al., 2021), são eficazes na análise de séries temporais univariadas de natureza simples e previsível, caracterizadas pela ausência de variações significativas ou ruídos. A precisão desses métodos, combinada com a facilidade de implementação e baixa demanda computacional, os torna especialmente adequados para essas situações. Contudo, encon-

tram dificuldades em identificar relações não lineares ou padrões complexos nas séries.

Algoritmos de aprendizado de máquina, como *Support Vector Machines* (SVM) (Cortes e Vapnik, 1995) e *Adaptive Boosting* (Freund, 1995), progrediram na superação de certas limitações. Esses métodos são capazes de modelar dependências temporais e lidar com características não lineares. Contudo, eles apresentam desafios em termos de generalização e por muitas vezes precisam de outros recursos para melhorar a acurácia preditiva (Chen et al., 2023).

O crescimento contínuo na complexidade e no volume de dados tem impulsionado a demanda por previsões de longo prazo, especialmente no contexto de séries temporais longas (STL) (Zhou et al., 2020). Nesse cenário, o desenvolvimento das *Recurrent Neural Networks* (RNN) (Rumelhart et al., 1986) representou um avanço. Diferentemente das abordagens anteriores, a RNN é projetada para lidar com dados sequenciais, onde informações passadas influenciam as saídas atuais, tornando-as interessantes para o problema que envolve séries. No entanto, as RNN tradicionais enfrentam dificuldades com sequências muito longas devido ao problema do desaparecimento e explosão do gradiente, e a limitações de memória que demandam uma grande quantidade de recursos computacionais (Sherstinsky, 2020). Para aprimorar a RNN, variantes como *Long Short-Term Memory* (LSTM) (Graves e Graves, 2012), *Bidirectional* LSTMs (BiLSTMs) (Siami-Namini et al., 2019) e *Gated Recurrent Units* (GRU) (Cho et al., 2014) foram desenvolvidas, que mitigam o problema do desaparecimento do gradiente e permitiu a RNN realizar previsões precisas em séries temporais. No entanto, o modelo recorrente não é paralelizável e tem dificuldades em lidar com um grande número de amostras (Zhou et al., 2022b).

Simultaneamente, as *Convolucional Neural Networks* (CNN) (Lecun et al., 2015) surgiram como uma alternativa para tarefas com séries temporais multivariadas e características espaciais. A CNN têm sido aplicadas com sucesso para previsão e classificação de séries temporais, devido à sua capacidade de capturar padrões temporais e espaciais nos dados (Wibawa et al., 2022; Zhao et al., 2017). Diversas arquiteturas combinadas com outros métodos foram propostas para extrair padrões em séries temporais para analisar a dependência entre variáveis, o que é importante para a previsão de séries temporais (Wang et al., 2019). Um exemplo foi o estudo conduzido por Bai et al. (2018), no qual empregou convolução causal para evitar vazamento de informações e convolução causal dilatada para lidar com sequências longas. Um avanço na utilização de redes CNN para análise de séries temporais foi o desenvolvimento do SCINet (Liu et al., 2021a). Esta arquitetura combina técnicas de redução de amostragem, convolução e interação de maneira recursiva, permitindo a

extração eficiente de características temporais distintas. Essa abordagem posicionou o SCINet como um modelo de estado da arte no campo da previsão de séries temporais.

Recentemente, os *Transformers*, proposto para processamento de linguagem natural, mostraram bons resultados de modelar dependências complexas em séries temporais (Vaswani et al., 2017a). O FEDformer é a variação mais recente (Zhou et al., 2022b) que combina um mecanismo de atenção com uma técnica de decomposição de componentes da série temporal utilizando médias móveis. Embora os trabalhos mais recentes em séries temporais longas (STL) empreguem *Transformers* (Zhou et al., 2020; Kitaev et al., 2020; Liu et al., 2021b; Wu et al., 2021), estes mostram desempenho inferior em séries temporais curtas devido à necessidade de aprimorar a integração de informações contextuais (Zhu et al., 2023).

A eficiência de previsão dos modelos atuais em RNN, CNN e *Transformers* de enfrentar o problema de STL deve-se em grande parte aos avanços no campo do aprendizado profundo, notavelmente influenciada pelas ResNets (Zagoruyko e Komodakis, 2016). As ResNets introduziram conexões residuais, que pode ajudar a preservar recursos de baixo nível e evitar a degradação do desempenho ao adicionar mais camadas (He et al., 2016). Essas conexões melhoraram a convergência dos modelos durante o treinamento, além da capacidade de generalização. Em continuidade a esses avanços, as DenseNets (Huang et al., 2017) deram um passo adiante, integrando entradas de todas as camadas anteriores em cada camada subsequente, resultando em redes que conseguem uma melhor representação e extração de dados (Yang et al., 2023).

Inicialmente desenvolvidas no contexto da visão computacional, as conexões densas foram utilizadas em vários domínios, como saúde (Zhou et al., 2022c), finanças (Mukherjee et al., 2023), agricultura (Mahum et al., 2023). No campo da previsão de séries temporais, a aplicação dessas conexões ainda é um território relativamente inexplorado. Considerando os benefícios observados em outras áreas investigar a implementação de conexões densas em arquiteturas voltadas para séries temporais apresenta um potencial significativo.

Até o momento, os estudos que exploram redes neurais nesse contexto geralmente utilizam uma variedade de técnicas combinadas para alcançar resultados eficazes, mas não focam especificamente nas conexões densas. Considerando os benefícios potenciais que as conexões densas oferecem no campo da visão computacional, esta pesquisa propõe uma investigação sobre como a incorporação de conexões densas pode otimizar o desempenho na previsão de séries temporais.

O potencial das conexões densas é explorado através do modelo proposto, o DESCINet, que integra essas conexões no modelo SCINet para aprimorar a captura de dependências temporais e o desempenho geral na previsão de séries temporais. O DESCINet destaca-se pela sua arquitetura, que permite uma melhor fluidez de informações, abordando singularidades nas matrizes de pesos e proporcionando uma convergência mais estável durante o treinamento.

Os resultados obtidos demonstram a eficácia do DESCINet, com uma melhoria significativa na precisão das previsões em comparação com modelos estado da arte. Em particular, observou-se que o método proposto em relação ao seu antecessor teve melhorias significativas em todos os conjuntos de dados testados e para vários horizontes de previsão. Este avanço representa uma contribuição significativa para o campo da previsão de séries temporais longas, abrindo caminhos para futuras pesquisas sobre a aplicação de conexões densas em outras arquiteturas e domínios.

1.1 Justificativa

O campo das séries temporais longas é marcado por padrões complexos, nos quais a habilidade de realizar previsões precisas a longo prazo desempenha um papel crucial. Essa tarefa é complicada pela presença de tendências não estacionárias, sazonalidades e eventos raros ou imprevisíveis, que podem induzir variações significativas nas previsões (Zhou et al., 2022a).

O SCINet representa um avanço na previsão de séries temporais, mas enfrenta desafios relacionados à eficiência no treinamento e generalização para novos padrões de dados devido à sua abordagem de *downsampling*. Tais desafios incluem a perda de detalhes informativos e um melhor ajuste nos hiperparâmetros, restringindo sua efetividade em diferentes contextos. Além disso, a eficácia do SCINet em detectar e preservar padrões tanto locais quanto globais declina com o aumento do horizonte de previsão, o que compromete a precisão das previsões futuras. A implementação de conexões residuais densas é vista como uma solução para esses desafios, prometendo melhorar a conservação de informações e a habilidade da rede em identificar dependências em várias escalas temporais. Espera-se que isso permita ao modelo manter uma precisão elevada mesmo em previsões de longo prazo, capturando assim a complexidade inerente às séries temporais.

Por fim, embora o conceito de conexões residuais tenha sido estabelecido em visão computacional, permitindo avanços significativos em tarefas como reconhecimento de imagens e detecção de objetos (Li et al., 2022), sua aplicação no campo de séries temporais ainda é pouco explorada. Esta lacuna na

literatura sugere um potencial inexplorado para essas conexões em melhorar a modelagem e a previsão de séries temporais longas.

Portanto, este trabalho se justifica pela necessidade de enfrentar os desafios inerentes às séries temporais longas e pela chance de investigar a influência das conexões residuais neste contexto. O objetivo da pesquisa é explorar como as conexões residuais densas podem ser adaptadas e refinadas para melhorar o desempenho preditivo de modelos de séries temporais, inspirando-se nos êxitos alcançados na visão computacional e ajustando-os às características e necessidades específicas dos dados temporais.

1.2 Hipótese da Pesquisa

A hipótese central deste trabalho é que o uso de conexões densas em uma rede neural pode melhorar a integração de informações ao longo da rede. Dentre os diversos algoritmos estado da arte, o SCINet foi o candidato a ser explorado nesta tese por não utilizar conexões densas e ainda assim estar entre os algoritmos estado da arte na previsão series temporais. Acredita-se que o emprego dessas conexões possa melhorar taxas de acerto e acelerar a convergência do modelo. De forma resumida, as seguintes hipóteses foram destacadas a seguir:

- **Hipótese 1 (Estabilidade de convergência)**: A adição de conexões residuais densas em rede neurais estado da arte resulta em uma curva de perda mais suave durante o treinamento, refletindo em uma convergência mais estável e previsível do modelo SCINet.

- **Hipótese 2 (Robustez contra variações abruptas)**: O uso de conexões densas aumenta a robustez do SCINet contra variações abruptas em séries temporais, melhorando a precisão das previsões mesmo com a presença de ruídos ou movimento atípicos.

- **Hipótese 3 (Aprimoramento da acurácia com conexões densas)**: Conexões densas em uma rede neural contribuem para um aumento na acurácia da previsão de dados, em comparação com o SCINet.

1.3 Objetivos

A presente tese visa contribuir para o campo da previsão de séries temporais longas através da integração e exploração de conexões densas em uma arquitetura de rede neural convolucional. Os objetivos específicos são:

1. Investigar o impacto do uso de conexões densas na previsão de séries temporais, com foco em sua influência na suavidade da convergência e na robustez do modelo através da curva de *loss*.

2. Avaliar a aplicabilidade das conexões densas em conjuntos de dados variados, medindo a capacidade do modelo em capturar padrões e tendências em diferentes horizontes.

3. Comparar o desempenho do modelo desenvolvido, chamado de DESCInet com o estado da arte, utilizando conjuntos de dados de referência para validar as melhorias na acurácia de previsão utilizando as métricas Mean Squared Error (MSE) e Mean Abolsute Error (MAE).

4. Examinar a generalização e a adaptabilidade do modelo proposto, em diferentes domínios de STL, destacando sua robustez e capacidade de previsão em face de anomalias e ruídos através de plotagens de gráficos de previsão.

1.4 Organização da Tese e Contribuições

Esta tese é organizada em seis capítulos, cada qual com um papel distinto na construção do argumento e na apresentação dos resultados do estudo. A seguir, delineia-se a estrutura dos capítulos e suas respectivas contribuições:

- O Capítulo 2 estabelece a base teórica, introduzindo conceitos essenciais de séries temporais e examinando as arquiteturas neurais mais relevantes para previsões em longo prazo. Além disso, este capítulo detalha as métricas utilizadas para avaliar a performance dos modelos propostos.

- No Capítulo 3 são explorados e comparados os modelos de aprendizado de máquina específicos para a previsão de séries temporais longas, estabelecendo um contexto cronológico da evolução dos modelos nesta área.

- O Capítulo 4 detalha a relevância e o impacto das conexões residuais densas na otimização de modelos neurais aplicados a séries temporais longas. Este capítulo oferece uma análise de como essas conexões melhoram o desempenho dos modelos, realçando sua contribuição na captação de padrões e na previsão de eventos futuros em séries temporais.

- O Capítulo 5 descreve em detalhes o modelo desenvolvido, DESCInet, incluindo sua estrutura e abordagem para previsão de séries temporais, além de apresentar os resultados alcançados pelo DESCInet, demonstrando sua precisão em uma variedade de conjunto de dados.

- Por fim, o Capítulo 6 sintetiza as principais contribuições da tese, aborda as limitações encontradas e sugere direções futuras para pesquisa na área de previsão de séries temporais de longa duração.

CAPÍTULO

2

Materiais e Métodos

Neste capítulo são delineados os conceitos fundamentais e as estruturas teóricas que formam o alicerce desta tese e dos experimentos realizados. Inicia-se com uma exposição detalhada do campo de previsão de séries temporais, seguida por uma análise das arquiteturas de aprendizado profundo aplicadas a séries temporais. O objetivo é oferecer uma compreensão abrangente tanto dos aspectos teóricos quanto práticos envolvidos na previsão de séries temporais, fornecendo uma base sólida para as investigações e aplicações propostas.

- Conceitos de séries temporais e séries temporais longas
- Arquiteturas para predições em séries temporais
- Métricas para avaliação de modelos

2.1 Séries Temporais

Uma série temporal $\mathbf{T}$ é uma sequência ordenada de n observações tais que $\mathbf{T} = \{x_1, x_2, \cdots, x_n\}$ e que x_t é o valor de uma observação, onde $x_t \in \mathbb{R}$ para qualquer $t \in [1, n]$. Em muitos fenômenos naturais, como variações na temperatura atmosférica, batimentos cardíacos ou flutuações no mercado financeiro, o elemento temporal é primordial. A correta sequência dos eventos em dados temporais é essencial e precisa ser assegurada no momento do registro. Estudar séries temporais implica em analisar esses registros cronológicos com o objetivo de projetar eventos futuros. As observações podem ser categorizadas

em várias unidades temporais, desde segundos e minutos até dias, meses ou anos (Chatfield, 2004).

Na organização desses dados, o tempo não atua como uma variável comum, mas como uma estrutura para organizar o conjunto de informações. Esta particularidade eleva a complexidade ao lidar com tais dados, exigindo dos profissionais da área técnicas especializadas de tratamento e manipulação. Este campo abre possibilidades para a extração de conhecimentos adicionais, como identificar padrões temporais e tendências em diversas áreas como finanças, saúde, indústria e tantas outras (Brockwell e Davis, 1991). Com isso, torna-se possível, por exemplo, projetar as vendas de produtos em supermercados, estimar valores de revenda de veículos, calcular números de passageiros em rotas aéreas ou prever o gasto de energia dos consumidores. Para exemplificar a relevância e a aplicabilidade da análise de séries temporais em situações práticas, considere o gráfico na Figura 2.1.

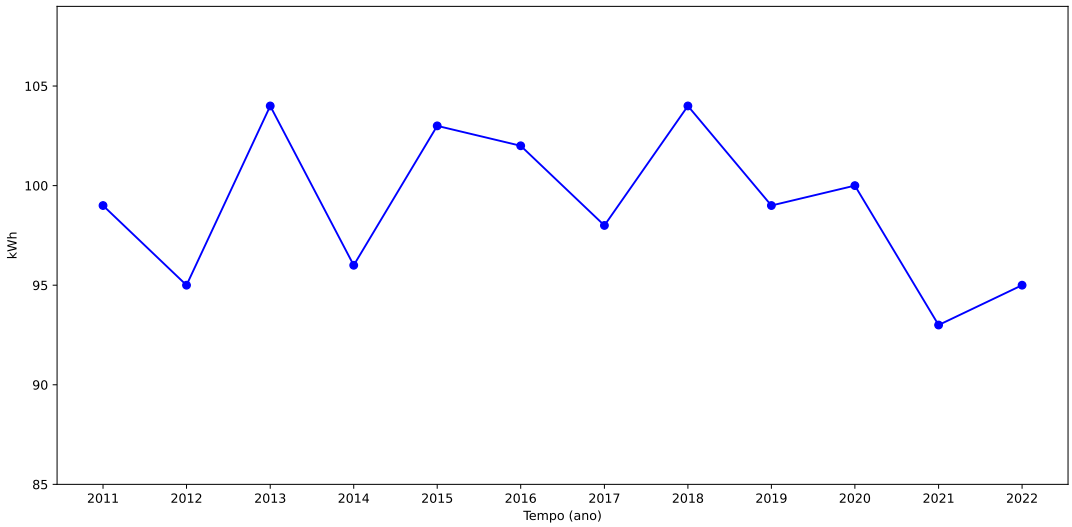


Figura 2.1: Série temporal fictícia do consumo de energia de uma casa.

Fonte: Elaborada pelo autor.

A Figura 2.1 mostra o consumo de energia elétrica em uma casa ao longo dos anos, servindo como um caso concreto que ilustra a utilidade deste campo de estudo no cotidiano. Existem basicamente dois tipos de séries temporais, de acordo com Hamilton (1994):

- Séries temporais regulares: este é o tipo mais comum, onde as observações são registradas em intervalos de tempo constantes, como a cada hora, mês ou ano.

- Séries temporais irregulares: neste tipo, as observações não ocorrem em intervalos fixos. Um exemplo seriam os resultados de exames médicos de um paciente, que são registrados apenas quando o paciente vai à clínica.

O foco principal deste trabalho são as séries temporais regulares, que são mais simples de analisar devido à uniformidade na coleta de dados. Em contraste, as séries temporais irregulares, como ilustrado na Figura 2.2, apresentam desafios únicos devido à variabilidade nos intervalos de tempo entre as observações, exigindo técnicas especializadas para sua análise eficaz (Shumway e Stoffer, 2017).

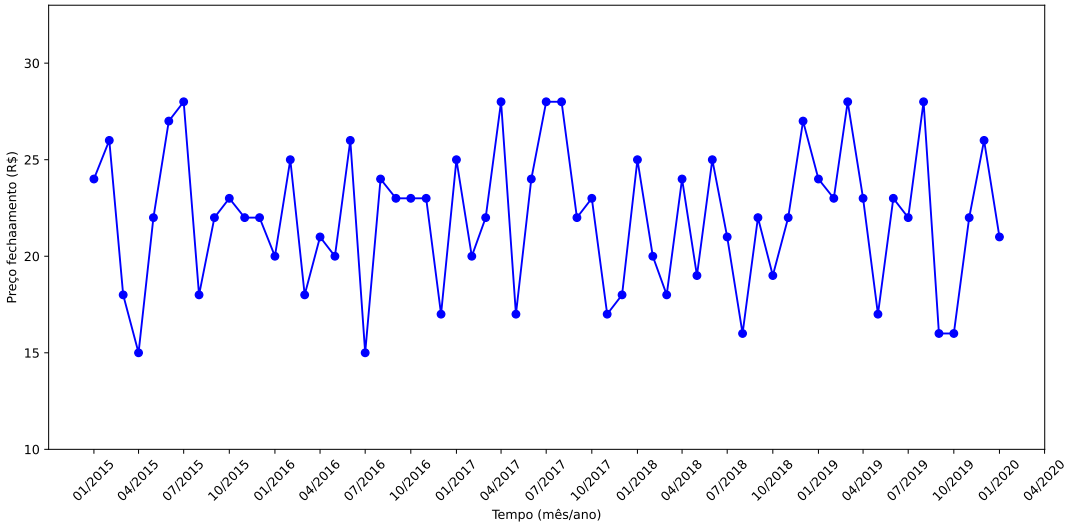


Figura 2.2: Série temporal irregular.

Fonte: Elaborada pelo autor.

Séries temporais são classificadas como contínuas ou discretas com base na natureza da coleta de dados. Uma série temporal contínua caracteriza-se por medições que ocorrem de forma contínua no tempo, idealmente sem lacunas, registradas em cada ponto no contínuo temporal $\mathbb{R}_+$. Este tipo de série é comum em fenômenos que exigem monitoramento constante, como dados meteorológicos ou sinais biomédicos.

Por outro lado, uma série temporal discreta envolve a coleta de dados em intervalos fixos e regulares de tempo. Esta abordagem é frequentemente encontrada em contextos onde a coleta contínua de dados não é viável ou necessária, como em dados econômicos ou estatísticas de uso. A regularidade dos intervalos em séries discretas simplifica a análise, permitindo o uso de métodos estatísticos e computacionais mais tradicionais. A diferença entre essas séries é ilustrada na Figura 2.3.

Embora as séries temporais contínuas e discretas sejam distintas em sua natureza, é importante notar que uma série contínua pode ser transformada em uma série discreta ao selecionar medições em intervalos uniformes. Esta conversão é uma via de mão única: enquanto é possível discretizar uma série contínua, o processo inverso, de recriar uma série contínua a partir de uma discreta, não é factível devido à perda de informações intrínsecas aos dados contínuos originais.

No entanto, é fundamental reconhecer que, no contexto da computação e análise de dados, todas as séries temporais são tratadas como discretas. Devido às limitações de coleta, armazenamento e processamento de dados em sistemas digitais, mesmo as séries originalmente contínuas são representadas e manipuladas por meio de amostras discretas. Esse reconhecimento sublinha a importância de técnicas de amostragem e análise adequadas para capturar a essência dos fenômenos estudados, mesmo quando operamos dentro das restrições impostas pela natureza discreta dos dados em ambientes computacionais.

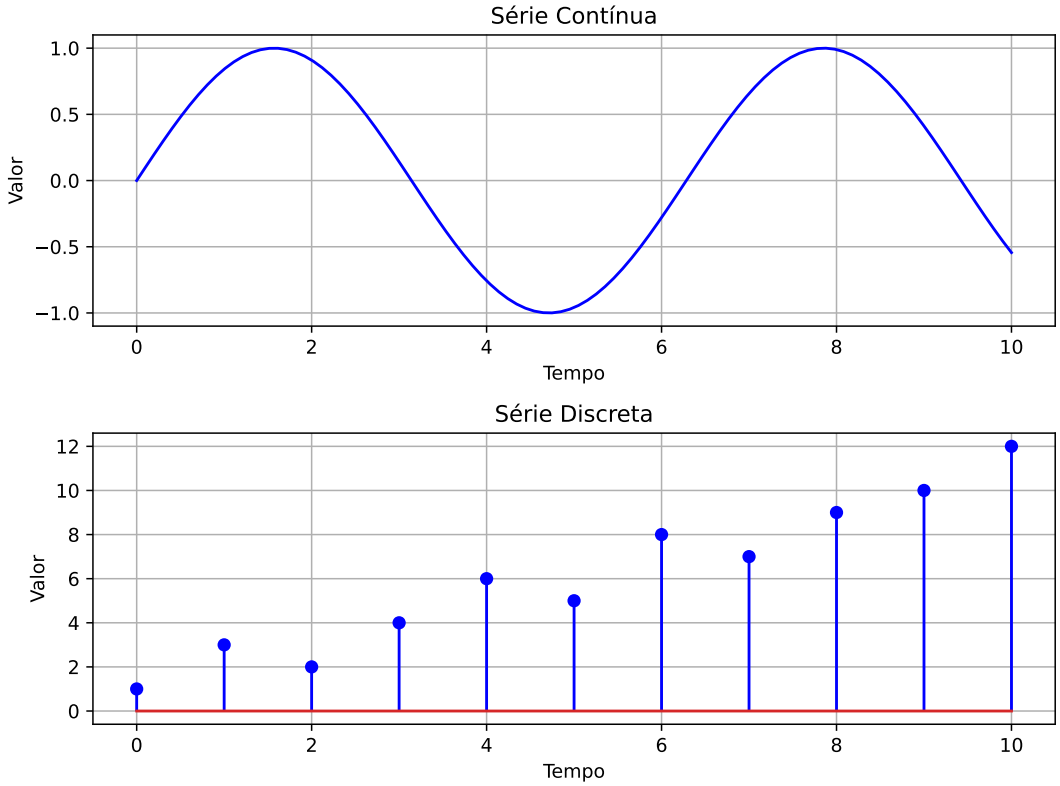


Figura 2.3: Série temporal apresentada em dois formatos distintos: contínuo e discreto, respectivamente

Fonte: Elaborada pelo autor.

Essa distinção entre séries contínuas e discretas estabelece uma base fundamental para compreender outro aspecto crucial das séries temporais: o número de variáveis registradas a cada momento. Assim como a natureza contínua ou discreta das séries temporais influencia a metodologia de análise, o mesmo acontece com o fato de uma série ser unidimensional ou multidimensional.

Se apenas uma variável é registrada, a série é considerada unidimensional ou univariada. Um exemplo de série univariada, onde os dados de entrada e saída são da mesma variável (temperatura), pode ser visto na Figura 2.4. Sendo assim, assuma que uma série univariada $T = [x_0, x_1, \cdots, x_{n-1}], x_t \in \mathbb{R}$.

Já a sua expressão matemática da tarefa de previsão univariada é representada na Equação 2.1.

$$\hat{y}_{t+1} = h(T_{t-n:t}), \tag{2.1}$$

onde, $\hat{y}_{t+1}$ representa a previsão ou estimativa do valor da série temporal T no momento $t+1$ e h é uma função que descreve como os valores futuros da série T são estimados com base nos dados históricos $T_{t-n:t} = [x_{t-n}, \cdots, x_t]$.

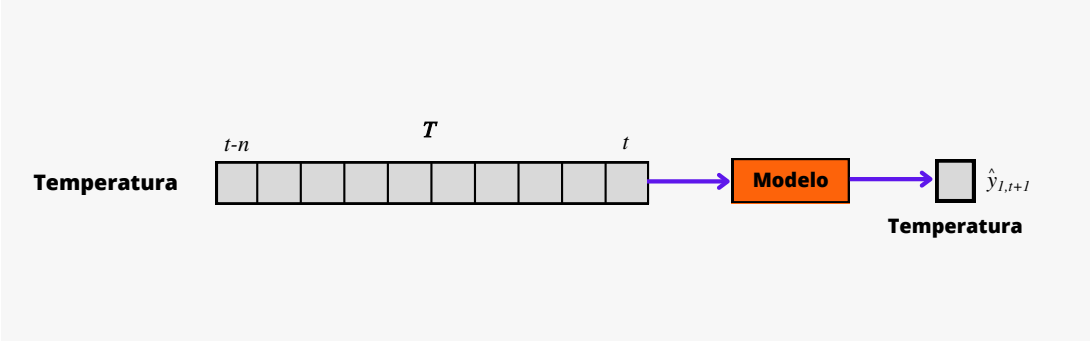


Figura 2.4: Série temporal com única entrada e realizando a previsão futura desta variável.

Fonte: Elaborada pelo autor.

Existe também a possibilidade de realizar previsões envolvendo características de múltiplas variáveis para prever o futuro de uma variável específica. Estas variáveis adicionais são conhecidas como multivariáveis. Uma série multivariada, ou uma série D-dimensional $T_m \in \mathbb{R}^{(D,n)}$ é um conjunto de séries univariadas de comprimento n. Note que $T_m = [T^{(0)}, \cdots, T^{(D-1)}]$ e $j \in [0, D-1]$, e que uma série univariada $T^{(j)} = [x_0, x_1, \cdots, x_{n-1}]$. Uma subsequência $T^{(j)}_{i,l} \in \mathbb{R}^l$ de dimensão T^j de uma série multivariada T_m é um conjunto continuo de valores de $T^{(j)}$ de comprimento l começando na posição i, formalmente, $T^{(j)}_{i,l} = [T^{(j)}_i, T^{(j)}_{i+1}, \cdots, T^{(j)}_{i+l-1}]$. Embora a previsão com multivariáveis envolva múltiplas características de entrada, o número de características que se pretende prever pode permanecer único ou múltiplos. Esta abordagem é ilustrada de duas maneiras, a primeira delas na Figura 2.5. Para previsões de um único passo à frente com multivariáveis, há a expressão matemática apresentada na Equação 2.2.

$$\hat{y}_{t+1} = h(T_{m[t-n:t]}) \tag{2.2}$$

O segundo caso é representado na Figura 2.6, que retrata um cenário multivariado, onde tanto as entradas quanto as saídas são multivariadas. Essa abordagem é fundamental para estudos que exigem uma compreensão mais detalhada das inter-relações entre múltiplas variáveis. A expressão matemática para previsão de um passo a frente é vista na Equação 2.3.

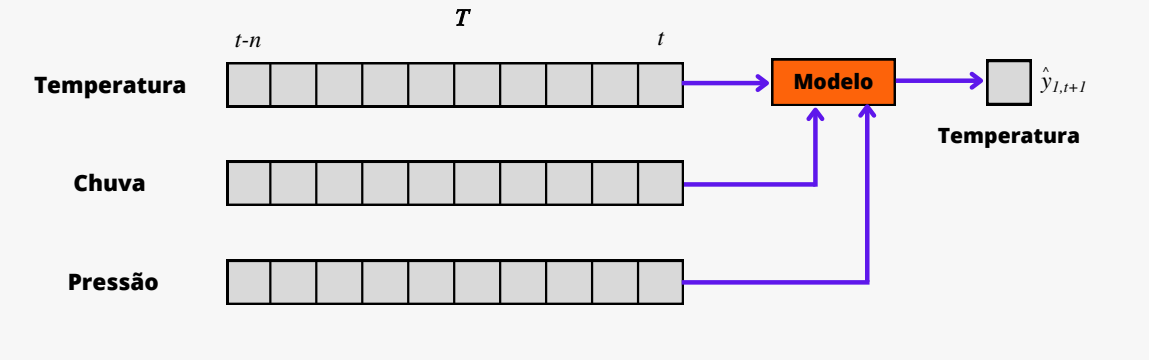


Figura 2.5: Série temporal multivariável tendo como entrada a temperatura, chuva e pressão e realizando a previsão futura da temperatura.

Fonte: Elaborada pelo autor.

$$\hat{\mathbf{Y}}_{t+1} = h(T_{[t-n:t]}) \tag{2.3}$$

onde $\hat{\mathbf{Y}}_{t+1} = [\hat{y}_{1,t+1}, ..., \hat{y}_{D,t+1}]$.

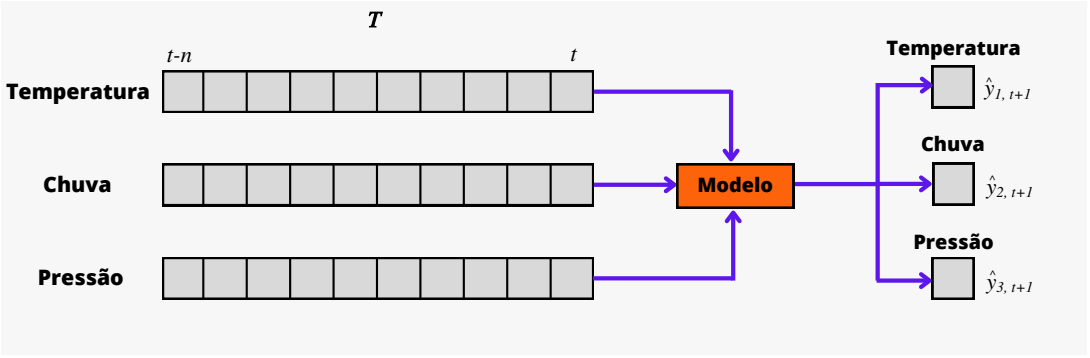


Figura 2.6: Série temporal tendo como entrada a temperatura, chuva e pressão e realizando a previsão futura da temperatura, chuva e pressão..

Fonte: Elaborada pelo autor.

Assim, compreender a complexidade das séries temporais, seja em sua dimensão univariada ou multivariada, permite uma análise mais profunda de suas características intrínsecas. Independentemente da dimensão, as séries temporais podem ser categorizadas com base em aspectos como determinismo ou estocasticidade, e linearidade. Antes de explorarmos conceitos avançados como a estacionariedade, é crucial entender os componentes básicos presentes em um conjunto de dados temporal. Esses componentes, que incluem tendência, sazonalidade e flutuações aleatórias, formam a estrutura fundamental sobre a qual as séries temporais são construídas e analisadas.

2.1.1 Componentes das Séries Temporais

Os componentes intrínsecos presentes em um conjunto de dados temporal referem-se aos padrões ou estruturas que, juntos, compõem os dados obser-

vados. Compreender esses componentes é fundamental para análises mais precisas e para realizar previsões confiáveis. Em geral, esses conjuntos podem ser categorizados em três aspectos principais: tendência, sazonalidade, e flutuações aleatórias (residual), conforme descrito por (Cleveland et al., 1990).

Tendência

Indica a trajetória geral dos valores, seja em ascensão ou queda, ao longo de um período prolongado. É comum observar mudanças na direção dessas tendências ao longo do tempo: elas podem aumentar, diminuir ou permanecer estáveis. No entanto, geralmente há uma tendência predominante. Exemplos de aplicações incluem a contagem populacional, a produção agrícola e a manufatura de produtos. Na Figura 2.7 b) são ilustradas três tendências: duas ascendentes e uma descendente. As tendências de alta ocorreram entre os anos de 1990 e 1992, e entre 1997 e 1998, enquanto a de baixa ocorreu entre 1993 e 1996 (Harvey, 1993).

Além da tendência, outro aspecto importante das séries temporais são as variações que ocorrem no curto ou longo prazo.

Sazonalidade

Variações sazonais referem-se a padrões temporais que ocorrem em intervalos regulares. Essas variações são caracterizadas por um período fixo e recorrente, frequentemente influenciadas por fatores como convenções sociais, mudanças sazonais e condições climáticas (Hylleberg, 1992). Por exemplo, a venda de guarda-chuvas aumenta durante a temporada de chuvas, enquanto a venda de sorvetes sobe no verão. A Figura 2.7 c) ilustra esse fenômeno, no qual, no início de cada ano, ocorre uma grande alta, seguida de uma grande baixa nos meses finais.

Variações sazonais referem-se a padrões temporais que ocorrem em intervalos regulares, caracterizados por um ciclo fixo e recorrente, muitas vezes influenciados por convenções sociais, mudanças estacionais e condições climáticas (Hylleberg, 1992). Por exemplo, a demanda por guarda-chuvas aumenta durante a temporada de chuvas, enquanto a procura por sorvetes cresce no verão. A Figura 2.7 c) ilustra esse fenômeno, mostrando que, no início de cada ano, há um aumento significativo na atividade, que posteriormente diminui nos meses finais.

Ciclos

Variações cíclicas diferem da sazonalidade principalmente pela ausência de um período fixo e previsível. Enquanto a sazonalidade se caracteriza por padrões que se repetem em intervalos regulares dentro de um ano, as variações

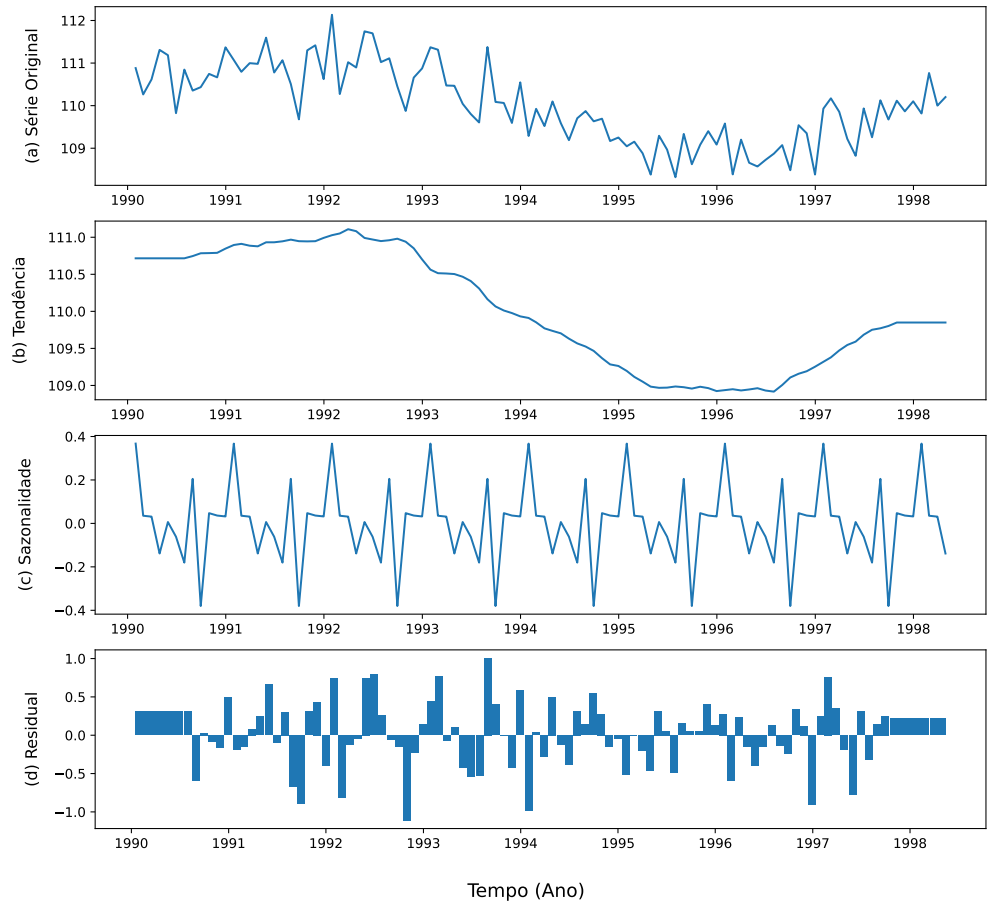


Figura 2.7: Componentes extraídos da série temporal de uma série gerada sinteticamente.

Fonte: Elaborada pelo autor.

cíclicas manifestam-se através de altas e baixas que não seguem um cronograma definido (Zarnowitz, 1992). Estes ciclos, embora recorrentes, variam em duração e são geralmente mais longos que um ano. Um exemplo clássico é o ciclo econômico, caracterizado por oscilações ascendentes e descendentes do Produto Interno Bruto (PIB) em torno de uma tendência de crescimento a longo prazo. Estes ciclos podem durar vários anos, com a duração específica de cada ciclo sendo imprevisível. A Figura 2.8 ilustra essa variação com o exemplo do ciclo econômico.

As variações cíclicas normalmente são vistos em prazos maiores do que um ano, enquanto as variações sazonais possuem movimentos de curto prazo.

Residual

As flutuações aleatórias são o último elemento a causar variações em dados de séries temporais. Essas flutuações são incontroláveis, imprevisíveis e erráticas, surgindo de fatores como desastres naturais, decisões políticas e

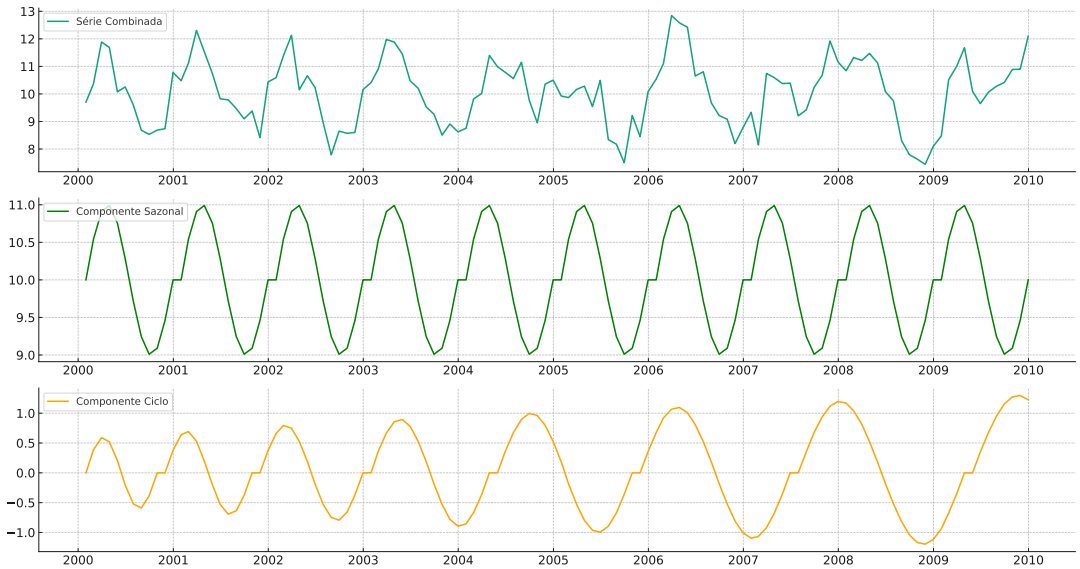


Figura 2.8: Exemplo comparativo de variação de ciclo e sazonalidade .

Fonte: Elaborada pelo autor.

guerras (Box e Jenkins, 1976). As flutuações são um processo estocástico fundamental, presente em todas as séries temporais naturais. Como será visto adiante, cada modelo teórico para esse tipo de dado pressupõe algum grau de aleatoriedade, manifestado pela adição de uma variável aleatória R_t a cada observação.

Na literatura sobre análise de séries temporais, é comum tratar as duas primeiras componentes - tendência, sazonalidade — como sinais nos dados, pois são indicadores determinísticos deriváveis dos dados. Autores como Brockwell e Davis (2002), Montgomery et al. (2015) e Box et al. (2015) exploram esses conceitos. Contrariamente, a última componente, flutuações aleatórias, representa variações arbitrárias e imprevisíveis, muitas vezes referidas como ruído. Este ruído é independente dos outros sinais e é impulsionado por variáveis latentes difíceis de observar, conforme ilustrado na Figura 2.7 d).

De forma resumida, é possível desmembrar as séries temporais através dessas três componentes, da seguinte maneira:

- D_t: tendência – o valor crescente ou decrescente ao longo do tempo,

- S_t: sazonalidade – o ciclo de curto prazo que se repete com uma frequência conhecida,

- C_t: ciclos – também são repetitivos, mas a frequência não é precisa, normalmente durando mais de dois anos,

- R_t : parte restante – engloba todos os outros aspectos.

Supondo uma decomposição aditiva, a série temporal original pode ser reconstruída pela soma desses componentes:

$$y_t = D_t + S_t + C_t + R_t \tag{2.4}$$

A decomposição aditiva é apropriada quando a variação em torno da tendência ou a magnitude das flutuações sazonais não difere do valor esperado da série temporal. Caso contrário, uma decomposição multiplicativa é mais adequada:

$$y_t = D_t \cdot S_t \cdot C_t \cdot R_t \tag{2.5}$$

É preciso identificar cuidadosamente em que medida cada componente está presente nos dados de séries temporais para poder construir uma solução precisa de previsão de aprendizado de máquina.

2.1.2 Estacionaridade e Linearidade

Uma série temporal composta por observações, pode ser caracterizada de forma determinística ou estocástica. Define-se uma série como determinística quando cada termo x_t é expresso como uma função determinada de observações anteriores, excluindo a influência de componentes aleatórios. Modelos dinâmicos, incluindo aqueles fundamentados na teoria do caos, como descrito em (Alligood et al., 1996; Kantz e Schreiber, 2004), são aplicáveis neste contexto. Alternativamente, uma série é caracterizada como estocástica quando x_t é influenciada por uma variável aleatória r_t, tornando modelos estatísticos, como o ARIMA, mais adequados.

Na realidade, séries temporais derivadas de fenômenos reais frequentemente compreendem tanto componentes determinísticas quanto estocásticas devido à influência de variáveis incontroláveis, tais como erros de medição ou variáveis latentes (Rios, 2013).

Além disso, a análise das séries estocásticas considera a estacionariedade. Uma série é estacionária se suas propriedades estatísticas, como média e variância, se mantêm constantes ao longo do tempo (Hamilton, 2020). A Figura 2.9 exemplifica uma série temporal estacionária. Em contraste, séries que exibem tendências ou variações sazonais são classificadas como não-estacionárias (Morettin e Toloi, 2006). Séries econômicas é um exemplo, que podem seguir um passeio aleatório, ou séries climáticas, que apresentam tendências lineares ou exponenciais, ilustradas nas Figura 2.9 b e c.

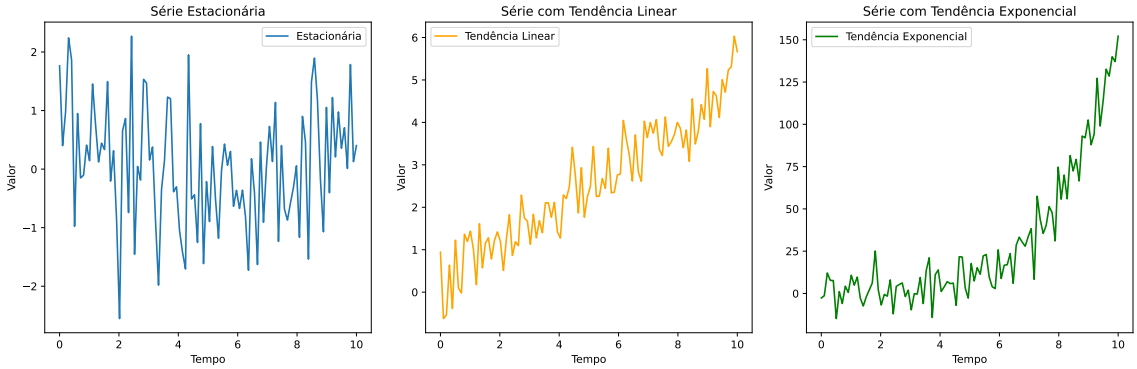


Figura 2.9: a) Série temporal estacionária. b) Série temporal não-estacionária com tendência linear. c) Série temporal não-estacionária com crescimento exponencial.

Fonte: Autoria própria.

2.1.3 Séries Temporais Longas

A análise de séries temporais longas desempenha um papel importante em uma variedade de campos, desde a previsão do tempo até o planejamento financeiro. Conforme ilustrado na Figura 2.10, as aplicações de previsão de séries temporais longas são vastas e impactam significativamente a tomada de decisões em setores como tráfego urbano, energia, finanças e medicina.

No tráfego urbano, por exemplo, a STL é essencial para prever padrões de deslocamento e otimizar o planejamento urbano. No setor energético, a capacidade de prever a demanda de energia e os preços do mercado é vital para a gestão e sustentabilidade dos recursos. Em finanças, as previsões de longo prazo são fundamentais para avaliar riscos e planejar investimentos. Na medicina, a STL pode ser utilizada para o desenvolvimento de modelos de previsão de doenças e alocação de recursos médicos. A meteorologia, talvez uma das áreas mais conhecidas de aplicação de STL, utiliza previsões de longo prazo para antecipar mudanças climáticas e desastres naturais, o que é essencial para a preparação e resposta a emergências.

Cada uma dessas áreas enfrenta desafios únicos ao aplicar STL devido às características inerentes dos dados, como a periodicidade e a presença de ciclos longos e complexos. No entanto, embora a necessidade de compreender e prever tais séries seja unanimemente reconhecida, não há um consenso claro sobre o que define uma 'longa' série temporal. Esta ambiguidade na definição pode influenciar diretamente a metodologia de previsão e a interpretação dos resultados em cada um desses campos de aplicação.

Segundo Chen et al. (2023), a caracterização de uma série temporal como longa ou curta é subjetivamente variável, dependendo do contexto específico e da aplicação. A ausência de uma definição padronizada torna desafiadora a comparação entre diferentes modelos e metodologias, especialmente ao avaliar

Figura 2.10: Aplicações em séries temporais.
Fonte: Autoria própria.

o desempenho preditivo em diversos horizontes temporais, ou seja, os períodos futuros até os quais as previsões se estendem. Ilustrando essa falta de consenso entre os autores, Manibardo et al. (2021) sugere que a definição de uma série longa depende da área de aplicação, enquanto Hyndman e Athanasopoulos (2018) associa o comprimento da série ao seu ciclo, considerando um elemento crucial nas tarefas de previsão. Segundo este último, uma série é considerada longa quando o intervalo de previsão supera o ciclo máximo presente nos dados. No setor financeiro, previsões que se estendem por mais de dois anos são classificadas como STL Harvey et al. (2007), refletindo os ciclos econômicos que variam de dois a dez anos. Na meteorologia, previsões com horizontes superiores a um mês são identificadas como STL Kim et al. (2016), enquanto na previsão de carga elétrica, horizontes acima de um ano são considerados longos Khuntia et al. (2016). Na previsão de tráfego, horizontes de alguns dias já são vistos como STL James et al. (2021), apesar dos ciclos máximos serem de apenas algumas horas ou um dia.

Adicionalmente, o estudo de Zhou et al. (2020) destaca que a maioria dos métodos existentes consegue prever até 48 passos à frente. A Figura 2.11 ilustra como o desempenho de uma rede LSTM se deteriora significativamente após esse horizonte de previsão, evidenciando uma redução na capacidade de inferência e na eficiência do processamento das predições. Isso sugere que, em horizontes de tempo mais extensos, a capacidade preditiva dos modelos geralmente diminui em comparação com horizontes mais curtos, ressaltando um desafio importante no estudo de séries temporais longas.

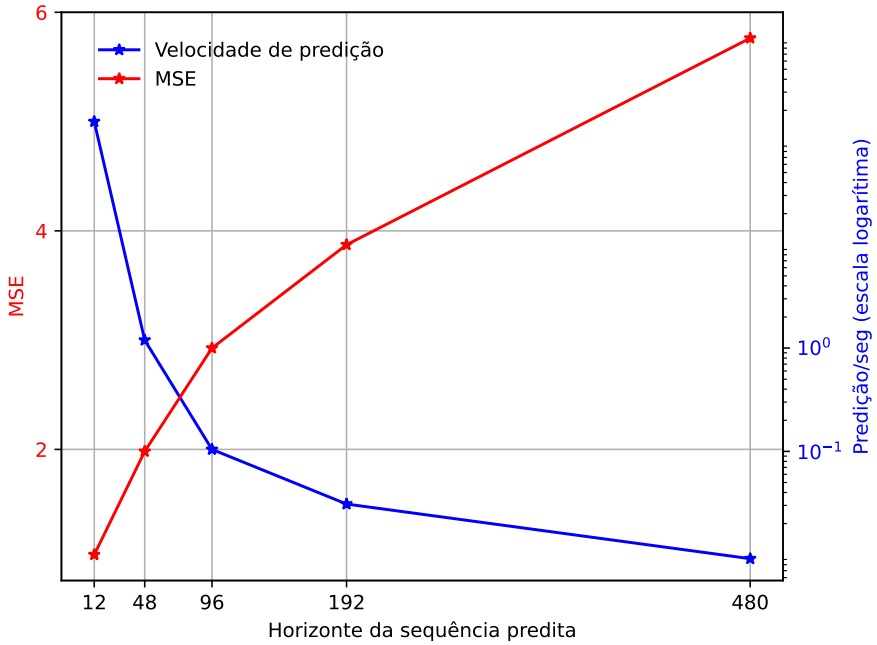


Figura 2.11: A capacidade de previsão de uma LSTM na previsão a longo prazo (STL). A partir de um comprimento de 48, o MSE aumenta e a velocidade de inferência cai rapidamente.

Fonte: Adaptação de Zhou et al. (2020).

Assim, após uma investigação detalhada das práticas e pesquisas em diversos campos, cujos detalhes serão mais aprofundados no Capítulo 3 de revisão literária, identificamos os principais modelos que se encaixam na definição de STL. Neste contexto, são particularmente notáveis os algoritmos capazes de gerar previsões para mais de 48 passos, utilizando conjuntos de dados frequentemente usados como referência. Estes conjuntos de dados, fundamentais para avaliar o desempenho e a aplicabilidade dos modelos STL, serão detalhadamente descritos no capítulo dedicado aos resultados.

2.2 Abordagens para previsão de Séries Temporais

Esta seção explora metodologias avançadas empregadas na previsão de séries temporais. Tais abordagens são fundamentais na antecipação de comportamentos futuros em séries temporais, permitindo previsões para horizontes de tempo estendidos. A seguir, são discutidos os principais métodos utilizados neste domínio, enfatizando suas respectivas formulações matemáticas, vantagens e limitações.

2.2.1 Múltiplos Passos Diretos (DMS)

A abordagem direta para previsão de séries temporais é um método que visa prever múltiplos passos à frente em uma única passagem para a frente através do modelo. Considerando uma série temporal definida na seção 2.1 e que deseja-se prever os próximos z passos à frente $\hat{y} = \{\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+z}\}$.

$$\begin{aligned} \hat{y}_{t+1} &= h_1(T,t) + \varepsilon_1 \\ \hat{y}_{t+2} &= h_2(T,t+1) + \varepsilon_2 \\ &\vdots \\ \hat{y}_{t+z} &= h_n(T,t+z-1) + \varepsilon_z \end{aligned} \tag{2.6}$$

onde ε_i representa o ruído associado a cada previsão, podendo incluir erros aleatórios, desvios sistêmicos ou outras formas de incerteza na previsão. Além disso, utilizando a propriedade das redes neurais que suportam múltiplas saídas, é possível substituir todos os modelos individuais h_i por um único modelo h_z mais complexo, que é capaz de prever todos os z valores futuros simultaneamente:

$$y_{t+z} = h_z(T) + \varepsilon_z \tag{2.7}$$

O processo de previsão é mostrado na Figura 2.12. Nesta abordagem, os valores de várias etapas de tempo são diretamente previstos sem acumular erros, embora a relação entre $x_{t+1}, x_{t+2}, ..., x_{t+z}$ não esteja sendo modelada neste modelo. Além disso, esta metodologia exige o treinamento de mais modelos ou um único modelo com mais parâmetros.

Vantagens

- Eficiência Computacional: Uma única passagem para a frente é necessária para gerar todas as previsões.
- Paralelismo: Pode ser facilmente paralelizado, já que cada previsão é independente.
- Simplicidade: Menos propenso a erros de acumulação, já que não há retroalimentação das previsões.

Desvantagens

- Complexidade do modelo: o modelo pode tornar-se complexo e difícil de treinar se o horizonte de previsão for muito longo.
- Incapacidade de adaptar-se a novas informações: uma vez que o modelo é treinado, ele não se adapta a novas informações sem retrabalho.

Nesta abordagem a previsão de séries temporais é viável empregar tanto a estratégia de usar múltiplos modelos para cada etapa de previsão quanto a de utilizar um único modelo para todas as etapas. Cada estratégia tem suas vantagens e desvantagens distintas. Optar por treinar um modelo separado para cada etapa do horizonte de previsão permite que cada modelo seja otimizado especificamente para sua etapa, potencializando a precisão das previsões. No entanto, essa abordagem pode ser mais onerosa em termos computacionais e complexa na manutenção, devido à necessidade de treinar e armazenar vários modelos.

Em contrapartida, o uso de um único modelo para prever todas as etapas do horizonte de previsão simultaneamente tende a ser mais eficiente em termos de custo computacional e armazenamento. Este modelo, com múltiplas saídas, é mais fácil de ser mantido e atualizado. Porém, essa estratégia pode comprometer a precisão das previsões, especialmente em horizontes de previsão mais longos, devido à complexidade e dificuldade de treinamento do modelo.

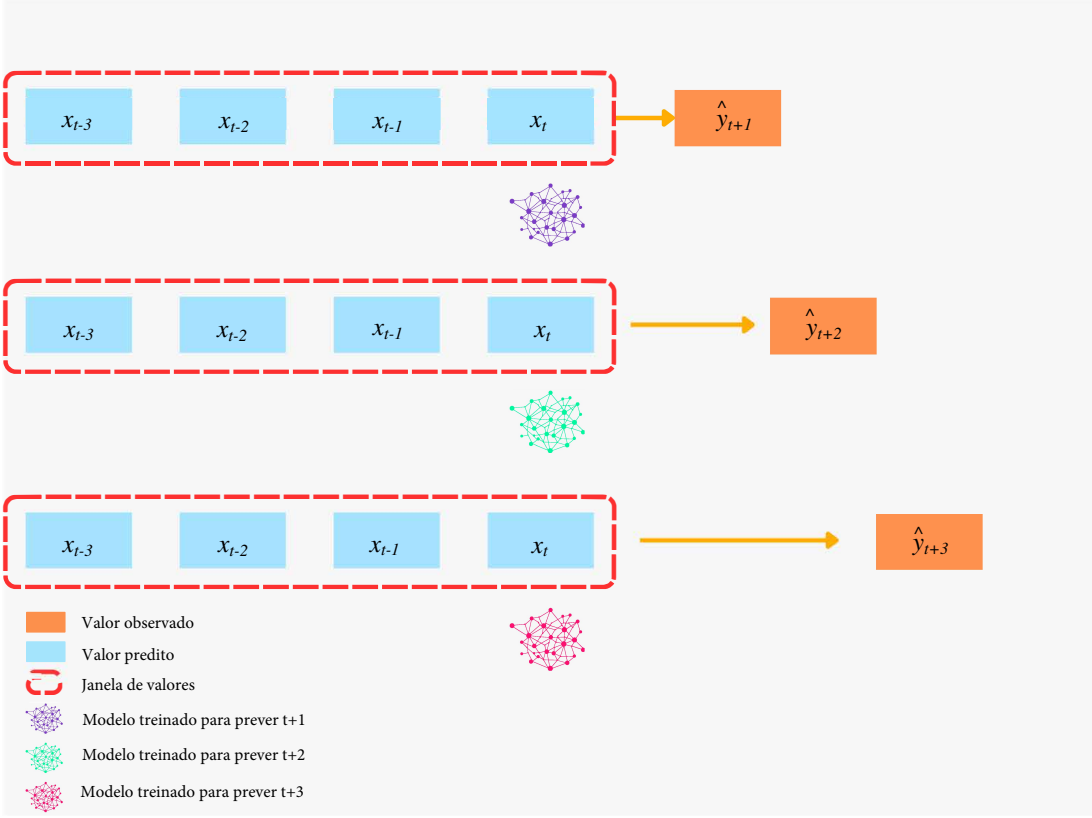


Figura 2.12: Ilustração do funcionamento do método DMS

2.2.2 *Múltiplos Passos Iterados (IMS)*

Neste método, um único modelo é desenvolvido e treinado para fazer previsões de um passo à frente. Entretanto, para fazer previsões de múltiplos

passos à frente, o modelo é usado iterativamente. A previsão para $t+1$ é feita e então alimentada de volta ao modelo para fazer a previsão para $t+2$ e assim por diante.

$$\begin{aligned} \hat{y}_{t+1} &= h(x_1, x_2, \ldots x_t) \\ \hat{y}_{t+2} &= h(x_2, x_3, \ldots \hat{y}_{t+1}) \\ &\vdots \\ \hat{y}_{t+n} &= h(x_{t+n-(n-1)}, x_{t+n-(n-2)}, \ldots, \hat{y}_{t+n-1}, \hat{y}_{t+n-2}, \ldots, \hat{y}_{t+1}) \end{aligned} \tag{2.8}$$

Essa abordagem pode ser menos precisa do que o DMS, especialmente para horizontes de tempo mais longos, pois os erros em previsões anteriores são propagados em previsões subsequentes. No entanto, é menos exigente em termos computacionais. Ao escolher entre as abordagens é importante considerar as necessidades específicas da aplicação, a disponibilidade de recursos computacionais e a precisão desejada para as previsões. A Figura 2.13 ilustra este processo.

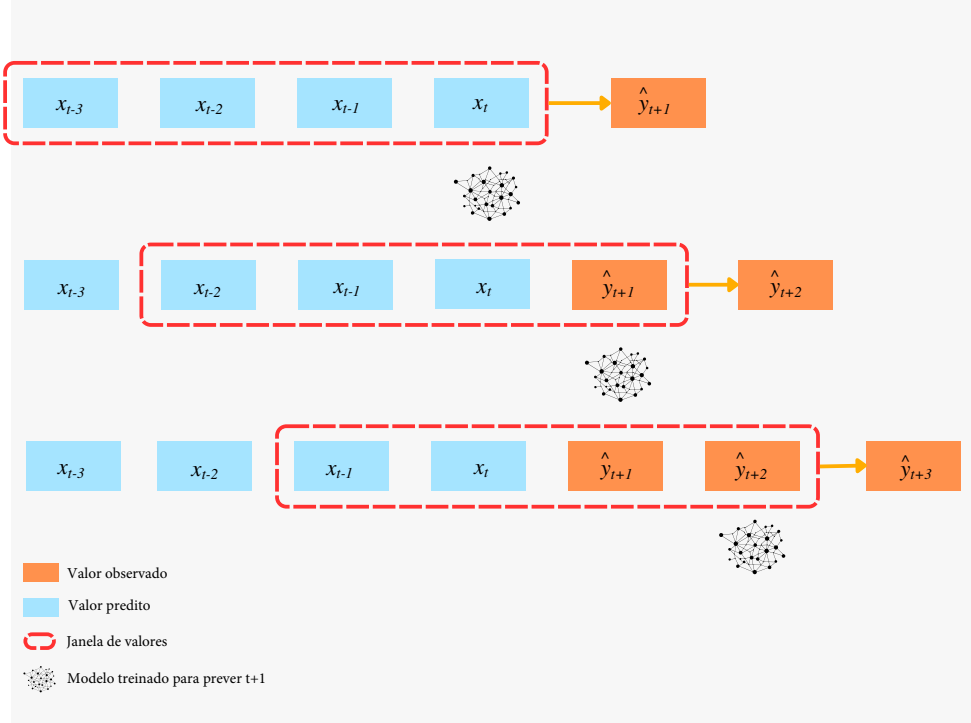


Figura 2.13: Ilustração do funcionamento do método IMS

Fonte: Elaborada pelo autor.

As vantagens e desvantagens do método de IMS para previsão de séries temporais estão listadas a seguir:

Vantagens

- **Simplicidade de Implementação:** O IMS requer apenas um modelo para realizar previsões de múltiplos passos à frente, simplificando a estrutura

do modelo e o processo de treinamento.

- **Flexibilidade:** Como o modelo é iterativamente alimentado com suas próprias previsões, ele pode adaptar-se dinamicamente às mudanças nos padrões de dados ao longo do tempo, o que é útil em séries temporais não estacionárias.

- **Eficiência Computacional:** Ao contrário do DMS, que pode exigir o treinamento de vários modelos para previsões de múltiplos passos à frente, o IMS utiliza um único modelo, potencialmente reduzindo o custo computacional e a necessidade de armazenamento.

Desvantagens

- **Acumulação de Erros:** A maior limitação do IMS é a propagação e acumulação de erros ao longo das previsões iterativas. Erros em previsões iniciais são alimentados de volta ao modelo, podendo amplificar imprecisões em previsões subsequentes.

- **Sensibilidade a Distúrbios:** A abordagem iterativa torna o modelo particularmente sensível a distúrbios ou anomalias nos dados. Variações abruptas ou pontos fora da curva podem afetar significativamente a qualidade das previsões futuras.

- **Dificuldade em Modelar Dependências de Longo Prazo:** Embora o IMS possa teoricamente capturar dependências dinâmicas, na prática, pode ser desafiador para o modelo manter a precisão ao longo de horizontes de previsão mais longos devido à propagação de erros mencionada anteriormente.

2.3 Aprendizado Profundo

O aprendizado profundo é uma subcategoria do aprendizado de máquina que se concentra na construção e treinamento de redes neurais com três ou mais camadas. Essas redes neurais são projetadas para emular o processo de aprendizagem do cérebro humano, lidando com grandes volumes de dados. Enquanto um cérebro humano possui bilhões de neurônios interconectados, uma rede neural contém nós, frequentemente referidos como neurônios artificiais, que são interligados por estruturas equivalentes a sinapses biológicas (LeCun et al., 2015).

Um dos aspectos mais notáveis do aprendizado profundo é sua capacidade de aprender características ou representações de maneira automática, diferenciando-se significativamente dos modelos tradicionais de aprendizado

de máquina. Nos métodos convencionais, a extração e a seleção de características eram processos majoritariamente manuais, exigindo uma intervenção humana considerável e sendo frequentemente suscetíveis a vieses e a erros (Bengio, 2012). No entanto, as redes neurais profundas revolucionaram este aspecto ao introduzir a automação dessa fase crítica. Por meio de uma sequência de transformações lineares e não lineares, essas redes são capazes de identificar e extrair características relevantes dos dados de forma independente. Este processo é otimizado utilizando métodos baseados em gradiente, permitindo que as redes neurais profundas aprendam representações de dados mais complexas e sutis de maneira eficiente e robusta (Goodfellow et al., 2016). A Figura 2.14 ilustra um comparativo entre o aprendizado profundo e o aprendizado de máquina tradicional.

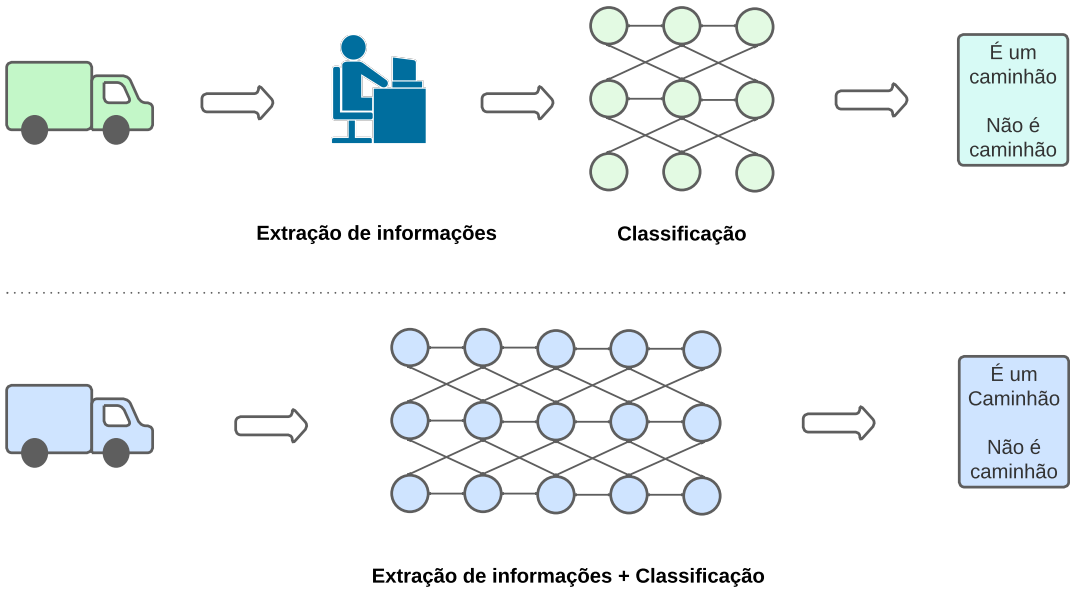


Figura 2.14: Diagrama comparativo entre aprendizado de máquina tradicional e aprendizado profundo, destacando a automação na extração de características por redes neurais profundas no aprendizado profundo.

Fonte: Elaborada pelo autor.

O aprendizado profundo tem aplicações em várias áreas, incluindo processamento de linguagem natural, visão computacional, medicina e até mesmo na criação de arte. A capacidade de processar grandes conjuntos de dados complexos torna o aprendizado profundo uma ferramenta valiosa para qualquer tarefa que requeira a interpretação de grandes quantidades de informações não estruturadas (Schmidhuber, 2015). No entanto, além das informações não estruturadas, o aprendizado profundo também demonstrou eficácia significativa no tratamento de dados estruturados. Dados estruturados, aqueles organizados em tabelas e com características bem definidas, são comuns em muitos contextos, como sistemas financeiros, registros médicos e transações

comerciais. A aplicação de modelos de aprendizado profundo a esses dados pode revelar padrões complexos e interações que métodos estatísticos tradicionais ou abordagens de aprendizado de máquina menos sofisticadas podem não capturar.

Como ilustrado na Figura 2.14, o aprendizado profundo pode ser entendido como um sistema que recebe dados brutos como entrada. Através de uma série de transformações lineares e não lineares, o sistema produz uma saída. Ele é capaz de ajustar seus parâmetros internos para que essa saída se aproxime o máximo possível da saída desejada. Para simplificar o conceito, adotamos um paradigma comum à maioria dos sistemas de aprendizado profundo. O processo inicia com os dados brutos de entrada, que são processados por N blocos de transformações lineares e não lineares. Estes blocos são responsáveis pelo aprendizado de representações. É importante salientar que, diferentemente das redes neurais tradicionais, as redes de aprendizado profundo caracterizam-se por terem um número significativamente maior desses blocos.

2.3.1 Redes Neurais Recorrentes

As Redes Neurais Recorrentes (RNN) são variantes de Redes Neurais Artificiais que surgiram nos anos 80 (Rumelhart et al., 1986; Elman, 1990; Werbos, 1988) e são especializadas no tratamento de dados sequenciais. Sua estrutura em cadeia as torna particularmente eficientes para tarefas de modelagem gerativa e processamento de sequências temporais. Essas redes têm sido fundamentais em aplicações como tradução automática (Al-Muzaini et al., 2018), interpretação de linguagem natural (Yao e Guan, 2018) e reconhecimento vocal (Shewalkar et al., 2019).

A RNN têm a habilidade de armazenar informações, usando dados de entradas passadas para influenciar decisões presentes. Ao contrário das redes neurais convencionais que tratam cada entrada e saída como independentes, as saídas da RNN são influenciadas por entradas anteriores. Mesmo que informações futuras pudessem ser valiosas para determinar a saída de uma sequência, as redes recorrentes padrão não levam em conta esses dados futuros em suas estimativas. Um traço marcante das RNN é o compartilhamento de parâmetros em suas camadas. Enquanto em redes *feedforward* cada nó possui pesos distintos, na RNN, um mesmo conjunto de pesos é usado em todas as camadas. No entanto, esses pesos são continuamente atualizados utilizando o *backpropagation* (BP).

Para compreender como uma rede recorrente funciona internamente, uma representação simplificada é ilustrada na Figura 2.15. A rede possui uma entrada x_t, que é um componente de uma sequência, e produz uma saída

y_t. Ao processar x_t, a rede gera um estado oculto, denominado H_t, que atua como uma memória acumulativa da atividade anterior da rede. Esse estado oculto é atualizado a cada novo componente da sequência e é usado pela própria rede para influenciar a saída correspondente ao próximo componente da série. Assim, a rede recorrente consegue integrar informações anteriores, fornecendo um contexto dinâmico para a previsão subsequente.

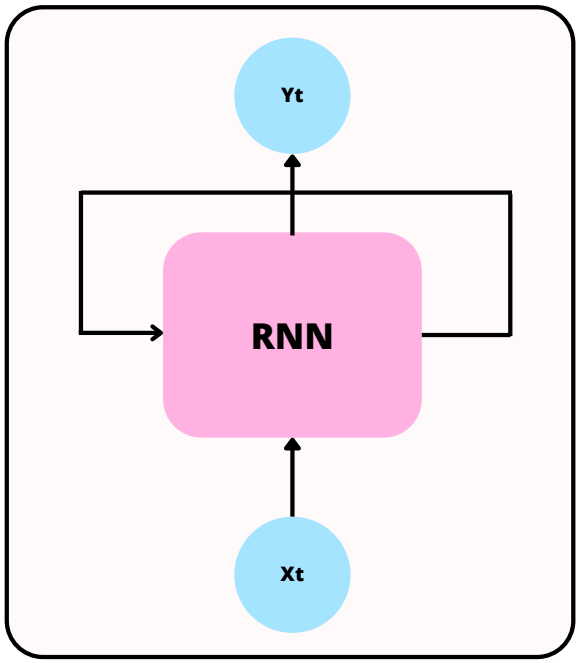


Figura 2.15: Uma representação simplificada de uma Rede Neural Recorrente mostrando como é calculado o estado oculto H_t, que é então retroalimentado na rede e combinado com a próxima entrada da sequência. Esse mecanismo permite que as RNN retenham informações de elementos anteriores da sequência e as utilizem para processar o elemento subsequente da série.

Fonte: Elaborada pelo autor.

Durante o processo de aprendizado, as RNN podem enfrentar desafios significativos, como os gradientes que se dissipam ou se amplificam excessivamente, fenômenos conhecidos como desvanecimento e explosão de gradientes, respectivamente. Estas questões surgem devido à natureza das próprias RNN em propagar informações ao longo de sequências, o que pode resultar em gradientes que se tornam muito pequenos ou excessivamente grandes, afetando adversamente o processo de aprendizado. Para mitigar esses problemas, diferentes variações de RNN foram desenvolvidas, cada uma com suas abordagens únicas para lidar com as limitações da RNN padrão e melhorar a eficiência do aprendizado em tarefas específicas.

Long Short-Term Memory - LSTM

A Long Short-Term Memory (LSTM) é uma variação avançada das RNNs, projetada para superar limitações como o desvanecimento do gradiente. Incorporando um estado celular, a LSTM consegue manter informações por mais tempo, sendo capaz de reter dados de elementos iniciais em uma sequência.

Esta arquitetura mais complexa que as RNNs padrão inclui três portões principais: *forget gate*, *input gate* e *output gate*. O *forget gate* decide quais informações antigas devem ser mantidas ou descartadas, enquanto o *input gate* determina quais informações do elemento atual são relevantes. Por fim, o *output gate* utiliza as informações armazenadas no estado celular para processar o elemento atual da sequência. Os portões foram ilustrados na Figura 2.16.

Estes portões permitem que a LSTM controle o fluxo de informações fazendo dela uma arquitetura capaz de analisar sequências temporais. A capacidade de reter informações importantes e determinar a relevância das informações atuais, as LSTMs são essenciais para tarefas como previsão de séries temporais e processamento de linguagem natural.

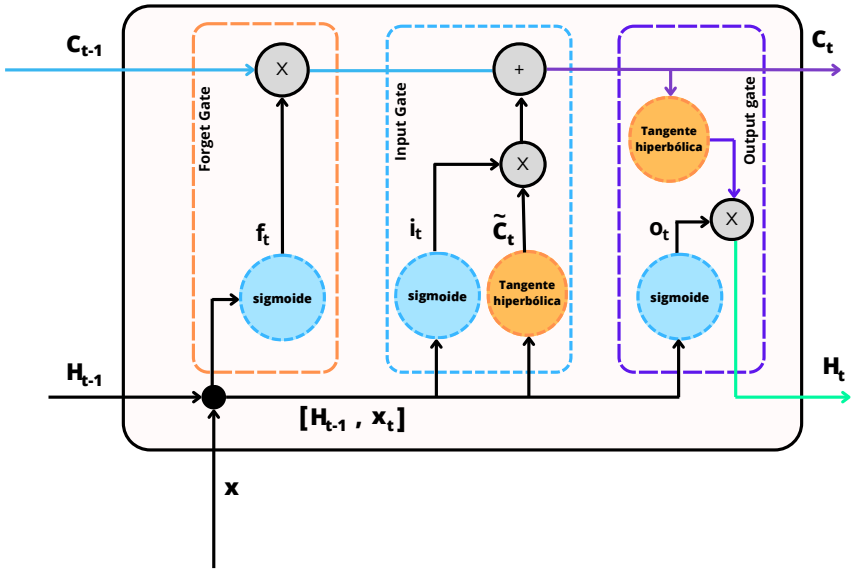


Figura 2.16: A arquitetura de um neurônio LSTM. O estado celular é representado como C, enquanto a entrada é x_t e o estado oculto é H_t.

Fonte: Elaborada pelo autor.

Gated Recurrent Unit (GRU)

Em 2014, Cho et al. (2014) propuseram uma variante da RNN que possui uma estrutura mais simples do que a LSTM, o que lhe torna mais fácil de treinar e requerendo menos cálculos (Yang et al., 2020). GRU não salva informações usando estado de célula, mas em vez disso usa uma condição oculta. Assim como a LSTM, a GRU utiliza mecanismos chamados de *gates* para aprender dependências de longo prazo. Esses portões são denominadas

update gate e *reset gate*, que realizam diferentes operações e são capazes de aprender quais informações devem ser adicionadas ou removidas do estado oculto. O *reset gate* da GRU controla se novas informações devem ser perdidas, enquanto o *update gate* é para lembrar (Yang et al., 2020). O modelo GRU é mostrado na Figura 2.17.

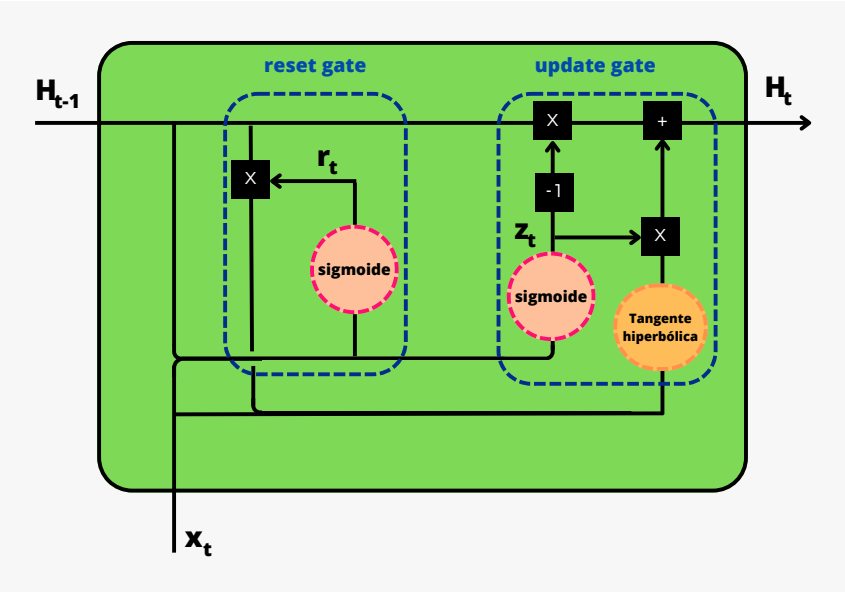


Figura 2.17: Arquitetura da Gated Recurrent Unit.

Fonte: Adaptada e modficada de Yang et al. (2020)

O portão de atualização ajuda o modelo GRU a determinar quanto das informações passadas precisam ser transmitidas para o futuro, como mostrado na Equação (2.9).

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}H_{t-1}) \tag{2.9}$$

onde x_t é o valor de entrada, ele é multiplicado pelo seu próprio peso $W^{(z)}$. O mesmo se aplica para H_{t-1}, que armazena informações sobre as unidades de tempo anteriores e é multiplicado pelo seu próprio peso $U^{(z)}$. O portão de reset seleciona quanto do conhecimento anterior esquecer. Esta fórmula é a mesma que a do portão de atualização; a distinção está nos pesos e no uso do portão expresso pela Equação (2.10).

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}H_{t-1}) \tag{2.10}$$

H_t é calculado como mostrado na Equação (2.11).

$$H_t = \tanh(Wx_t + r_t \odot UH_{t-1}) \tag{2.11}$$

onde na Equação (2.11), H_{t-1} é o estado oculto, U é a matriz de peso, e $\odot$ denota o produto Hadamard (produto elemento a elemento).

2.3.2 Redes Convolucionais

As Redes Neurais Convolucionais (CNNs) são especializadas no processamento de dados organizados em matrizes. Esses dados podem ser bidimensionais, como no caso de imagens, ou unidimensionais, como em séries temporais. Inspiradas na visão humana, as CNNs foram desenvolvidas há algum tempo, mas só ganharam destaque em 2012. Esse reconhecimento veio com a boa performance da arquitetura AlexNet (Krizhevsky et al., 2012) na competição ImageNet, voltada para o reconhecimento de imagens. Uma CNN é um tipo particular de rede neural artificial voltada para preservar relações espaciais nos dados. No campo da previsão de series temporais, o modelo *Temporal Convolutional* (TCN) (Bai et al., 2018), que é uma variante das CNNs, obteve bons resultados. Ele consegue predizer sequências de comprimento variável com menor uso de memória e complexidade reduzida, além de aproveitar o paralelismo, não visto nas RNNs.

A entrada típica para uma CNN é uma matriz que passa por sucessivos filtros. Esses filtros capturam relações desde a granularidade mais básica até detalhes de alto nível. Por exemplo, na análise de imagens, características de baixo nível seriam arestas, cantos, cores ou orientação de gradiente; uma característica de alto nível poderia ser um rosto completo. Camadas consecutivas de convoluções e ativações, geralmente intercaladas com camadas de agrupamento, constroem uma CNN profunda. A ilustração da Figura 2.18 demonstra esse processo:

- A matriz de entrada (24×24) é submetida a uma primeira camada de convolução que aplica filtros para extrair características de baixo nível, resultando em uma nova representação espacial (21×21).
- Uma segunda camada de convolução atua sobre a saída da primeira, capturando características mais complexas, reduzindo ainda mais a dimensão da representação (7×7).
- Após as camadas de convolução, a representação aprendida é achatada em um vetor (4×4) que alimenta uma camada totalmente conectada, típica das redes neurais tradicionais, onde todas as entradas são conectadas a cada neurônio.
- Finalmente, a camada totalmente conectada produz a saída do modelo, como um vetor de predições ou classificações (1×1), dependendo da tarefa em questão.

A Figura 2.18 detalha o fluxo de dados através da CNN, evidenciando como a informação é transformada e refinada em cada etapa para extrair e utilizar eficientemente as relações espaciais intrínsecas aos dados de entrada.

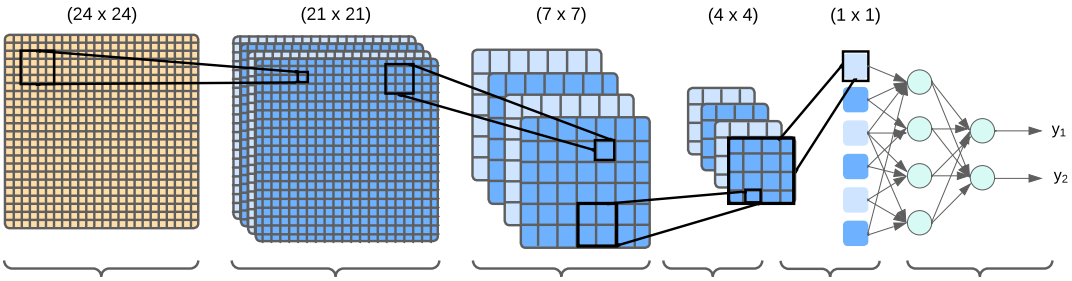


Figura 2.18: A ilustração de uma amostra de CNN com 2 camadas de convolução e uma camada totalmente conectada.

Fonte: Elaborada pelo autor.

Convolução e Redução de dimensão

Em cada camada convolucional, a operação de convolução é realizada por um filtro específico, também conhecido como *kernel*, que é aplicado sobre a saída da camada anterior, chamado de mapa de características. Esses mapas de características são passados adiante por novas convoluções gerando também novos mapas. Uma operação de convolução envolve pesos. Todas as unidades no novo mapa de características são processadas usando os mesmos pesos. Assim, é garantido que todas as unidades no novo mapa de características detectarão exatamente o mesmo padrão. Outra consequência não menos importante do compartilhamento de pesos é a diminuição no número de parâmetros aprendíveis, que se traduz em um procedimento de aprendizagem mais eficiente. Nas Redes Neurais Convolucionais Temporais, as características autoregressivas implicam que o valor no passo de tempo t deve depender apenas dos passos de tempo anteriores e não dos futuros. Para garantir esse comportamento, a operação de convolução padrão é substituída por convolução causal. Além disso, a convolução causal dilatada é uma técnica usada para aumentar o campo receptivo do TCN, permitindo aprender dependências de longo prazo dentro dos dados. Outro bloco de construção da CNN é a camada de agrupamento, cujo papel é subamostrar o mapa de características obtido. Um dos objetivos é reduzir o tamanho espacial da representação de dados. Além desse impacto na complexidade do aprendizado, um segundo objetivo é tornar a característica capturada pela camada convolucional invariante a anomalias locais, como distorções na análise de imagens.

Embora inicialmente projetadas para lidar com dados de imagem bidimensionais, as CNNs podem ser usadas para modelar séries temporais univariadas e multivariadas. Uma CNN unidimensional operará apenas sobre uma sequência em vez de uma matriz. Enquanto uma série temporal multivariada será alimentada à CNN como vários vetores correspondentes aos canais iniciais. A Figura 2.19 ilustra o processo de uma CNN 1D lidando com dados

econômicos multivariados. Neste exemplo, cada canal da entrada multivariada corresponde a uma série temporal diferente: o primeiro canal captura a taxa de desemprego, o segundo canal representa o Produto Interno Bruto (PIB), e o terceiro canal registra o número de empresas criadas durante o ano. Estes canais são tratados como camadas distintas dentro da matriz de entrada, permitindo que a rede processe múltiplas séries temporais simultaneamente.

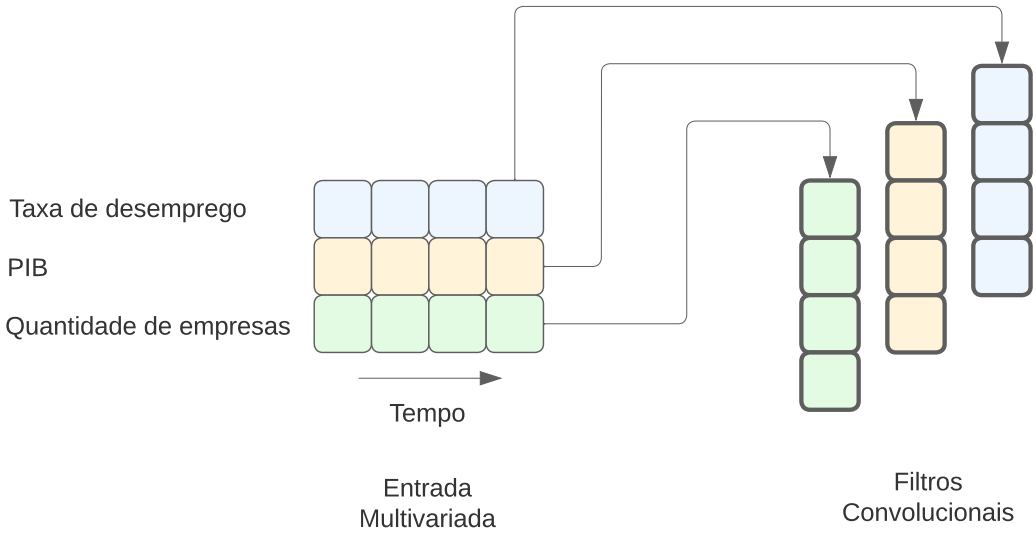


Figura 2.19: Ilustração de funcionamento de uma CNN 1D com entrada multivariada.

Fonte: Elaborada pelo autor.

Os filtros convolucionais são aplicados ao longo do eixo do tempo, movendo-se sobre cada canal para extrair recursos temporais relevantes. Esses filtros são capazes de capturar padrões como tendências, sazonalidades e relações entre os canais - por exemplo, como a taxa de desemprego pode influenciar ou ser influenciada pelo PIB e pela atividade empresarial. Esta abordagem permite que a CNN identifique relações complexas e interdependências entre diferentes indicadores econômicos ao longo do tempo.

2.3.3 *Transformers*

Os *Transformers*, introduzidos por Vaswani et al. (2017a), apresentaram uma mudança no processamento de sequências, especialmente em tarefas de *Natural Language Processing* (NLP) (Devlin et al., 2018). Esta arquitetura se juntou as Redes Neurais Recorrentes, *Long Short-Term Memory* e *Gated Recurrent Units*, que eram as abordagens padrão até então na tarefa de NLP. Com os *Transformers*, o processamento sequencial dos dados foi substituído por um método paralelo mais eficiente.

A crescente popularidade dos modelos de *Transformers* impulsionou o desenvolvimento de variantes. O termo *X-formers* é a maneira sob o qual diversas iterações e inovações do modelo *Transformer* original são agrupadas. Cada variante visa aprimorar o modelo base de diferentes maneiras, seja por eficiência computacional, precisão aprimorada, ou adaptabilidade a diferentes tipos de tarefas de processamento de dados. Estes modelos também demonstraram uma boa capacidade para modelar dependências e interações de longo alcance em dados sequenciais, tornando-os uma escolha interessante para a modelagem de séries temporais. Alguns modelos foram desenvolvidos para séries temporais, alcançando êxito em algumas aplicações, incluindo previsão (Li et al., 2019; Zhou et al., 2022b), detecção de anomalias (Xu et al., 2021a; Tuli et al., 2022) e classificação (Zerveas et al., 2021; Yang et al., 2021). Em particular, a sazonalidade ou periodicidade é um aspecto importante em séries temporais (Wen et al., 2021). Ainda é um desafio modelar efetivamente dependências temporais de longo e curto prazo e, ao mesmo tempo, capturar a sazonalidade (Wu et al., 2021; Wen et al., 2022).

A arquitetura deste modelo é composta por um componente de codificação e um de decodificação, cada um melhorado com mecanismos de atenção de múltiplas cabeças, como ilustrado na Figura 2.20. Essa arquitetura difere significativamente das anteriores, como RNN e LSTMs, que processavam informações de forma sequencial, limitando a captura de dependências de longo alcance. Em contraste, o *Transformer* processa dados de forma paralela, o que aumenta a eficiência e eficácia na modelagem de dependências complexas em tarefas de NLP e análise de séries temporais.

O *Encoder* e o *Decoder* são componentes centrais da arquitetura do *Transformer*. O Encoder é responsável por converter a sequência de entrada em um conjunto de representações contextuais. Ele é composto por várias camadas idênticas, cada uma contendo uma camada de auto-atenção de múltiplas cabeças e uma rede feedforward. A auto-atenção permite que cada parte da sequência de entrada se relacione de forma diferenciada, enquanto a rede *feedforward* processa os sinais sequencialmente. As conexões residuais e a normalização de camada em cada sub-camada ajudam no fluxo de gradientes durante o treinamento.

Por outro lado, o *Decoder* trabalha na sequência de saída e tem uma estrutura semelhante ao *Encoder*, com múltiplas camadas idênticas. Ele inclui uma camada de auto-atenção de múltiplas cabeças mascarada, que garante a autoregressividade do modelo, e uma camada de atenção que recebe as saídas do Encoder. Isso permite que o Decoder se concentre em partes relevantes da entrada para gerar a saída. Uma rede *feedforward* semelhante à do *Encoder* processa as entradas da camada de atenção.

Essa configuração do *Transformer*, com seu processamento paralelo e eficiente, torna-o particularmente adequado para capturar dependências complexas em tarefas de processamento de linguagem natural e análise de séries temporais.

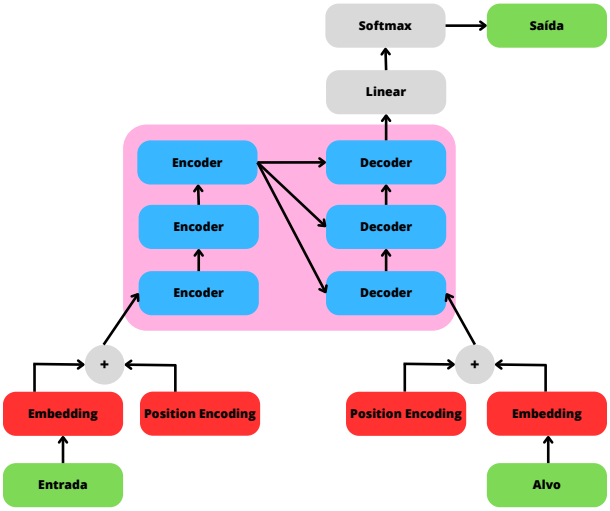


Figura 2.20: Representação esquemática da arquitetura do *Transformer* mostrando o componente de codificação (*encoders*) e o componente de decodificação (*decoders*). A transformação da entrada é ilustrada, com as setas indicando o fluxo de dados do modelo até a saída.

Fonte: Adaptação de Al-Mammar, J.

Esse mecanismo de atenção multicabeça em ambos os componentes, codificador e decodificador, permite que o modelo foque em diferentes partes da entrada para diferentes "cabeças", proporcionando uma visão mais abrangente dos dados.

No contexto de modelagem de séries temporais, a entrada do codificador pode ser uma sequência de observações passadas, enquanto a entrada do decodificador seria uma sequência projetada para previsões futuras, inicialmente baseada em dados históricos e atualizada iterativamente com previsões progressivas. O resultado do decodificador seria a previsão do próximo ponto na série temporal. Durante o treinamento, técnicas como 'teacher forcing' podem ser adaptadas, onde o modelo é alimentado com as respostas corretas durante fases anteriores do treinamento, para melhorar a aprendizagem e a precisão das previsões futuras. Este método facilita o aprendizado do modelo na captura de padrões temporais e sazonalidades complexas, essenciais na análise de séries temporais.

Mecanismo de Atenção

Os modelos *Transformer*, introduzidos por Vaswani et al. (2017b), revolucionaram o processamento de dados sequenciais ao implementar um mecanismo chamado "atenção de produto escalar dimensionado". No mecanismo

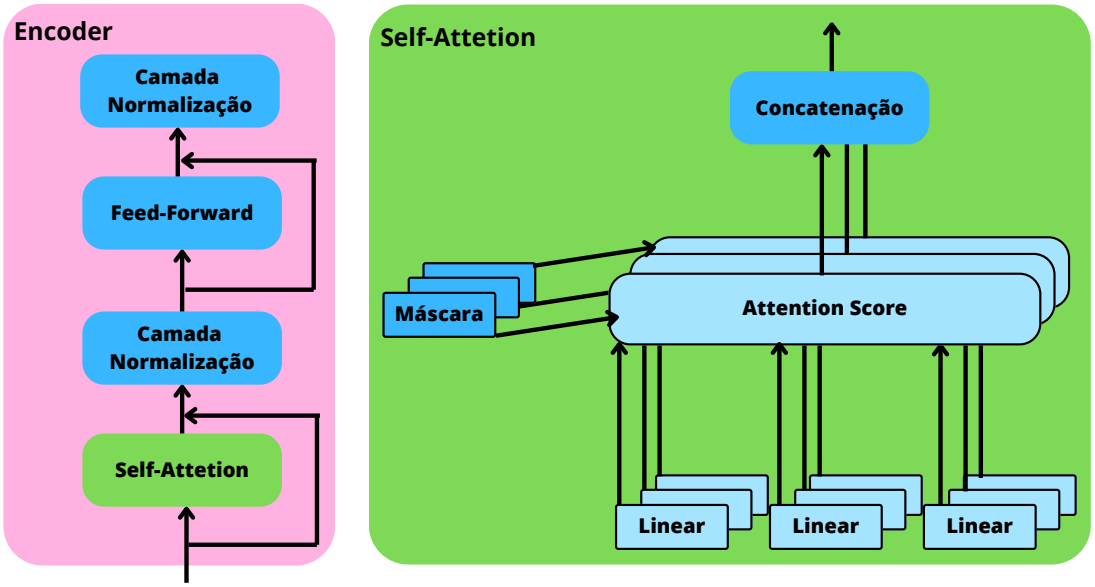


Figura 2.21: Representação da arquitetura *Multi-Head Attention* em um *Transformer*, ilustrando como o módulo de Atenção divide suas consultas (Query), chaves (Key) e valores (Value) em múltiplas cabeças de atenção. Cada cabeça processa esses elementos de forma independente, permitindo ao *Transformer* capturar múltiplas relações e nuances para cada palavra, e combinando-as para formar um score de atenção final.

Fonte: Adaptação de Al-Mammar, J.

de atenção, especialmente no contexto dos *Transformers*, as letras Q, K, e V representam consulta (*Query*), chave (*Key*) e valor (*Value*), respectivamente. Esses componentes são essenciais para o funcionamento do mecanismo de atenção, que é central na arquitetura do *Transformer*. A consulta refere-se aos dados que estamos tentando entender ou para os quais estamos buscando informações. A chave representa os dados aos quais estamos comparando a consulta para avaliar a relevância. O valor é a informação real que queremos recuperar com base na correspondência entre a consulta e a chave. A atenção é computada usando a Equação 2.12,

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

onde $\sqrt{d_k}$ é um fator de escala derivado da dimensão das chaves para evitar gradientes extremamente pequenos durante o treinamento.

Esse conceito é comum em sistemas de busca. Imagine uma busca em uma plataforma de comércio eletrônico, onde você insere preferências (consulta) que são comparadas com as especificações dos produtos (chave), resultando na apresentação de produtos (valor) que mais correspondem às suas preferências. No *Transformer*, este processo é realizado matematicamente através do cálculo da semelhança entre consulta e chave, determinando a relevância de cada valor.

Um dos fundamentos que ajudam os modelos *Transformers* em sequências temporais é devido a condição posicional, no qual incorporam a noção de ordem na sequência, os *Transformers* utilizam codificações posicionais. Em analogia às CNNs, onde a posição espacial das características é importante, a codificação posicional nos *Transformers* assegura que a sequência dos dados seja considerada.

É importante destacar que tanto a atenção própria quanto a atenção de múltiplas cabeças são invariantes à permutação de suas entradas. Ou seja, a ordem dos elementos na sequência não afeta o resultado, uma vez que o mecanismo de atenção utiliza operações de produto escalar. Para incorporar informações sobre a posição sequencial dos elementos na entrada, o *Transformer* utiliza codificações posicionais. Isso é essencial, pois o mecanismo de atenção por si só não possui uma noção de ordem sequencial, o que pode ser crítico para entender a estrutura inerente dos dados processados, como linguagem natural ou séries temporais. A equação posicional é dada por:

$$\tilde{X} = X + PE \tag{2.13}$$

onde X representa a entrada para a camada de atenção, PE denota a codificação posicional adicionada a X, e $\tilde{X}$ é a saída que combina a entrada original com informação posicional.

A codificação posicional pode ser implementada de várias maneiras, incluindo o uso de funções seno e cosseno com frequências variáveis, conforme proposto por Vaswani et al. (2017b). Essa abordagem permite que o modelo distinga não apenas elementos diferentes na sequência, mas também suas posições relativas.

2.3.4 Comparativo Entre Arquiteturas

Uma distinção fundamental entre as RNNs e os *Transformers* reside no modo de entrada dos dados sequenciais. Enquanto as RNNs processam os dados sequencialmente, elemento por elemento, os *Transformers* inserem toda a sequência de uma vez, utilizando codificação posicional para manter a sequencialidade dos dados. Essa diferença é ilustrada na Figura 2.22. Em tarefas de NLP, os *Transformers* se baseiam em codificações de palavras e frases para representar os dados de entrada, o que resulta em um aumento na acurácia. De maneira mais abrangente, codificar uma sequência de entrada significa converter cada elemento de entrada em um vetor de identificadores. Em aplicações de séries temporais, o codificador sintetiza informações e incorpora comportamento temporal das séries em análise.

Outra diferença é que os *Transformer* não precisam processar os dados

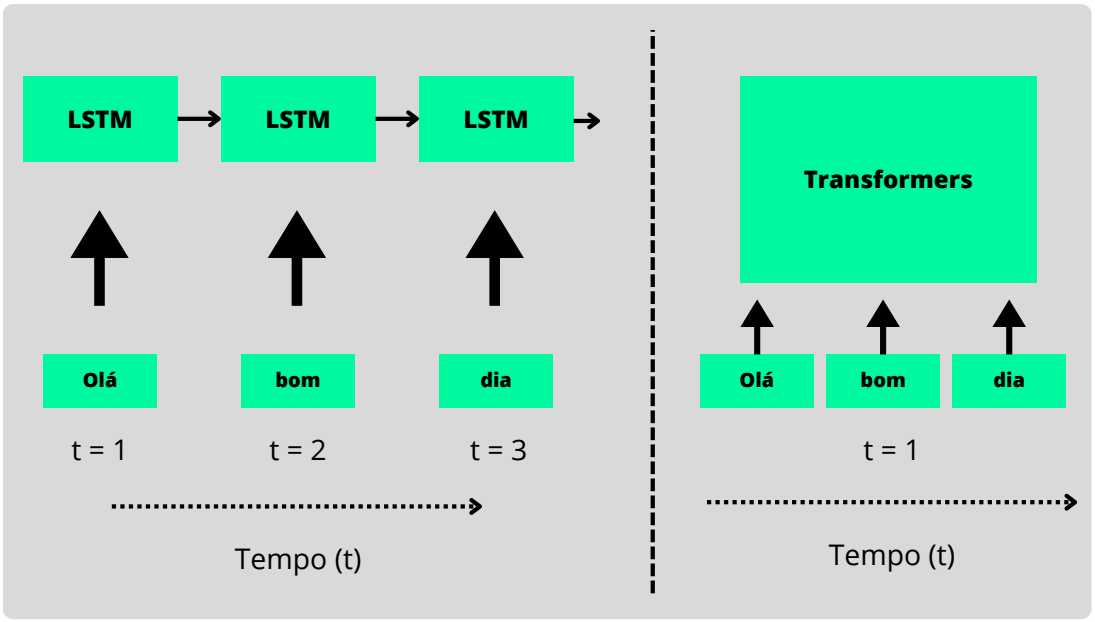


Figura 2.22: Comparativo entre RNN e *Transformers*

Fonte: Elaborada pelo autor.

de entrada de forma sequencial. Isto significa que os *Transformer* podem processar o final da sequência de entrada antes do início, permitindo uma paralelização mais extensa em comparação com as RNNs, o que reduz o tempo de treinamento.

Quanto à captura de dependências em dados sequenciais para previsão, as RNNs e os *Transformers* também divergem. As unidades RNN têm a capacidade de lembrar ou esquecer partes das informações armazenadas anteriormente, dependendo da importância. Confiar em uma única unidade RNN exige que ela memorize eventos passados importantes para prever valores futuros. Em contraste, os *Transformers* usam duas unidades especializadas, formando a arquitetura de codificador-decodificador. No contexto da previsão de séries temporais, o codificador extrai características importantes dos eventos passados, enquanto o decodificador utiliza essas informações para prever valores futuros. Essa capacidade de focar em diferentes partes dos dados e aprender relações complexas é comparável à habilidade das CNNs de extrair características espaciais em imagens.

Um problema com as LSTMs é a incapacidade de atribuir maior importância a certas partes da sequência de entrada durante o processamento. O mecanismo de atenção, fundamental no modelo *Transformer*, foi desenvolvido como uma melhoria para capturar dependências, particularmente aquelas de longo alcance, em uma sequência. No modelo tradicional de codificador-decodificador, somente o vetor final produzido pelo codificador é utilizado para iniciar o decodificador. Entretanto, resumir uma longa sequência de entrada em um único vetor reduz o desempenho do transdutor. A atenção se concen-

tra em usar todas as codificações intermediárias geradas pelo codificador para enriquecer as informações transmitidas ao decodificador.

Outra distinção é que as RNNs permitem apenas previsões de um passo à frente. Assim, previsões de múltiplos passos à frente são realizadas em RNNs por meio da repetição de previsões de um passo à frente, onde o valor previsto é reinserido como entrada para o próximo valor da série. Este método iterativo tem a desvantagem da propagação de erro. Em contraste, os transformadores permitem previsões diretas de $x_T + h$, onde $h \in \{1, 2, \ldots, H\}$. A distinção dessas arquiteturas pode ser vista na Figura 2.23.

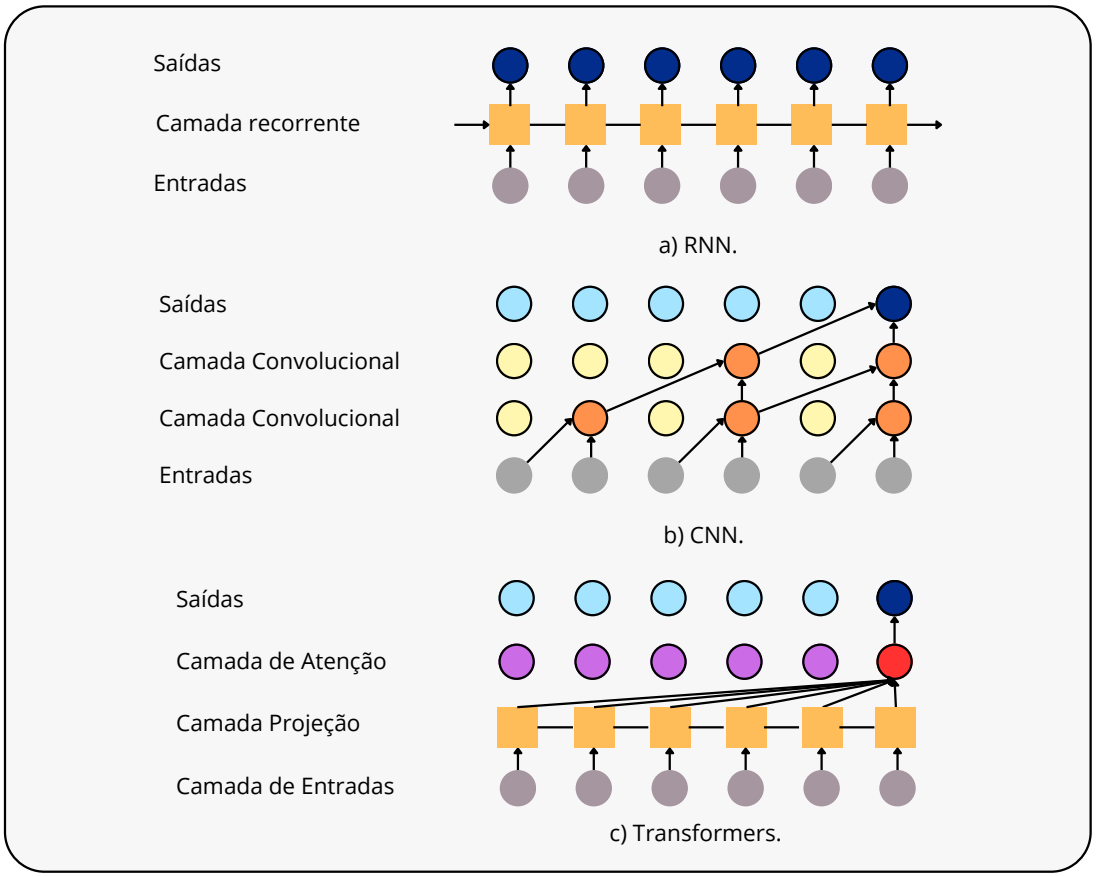


Figura 2.23: Ilustração das arquiteturas RNN, CNN e *Transformers*

Fonte: Elaborada pelo autor.

A imagem ilustra as três arquiteturas, e pode-se comentar que em relação ao *Transformer*, ela está simplificada com foco em gerar uma única saída. No entanto, uma arquitetura de *Transformer* típica para tarefas de processamento de sequências, como tradução automática, normalmente geraria uma sequência de saídas, uma para cada posição de tempo na sequência alvo. Cada saída é gerada com base no cálculo de atenção, que envolve três componentes principais: consultas (*queries*), chaves (*keys*) e valores (*values*). Para cada posição de saída, o modelo calcula um conjunto de pontuações de atenção que determinam a importância relativa de cada posição de entrada para a produção daquela saída específica. Essas pontuações são calculadas usando

um produto escalar entre a consulta (associada à posição de saída) e todas as chaves (associadas às posições de entrada). Após a aplicação de uma função *softmax* para normalizar essas pontuações em uma distribuição de probabilidade, elas são usadas para ponderar os valores (que também estão associados às posições de entrada) antes de serem somados para produzir a saída final para aquela posição.

2.4 Métricas de Erro de previsão

No campo do aprendizado de máquina, métricas como o Erro Médio Absoluto (MAE) e o Erro Quadrático Médio (MSE) são frequentemente empregadas para avaliar a qualidade de modelos com saídas contínuas. Contudo, no contexto de previsões em séries temporais, a escolha de métricas se torna mais complexa devido à existência de uma ampla gama de métricas disponíveis. Embora estudos anteriores tenham frequentemente abordado o MAE e o MSE, é importante reconhecer que não existe um padrão universalmente aceito para determinar a métrica mais adequada. Essa diversidade de métricas se deve ao fato de que cada uma possui suas próprias limitações e não é capaz de capturar todos os aspectos relevantes de uma previsão de maneira isolada. Neste contexto, é planejado explorar várias dessas métricas para avaliar a eficácia das previsões, sem atribuir peso excessivo a qualquer uma delas em particular.

Erro Médio Absoluto

Erro Médio Absoluto ou *Mean Absolute Error* (MAE) é uma medida de erros entre observações pareadas que expressam o mesmo fenômeno. Logo, é a diferença absoluta média entre y_i e $\hat{y}_i$, descrito pela fórmula Equação 2.14:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \tag{2.14}$$

Onde $\hat{y}_i$ representa o valor previsto, y_i o valor verdadeiro e m o tamanho da amostra.

Erro Quadrático Médio

Mean Squared Error (MSE) ou Erro Quadrático Médio é uma métrica muito utilizada em problemas de regressão linear, quando há uma linearidade entre os dados de entrada e os dados a serem preditos. Sua abordagem, calcula a diferença entre os resultados obtidos e o resultado real, eleva cada diferença ao quadrado, e depois calcula a média.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{2.15}$$

2.5 Considerações Finais

Este capítulo explorou as bases teóricas do aprendizado de máquina e sua aplicação na previsão de séries temporais, concentrando-se nas arquiteturas de redes neurais como RNN, LSTMs, GRUs, *Transformers* e CNNs. A discussão abrangeu a utilidade de cada tecnologia no tratamento de dados sequenciais, ressaltando suas características e aplicações.

As RNNs são úteis para processar sequências de dados, enquanto LSTMs e GRUs oferecem melhorias na capacidade de capturar dependências de longo prazo, lidando com o desafio do desvanecimento do gradiente. Os *Transformers*, por sua vez, trouxeram uma nova perspectiva para o processamento paralelo de sequências, provando ser valiosos no processamento de linguagem natural.

No contexto das séries temporais, as CNNs, conhecidas por sua aplicação em visão computacional, também demonstraram ser bastante eficazes. Elas se destacam na identificação de características locais e padrões temporais, aproveitando a arquitetura de convolução para aprender automaticamente os recursos mais relevantes diretamente dos dados brutos. Essa capacidade de extração automática de características diminui a necessidade de engenharia de recursos manual, tornando as CNNs uma opção interessante para a previsão de séries temporais.

Além disso, as CNNs mostram uma menor tendência ao problema de desvanecimento do gradiente, comparativamente a outras arquiteturas, o que favorece um treinamento mais estável e eficiente. Sua eficiência computacional, permitindo o processamento paralelo de dados, as torna adequadas para lidar com grandes volumes de dados, característica valiosa ao trabalhar com séries temporais complexas.

A discussão incluiu também a importância de métricas de desempenho na avaliação de modelos de previsão de séries temporais, enfatizando a necessidade de selecionar cuidadosamente a arquitetura de rede adequada para obter os melhores resultados.

O capítulo seguinte se voltará para o estado da arte em séries temporais, apresentando estudos de caso e aplicações práticas de redes neurais. Serão detalhados os principais algoritmos, organizados de forma a oferecer uma visão ampla das tendências e inovações no campo.

CAPÍTULO

3

Revisão Literária

Este capítulo apresenta uma revisão de literatura sobre os principais algoritmos aplicados em séries temporais, assim como os trabalhos que fundamentaram o estudo realizado com a utilização de conexões densas.

A revisão da literatura realizada neste estudo foi influenciada pelos métodos sugeridos por Khan et al. (2003), embora com ajustes para se adequar melhor às especificidades da previsão de séries temporais.

1. Compreender os principais elementos envolvidos na previsão de séries temporais.
2. Identificar os fatores que influenciam a precisão dessas previsões.
3. Reconhecer os métodos/modelos utilizados para prever séries temporais e outros instrumentos de análise.
4. Estudar os algoritmos estado da arte na previsão de séries temporais longas.
5. Avaliar como as conexões foram úteis em outras áreas e como aplicá-las em séries temporais.

Em seguida, foram utilizados dois mecanismos de busca para coletar informações: o site Periódicos Capes e Google Acadêmico. Os artigos foram avaliados com base em: (i) sua relevância para as questões de pesquisa mencionadas; (ii) se passaram por revisão por pares; (iii) o ano de sua publicação; e (iv) o número de citações recebidas. Por último, as informações mais relevantes de cada artigo foram extraídas para planejar e comparar os experimentos que seriam realizados no estudo atual.

Para a busca de algoritmos relacionados a séries temporais longas, as palavras-chave utilizadas foram: "long time series forecasting", "deep learning for time series", "residual connections in neural networks", "dense connections in time series", e "time series models". Essas buscas foram direcionadas a literatura publicada entre os anos de 2000 e 2022, visando capturar tanto os desenvolvimentos mais recentes quanto trabalhos fundamentais que estabeleceram as bases para técnicas modernas de previsão de séries temporais.

3.1 Evolução das Séries Temporais Longas

Nesta seção são avaliados os avanços recentes em aprendizado profundo aplicados a sequências longas de predição (LSTF), organizando os modelos desenvolvidos conforme as abordagens técnicas de suas arquiteturas. Assim, a partir das arquiteturas predominantes é possível categorizar os modelos baseados em: Redes Neurais Recorrentes, Redes Neurais Convolucionais, *Transformer* e compostos. Modelos que não se enquadram em nenhum dos grupos citados acima, mas que conseguiram contribuir de alguma forma para essa área, foram agrupados como diversos.

3.1.1 Redes Recorrentes

As redes LSTM e GRU têm sido utilizadas para a previsão de séries temporais devido à sua capacidade de capturar dependências de longo prazo. O LSTMa, uma variante do LSTM que incorpora um mecanismo de atenção, que foi introduzido por Bahdanau et al. (2014) para melhorar a capacidade do modelo de focar em aspectos relevantes da sequência de entrada, mitigando assim o problema do gradiente desvanecente. Apesar dos benefícios, a inclusão de um mecanismo de atenção pode introduzir uma complexidade adicional ao modelo, impactando o tempo de treinamento e exigindo um ajuste cuidadoso dos componentes de atenção para otimizar o desempenho. Além disso, para séries multivariadas, o modelo não pode selecionar explicitamente séries quais delas são mais relevantes para fazer previsões.

Para solucionar possíveis problemas, Chang et al. (2017) propôs a o modelo DilatedRNN que foi inspirada na eficiência das CNNs dilatadas no estudo de Yu e Koltun (2015), porém transposta para um cenário recorrente. O DilatedRNN possui conexões residuais recorrentes dilatadas de multi-resolução que são empilhadas para melhorar a análise de dependências entre séries longas, já que diferentes camadas se concentram em diferentes resoluções ao longo da sequência, além de ajudar nos problemas relacionados ao gradiente e na redução no número de parâmetros necessários.

O MQ-RNN, desenvolvido por Wen et al. (2017), é um modelo de sequência

a sequência que utiliza um codificador e decodificador LSTM. Este modelo é treinado utilizando uma função de perda de quantil, com o objetivo de aprender incertezas. No entanto, o MQ-RNN limita-se à estimação de quantis univariados e não modela diretamente as correlações entre múltiplas estimativas de quantis, realizando apenas através de espaços de parâmetros compartilhados. A arquitetura deste modelo pode ser complexa, exigindo um ajuste nos dados de treinamento para possibilitar a captura efetivamente as características de diferentes horizontes de previsão.

Proposto por Qin et al. (2017), o DA-RNN é uma rede neural recorrente de duas etapas, utilizando um mecanismo de atenção. No codificador, o DA-RNN introduz um mecanismo de atenção de entrada inovador que permite ao modelo focar adaptativamente em séries de influência relevantes e atribuir-lhes pesos de forma dinâmica. Já no decodificador, é implementado um mecanismo de atenção temporal que se concentra de forma adaptativa na saída do codificador em todas as etapas temporais. No entanto, o modelo apresenta desafios ao lidar com sequências que exibem periodicidade e autocorrelação.

O DeepAR, apresentado por Salinas et al. (2020), é um método constituído por RNNs para previsão de séries temporais univariadas que emprega uma abordagem probabilística. A entrada da RNN consiste no valor defasado da sequência e multivariáveis. O treinamento e a previsão seguem o método convencional de modelos autorregressivos. Ele é capaz de aprender um modelo global a partir de séries temporais correlacionadas e de capturar padrões complexos, como sazonalidade. No entanto, o modelo pode exigir uma quantidade substancial de dados de treinamento para aprender as distribuições com precisão, e seu treinamento pode ser desafiador devido à sua natureza probabilística, além de tratar apenas séries univariadas.

Por fim, o MTSMFF apresentou uma estrutura de aprendizado profundo para previsão de séries temporais multivariadas utilizando aprendizagem encoder-decoder com Bi-LSTM, aliada com um mecanismo de atenção temporal. Esta abordagem representou um método inovador para a representação dinâmica de dados de séries temporais multivariadas, capaz de aprender conjuntamente padrões de dependências temporais de longo prazo e características de correlação não-lineares dos dados temporais multivariados. Os experimentos foram conduzidos em cinco conjuntos de dados de séries temporais multivariadas. A abordagem se destaca pela sua capacidade de lidar com complexidades e inter-relações inerentes aos dados multivariados, oferecendo uma previsão mais precisa e detalhada em diversos contextos temporais.

Tabela 3.1: Visão geral de trabalhos publicados em predição de séries temporais utilizando como base em modelos RNN.

Modelo	Ano	Técnica	Comparativo
LSTMa	2014	LSTM	-
Dilated RNN	2017	Dilated Connection+RNN	Vanilla RNN, Vanilla LSTM, VanillaGRU, Stacke RNN, Stack LSTM Stack GRU, Skip RNN
MQ-RNN	2017	LSTM+Encoder–Decoder+MLP	-
DA-RNN	2017	Attention+LSTM	ARIMA, NARX RNN, Encoder–Decoder, Attention RNN, Input-Attn-RNN
LSTNet	2018	RNN + CNN	AR, LRidge, LSVR, TRMF, GP, VARMLP, RNN-GRU
DeepAR	2020	RNN + AR	Snyder, Croston, ISSM, ETS, rnn-gaussian, rnn-negbin
MTSMFF	2020	LSTM + Encoder–Decoder	SVR, RNN, CNN, LSTM, GRU, seq2seq, seq2seq-BI, seq2seq-ATT

3.1.2 Redes Convolucionais

Com bons resultados na área de visão computacional, as redes neurais convolucionais chamaram a atenção devido a sua capacidade de redução de características e processamento paralelo em sequências longas tem motivado a aplicação dessas estruturas no campo de LSTF. Um dos desafios era a captura de dependências locais de curto prazo, levantando questões sobre como capturar dependências globais de longo prazo de forma eficaz. Avanços foram alcançados com o advento da convolução causal (Oord et al., 2016) e da convolução causal dilatada (Yu e Koltun, 2015), mitigando esse problema. Nesta seção, são apresentados os trabalhos recentes sobre modelos baseados em CNN, e que estão sumarizados na Tabela 3.2.

O modelo TCN Bai et al. (2018) utilizou uma estrutura de rede convolucional dilatada obtendo resultados melhores que redes LSTM. Além disso, a compressão de recursos do TCN o tornou mais apropriado para LSTF do que as RNNs convencionais. A causalidade é importante em tarefas de previsão, e o problema de vazamento de informações futuras surge quando há sobreposições temporais entre o resultado e a entrada. As convoluções causais são pertinentes em previsões autorregressivas, nas quais o resultado anterior é utilizado como entrada para a previsão futura. Esta arquitetura apresenta limitações, incluindo: 1) partilha de um único filtro convolucional em cada camada, o que leva a uma tendência de extrair características temporais médias dos dados ou características da camada anterior. A complexidade de séries temporais, que muitas vezes contêm dinâmicas temporais, torna-se necessário a extração de características distintas com um conjunto diversificado de filtros convolucionais. 2) limitação dos campos receptivos efetivos das camadas intermediárias, especialmente aquelas mais próximas às entradas, resultando na perda de relações temporais durante o processo de extração de características.

O DSANet (Huang et al., 2019) utilizou CNN com Atenção, para diminuir os efeitos de causados nos modelos LSTM e GRU em previsões de longo prazo de

séries multivariadas. Uma rede de autoatenção dual para lidar com sequências dinâmicas periódicas ou aperiódicas. O modelo utiliza duas convoluções paralelas (convolução temporal global e convolução temporal local) para capturar padrões temporais globais e locais. Além disso, o mecanismo de autoatenção serve para aprender as conexões entre múltiplas séries temporais. Apesar do DSANet conseguir realizar predições em séries temporais multivariadas, o modelo tem limitações quanto ao horizonte. Nos testes de desempenho, o modelos demonstrou uma limitação em alguns conjuntos de dados ao prever além de 48 passos.

Diferentemente do TCN e outros modelos que utilizaram convolução causal, o MICN (Wang et al., 2022a) são utilizadas múltiplas ramificações de núcleos convolucionais distintos para modelar separadamente diferentes padrões da sequência. Em cada ramificação, as características locais da sequência são extraídas por meio de um módulo local baseado em convolução de subamostragem, e, sobre isso, a correlação global é modelada por um módulo global fundamentado em convolução isométrica. Este modelo conseguiu reduzir a complexidade temporal e espacial para uma linear, diferente dos seus antecessores que utilizam como base a CNN.

O SCINet (Liu et al., 2021a) é uma abordagem inovadora para a modelagem de séries temporais, especialmente aquelas com dinâmicas temporais complexas. Diferente de outros modelos que utilizam a convolução causal para evitar o vazamento de informações futuras durante o treinamento, o SCINet opta por eliminar essa restrição. Ao invés disso, introduz uma arquitetura recursiva de redução de amostragem junto com convoluções para capturar as dependências temporais nas séries temporais. O *SCI-Block* é um componente inovador do modelo SCINet, que desempenha um papel importante na análise e previsão de séries temporais complexas ao empregar uma estratégia hierárquica e decomposição para lidar com dados em múltiplas escalas de tempo. Este bloco utiliza uma técnica de *downsampling*, que permite reduzir a dimensão temporal dos dados enquanto preserva informações essenciais, abordando assim o desafio de processar sequências longas e capturar dependências de longo prazo de maneira eficiente. Cada *SCI-Block* é estruturado em quatro submódulos: `modules_P`, `modules_U`, `modules_phi` e `modules_psi`. Os submódulos são construídos com uma sequência de camadas, visando realizar extração de padrões nas subsequências. Eles fazem parte do *Interactive Learning*, no qual foi descrito pelos autores. Algumas operações matemáticas são realizadas em cada módulo, mas não foram explicadas pelos autores o motivo delas serem daquela maneira. A abordagem de redução de amostragem ajuda a concentrar os recursos de modelagem nas partes mais informativas da série temporal, o que pode melhorar a eficiência e a eficácia do modelo. No entanto, um desafio

mencionado é que, apesar dessas inovações, o SCINet ainda pode enfrentar dificuldades para capturar dependências de longo prazo de forma eficaz. Isso se deve ao campo receptivo limitado, que é o alcance dos dados de entrada que o modelo pode acessar ao fazer previsões para um ponto específico na série temporal. Embora a arquitetura de redução de amostragem possa ajudar a mitigar esse problema até certo ponto, ainda existe um limite para quão bem o modelo pode capturar relações temporais estendidas sem a capacidade de acessar diretamente informações de entrada mais distantes no tempo.

Tabela 3.2: Visão geral de trabalhos publicados em predição de séries temporais utilizando como base em modelos CNN.

Modelo	Ano	Técnica	Comparativo
TCN	2018	Causal CNN + Dilated CNN	LSTM, GRU, RNN
DSANet	2019	CNN + Attention	VAR, LRidge, LSVR, GP, GRU, LSTNet, TPA
SCINet	2022	CNN + Encoder–Decoder	Autoformer, Informer, Transformer, TCN, LSTNet, TPA-LSTM
MICN	2022	Highway-CNN+MLP	LogTrans, LSTNet, LSTM, Autoformer, FEDformer,Informer, Transformer, TCN

3.1.3 Transformers

O modelo Transformer, proposto Vaswani et al. (2017a), representa uma mudança paradigmática na arquitetura de aprendizado profundo para tarefas de sequência. Este modelo se diferencia dos tradicionais CNNs e RNNs ao utilizar um mecanismo de autoatenção para capturar correlações em dados sequenciais. Sua arquitetura oferece melhor paralelismo e uma interpretabilidade aprimorada em comparação com estruturas anteriores. Contudo, enfrenta desafios como alta complexidade computacional e dificuldades em capturar informações de longa distância.

O LogTrans (Li et al., 2019) introduz a convolução local ao *Transformer* e propõe a atenção LogSparse para selecionar intervalos de tempo seguindo intervalos que aumentam exponencialmente, o que reduz a complexidade para $O(L(logL)^2)$. O Reformer (Kitaev et al., 2020) apresenta a atenção de hashing sensível a localização (LSH) e reduziu a complexidade para $O(LlogL)$. O Informer (Zhou et al., 2020) expandiu o Transformer com a atenção ProbSparse baseada na divergência KL, alcançando também uma complexidade de $O(LlogL)$. Vale ressaltar que esses métodos são baseados no Transformer original e buscam aprimorar o mecanismo de auto-atenção para uma versão esparsa, que ainda segue a dependência e agregação ponto a ponto.

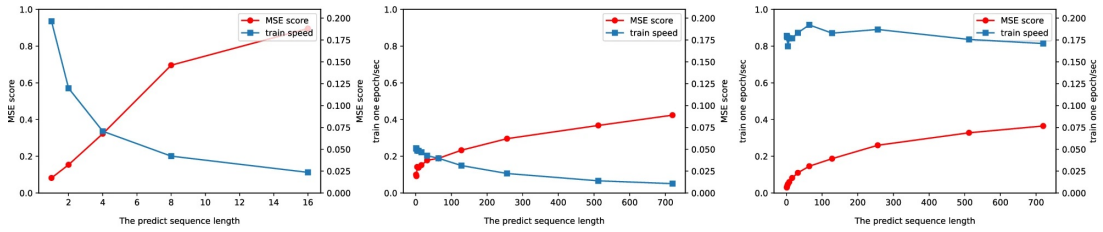
O modelo ProbSparse (Lee, 2020) propuseram um mecanismo de atenção baseado em correlação parcial, e o Pyraformer (Liu et al., 2021b) introduziu um mecanismo de atenção piramidal para representações multirresolução.

Na área de análise temporal, o Autoformer (Wu et al., 2021) se destacou ao integrar estratégias de decomposição de sequência em sua estrutura. O

FEDformer Zhou et al. (2022b), por sua vez, otimizou o *Transformer* do ponto de vista do domínio de frequência, alcançando complexidade linear $O(L)$ e melhorando a precisão de previsão. Chen et al. (Song et al., 2022) apresentaram o Quatformer, que incorpora aprendizado de rotação na atenção, e (Fan et al., 2022) desenvolveram o Sepformer para aprimorar a extração de características e reduzir a complexidade computacional.

Essas inovações demonstram um progresso contínuo na otimização dos modelos *Transformers*, abordando questões de complexidade, capacidade de captura de informação a longo prazo e interpretabilidade.

A arquitetura *Transformers* tem se destacado na área de previsão de séries temporais de longo prazo. Entretanto, o estudo de Zhu et al. (2023) aponta limitações no Autoformer Wu et al. (2021), especialmente na previsão de séries temporais curtas. Esta questão é evidenciada na Figura 3.1, onde se observa que, embora o Autoformer apresente um MSE inferior ao da rede LSTM em previsões de longo prazo, o desempenho do Autoformer é superado pela LSTM em previsões de curto prazo. Quanto à velocidade de treinamento, o Autoformer demonstra ser mais lento, porém, a sua velocidade de treino não varia significativamente com a mudança no comprimento das séries previstas.



(a) LSTM (b) Autoformer (c) DRCNN

Figura 3.1: Desempenho do modelo no ETTm2 em várias extensões de saída.

Fonte: DRCNN.

3.1.4 Compostos

A LSTNet, proposta por Lai et al. (2018), mescla um modelo linear autoregressivo tradicional com RNN. Essa abordagem supera outras RNNs e métodos estatísticos tradicionais ao integrar CNNs para detecção de tendências de curto prazo e RNNs para dependências de longo prazo. No entanto, o uso de RNN para previsões de longo prazo pode resultar em erros acumulativos. Além disso, a LSTNet captura apenas dependências lineares sequenciais, deixando as dependências lineares de padrões periódicos temporais não abordadas. A complexidade de sua arquitetura dupla torna a LSTNet computacionalmente intensiva, o que também eleva o potencial de problemas de sobreajuste, especialmente em conjuntos de dados menores. Além disso, esses modelos funcionam apenas para séries univariadas.

Tabela 3.3: Visão geral de trabalhos publicados em predição de séries temporais utilizando como base modelos Transformers.

Modelo	Ano	Técnica	*Baseline*	Descrição
LogTrans	2019	LogSparse self-attention + *Transformer*	ARIMA, ETS, TRMF, DeepAR	Autoatenção LogSparse para redução da complexidade temporal, demonstrando a capacidade do *Transformer* de lidar com dependências de longo prazo.
Informer	2021	Convolution + Transformer, ProbSparse Self-attention	LogTrans, Reformer, LSTM, ARIMA, Prophet, LSTM	Arquitetura do *Transformer* eficaz e computacionalmente econômica.
Autoformer	2021	Sequence Decomposition + Auto-Correlation + Self-attention	DeepAR, LogTrans, Informer, ARIMA, Prophet, LSTM Reformer,	Arquitetura de autocorrelação e decomposição.
Pyraformer	2022	Pyramidal Attention Module + *Transformer*	Informer, LogTrans, Longformer, Reformer, ETC	Módulo de Atenção Piramidal para representação multirresolução.
FEDformer	2022	Fourier Enhanced + Wavelet Enhanced + *Transformer*	Transformer, Autoformer, Informer, LogTrans, Reformer	Redução da complexidade temporal com decomposição do domínio de frequência baseada na arquitetura do Autoformer.

3.1.5 Diversos

Além dos modelos de referência mencionados anteriormente e dos métodos compostos associados, a pesquisa na área de (LSTF) tem se expandido para incluir abordagens menos convencionais, conforme ilustrado na Tabela 3.4. Cui et al. (2021) propõe o conceito de Inércia Histórica (HI) como uma nova linha de base para previsão de séries temporais de longa sequência. HI refere-se ao uso dos dados históricos mais recentes na série temporal como previsão direta. O estudo avaliou de forma experimental o poder da HI em quatro conjuntos de dados públicos do mundo real, demonstrando melhorias significativas de até 82% em relação aos modelos estado-da-arte. A pesquisa comparou o modelo proposto com derivações do *Transformes* em previsões univariadas e multivariadas, mostrando que a HI muitas vezes supera esses modelos, especialmente em previsões multivariadas. De acordo com os autores, um dos principais benefícios da HI é que ela garante que as saídas estejam em magnitude similar às entradas, o que é particularmente verdadeiro em cenários de previsão de longa sequência. No entanto, os autores também reconhece limitações. O HI pode não ser eficaz quando não há padrões periódicos na série temporal, ou quando esses padrões não estão incluídos nos dados históricos. Além disso, a pesquisa destaca que o modelo pode ser menos eficaz em previsões de curto prazo, em comparação com modelos de aprendizado profundo.

Zhou et al. (2022a) abordaram o problema do LSTF atribuído ao sobreajuste do ruído histórico. Eles projetaram uma arquitetura chamada Modelo de Memória Legendre Melhorado por Frequência (FiLM), que utiliza projeções polinomiais de Legendre para aproximar informações históricas, projeções de

Fourier para eliminar ruídos e uma aproximação de baixo ranque para acelerar os cálculos. Este modelo mostrou-se eficaz na mitigação do sobreajuste de ruído no LSTF, além de acelerar a inferência.

O modelo DLinear (Zeng et al., 2022) destaca-se pela sua capacidade de decompor e analisar de maneira distinta as componentes sazonais e de tendência em dados temporais utilizando uma abordagem simples frente ao demais modelos. O DLinear emprega um bloco de decomposição de séries, que utiliza uma média móvel para identificar e separar a tendência subjacente da série temporal. Este processo resulta em duas partes distintas: uma refletindo a tendência (média móvel) e outra representando os resíduos, ou seja, o que sobra após a remoção da tendência.

Para cada uma dessas componentes - sazonal e de tendência - o modelo DLinear aplica camadas lineares específicas. Estas camadas são responsáveis por modelar e entender os padrões sazonais e as tendências ao longo do tempo. Um aspecto notável do modelo é sua flexibilidade em lidar com canais de dados. Dependendo da configuração, ele pode tratar cada canal da série temporal de maneira independente, empregando um conjunto de camadas lineares para cada canal, ou pode optar por uma abordagem mais agregada, utilizando uma única camada linear para todos os canais.

A previsão final gerada pelo modelo é uma soma das saídas das camadas lineares para tendência e sazonalidade. Esta abordagem não só permite que o DLinear capture as nuances dos padrões temporais, mas também oferece uma análise detalhada das diferentes forças que influenciam a série temporal. Em suma, o DLinear é um modelo que combina técnicas de decomposição de séries temporais com aprendizado linear, adaptando-se para tratar os dados de forma individualizada ou coletiva, conforme necessário.

Tabela 3.4: Visão resumida dos trabalhos publicados em predição de séries temporais utilizando como base modelos *Transformers*.

Modelo	Ano	Técnica	Comparativo
N-Beats	2020	MLP	-
HI	2021	HI	Prophet, ARIMA, DeepAR, LSTMa, Reformer, LogTrans, Informer
N-HiTs	2022	MLP	N-BEATS, FEDformer, Autoformer, Informer, LogTrans, Reformer, DilRNN, ARIMA
FiLM	2022	Legendre projection + Fourier analysis	FEDformer, Autoformer, S4, Informer, LogTrans, Reformer
DLInear	2022	MLP	Informer, Autoformer, Pyraformer, FEDformer

Além disso, a meta-aprendizagem tem sido amplamente utilizada em diversos campos como uma técnica popular. Algumas das abordagens mencionadas acima também incorporaram a meta-aprendizagem para auxiliar no processo de previsão dos modelos. O trabalho (Oreshkin et al., 2021) propôs um *framework* genérico de meta-aprendizagem, que permite treinar redes neurais em conjuntos de dados de séries temporais e, posteriormente, aplicá-las a

diferentes conjuntos de dados de séries temporais-alvo sem a necessidade de re-treinamento. Isso é particularmente útil quando o número de amostras históricas disponíveis para algumas séries temporais-alvo é limitado, ajudando a superar a escassez de dados históricos no LSTF.

Oreshkin et al. (2019) exploraram como realizar tarefas complexas de previsão temporal utilizando modelos de aprendizagem profunda (DL) fortes e interpretáveis. Eles propuseram o modelo N-Beats, baseado na sobreposição de redes MLP e conexões residuais. Seu trabalho subsequente, N-HiTs Challu et al. (2022), aprimora o N-Beats para a tarefa LSTF, incorporando técnicas inovadoras de interpolação hierárquica e amostragem de dados em múltiplas taxas, enfatizando componentes de diferentes frequências e escalas durante a decomposição do sinal de entrada e a síntese das previsões.

3.2 Considerações Finais e Gaps das Literaturas

A previsão de séries temporais longas, observa-se que, apesar dos avanços significativos alcançados, cada metodologia apresenta seus próprios desafios. As RNNs, por exemplo, lidam com problemas como o gradiente desvanecente, enquanto as CNNs enfrentam dificuldades em estabelecer relações de longo alcance nos dados. Por sua vez, os *Transformers*, apesar de oferecerem vantagens em termos de interpretabilidade e capacidade de processamento paralelo, ainda enfrentam desafios relacionados à complexidade computacional e à captação de informações distantes no tempo.

A Figura 3.3 apresenta uma visão clara da progressão nas arquiteturas usadas em séries temporais, mostrando uma trajetória que se inicia com as Redes Neurais Recorrentes e modelos compostos, seguida pela integração das Redes Neurais Convolucionais. Essa sequência evolui para a inclusão de redes generativas, desembocando na adoção gradual dos *Transformers*. Esta sequência ilustra a tendência de desenvolvimento e aprimoramento contínuo no campo da previsão de séries temporais, refletindo uma mudança gradual de técnicas mais tradicionais para abordagens mais recentes.

No entanto, um *gap* na literatura atual diz respeito à investigação do papel das conexões residuais em modelos de previsão de séries temporais. Embora as conexões residuais tenham sido amplamente exploradas no contexto de tarefas de visão computacional, seu impacto em modelos de previsão de séries temporais ainda é uma área pouco estudada. Esta lacuna motivou a análise presente nesta tese, focada na exploração de como as conexões residuais podem melhorar a performance de modelos preditivos em séries temporais, particularmente em contextos onde a captura de dependências temporais de longo prazo é importante.

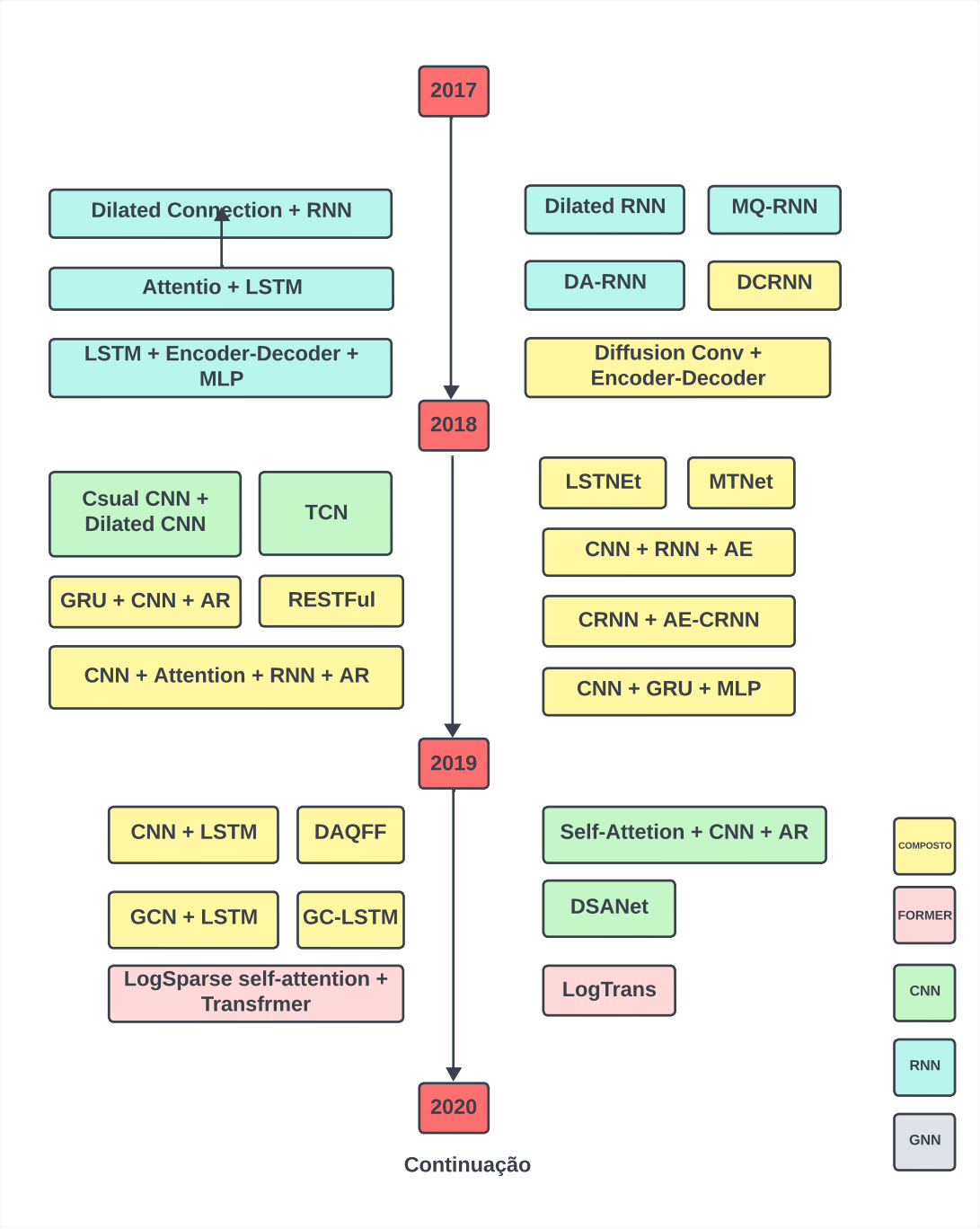


Figura 3.2: Evolução dos trabalhos em LSTF.

Fonte: (Benidis et al., 2022).

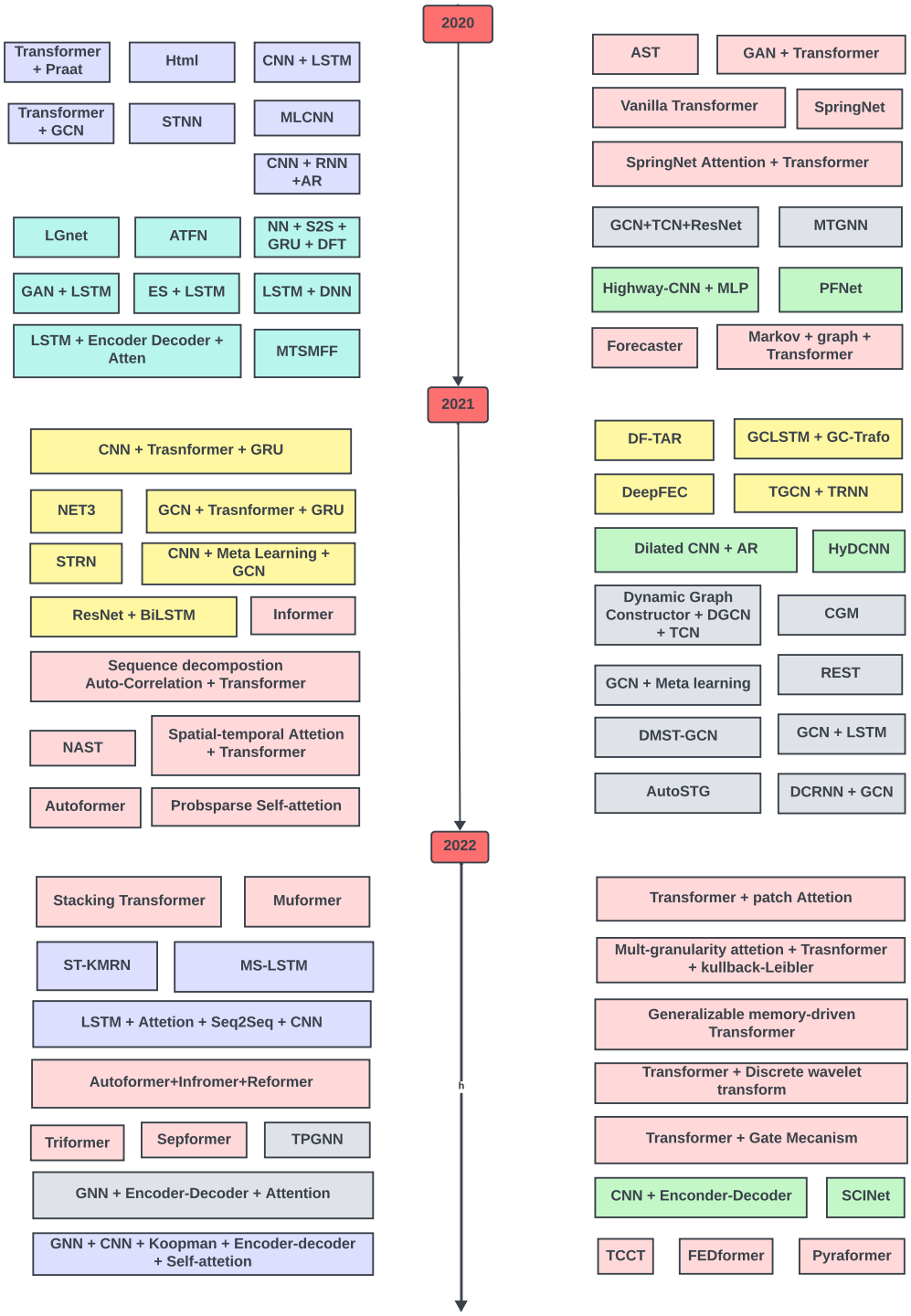


Figura 3.3: Continuação Evolução dos trabalhos em LSTF.

Fonte: (Benidis et al., 2022).

CAPÍTULO

4

Estudo de conexões residuais

No desenvolvimento inicial das redes neurais profundas, modelos com poucas camadas predominavam e foram apelidadas de redes rasas ou *shallow netorks* (Xu et al., 2021b). Contudo, observou-se um impulso constante para o treinamento de modelos mais profundos com o intuito de melhorar a generalização do modelo. Essa tendência pode ser vista na evolução dos resultados no conjunto de dados ImageNet (Russakovsky et al., 2015), onde modelos mais profundos demonstraram superioridade, conforme Figura 4.1. Observando a Figura 4.1, as redes AlexNet (8 camadas) (Krizhevsky et al., 2012), VGG (16 - 19 camadas) (Simonyan e Zisserman, 2014) e GoogleNet (Szegedy et al., 2015) representam redes sem conexões com baixa profundidade, em contraste com a ResNet que alcançou 152 camadas com um erro menor que suas concorrentes devido a adição de conexões residuais.

Recentemente, pesquisas como as apresentadas em (Oyedotun et al., 2020) exploraram métodos para aumentar a profundidade das redes sem o uso de conexões residuais, superando assim limitações anteriormente existentes. Antes destes avanços, era desafiador treinar redes neurais com mais de onze camadas, uma vez que a complexidade e a natureza do problema muitas vezes impunham barreiras significativas. Essa dificuldade é exemplificada no trabalho de Simonyan e Zisserman (2014) no modelo VGG-11, com suas 11 camadas treinadas com sucesso, contrastando com as falhas na otimização dos modelos mais profundos VGG-13, VGG-16 e VGG-19. Diante desses desafios, arquiteturas mais recentes, com mais de quinze camadas, passaram a incorporar conexões residuais, conectando saídas de camadas anteriores às posteriores para mitigar problemas de treinamento. Essa abordagem promoveu a criação de modelos profundos variados, como ResNet (He et al., 2016),

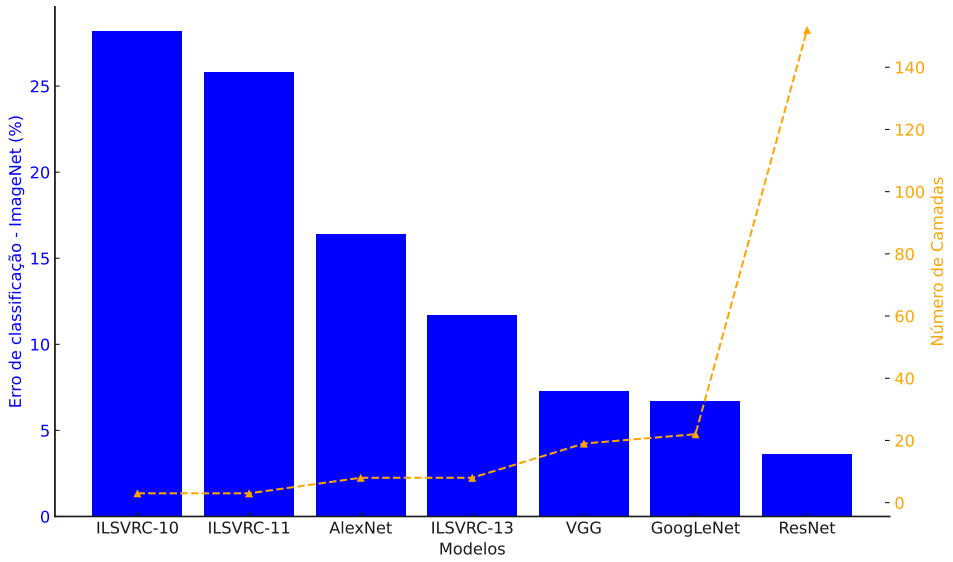


Figura 4.1: Evolução da quantidade de camadas em modelos neurais.

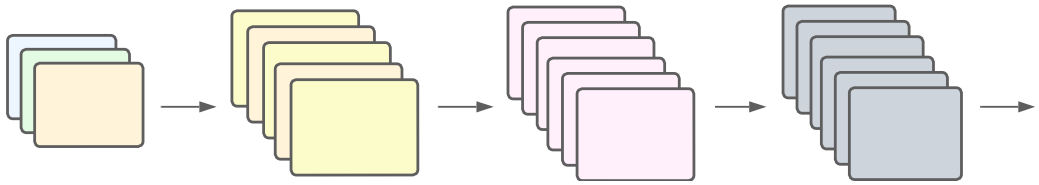
DenseNet (Huang et al., 2017), U-NET++ (Zhou et al., 2018) e *Transformers* (Vaswani et al., 2017b). Embora existam diversos trabalhos empíricos (Liu et al., 2020; Wu et al., 2020a; Oyedotun e Aouada, 2020) que destacam o papel positivo das conexões residuais na facilitação do treinamento e na melhoria da generalização, a compreensão completa desses mecanismos ainda é um tema em aberto. Por isso, é importante analisar as características distintas que conferem superioridade a um modelo em relação a outro.

Este capítulo concentra-se em uma rápida introdução aos tipos de conexões residuais e em seguida uma análise teórica e experimental da ResNet (He et al., 2016) em comparação com redes rasas.

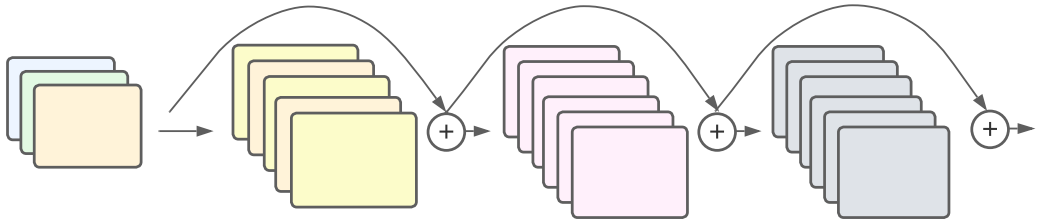
Para ilustrar mais claramente as diferenças entre os tipos de conexões em redes neurais, as seguintes figuras são apresentadas: a Figura 4.2a exibe o modelo de conexões simples, onde cada camada se conecta apenas com a subsequente de forma direta. A Figura 4.2b demonstra conexões residuais, caracterizadas por permitir que a informação de camadas anteriores seja reintroduzida em camadas mais profundas, efetivamente pulando uma ou mais camadas intermediárias. Por fim, a Figura 4.2c destaca as conexões densas, em que cada camada recebe informações de todas as camadas anteriores, formando uma rede altamente integrada e complexa. Estas visualizações auxiliam na compreensão das nuances arquitetônicas e funcionais de cada tipo de conexão.

4.1 Conexões Residuais

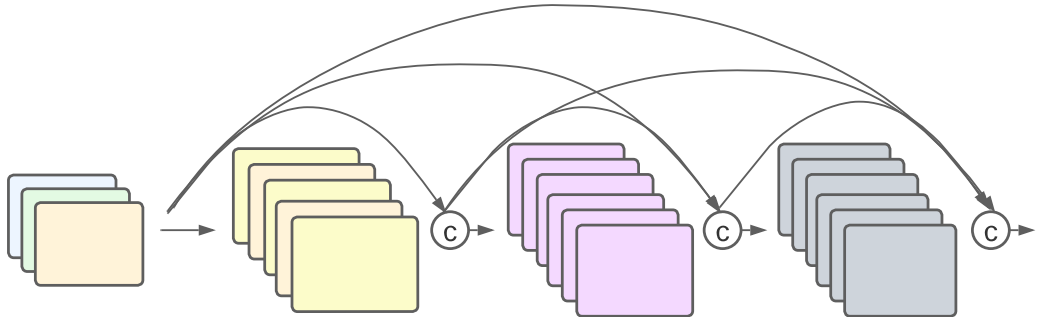
As conexões residuais, também conhecidas como conexões de salto, *skip connections*, entre outros nomes contribuíram para o avanço das arquitetu-



(a) Conexão simples



(b) Conexões residuais



(c) Conexões densas

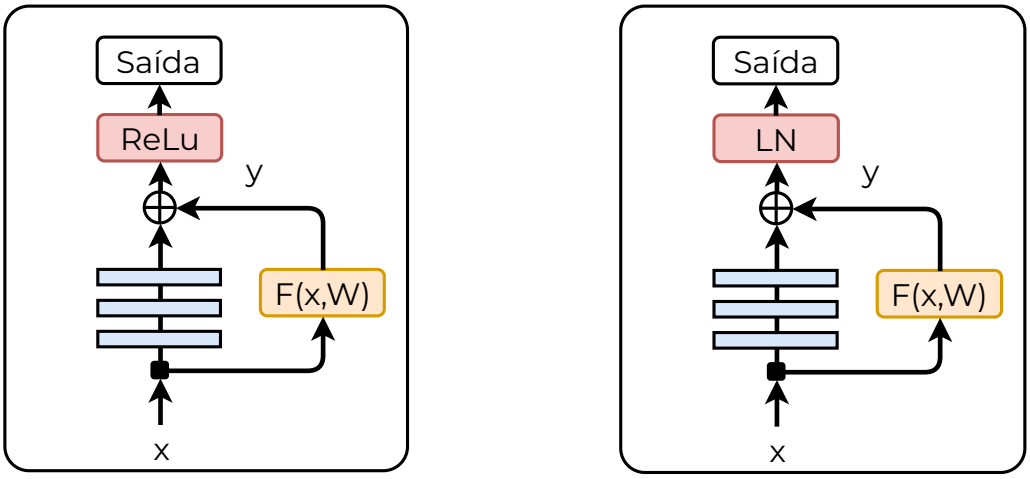
Figura 4.2: Ilustração das diferentes arquiteturas de conexões em redes neurais: (a) apresenta uma arquitetura de conexão simples, (b) exibe conexões residuais, e (c) mostra conexões densas.

ras de aprendizado profundo, exemplificadas pelas Redes Neurais Residuais (ResNets). Apresentadas inicialmente no trabalho de He et al. (2016), essas conexões foram a solução para o problema de diminuição de desempenho em modelos de redes neurais mais profundos. Em modelos com muitas camadas frequentemente não se obtinham melhorias significativas, uma vez que mais camadas podiam levar a uma redução na eficácia do modelo. O estudo de He et al. (2016) incluiu a análise de redes com 20 e 56 camadas, treinadas com o conjunto de dados CIFAR-10. Observou-se que tanto o erro de treinamento quanto o de teste eram impactados pela degradação no desempenho.

O uso dessas conexões envolve a atualização da retropropagação através da função identidade, incorporando uma matriz que contém elementos da matriz identidade. Este método assegura que o gradiente seja multiplicado por um, mantendo assim sua integridade. A visualização na Figura 4.3 demonstra como essa técnica facilita a transmissão da saída de uma camada anterior diretamente para uma ou mais camadas subsequentes, otimizando o fluxo de informações e o processo de aprendizado na rede. Ainda nesta imagem, x representa a entrada do bloco junto com a conexão residual, F realiza a transformação não linear induzida pela rede neural parametrizada por W.A

importância dessas conexões em uma rede neural é que à medida que mais camadas são adicionadas em um modelo, as informações tendem a se perder entre as camadas. Esse problema afeta o gradiente usado para ajustar os pesos da rede, pois seu valor diminui conforme a informação é propagada pela rede, uma disfunção chamada degradação do gradiente (Grosse, 2017; Hanin, 2018). A consequência para o ajuste do gradiente é que as camadas iniciais na etapa de retropropagação podem não ser atualizadas com informações relevantes para fazer os ajustes significativos nos pesos, gerando uma perda de desempenho na rede neural.

A importância dessas conexões em uma rede neural é que à medida que mais camadas são adicionadas em um modelo, as informações tendem a se perder entre as camadas. Esse problema afeta o gradiente usado para ajustar os pesos da rede, pois seu valor diminui conforme a informação é propagada pela rede, uma disfunção chamada degradação do gradiente (Grosse, 2017; Hanin, 2018). A consequência para o ajuste do gradiente é que as camadas iniciais na etapa de retropropagação podem não ser atualizadas com informações relevantes para fazer os ajustes significativos nos pesos, gerando uma perda de desempenho na rede neural.



(a) ResNet. (b) Transformer.

Figura 4.3: Representação da conexão residual: (a) Unidade Residual Convencional (He et al., 2016), (b) Unidade Residual em *Transformers* (LN = Normalização de Camadas) (Vaswani et al., 2017a). Adaptado de Liu et al. (2020) .

Outro fator relevante é que a preservação da informação é um aspecto que justifica o uso do conexões residuais, pois possibilita que a informação seja enviada diretamente para as camadas intermediárias ou finais, permitindo ajustes significativos na atualização dos pesos e na rede neural como um todo (Liu et al., 2020). Seu uso é comum em arquiteturas de redes neurais profundas, como as utilizadas em tarefas de processamento de imagem, texto e áudio. O uso de conexões residuais também pode auxiliar redes neurais com

conexões de atenção, já utilizadas na arquitetura *Transformer* (Vaswani et al., 2017a). Pois permitem que a rede concentre sua atenção em diferentes partes da entrada durante a fase de processamento, portanto, usar conexões residuais garante que informações importantes sejam retidas e possam ser usadas por camadas posteriores da rede (Schneider e Vlachos, 2023). A Figura 4.3 ilustra duas alternativas para lidar com o problema de otimização nos modelos *ResNet* e *Transformer*, explorando conexões residuais juntamente com outros métodos de normalização.

4.2 Conexões Densas

As conexões densas introduzidas por Huang et al. (2017) marcam um progresso notável na arquitetura das redes neurais profundas. As DenseNets se distinguem por uma propriedade singular: cada camada recebe como entrada a saída de todas as camadas anteriores, formando uma estrutura altamente interligada. Essa abordagem de conexões densas não só permite à rede manter informações ao longo de seu processo, enfrentando o desafio do desaparecimento do gradiente - um problema comum em redes profundas -, mas também facilita a reutilização de características, proporcionando a cada camada acesso direto a todas as características previamente aprendidas.

A adoção de conexões densas é motivada, sobretudo, pela sua capacidade de aprimorar a eficiência no aprendizado, efetivando a reutilização de características e minimizando a necessidade de aprender parâmetros redundantes. Este atributo torna as DenseNets ferramentas especialmente valiosas em tarefas de visão computacional, como classificação de imagens, detecção de objetos e segmentação. Por exemplo, em desafios como o ImageNet, os modelos baseados em DenseNet demonstraram precisão significativamente superior em comparação com arquiteturas anteriores.

Desde sua introdução, as DenseNets ganharam ampla adoção em diversas aplicações. Na área de análise de imagens médicas, por exemplo, mostraram-se eficientes na identificação de patologias em imagens de raios-X e ressonância magnética. Grandes nomes como *Google* e *DeepMind* utilizaram variantes dessas redes em sistemas de diagnóstico. Além disso, no campo do processamento de linguagem natural, a incorporação de conexões densas tem sido fundamental para melhorar a capacidade dos modelos de compreender contextos complexos e nuances linguísticas.

As DenseNets se destacam pela eficiência na propagação de gradiente, crucial para o treinamento de redes mais profundas, e pela maior eficiência computacional quando comparadas a outras arquiteturas, graças à redução da redundância de parâmetros. Isso acelera o treinamento e melhora a generali-

zação do modelo, diminuindo o risco de *overfitting*.

Como dito anteriormente, a ideia fundamental por trás das camadas densas é conectar cada camada com as saídas de todas as camadas anteriores, formando uma rede altamente interconectada. Ao contrário das camadas convolucionais, onde os neurônios são conectados apenas a um subconjunto de neurônios na camada anterior, as camadas densas abrangem todas as saídas anteriores, promovendo um mecanismo de aprendizado global. Isso permite que a rede modele interações complexas entre os recursos extraídos das camadas convolucionais. Em uma rede de L camadas, isso resulta em

$$\frac{L(L+1)}{2}$$

conexões diretas, onde a camada i^{th} X_i processa os mapas de características de todas as camadas anteriores como entrada:

$$X_i = H_i([X_0, X_1, ..., X_{i-1}])$$

com H_i sendo uma função composta de *Batch-Normalization*, *ReLU* e *convolução*.

Apesar das vantagens notáveis das conexões densas em redes neurais profundas, existem desvantagens que devem ser consideradas. Uma das principais críticas a essa abordagem é o aumento substancial na complexidade computacional e no uso de memória. Devido à característica de cada camada receber como entrada a saída de todas as camadas anteriores, o número de conexões cresce rapidamente à medida que a rede se aprofunda. Isso pode resultar em um aumento exponencial na quantidade de cálculos e na demanda por memória, tornando o treinamento e a inferência particularmente desafiadores em dispositivos com recursos limitados (Huang et al., 2017).

Outro ponto de atenção é a potencial redundância de características. Embora a reutilização de características seja um dos principais atrativos das DenseNets, ela também pode levar à redundância de informações, onde características semelhantes são processadas múltiplas vezes sem contribuir significativamente para a melhoria do desempenho do modelo. Isso não apenas aumenta o custo computacional, mas também pode dificultar a otimização do modelo, uma vez que a rede pode se tornar menos capaz de distinguir entre características úteis e redundantes (Li et al., 2018).

A integração de conexões densas com outras arquiteturas também pode apresentar desafios. A natureza altamente específica das conexões densas pode limitar sua compatibilidade com outras estruturas de rede, dificultando a implementação de modelos híbridos que combinem as forças de diferentes arquiteturas. Isso pode ser um obstáculo para a inovação e a experimentação

em campos de pesquisa onde a combinação de abordagens é fundamental para avanços significativos (Sandler et al., 2018).

Além disso, a gestão eficiente do fluxo de gradiente, apesar de ser uma vantagem em teoria, pode se tornar um desafio em prática. A propagação de gradiente através de um grande número de camadas interconectadas pode, paradoxalmente, levar a problemas de estabilidade numérica e dificultar a convergência do modelo durante o treinamento, especialmente em configurações com alta variabilidade de dados (Zoph et al., 2018).

Em resumo, enquanto as conexões densas oferecem vantagens notáveis em termos de eficiência do aprendizado e capacidade de generalização, suas desvantagens, como o aumento da complexidade computacional, a redundância de informações, a limitada compatibilidade com outras arquiteturas e potenciais desafios na gestão do fluxo de gradiente, necessitam ser cuidadosamente consideradas no design de redes neurais profundas.

4.3 Prevenção de Singularidades

O estudo de singularidades é uma dos fundamentos usado para análise de convergência de redes neurais. As singularidades são provenientes de um conceito fundamental de álgebra linear sobre matrizes singulares. Uma matriz singular é dita singular quando seu determinante é zero. Isso ocorre quando a matriz contém vetores linha ou coluna linearmente dependentes ou mesmo quando linhas ou colunas inteiras são iguais a zero. Um exemplo deste conceito é sobre a matriz de pesos de uma determinada camada. Quando esta matriz é singular, resultados teóricos e práticos indicam que isso causa regiões de plateau na otimização do gradiente descendente (Wei et al., 2008).

As singularidades encontradas nos artigos de redes neurais não se restringem a matriz de peso mas também sobre outras matrizes que podem ser produzidas pela rede neural como singularidades na matriz Hessiana obtida pela função de perda (Orhan e Pitkow, 2017).

A adoção de conexões residuais trouxe avanços na eficiência do treinamento e na acurácia dos modelos de aprendizado profundo. Treinar redes mais profundas sem enfrentar o desaparecimento do gradiente resultou em aprendizado mais efetivo e representações de dados mais ricas. Adicionalmente, simplificou o processo de treinamento, reduzindo a necessidade de inicializações meticulosas e técnicas de regularização complexas. Embora facilitem o treinamento de redes profundas, não garantem automaticamente melhorias na performance para todas as tarefas, podendo em alguns casos levar a uma estagnação em termos de aumento de precisão.

Orhan e Pitkow (2017) estudaram conexões residuais apresentando argu-

mentos de que elas podem melhorar o treinamento de redes profundas, em parte, eliminando as singularidades inerentes a este tipo de rede. De acordo com Orhan e Pitkow (2017) a dificuldade de treinar redes profundas é parcialmente devido às singularidades causadas pela não identificabilidade do modelo. As singularidades identificadas em trabalhos anteriores: (i) singularidades de sobreposição causadas pela simetria de permutação de nós em uma determinada camada, (ii) singularidades de eliminação correspondentes à eliminação, ou seja, desativação consistente, de nós, (iii) singularidades geradas pela dependência linear dos nós. Elas são ilustradas na 4.4.

Saad e Solla (1995); Amari et al. (2006); Wei et al. (2008) mostram que singularidades retardam significativamente o aprendizado em redes rasas. As singularidades de eliminação surgem quando uma unidade oculta é efetivamente morta, ou seja, quando a soma dos seus pesos (J) de entrada (ou saída) tornam-se zero, Figura 4.4 a. Isto torna as conexões de saída das unidades não identificáveis. As singularidades sobrepostas são causadas pela simetria de permutação das unidades ocultas em uma determinada camada e surgem quando duas unidades se tornam idênticas, ou seja, quando seus pesos (w) de entrada se tornam idênticos Figura 4.4 b. Singularidades de dependência linear surgem quando um subconjunto de unidades ocultas em uma camada se torna linearmente dependente Figura 4.4 c. Novamente, as conexões de saída destas unidades não possuem mais valores e apenas uma combinação linear delas é identificável.

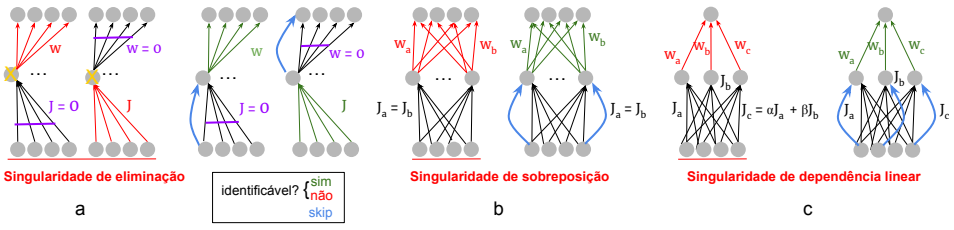


Figura 4.4: Singularidades em uma camada totalmente conectada e como pular conexões evita isso. Adaptado de Orhan (2017).

Embora observações empíricas tenham sido apresentadas para motivar o uso de conexões residuais, uma compreensão clara de como essas conexões melhoram o treinamento ainda não estava definida até o surgimento recente de estudos focados exclusivamente na análise de seu uso. A próxima seção apresenta um estudo simplificado de como as conexões residuais podem ajudar no treinamento das redes neurais.

4.4 Análise Matemática das Conexões Residuais

4.4.1 Sem conexões residuais

Considere uma rede neural simplificada de duas camadas l_1 e l_2 com pesos w_1 e w_2 respectivamente. As saídas de l_1 e l_2 são representadas por s_1 e $\hat{y}$. A função de ativação ϕ é uma ReLU. A entrada desta rede é dada por um escalar x e a saída de $\hat{y}$. A saída desejada (classe) de y. Desse modo a função de perda (loss) é definida como:

$$\text{loss} = (\hat{y} - y)^2 \tag{4.1}$$

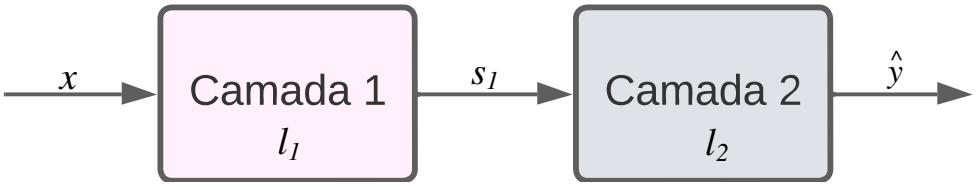


Figura 4.5: Ilustração de rede sem conexões residuais.

- Primeira Camada 1 (l_1):
  - Entrada: x
  - Peso: w_1
  - Saída: $s_1 = \phi(w_1 \cdot x)$
- Segunda Camada 2 (l_2):
  - Entrada: s_1
  - Peso: w_2
  - Saída: $\hat{y} = \phi(w_2 \cdot s_1)$

Para efeitos de demonstração será feita a análise da atualização do gradiente para w_1. A atualização de w_1 requer o cálculo da derivada $\frac{\partial \text{loss}}{\partial w_1}$

$$\frac{\partial \text{loss}}{\partial \hat{y}} = 2(\hat{y} - y) \tag{4.2}$$

$$\frac{\partial \hat{y}}{\partial s_1} = \mathbf{1}_{[w_2 \cdot s_1 \geq 0]} \cdot w_2 \tag{4.3}$$

$$\frac{\partial s_1}{\partial w_1} = \mathbf{1}_{[w_1 \cdot x \geq 0]} \cdot x \tag{4.4}$$

$$\frac{\partial \text{loss}}{\partial w_1} = \frac{\partial \text{loss}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial s_1} \cdot \frac{\partial s_1}{\partial w_1} \tag{4.5}$$

$$\frac{\partial \text{loss}}{\partial w_1} = 2(\hat{y} - y) \cdot (\mathbf{1}_{[w_2 \cdot s_1 \geq 0]} \cdot w_2) \cdot (\mathbf{1}_{[w_1 \cdot x \geq 0]} \cdot x) \tag{4.6}$$

ReLU não é uma função diferenciável no zero. No entanto, seguindo o estudo (Bertoin et al., 2021), nesta seção assume-se que a derivada da ReLU no zero é um. Isso permite a utilização da função indicadora $\mathbf{1}_{[prop]}$ (*indicator function*) para representar a derivada da ReLU. A função indicadora retorna 1 (um) quando *prop* é verdadeira e 0 (zero) caso contrário.

Quando $w_1 \cdot x$ ou $w_2 \cdot s_1$ da Equação 4.6 são negativos, a função indicadora torna-se zero. Por ser uma multiplicação de termos, quando um termo é zero a resultante da multiplicação é zero. Em termos práticos, quando a saída de uma camada é expressa por valores negativos, a função indicadora retorna zero e zera a atualização do parâmetro, impedindo o treinamento. Esse resultado corrobora com alguns cenários de singularidades expostos anteriormente na qual a função indicadora retorna zero e ocorre um bloqueio na atualização do peso.

A análise realizada nesta seção pode ser generalizada para redes com mais camadas. Na equação $\frac{\partial \text{loss}}{\partial w_k}$, o número de funções indicadoras é proporcional ao número de camadas. Assim, maior o número de funções indicadoras multiplicando os termos, maior a chance de ter um termo zerado no cálculo de $\frac{\partial \text{loss}}{\partial w_k}$. Isso também é uma possível justificativa para a dificuldade de treinar redes neurais mais profundas. As conexões residuais são uma alternativa para diminuir as chances de ter a equação $\frac{\partial \text{loss}}{\partial w_k}$ zerada por multiplicações de funções indicadoras iguais a zero.

4.4.2 Com duas conexões residuais

Um outro estudo realizado foi em relação ao uso de conexões residuais. Considere uma rede neural simplificada de duas camadas l_1 e l_2 com pesos w_1 e w_2 respectivamente. As saídas de l_1 e l_2 são representadas por s_1 e $\hat{y}$. A função de ativação ϕ é uma ReLU. A entrada desta rede é dada por um escalar x e a saída de $\hat{y}$. A conexão residual da segunda camada é s_{k1}. Por fim, a saída desejada (classe) de y. A Figura 4.6 ilustra esse estudo.

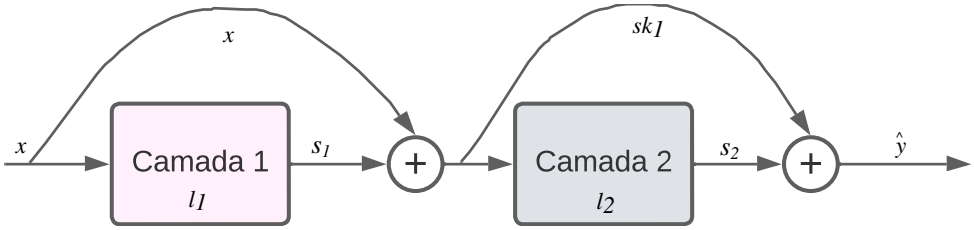


Figura 4.6: Ilustração de rede sem conexões residuais.

Desse modo a função de perda (loss) é definida como:

- **Primeira Camada** (l_1):
  - Entrada: x
  - Peso: w_1
  - Saída: $s_1 = \phi(w_1 \cdot x)$
- **Conexão residual após** l_1:
  - Saída modificada de l_1: $sk_1 = s_1 + x$
- **Segunda camada** (l_2):
  - Entrada: sk_1
  - Peso: w_2
  - Saída: $s_2 = \phi(w_2 \cdot sk_1)$
- **Conexão residual após** l_2:
  - Saída final: $\hat{y} = s_2 + sk_1 = \phi(w_2 \cdot sk_1) + sk_1$
- **Função de *loss***:
  - $\text{loss} = (\hat{y} - y)^2$

$$\frac{\partial s_1}{\partial w_1} = \mathbf{1}_{[w_1 \cdot x \geq 0]} \cdot x \tag{4.7}$$

$$\frac{\partial s_{k1}}{\partial s_1} = 1 \tag{4.8}$$

$$\frac{\partial \hat{y}}{\partial s_{k1}} = \mathbf{1}_{[w_2 \cdot s_{k1} \geq 0]} + 1 \tag{4.9}$$

$$\frac{\partial \text{loss}}{\partial \hat{y}} = 2(\hat{y} - y) \tag{4.10}$$

Aplicação da Regra da Cadeia:

$$\frac{\partial \text{loss}}{\partial w_1} = \frac{\partial s_1}{\partial w_1} \cdot \frac{\partial s_{k1}}{\partial s_1} \cdot \frac{\partial \hat{y}}{\partial s_{k1}} \cdot \frac{\partial \text{loss}}{\partial \hat{y}} \tag{4.11}$$

Substituindo os termos:

$$\frac{\partial \text{loss}}{\partial w_1} = \mathbf{1}_{[w_1 \cdot x \geq 0]} \cdot x \cdot 1 \cdot (\mathbf{1}_{[w_2 \cdot sk_1] \geq 0]} + 1) \cdot 2(\hat{y} - y) \tag{4.12}$$

Considerando as condições apresentadas nesse estudo, uma rede sem conexão residual tem problemas quando a saída de um neurônio é negativa, resultando em um gradiente zero para esse neurônio. Com a conexão residual,

representada pelo "+1" proveniente da $\frac{\partial \hat{y}}{\partial s_{k1}} = (\mathbf{1}_{[w_2 \cdot s_{k1} \geq 0]} + 1)$, a saída da camada anterior é adicionada à saída da camada atual. Isso significa que, mesmo se a ativação $w_2 \cdot sk_1$ for negativa o termo $(\mathbf{1}_{[w_2 \cdot sk_1 \geq 0]} \cdot x + 1)$ é igual a um (1). Isso impede que o gradiente fique zero. Assim, os neurônios têm uma chance maior de receber gradientes não nulos e continuar o processo de aprendizado.

4.5 Considerações Finais

Este capítulo analisou a transição dos modelos de redes neurais, de estruturas mais rasas para aquelas mais profundas, destacando o papel das conexões residuais na eficiência do treinamento e na capacidade de generalização desses modelos.

Do ponto de vista teórico e experimental, observou-se que as conexões residuais ajudam a atenuar o problema do desaparecimento do gradiente, um desafio presente em arquiteturas mais antigas. A adoção dessas conexões em arquiteturas mais recentes permitiu avanços em diversas tarefas de aprendizado profundo, especialmente em áreas como visão computacional e processamento de linguagem natural.

Assim, surge a necessidade contínua de inovações na eficiência computacional e otimização de memória para aplicar essas arquiteturas avançadas de forma eficaz em situações reais. Este capítulo também levanta a importância de compreender as singularidades emergentes nessas redes. As conexões residuais, ao permitirem a propagação direta do gradiente através das camadas, ajudam a evitar singularidades como a sobreposição e eliminação de nós, assim como a dependência linear entre eles. Tais singularidades podem retardar significativamente o aprendizado em redes profundas, tornando o treinamento ineficaz. A compreensão das singularidades e sua mitigação por meio das conexões residuais é, portanto, um aspecto importante para o treinamento dessas redes.

Além disso, a complexidade das funções de perda, particularmente nas DenseNets com seus múltiplos mínimos locais, sugere a necessidade de estratégias de treinamento mais avançadas e métodos de otimização adaptados.

Concluindo, as conexões residuais se mostraram um avanço no campo do aprendizado profundo. No próximo capítulo, serão detalhados os resultados da adição dessas conexões em um modelo pré-existente.

CAPÍTULO

5

DESCINet

Este capítulo detalha o funcionamento do modelo DESCINet, uma evolução do *framework* SCINet com a inclusão de conexões residuais densas. Em seguida, aborda-se a seleção dos conjuntos de dados para treinamento e teste, fundamentada na necessidade de avaliar o DESCINet em variados contextos de séries temporais. Por fim, apresenta-se uma comparação com algoritmos relevantes, estabelecendo uma base sólida para avaliar a capacidade de previsão do desenvolvido.

5.1 Modelo

O modelo *DESCINet*, ilustrado na Figura 5.3, é uma evolução do *SCINet* que mantém sua estrutura de árvore binária e aplica a técnica de *downsampling* na sequência de entrada para dividir em subsequências de elementos pares e ímpares, visando a extração de informações da série temporal. Conforme discutido no capítulo de Revisão da Literatura, neste contexto destacaremos as inovações implementadas no bloco estrutural *SCI-Block* e no modelo como um todo. O bloco aprimorado é denominado *DesBlock* e o modelo resultante, *DESCINet*. Este novo bloco é desenhado para processar entradas de sequências temporais com características diversificadas. Na prática, cada camada de entrada de tamanho n é dividida em dois subconjuntos de tamanho $\frac{n}{2}$. A separação da entrada em duas subsequências é realizada utilizando índices, com índices pares seguindo pelo lado direito da árvore e ímpares pelo esquerdo. Cada subconjunto é processado por módulos no *DesBlock*, que intercalam as subsequências para extrair informações abrangentes. Devido à introdução de conexões densas no novo modelo, foram adicionadas ao *SCI-Block* original

duas normalizações de *batch* para prevenir o *overfitting* da rede neural e ajustes de dimensão são realizados para compatibilizar o tamanho das subsequências em diferentes níveis da rede. Na Figura 5.1, os componentes distintivos do *DesBlock* em relação ao *SCI-Block* estão realçados em vermelho.

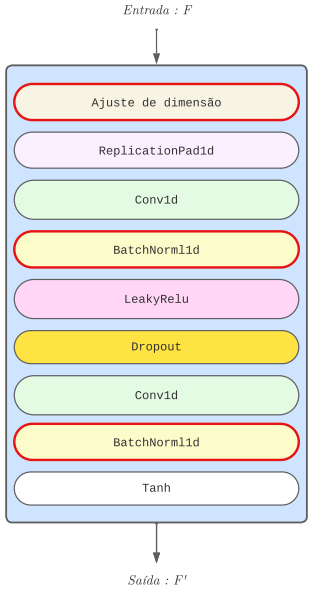


Figura 5.1: Arquitetura geral do DESBlock.
Fonte: Autoria própria.

Em resumo, o `DesBlock` combina técnicas de *padding*, convolução, função de ativação e regularização para processar e extrair características relevantes de séries temporais, servindo como um componente fundamental na rede neural para análise de dados temporais.

Para um melhor entendimento de como funciona o DESCINET e apontar melhor as contribuições do DESCINET, um exemplo de funcionamento do modelo será descrito a seguir e ilustrado uma parte desse passo a passo na Figura 5.2.

Inicialmente, a sequência de entrada do modelo para treinamento, denotada por F, uma subsequência do conjunto de treinamento, alimenta o primeiro nó da árvore (raiz), destacado em laranja, que separa em dois subconjuntos: F_1 (amarelo) e F_2 (azul), com base na paridade dos índices. F_1 encapsula os elementos de índices pares, enquanto F_2 contém aqueles de índices ímpares da série original. Esta divisão é importante para o processamento subsequente, que visa explorar e amplificar as nuances distintas dos subconjuntos separados.

Após a separação, cada subsequência alimenta o `DesBlock` direito e esquerdo, passando pelo processamento mencionado anteriormente e ilustrado na Figura 5.1. Nesta etapa, a subsequência F_1 também é enviada para ser processada por um `DesBlock` em níveis mais baixos da árvore binária. Após

o processamento de F_1, cria-se uma nova subsequência de mesmo tamanho e dimensão, chamada F_1'. Esta nova subsequência é copiada e passa por um novo bloco de divisão de série, originando a subsequência F_3. Antes de F_3 ser processada por um novo bloco, ela é concatenada com as subsequências anteriores. Neste nível, F_3 possui um tamanho menor em relação a F_1 e F_1'. Para resolver essa discrepância de tamanho, um módulo de ajuste de tamanho foi incorporado ao `DesBlock`. Dependendo da posição da subsequência (lado direito ou esquerdo), ele selecionará índices pares ou ímpares das sequências a serem concatenadas. Esse processo se repete até chegar ao último nível da árvore, onde a saída de cada `DesBlock` alimenta uma camada convolucional que reduz a dimensão do sinal de saída, prosseguindo para uma concatenação e realinhamento dos valores da sequência.

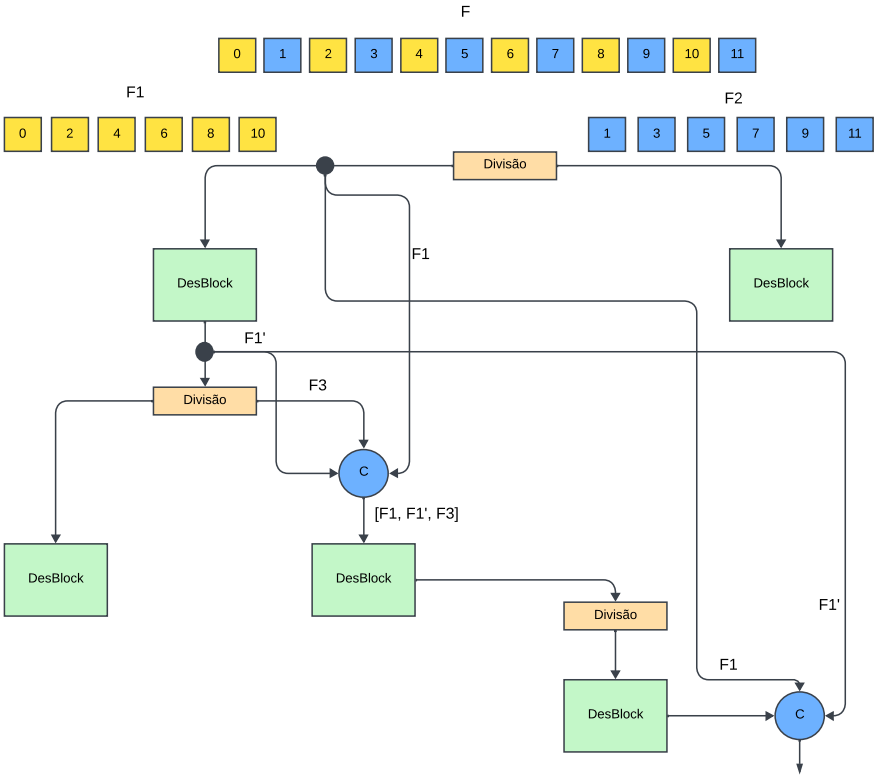


Figura 5.2: Passo a passo do fluxo de informações através do modelo DESCINet.

Fonte: Autoria própria.

Por fim, a saída gerada pela camada linear será somada ao sinal de entrada. Esta conexão residual foi mantida em relação ao algoritmo original, pois continuou contribuindo, de certa forma, para uma melhor performance do modelo nas previsões. Testes foram realizados com e sem essa conexão, e, embora a contribuição tenha sido pequena, ainda se mostrou interessante em alguns cenários. A estrutura completa do modelo pode ser vista na Figura 5.3.

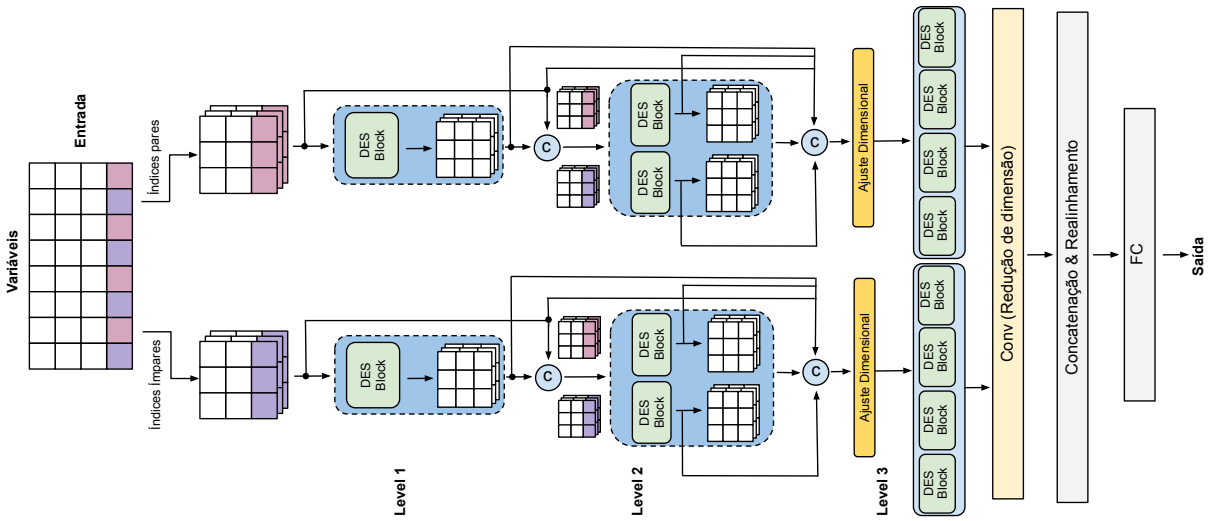


Figura 5.3: Arquitetura geral do DESCINet.

Fonte: Autoria própria.

5.2 Base de dados

Os conjuntos de dados selecionados para este estudo, como *Electricity Transformer Temperature* (ETT), *Electricity Consumption* (ECL), *Traffic*, *Exchange Rate*, *Weather*, e *Influenza-like illness* (ILI), são amplamente reconhecidos e utilizados na área de séries temporais. A escolha desses conjuntos de dados não é arbitrária; ela é baseada na prática comum de outros trabalhos na área de séries temporais que consistentemente os utilizam como *benchmarks* para avaliar e comparar o desempenho de modelos de aprendizado de máquina. Essa consistência na escolha dos conjuntos de dados facilita a comparação direta entre diferentes abordagens e técnicas, permitindo uma avaliação mais objetiva do progresso na área.

Além disso, é importante detalhar o pré-processamento realizado antes do treinamento. Inicialmente, os dados são carregados de arquivos CSV para depois serem normalizados. Diferenças significativas nas escalas das características podem levar a desafios no treinamento de modelos, como a convergência lenta ou a incapacidade de aprender padrões sutis. A utilização de um escalador padrão para ajustar os dados de treinamento e aplicá-lo a todo o conjunto de dados assegura que todas as características contribuam igualmente para o aprendizado do modelo, sem que nenhuma domine devido à sua magnitude.

A natureza sequencial das séries temporais exige uma abordagem cuidadosa na divisão dos dados em conjuntos de treinamento, validação e teste. Esta separação é feita com base no tempo, garantindo que a ordem temporal seja mantida e que o modelo possa ser avaliado em condições realistas, prevendo dados futuros não vistos durante o treinamento. A seleção de características é outro aspecto crítico do pré-processamento. Dependendo do

objetivo da análise, pode-se escolher trabalhar apenas uma série (univariada) ou incluir múltiplas séries que possam contribuir para a precisão das previsões. A flexibilidade na seleção de características permite que o modelo seja adaptado a diferentes cenários e objetivos de previsão.

Além disso, a extração de características temporais a partir das datas é uma prática valiosa para capturar padrões sazonais ou cíclicos, que são comuns em muitos conjuntos de dados de séries temporais. Essas características temporais enriquecem o conjunto de dados com informações contextuais que podem melhorar significativamente a capacidade do modelo de fazer previsões precisas. Finalmente, a preparação dos dados para o modelo envolve a criação de sequências de entrada e saída apropriadas para o treinamento. Esta etapa assegura que o modelo receba os dados no formato correto para aprender a relação entre as entradas passadas e as futuras previsões. A capacidade de reverter os dados para sua escala original também é realizada para interpretar as previsões do modelo, permitindo uma avaliação precisa do seu desempenho em termos práticos.

Essa abordagem metodológica, adotando conjuntos de dados padronizados e aplicando técnicas de normalização, assegura a integridade e a comparabilidade dos resultados obtidos, alinhando este trabalho com as melhores práticas na pesquisa de séries temporais.

Electricity Transformer Temperature (ETT)

O conjunto de dados ETT consiste em registros de séries temporais relacionados à temperatura de dois transformadores de eletricidade, nomeados de 1 e 2. O estudo dessas séries visa monitorar e garantir a eficiência e a segurança das operações dos transformadores em redes elétricas. Uma falha ou superaquecimento pode levar a interrupções de energia ou até mesmo a danos em equipamentos.

Os dados foram coletados durante dois anos, sendo cada ponto de dados registrado a cada minuto (marcado por m), ou a cada hora (marcado por h). Cada um dos mais de 70.000 pontos consiste em 8 características, incluindo a data, o valor preditivo temperatura do óleo e 6 diferentes tipos de características. O significado de cada variável está detalhado na Tabela 5.1. Especificamente, esse conjunto combina padrões periódicos de curto e longo prazos, tendências de longo prazo e muitos padrões irregulares.

Electricity Consumption (ECL)

O ELC armazena consumo de eletricidade em kWh a cada 15 minutos de 2011 a 2014. Devido à presença de colunas com valores zero, os registros de 2011 foram eliminados, já que alguns clientes foram adicionados a base de

Tabela 5.1: Descrição dos campos do conjunto ETT.

Campo	Descrição
Data	Data registrada
HUFL	Uso Alto Útil
HULL	Uso Alto Inútil
MUFL	Uso Médio Útil
MULL	Uso Médio Inútil
LUFL	Uso Baixo Útil
LULL	Uso Baixo Inútil
OT	Temperatura do Óleo

dados depois. O conjunto final inclui 321 clientes de 2012 a 2014, sendo útil para estudos sobre padrões de consumo de energia e análises de demanda energética. O conjunto de dados está disponível em https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

Traffic

Esta coleção abrange dados horários de 48 meses, de 2015 a 2016, do Departamento de Transporte da Califórnia. Os dados incluem taxas de ocupação de rodovias (0 ou 1), medidas por sensores na área da baía de São Francisco. Esses dados são úteis para análises de tráfego, planejamento urbano e desenvolvimento de soluções de mobilidade. Disponível em http://pems.dot.ca.gov.

Exchange Rate

Neste conjunto inclui as taxas de câmbio de oito países: Austrália, Grã-Bretanha, Canadá, Suíça, China, Japão, Nova Zelândia e Singapura, de 1990 a 2016. É utilizado para estudos econômicos, análises financeiras e empresas internacionais, oferecendo uma visão das variações cambiais históricas.

Weather

O conjunto de dados *Weather* documenta 21 características climáticas da cidade de Jena em 2020, incluindo temperatura do ar e umidade. Os dados são coletados em intervalos de 10 minutos, resultando em 144 medições diárias e cerca de 4320 mensais. A alta frequência de dados apresenta um desafio para a capacidade dos modelos em processar sequências longas.

Influenza-like illness (ILI)

O conjunto de dados ILI apresenta relatórios semanais de pacientes com doenças semelhantes à influenza nos Estados Unidos, de 2002 a 2021, coletados pelos Centros de Controle e Prevenção de Doenças. Representa a proporção de pacientes com influenza em relação ao total semanal.

De forma resumida, os conjuntos de dados podem ser vistos na tabela abaixo.

Conjunto de Dados	Nº de Variáveis	Número de Pontos	Frequência	Data Inicial
ETTh1	7	17.420	1 hora	1/7/2016
ETTh2	7	17.420	1 hora	1/7/2016
ETTm1	7	69.680	15 minutos	1/7/2016
Traffic	862	17.544	1 hora	1/1/2015
Electricity	321	26,304	1 hora	1/1/2012
Exchange-Rate	8	7.588	Diária	1/1/1990
Influenza-like illness	8	966	Semanal	1/1/2002
Weather	21	52.560	10 minutos	1/1/2020

Tabela 5.2: Detalhes resumidos das características dos oito conjuntos de dados.

Durante o processo de treinamento, validação e teste dos modelos, os conjuntos de dados foram divididos seguindo uma proporção específica para garantir uma avaliação justa e eficaz da capacidade de previsão de cada modelo. Especificamente, 60% dos dados foram destinados ao treinamento dos modelos, 20% foram utilizados para a validação, e os 20% restantes foram reservados para o teste. Esta divisão foi aplicada uniformemente em todos os conjuntos de dados listados da Tabela 5.2.

A Figura 5.4 ilustra as complexidades associadas à realização de previsões com estes conjuntos de dados. Os conjuntos de dados ETTh1, ETTh2 e ETTm1 apresentam significativa variabilidade, com diferenças na quantidade de pontos de dados devido à granularidade temporal. Para os conjuntos de dados *Traffic* e *Weather*, o número de amostras foi restrito a 850 e 6500 pontos, respectivamente, para facilitar a visualização e interpretação de seus gráficos.

5.2.1 Configurações do Modelo

Os experimentos para a geração dos resultados foram conduzidos em um computador portátil equipado com processador Apple M1, 8 GB de memória RAM e 256 GB SSD, operando sob o sistema macOS Mojave. Para o desenvolvimento e treinamento dos modelos, optou-se pela utilização da biblioteca *PyTorch*, devido à sua flexibilidade e eficiência. Os resultados discutidos neste capítulo, especificamente as métricas MSE e MAE, foram baseados nos estudos de Zeng et al. (2022), Zhou et al. (2022b) e Liu et al. (2021a). Os gráficos apresentados foram gerados a partir de análises locais.

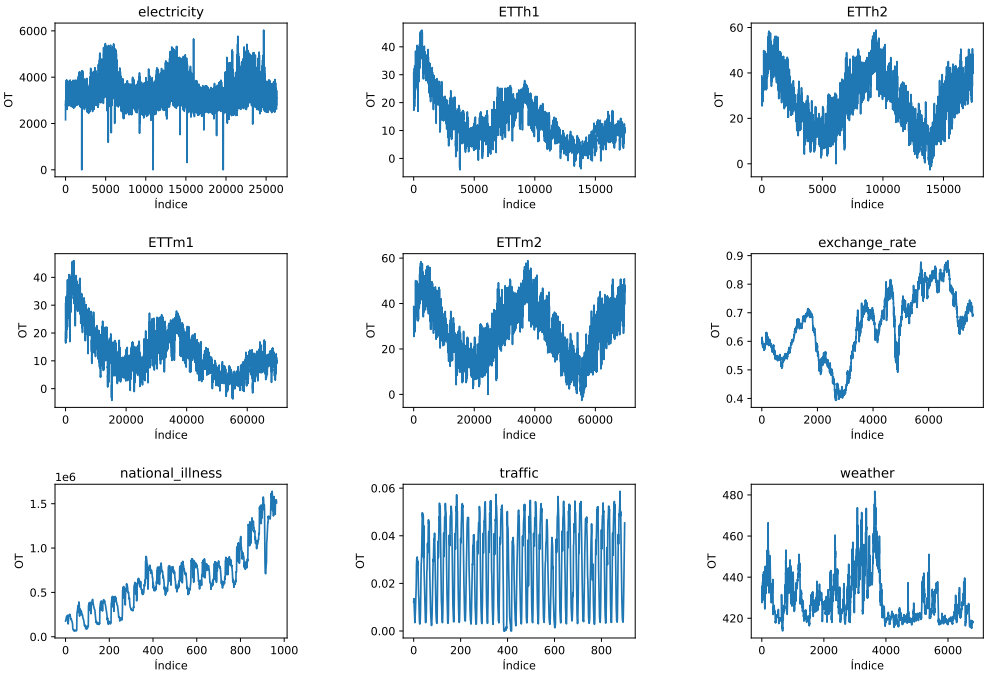


Figura 5.4: Gráfico dos conjuntos de dados utilizados para previsão.

Fonte: Elaborada pelo autor.

Na análise comparativa, selecionaram-se quinze modelos de destaque na literatura, que variam desde técnicas estatísticas tradicionais até métodos avançados de aprendizado profundo para a previsão de séries temporais, tanto univariadas quanto multivariadas. A seleção desses modelos foi guiada pela data de submissão do DESCINet, que atuou como um marco temporal. Adicionalmente, a análise restringiu-se aos modelos frequentemente citados e utilizados como referência nos principais estudos publicados na área de séries temporais, visando uma gestão eficaz da ampla gama de algoritmos disponíveis.

Para os conjuntos de dados analisados, adotaram-se horizontes de previsão variados para uma avaliação mais abrangente. No caso do conjunto de dados ETT, foram selecionados cinco intervalos distintos (24, 48, 168, 336, 720 horas). Para os demais conjuntos de dados, quatro intervalos foram escolhidos (96, 192, 336 e 720 horas).

A Tabela 5.3, detalhando os hiperparâmetros utilizados, é importante para entender os experimentos realizados e para a reprodução desta pesquisa. O otimizador *Adam* foi selecionado para o processo de otimização, enquanto a função de perda *MSELoss* foi empregada, considerando-se sua adequação ao problema em questão. Com o objetivo de facilitar a compreensão do código e a replicação dos resultados no futuro, as colunas desta tabela são descritas a seguir.

Tabela 5.3: Hiperparâmetros do modelo para cada conjunto de dados

Dataset	Horizonte	*WS*	*BS*	*LR*	*Hidden size* (h)	*Dropout*	Épocas	Nível (L)	Tipo
ETTh1	24	48	64	7e-3	8	0,25	11	3	Uni.
	48	48	8	1e-4	4	0,5	11	4	
	168	168	8	5e-5	4	0,5	11	4	
	336	336	128	1e-3	1	0,5	11	4	
	720	720	32	1e-4	4	0,5	11	5	
ETTh2	24	48	8	1e-3	4	0,15	11	3	Uni.
	48	96	16	1e-4	4	0,25	11	4	
	168	336	8	1e-4	2	0,15	11	3	
	336	336	64	5e-4	4	0,25	11	3	
	720	736	32	1e-5	4	0,25	11	3	
ETTm1	24	96	32	1e-3	4	0,5	15	4	Uni.
	48	96	16	5e-4	4	0,5	15	3	
	96	384	32	1e-5	0,5	0,5	15	4	
	288	672	32	1e-5	4	0,5	15	4	
	672	672	32	1e-4	4	0,5	15	5	
ETTh1	24	48	8	3e-3	4	0,5	15	3	Mult.
	48	96	16	9e-3	4	0,25	15	3	
	168	336	32	5e-4	4	0,5	15	3	
	336	336	512	1e-4	1	0,5	15	3	
	720	736	256	5e-5	1	0,5	15	5	
ETTh2	24	48	16	7e-3	8	0,25	15	3	Multi.
	48	96	4	7e-3	4	0,5	15	3	
	168	336	16	5e-5	0,5	0,5	15	4	
	336	336	128	5e-5	1	0,5	15	4	
	720	736	128	1e-5	4	0,5	15	4	
ETTm1	24	48	32	5e-3	4	0,5	15	5	Multi.
	48	96	16	1e-3	4	0,5	15	3	
	96	384	32	5e-5	0,5	0,5	15	3	
	288	672	32	1e-5	4	0,5	15	5	
	672	672	32	1e-5	4	0,5	15	5	
Traffic	96	168	16	5e-4	1	0,25	18	3	Multi.
	192	168	16	5e-4	2	0,5	18	3	
	336	168	16	5e-4	2	0,25	18	3	
	720	168	16	5e-4	1	0,5	18	3	
Weather	96	96	36	2e-4	1	0,5	25	4	Multi.
	192	96	36	3e-4	2	0,5	25	3	
	336	96	36	3e-4	2	0,5	25	3	
	720	96	36	2-e4	2	0,5	30	4	
Electricity	96	96	32	9e-4	8	0	15	3	Multi.
	192	96	32	9e-4	8	0	15	3	
	336	96	32	9e-4	8	0	15	3	
	720	96	32	9e-4	8	0	15	3	
Exchange-rate	96	96	4	5e-4	2	0,5	16	3	Multi.
	192	96	4	5e-4	2	0,5	16	3	
	336	96	4	5e-4	2	0,5	16	3	
	720	96	4	5e-4	2	0,5	16	3	
Illness	96	96	4	3e-4	4	0,25	12	3	Multi.
	192	96	4	3e-4	4	0,25	12	3	
	336	96	4	3e-4	4	0,25	12	3	
	720	96	4	3e-4	4	0,25	12	3	

- *Dataset*: O conjunto de dados usado.
- Horizonte: Quantos passos temporais à frente a previsão é realizada.
- *WS (Window Size)*: Quantos passos temporais anteriores são usados para prever.
- *BS (Batch Size)*: Número de amostras treinadas antes da atualização dos parâmetros.
- *LR (Learning Rate)*: Determina o tamanho dos ajustes nos parâmetros

durante o treinamento.

- *Hidden size (h)*: Número de unidades nas camadas ocultas.
- *Dropout*: Ajuste de *overfitting*.
- Épocas: Número vezes que o modelo é treinado com o conjunto de dados fornecido.
- *Nível (L)*: Profundidade da árvore binária.
- Tipo: Configuração aplicada a uma série univariada ou multivariada.

Após a definição do funcionamento do modelo proposto, apresenta-se a seguir uma lista abrangente de modelos comparados no artigo original em relação ao SCINet, complementada por adições de modelos lançados até o final de 2022, como o FEDformer e o DLinear. Para cada modelo, fornece-se uma descrição breve, visando esclarecer o tipo específico de rede neural empregado e suas principais características distintivas.

- LSTMa (Bahdanau et al., 2014): Um modelo de tradução automática neural baseado em rede neural recorrente projetado para frases longas.
- ARIMA (Box et al., 2015): Um modelo autoregressivo integrado baseado em média móvel para previsão de preços de ações.
- DeepAR (Salinas et al., 2020): Uma rede neural recorrente autorregressiva.
- LSTNet Lai et al. (2017): Modelagem de padrões temporais de longo e curto prazo com redes neurais profundas
- Prophet (Taylor e Letham, 2018): Um modelo de regressão que modela características comuns de séries temporais de maneira consciente de escala.
- LogTrans (Li et al., 2019): Uma variante do *Transformer* usando atenção convolucional e atenção esparsa.
- N-Beats (Oreshkin et al., 2019): Arquitetura neural profunda baseada em links residuais para trás e para frente e uma pilha muito profunda de camadas totalmente conectadas.
- RNN-GRU (Wu et al., 2020b): Modelo de Rede Neural Recorrente usando célula GRU.
- Reformer (Kitaev et al., 2020): Uma variante do *Transformer* usando hashing sensível à localidade e camadas residuais reversíveis.

- Informer (Zhou et al., 2020): Uma variante do *Transformer* usando autoatenção ProbSparse e destilação de autoatenção.
- SCINet (Liu et al., 2021a): Modelo convolucional com subamostragem de amostras para modelagem temporal e previsão.
- Autoformer (Wu et al., 2021): Uma variante do *Transformer* que utiliza uma arquitetura de decomposição com mecanismo de Auto-Correlação para capturar dependências cruzadas no tempo para previsão.
- Pyraformer (Liu et al., 2021b): Uma variante do *Transformer* que aprende representações multi-resolução de séries temporais por meio do módulo de atenção piramidal para capturar dependências cruzadas no tempo para previsão.
- FEDformer (Zhou et al., 2022b): Uma variante do *Transformer* que utiliza a decomposição sazonal-tendência com blocos aprimorados por frequência para capturar dependências cruzadas no tempo para previsão.
- DLinear (Zeng et al., 2022): Modelos lineares para previsão de sequências longas.

Nem todos os modelos foram aplicáveis aos estudos de previsão em séries temporais univariadas e multivariadas. Isso se deve a duas razões principais: alguns modelos foram desenvolvidos exclusivamente para previsão em séries univariadas, não sendo adequados para o contexto multivariado, e outros modelos apresentaram um desempenho insatisfatório, com elevados erros MSE e MAE. Essas limitações restringiram a seleção de modelos adequados para a análise comparativa nos diferentes contextos de séries temporais.

5.3 Estudo em Séries Univariadas

Nesta seção, realizou-se uma análise detalhada dos conjuntos de dados ETT, abrangendo ETTh1, ETTh2 e ETTm1, sob diversos horizontes de previsão: 24, 48, 168, 336 e 720 para ETTh1 e ETTh2; e 24, 48, 96, 288 e 672 para ETTm1. Esta análise minuciosa possibilitou uma comparação aprofundada dos resultados, apresentados na Tabela 5.4.

O modelo desenvolvido demonstrou uma superioridade notável em relação aos principais concorrentes, SCINet e DLinear. Observou-se que o SCINet enfrenta limitações em horizontes de previsão mais extensos, como 672 e 720 passos, possivelmente devido à perda de informações na decomposição das séries temporais. Em contrapartida, o DESCINet exibiu um desempenho

superior em comparação às redes neurais recorrentes, como LSTMa e LSTNet, evidenciando reduções significativas no MAE para diversos horizontes de previsão. A eficácia do DESCINet é atribuída à sua capacidade de capturar características temporais de curto e longo prazos, beneficiando-se tanto da decomposição de séries quanto das conexões de salto. Além disso, superou modelos baseados em *Transformer*, como Reformer e Informer, no MAE, com reduções consideráveis para vários horizontes de previsão, utilizando o conjunto de dados ETTh1 como referência. Notavelmente, apenas no conjunto de dados ETTm1, o DLinear apresentou um desempenho superior em todos os horizontes de previsão.

Para elucidar os resultados exibidos na Tabela 5.4, conduziram-se experimentos de previsão nos horizontes temporais de 96, 192 e 336, empregando os modelos FEDformer, SCINet e DLinear. Estes modelos foram selecionados com base em sua comprovada eficácia na previsão de séries temporais, conforme indicado pelos baixos valores de MSE e MAE. A comparação de desempenho entre estes modelos é visualizada na Figura 5.5, destacando a robustez e eficiência do DESCINet em diversos cenários de previsão.

Os resultados apresentados enfatizam a melhoria proporcionada pelo DESCINet em comparação a outros modelos, com melhorias percentuais significativas em MSE e MAE para vários horizontes de previsão. Essas melhorias reforçam a eficácia do DESCINet na manipulação da complexidade inerente às séries temporais, oferecendo insights valiosos para futuras investigações e aplicações práticas na área de previsão de séries temporais.

Tabela 5.4: Resultados de previsão de série temporal univariada nos conjuntos de dados ETT.

Métodos	Métricas	ETTh1					ETTh2					ETTm1				
		Horizontes de previsão					Horizontes de previsão					Horizontes de previsão				
		24	48	168	336	720	24	48	168	336	720	24	48	96	288	672
ARIMA	MSE	0,108	0,175	0,396	0,468	0,659	3,554	3,190	2,8	2,753	2,878	0,090	0,179	0,272	0,462	0,639
	MAE	0,284	0,424	0,504	0,593	0,766	0,445	0,474	0,595	0,738	1,044	0,206	0,306	0,399	0,558	0,697
Prophet	MSE	0,115	0,168	1,224	1,549	2,735	0,199	0,304	2,145	2,096	3,355	0,120	0,133	0,194	0,452	2,747
	MAE	0,275	0,330	0,763	1,820	3,253	0,381	0,462	1,068	2,543	4,664	0,290	0,305	0,396	0,574	1,174
DeepAR	MSE	0,107	0,162	0,239	0,445	0,658	0,098	0,163	0,255	0,604	0,429	0,091	0,219	0,364	0,948	2,437
	MAE	0,280	0,327	0,422	0,552	0,707	0,263	0,341	0,414	0,607	0,580	0,243	0,362	0,496	0,795	1,352
NBEats	MSE	0,042	0,065	0,106	0,127	0,269	0,078	0,123	0,244	0,270	0,281	0,031	0,056	0,095	0,157	0,207
	MAE	0,156	0,2	0,255	0,284	0,422	0,210	0,271	0,393	0,418	0,432	0,117	0,168	0,234	0,311	0,370
LSTMa	MSE	0,114	0,193	0,236	0,590	0,683	0,155	0,190	0,385	0,558	0,640	0,121	0,305	0,287	0,524	1,064
	MAE	0,272	0,358	0,392	0,698	0,768	0,307	0,348	0,514	0,606	0,681	0,233	0,411	0,420	0,584	0,873
LSTNet	MSE	1,293	1,456	1,997	2,655	2,143	2,742	3,567	3,242	2,544	4,625	1,968	1,999	2,762	1,257	1,917
	MAE	0,901	0,960	1,214	1,369	1,380	1,457	1,687	2,513	2,591	3,709	1,700	1,215	1,542	2,076	2,941
Reformer	MSE	0,991	1,313	1,824	2,117	2,415	1,531	1,871	4,660	4,028	5,381	0,724	1,098	1,433	1,820	2,187
	MAE	0,754	0,906	1,138	1,280	1,520	1,613	1,735	1,846	1,688	2,015	0,607	0,777	0,945	1,094	1,232
LogTrans	MSE	0,103	0,167	0,207	0,230	0,273	0,102	0,169	0,246	0,267	0,303	0,065	0,078	0,199	0,411	0,598
	MAE	0,259	0,328	0,375	0,398	0,463	0,255	0,348	0,422	0,437	0,493	0,202	0,220	0,386	0,572	0,702
Informer	MSE	0,098	0,158	0,183	0,222	0,269	0,093	0,155	0,232	0,263	0,277	0,030	0,069	0,194	0,401	0,512
	MAE	0,247	0,319	0,346	0,387	0,435	0,240	0,314	0,389	0,417	0,431	0,137	0,203	0,372	0,554	0,644
Autoformer	MSE	0,057	0,103	0,090	0,116	0,120	0,110	0,123	0,188	0,225	0,257	0,025	0,039	0,057	0,103	0,110
	MAE	0,188	0,257	0,235	0,254	0,277	0,259	0,271	0,340	0,376	0,402	0,122	0,156	0,184	0,253	0,261
FEDformer	MSE	0,045	0,055	0,105	0,12	0,127	0,156	0,166	0,238	0,271	0,288	0,023	0,024	0,036	0,061	0,105
	MAE	0,159	0,179	0,256	0,269	0,28	0,301	0,336	0,38	0,412	0,438	0,091	0,149	0,206	0,209	0,248
SCINet	MSE	0,031	0,047	0,088	0,111	0,099	0,158	0,180	0,065	0,180	0,411	0,022	0,047	0,072	0,117	0,110
	MAE	0,132	0,168	0,233	0,264	0,364	0,191	0,240	0,312	0,342	0,518	0,096	0,147	0,199	0,266	0,261
DLinear	MSE	**0,028**	0,065	0,121	0,149	0,235	0,070	**0,093**	0,182	0,196	**0,285**	**0,011**	**0,020**	**0,031**	**0,051**	**0,085**
	MAE	**0,123**	0,196	0,271	0,307	0,398	0,198	**0,229**	0,332	0,358	**0,430**	**0,076**	**0,105**	**0,132**	**0,168**	**0,220**
DESCINet	MSE	0,032	**0,046**	**0,080**	**0,109**	**0,174**	**0,063**	0,079	**0,157**	**0,179**	0,363	0,019	0,045	0,067	0,114	0,101
	MAE	0,135	**0,166**	**0,221**	**0,261**	**0,340**	**0,193**	0,240	**0,311**	**0,340**	0,482	0,088	0,145	0,191	0,261	0,240
Melhoria	MSE	-12,50%	**2,13%**	**33,88%**	**1,80%**	**25,96%**	**2,86%**	-6,06%	**0,63%**	**0,56%**	-21,49%	-72,73%	-57,45%	-53,73%	-55,26%	-42,95%
	MAE	-8,89%	**1,19%**	**18,45%**	**1,14%**	**14,57%**	**2,53%**	-4,58%	**0,32%**	**0,58%**	-10,79%	-15,79%	-27,59%	-30,89%	-35,63%	-27,15%

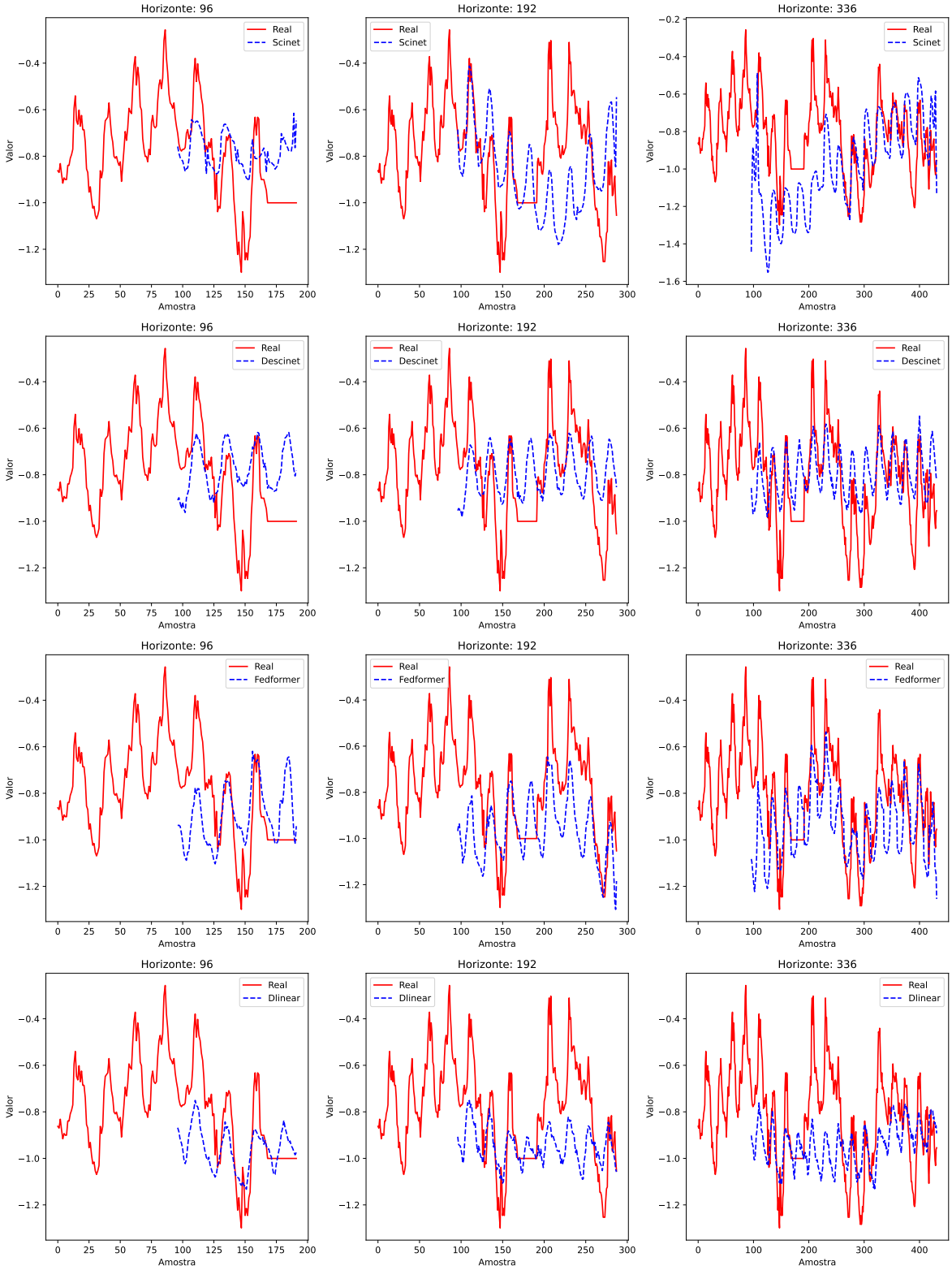


Figura 5.5: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados univariado ETTh1.

Fonte: Elaborada pelo autor.

Na análise dos dados das previsões de séries temporais realizadas pelos algoritmos SCINet, DESCINet, FEDFormer e DLinear, observa-se uma tentativa de todos os modelos em capturar a tendência geral da série real, embora

com variações significativas em termos de precisão e sensibilidade às flutuações inerentes à série. A complexidade da série real, caracterizada por sua variação considerável com picos e vales pronunciados, apresenta um desafio notável para a previsão, exigindo dos modelos uma capacidade robusta de adaptação às mudanças dinâmicas no comportamento dos dados.

Ao considerar a adequação de cada modelo para a tarefa em questão, nota-se que o SCINet parece capturar algumas das tendências gerais da série real. No entanto, este modelo pode ter dificuldades com as flutuações mais acentuadas, sugerido pelos valores negativos consistentes e pela falta de correspondência com os picos mais significativos da série real. Isso pode indicar uma tendência do modelo para suavizar excessivamente as variações, potencialmente perdendo detalhes cruciais que são fundamentais para uma previsão precisa.

Similarmente, o DESCINet demonstra uma tentativa de seguir a série real, mas parece também suavizar as flutuações, o que pode ser indicativo de um modelo que não captura totalmente a volatilidade da série temporal. Essa característica sugere uma limitação na capacidade do modelo de ajustar-se às nuances específicas e às mudanças abruptas que podem ocorrer na série temporal analisada. Por outro lado, o FEDFormer, com seus valores negativos mais extremos, sugere uma tentativa de capturar a volatilidade da série. No entanto, o modelo pode estar exagerando as tendências de queda ou não estar adequadamente ajustado para as nuances da série real, o que pode levar a previsões que, embora tentem refletir a dinâmica da série, falham em capturar sua verdadeira essência. O DLinear, apresentando uma variação mais ampla nos seus valores previstos, indica uma tentativa de capturar a volatilidade da série real. Esse modelo parece oferecer uma abordagem mais dinâmica na tentativa de ajustar-se às flutuações da série, o que pode ser um indicativo de sua capacidade potencialmente superior de lidar com a complexidade e a imprevisibilidade inerentes à série temporal em análise devido ao módulo de decomposição de componentes.

Analisando individualmente cada cenário, na previsão de 96 períodos, o modelo DESCINet e FEDFormer se destacaram, especialmente entre os pontos 100 e 160, ao prever valores próximos aos reais e acompanhar com precisão as tendências de subida e descida. Contudo, não conseguiu antecipar com exatidão o término desses movimentos, errando ao prever antecipadamente os picos e vales da amostra. Por outro lado, o SCINet, que inicialmente parecia ter um desempenho similar para este intervalo, revelou diferenças significativas na previsão dos picos da série original. Enquanto o máximo esperado era em torno de -0,4, o SCINet previu -0,6 e o DESCINet -0,55. Este padrão se repetiu em outros modelos. Para o mesmo período, o DLinear foi menos eficaz,

acompanhando apenas a primeira queda na série, no ponto 130 do eixo horizontal, e depois tendo dificuldades para predizer do padrão real, se comparado com os demais algoritmos. Já para as previsões de 192 e 336, o SCINet apresentou resultados menos precisos, afastando-se dos valores reais. Contudo, tanto o SCINet quanto o FEDformer conseguiram identificar uma queda acentuada no ponto 150 da amostra. Em contraste, o DESCINet realizou previsões mais consistentes, embora não tenha capturado variações abruptas tão eficientemente. Apesar disso, foi mais preciso em pontos específicos para estes intervalos de previsão.

A Figura 5.6 ilustra o resultado das predições na base de dados ETTh2, onde a série original, caracterizada por suas flutuações e tendências variáveis ao longo do tempo, serve como um teste rigoroso para a precisão e a eficácia dos modelos. A análise das previsões revela que cada modelo tem seus pontos fortes e limitações, refletidos na proximidade de suas previsões aos valores reais. O SCINet, por exemplo, mostra uma tendência a suavizar as flutuações mais extremas, o que pode ser benéfico em séries com ruído significativo, mas também pode resultar na perda de detalhes críticos em momentos de mudanças abruptas. Por outro lado, o DESCINet oferece uma abordagem mais equilibrada, tentando capturar tanto as tendências gerais quanto as variações menores, embora ainda possa haver desvios notáveis em relação à série real, especialmente em pontos de inflexão críticos. O FEDFormer, com sua abordagem potencialmente mais adaptativa, tenta abordar a volatilidade da série temporal, o que pode ser visto em suas tentativas de ajustar-se às mudanças mais rápidas na série. No entanto, isso também pode levar a sobreajustes, onde o modelo pode interpretar incorretamente o ruído como uma tendência significativa, resultando em previsões que desviam da realidade. O DLinear, por sua vez, apresenta uma abordagem que parece focar na captura de tendências de longo prazo, possivelmente à custa de perder detalhes de curto prazo. Isso pode ser particularmente visível em séries temporais com muitas variações de curto prazo, onde o modelo pode não responder adequadamente a todas as flutuações.

Analisando partes dos gráficos, pode-se destacar o desempenho do SCINet, que se sobressaiu em relação aos demais modelos, particularmente nos horizontes de previsão de 96, 192 e 336. Esta superioridade é mais evidente no gráfico correspondente ao horizonte de 336, onde se observa que o SCINet, apesar de apresentar algumas discrepâncias em relação aos valores reais, conseguiu aproximar-se mais deles do que os outros modelos. Notadamente, em valores próximos de 0,75 no eixo vertical, o SCINet aproximou-se mais da série real, e nos vales observados, o modelo tendeu a superestimar os valores reais apenas em uma região específica, próxima ao ponto 50 no eixo horizontal.

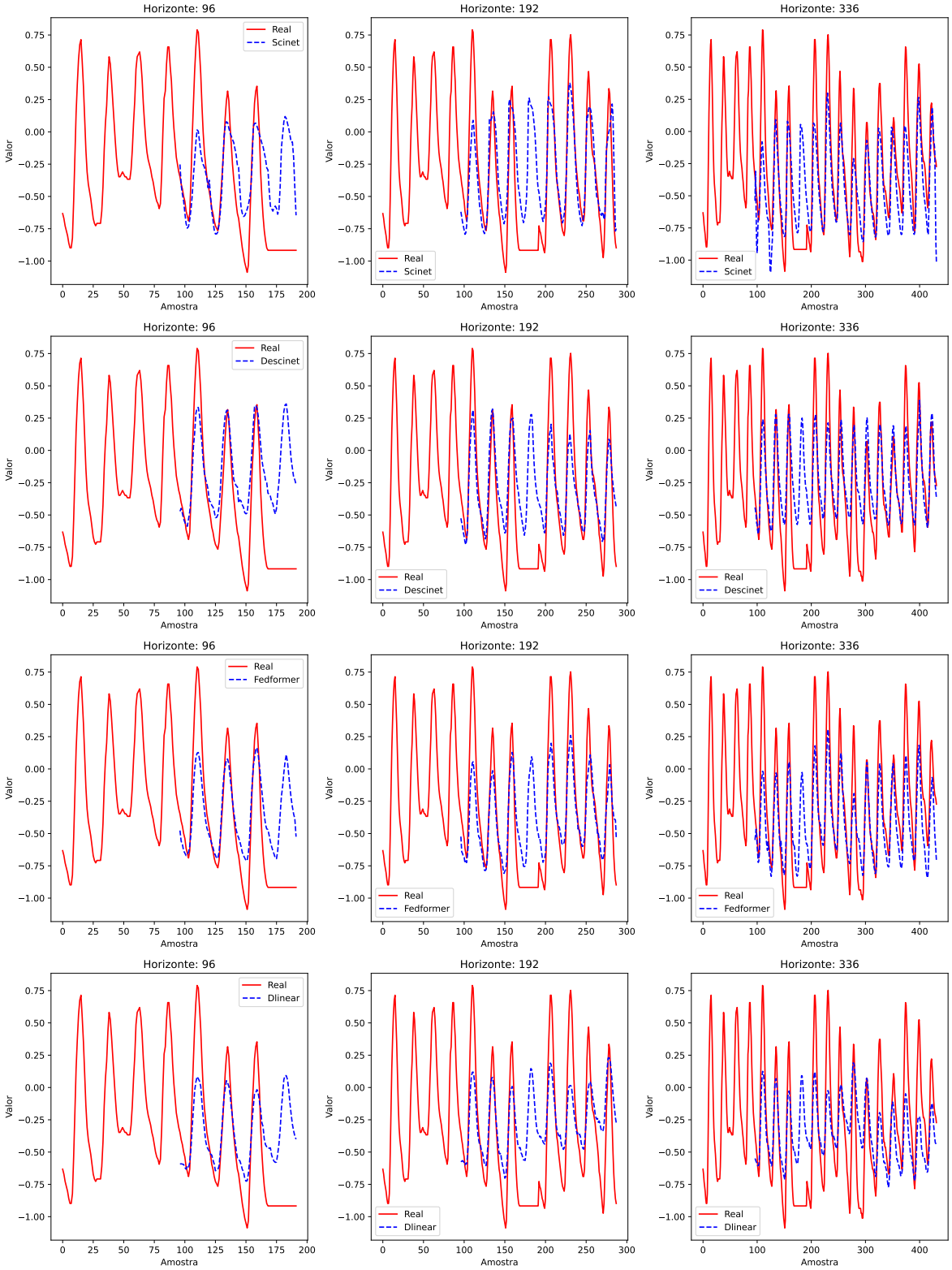


Figura 5.6: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados univariado ETTh2.

Fonte: Elaborada pelo autor.

Em outras áreas, conseguiu acompanhar ou chegar perto da série real. É importante ressaltar que nenhum dos modelos testados foi capaz de capturar integralmente uma variação atípica na série temporal, ocorrida entre os pon-

tos 160 e 210 no eixo horizontal. Essa região representa um desafio maior em termos de previsão, onde tanto o SCINet quanto o FEDformer exibiram menores erros, indicando uma maior adaptabilidade desses modelos às mudanças súbitas na série temporal. Tal observação sugere que, embora os modelos apresentem competência na previsão de séries temporais com variações suaves, há ainda espaço para melhorias na previsão de mudanças abruptas e atípicas na série temporal.

5.4 Estudo em Séries Multivariadas

No estudo de séries temporais multivariadas, os conjuntos de dados utilizados foram: *ETT*, *Electricity*, *Exchange Rate*, *Weather*, *Traffic* e *Influenza-like illness*. Os resultados desses estudos estão detalhados em tabelas que apresentam valores de MSE e MAE, complementados por gráficos que ilustram as predições nos horizontes entre 96 e 336.

Na Tabela 5.5, constatou-se que, em comparação com métodos baseados em RNN, como LSTMa e LSTnet, houve uma redução significativa no MAE de 43,6% (em 168), 62,3% (em 336) e 55,7% (em 720). Em relação aos métodos baseados em *Transformer*, tais como *Reformer* e *Informer*, notórios pela habilidade de capturar padrões latentes de longo prazo em dados históricos, observou-se uma diminuição do MAE nos horizontes de 24 (59,84%), 48 (54,98%), 168 (52,62%) e 336 (40,93%). Métodos que empregam redes convolucionais demonstraram superioridade em relação aos baseados em *Transformer*, pois as camadas convolucionais empilhadas proporcionam um aprendizado mais eficiente de relações temporais locais e globais para séries temporais multivariadas, superando os métodos RNN e *Transformers*.

Tabela 5.5: MSE / MAE para diferentes horizontes de previsão em séries multivariadas no conjunto de dados ETT.

Métodos	Métricas	ETTh1					ETTh2					ETTm1				
		Horizontes de previsão					Horizontes de previsão					Horizontes de previsão				
		24	48	168	336	720	24	48	168	336	720	24	48	96	288	672
LSTMa	MSE	0,114	0,193	0,236	0,590	0,683	0,155	0,190	0,385	0,558	0,640	0,121	0,305	0,287	0,524	1,064
	MAE	0,272	0,358	0,392	0,698	0,768	0,307	0,348	0,514	0,606	0,681	0,233	0,411	0,420	0,584	0,873
Reformer	MSE	0,991	1,313	1,824	2,117	2,415	1,531	1,871	4,660	4,028	5,381	0,724	1,098	1,433	1,820	2,187
	MAE	0,754	0,906	1,138	1,280	1,520	1,613	1,735	1,846	1,688	2,015	0,607	0,777	0,945	1,094	1,232
LogTrans	MSE	0,103	0,167	0,207	0,230	0,273	0,102	0,169	0,246	0,267	0,303	0,065	0,078	0,199	0,411	0,598
	MAE	0,259	0,328	0,375	0,398	0,463	0,255	0,348	0,422	0,437	0,493	0,202	0,220	0,386	0,572	0,702
LSTNet	MSE	1,293	1,456	1,997	2,655	2,143	2,742	3,567	3,242	2,544	4,625	1,968	1,999	2,762	1,257	1,917
	MAE	0,901	0,960	1,214	1,369	1,380	1,457	1,687	2,513	2,591	3,709	1,232	1,215	1,542	2,076	2,941
Informer	MSE	0,498	0,558	0,183	0,622	0,669	1,493	2,355	2,232	1,463	0,377	0,430	0,669	0,694	0,801	0,912
	MAE	0,347	0,389	0,446	0,487	0,498	0,740	0,814	0,789	0,817	0,931	0,237	0,303	0,472	0,494	0,644
Autoformer	MSE	0,439	0,429	0,493	0,509	0,539	0,288	0,338	0,456	0,482	0,515	0,410	0,485	0,502	0,604	0,607
	MAE	0,440	0,442	0,479	0,492	0,537	0,348	0,377	0,452	0,486	0,511	0,428	0,464	0,476	0,522	0,530
Pyraformer	MSE	0,493	0,554	0,781	0,912	0,993	0,610	0,625	0,788	0,907	0,963	0,310	0,465	0,520	0,729	0,980
	MAE	0,507	0,544	0,675	0,747	0,792	0,554	0,577	0,683	0,747	0,783	0,371	0,464	0,504	0,657	0,678
FEDFormer	MSE	0,318	0,342	0,412	0,456	0,521	0,303	0,326	0,429	0,496	0,463	0,290	0,342	0,366	0,398	0,455
	MAE	0,384	0,396	0,449	0,474	0,515	0,357	0,388	0,439	0,487	0,474	0,364	0,396	0,412	0,433	0,464
SCINet	MSE	0,034	0,047	0,091	0,112	0,193	0,069	0,119	**0,156**	0,177	0,394	0,020	0,046	0,071	0,1969	0,178
	MAE	0,137	0,166	0,234	0,265	0,363	0,190	0,260	**0,313**	0,340	0,5069	0,094	0,145	0,196	0,296	0,328
DLinear	MSE	0,315	0,345	0,400	0,469	0,516	0,192	0,243	0,372	0,433	0,874	0,536	0,315	0,303	0,373	0,470
	MAE	0,352	0,374	0,412	0,486	0,522	0,291	0,326	0,411	0,459	0,663	0,453	0,355	0,345	0,400	0,479
DESCINet	MSE	**0,030**	**0,046**	**0,081**	**0,071**	**0,173**	**0,068**	**0,094**	0,166	**0,173**	**0,255**	**0,018**	**0,045**	**0,069**	**0,116**	**0,159**
	MAE	**0,131**	**0,166**	**0,220**	**0,207**	**0,336**	**0,190**	**0,232**	0,322	**0,336**	**0,403**	**0,088**	**0,143**	**0,182**	**0,264**	**0,311**
Melhoria	MSE	**10,41%**	**0,21%**	**11,73%**	**36,51%**	**10,36%**	**1,16%**	**20,67%**	-6,13%	**2,42%**	**35,37%**	**11,00%**	**2,597%**	**2,68%**	**41,03%**	**10,78%**
	MAE	**5,02%**	**0,36%**	**5,94%**	**21,85%**	**7,48%**	**0,05%**	**10,94%**	-2,84%	**1,17%**	**20,41%**	**6,50%**	**1,03%**	**7,19%**	**10,93%**	**4,97%**

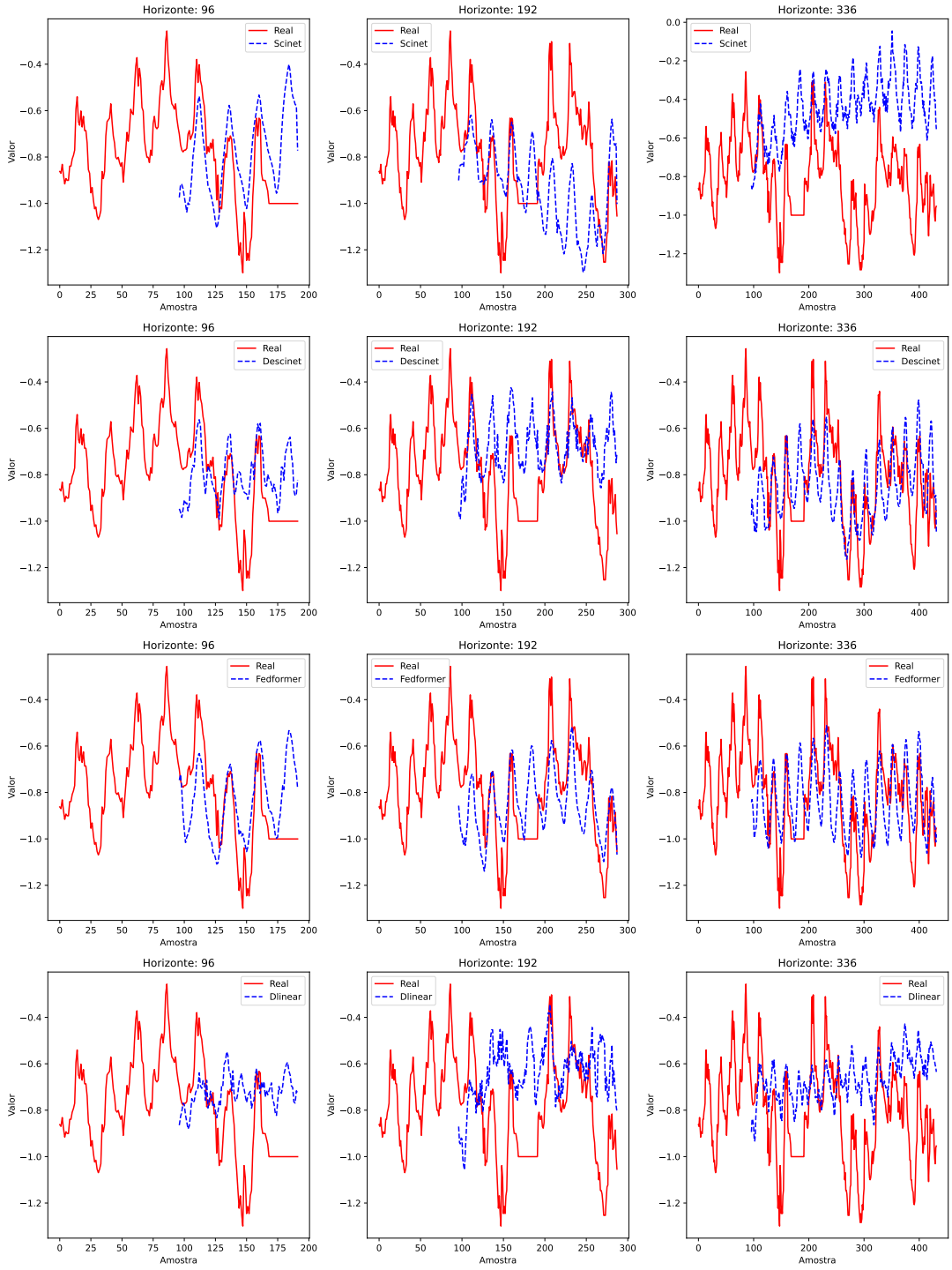


Figura 5.7: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariados ETTh1.

Fonte: Elaborada pelo autor.

Para ilustrar os resultados estabelecidos na Tabela 5.5, foram realizados testes de previsão com o conjunto de dados multivariado ETT. Todos os gráficos apresentados a seguir ilustram a variável *OT* para três horizontes temporais: 96, 192 e 336.

O primeiro conjunto de dados a ser analisado é o ETTh1 (multi) e ele é ilustrado na Figura 5.7. Para o primeiro horizonte, todos os modelos apresentam

variações significativas em relação à série real, com tendências de subestimação ou superestimação em diferentes pontos. O SCINet, por exemplo, mostra uma tendência de acompanhar a direção da série real, mas com claras diferenças de magnitude, indicando uma sensibilidade às tendências gerais, mas com dificuldades em capturar a amplitude exata das variações. O DESCINet, por sua vez, apresenta variações que, embora sigam a tendência da série real, desviam-se em pontos específicos, sugerindo uma resposta talvez mais atenuada às mudanças abruptas na série temporal.

O FEDformer consegue ter uma aproximação que parece capturar melhor as tendências de médio prazo devido a capacidade dos *Transformers*, mas possui algumas superestimações e subestimações, refletindo uma capacidade de adaptação às mudanças da série. O DLinear, por outro lado, mostra-se mais conservador, com previsões que tendem a suavizar as variações reais, indicando uma possível dificuldade em ajustar-se rapidamente às mudanças mais abruptas na série temporal e limita sua eficácia em capturar picos e vales acentuados.

Na análise do conjunto de dados ETTh2, ilustrada pela Figura 5.8, os resultados obtidos pelos algoritmos testados mostraram-se consistentes em todos os horizontes temporais, com pequenas variações entre eles.

No horizonte de 96, especificamente, o FEDformer superou os vales tanto nos horizontes de 96 quanto de 192, enquanto nos picos, seu desempenho foi semelhante ao do DLinear, apesar de não alcançar a mesma proximidade com os valores reais.

Apesar das diferenças sutis observadas no horizonte de 96, alguns aspectos destacam a superioridade do DESCINet, como sua previsão mais precisa e alinhada com os valores reais. Este algoritmo evitou desvios significativos tanto no eixo horizontal (amostra) quanto no vertical (previsão), mantendo-se mais próximo dos valores mais elevados. Por outro lado, tanto o SCINet quanto o DLinear apresentaram picos próximos ao ponto zero do eixo Y, enquanto o SCINet se aproximou mais dos valores da série original, que variam aproximadamente entre -0,8 e 0,6.

Essa análise detalhada revela *nuances* importantes no desempenho dos algoritmos, enfatizando a relevância do DESCINet em fornecer previsões mais acuradas e alinhadas com os dados reais, um fator importante para a confiabilidade e eficácia em aplicações práticas de modelagem de séries temporais.

Na análise do conjunto ETTm1 ilustrado na Figura 5.9, observa-se uma distinção clara no desempenho dos modelos analisados, particularmente evidenciada nas comparações entre os horizontes de previsão de 96, 192 e 336. O DLinear demonstrou limitações significativas, não conseguindo capturar adequadamente os movimentos da série temporal, o que reforça as observa-

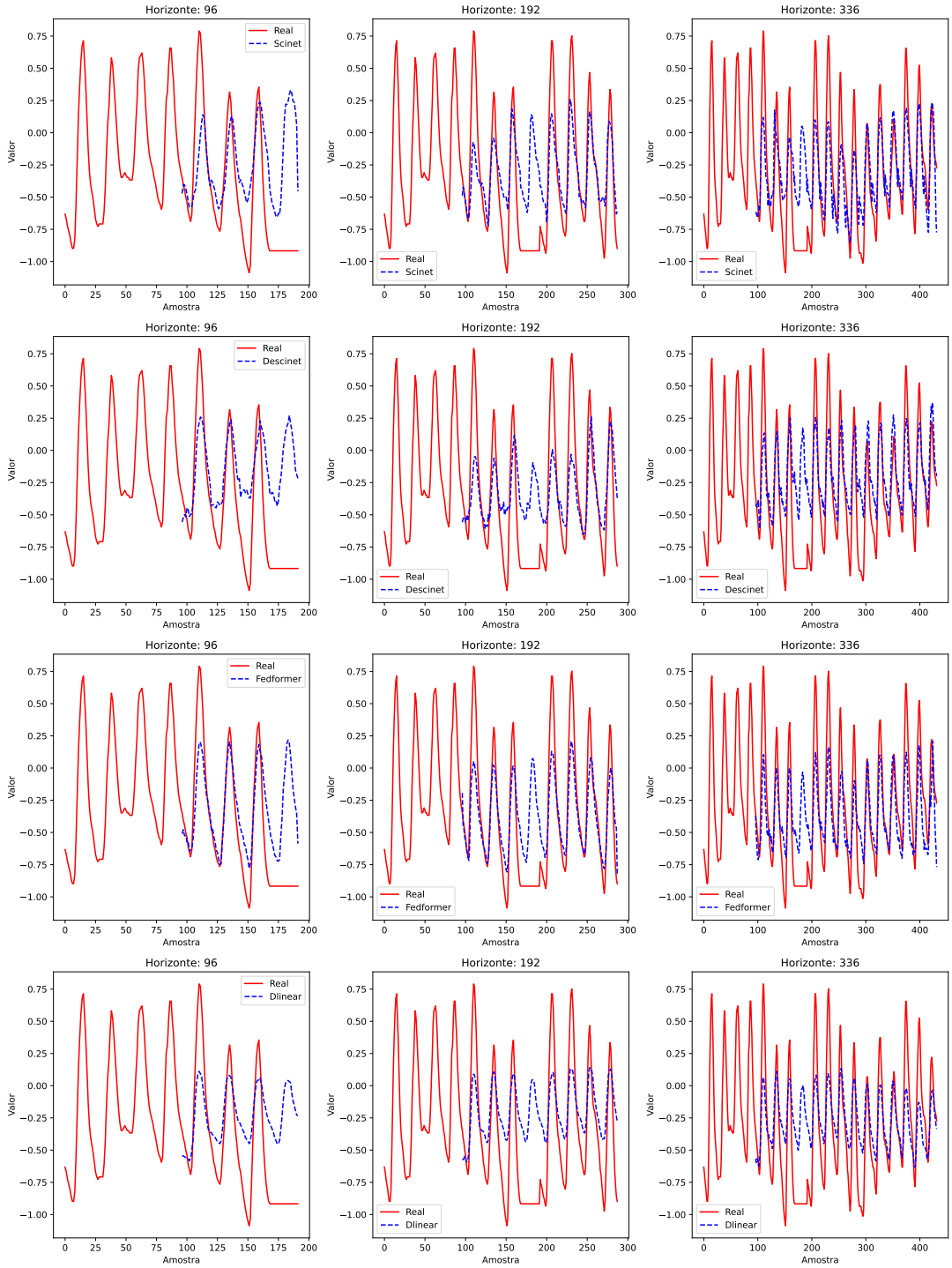


Figura 5.8: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariados ETTh2.

Fonte: Elaborada pelo autor.

ções sobre sua performance quanto ao uso desse modelo em algumas séries multivariadas.

Por outro lado, o SCINet emergiu como um modelo de destaque, especialmente no horizonte de 96, mostrando uma capacidade notável de acompanhar a série temporal com eficácia. Esta habilidade se estendeu, ainda que de forma

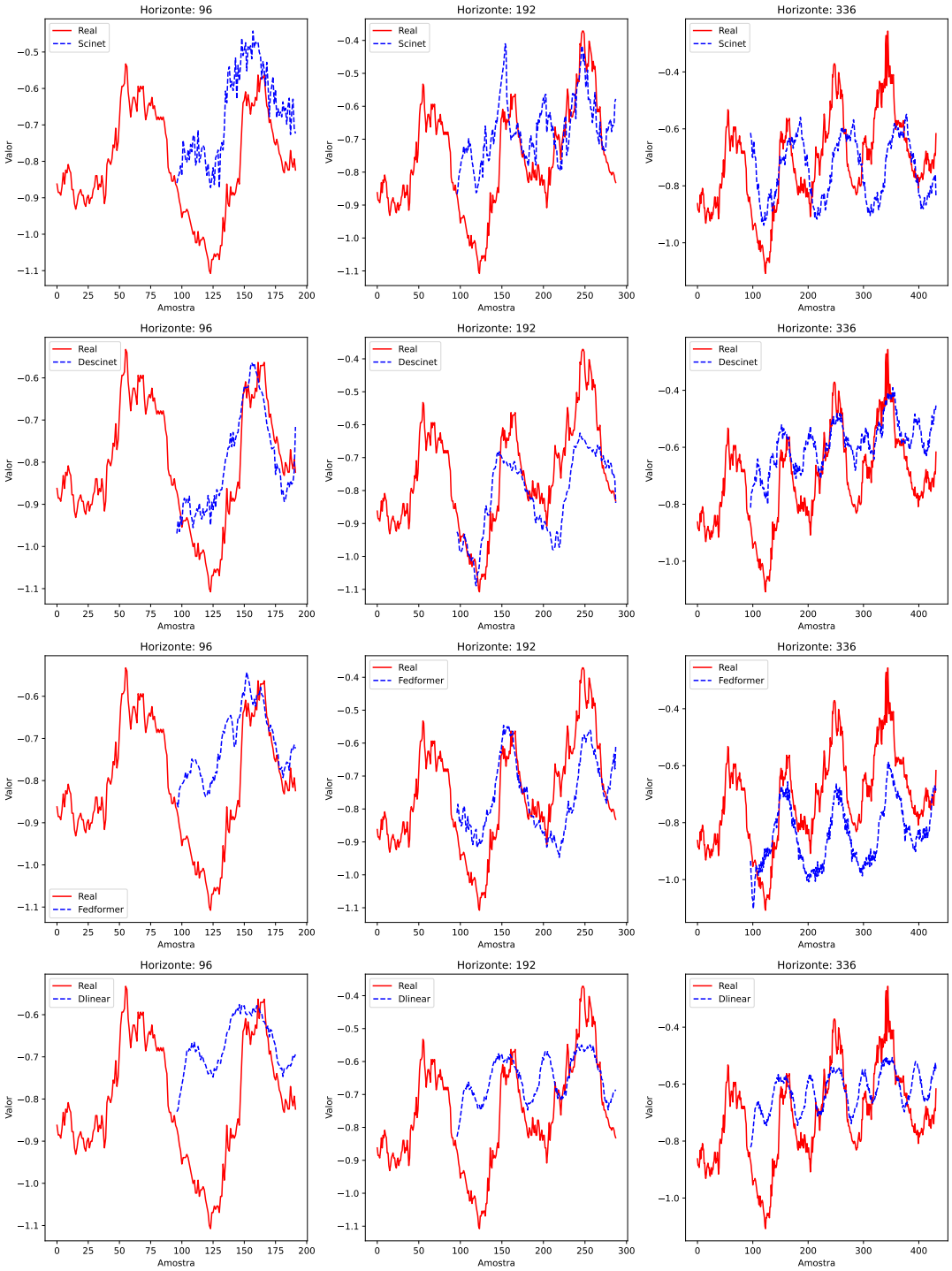


Figura 5.9: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariados ETTm1.

Fonte: Elaborada pelo autor.

atenuada, ao horizonte de 192, posicionando o SCINet como um modelo competente para previsões de curto a médio prazo. Este desempenho contrasta com o dos modelos FEDformer e DESCINet, com o FEDformer apresentando uma superioridade notável nos horizontes de 96 e 192. Esta vantagem reflete sua eficiência em capturar tanto o movimento quanto a tendência geral

da série, sugerindo uma sintonia fina com as dinâmicas subjacentes da série temporal analisada.

Ao expandir a análise para o horizonte de 336, a vantagem do FEDformer se atenua frente ao DESCINet, que demonstra uma melhoria progressiva em suas previsões. Este ajuste no desempenho do DESCINet sugere uma adaptabilidade e uma capacidade de refinamento das previsões em horizontes temporais mais longos, evidenciando uma potencialidade para análises mais extensivas de séries temporais.

Em relação as dificuldades apresentadas pelos modelos, no horizonte de 96, o SCINet, por exemplo, acaba subestimando os valores da série real, o que pode ser indicativo de uma certa rigidez do modelo em capturar picos ou fundos mais acentuados. O DESCINet, por sua vez, apresenta previsões que também tendem a desviar dos valores reais, mas de maneira que sugere uma diferente abordagem ou sensibilidade às variações da série temporal.

O FEDformer e o DLinear, embora também apresentem discrepâncias em relação à série real, cada um deles exibe características únicas em suas previsões. O FEDformer, com suas previsões, pode estar mostrando uma capacidade de capturar tendências de médio prazo, mas com dificuldades em ajustes finos para flutuações de curto prazo. O DLinear, sendo um modelo mais simples em comparação com os demais, ainda assim consegue oferecer previsões razoavelmente alinhadas com a série real, o que pode indicar uma boa eficiência em capturar a essência da série temporal sem necessariamente adentrar em complexidades.

Expandindo a análise para os horizontes de 192 e 336, é possível observar que as discrepâncias entre as previsões dos modelos e a série real tendem a se ampliar, o que é esperado devido ao aumento da dificuldade inerente à previsão de longo prazo. Neste contexto, cada modelo revela diferentes graus de robustez e adaptabilidade às dinâmicas de longo prazo da série temporal. Modelos que apresentam menor desvio em relação à série real podem estar melhor equipados com mecanismos para entender e projetar tendências de longo prazo, enquanto aqueles com maior desvio podem enfrentar desafios em manter a precisão à medida que o horizonte de previsão se estende.

5.4.1 Exchange Rate, Electricity e Traffic

Para os conjuntos de dados *Electricity*, *Exchange Rate* e *Traffic*, os comparativo para cada horizonte foram inseridos na Tabela 5.6. Começando a análise pelo conjunto de dados *Exchange Rate*, e teve as previsões ilustradas na Figura 5.10.

O modelo DESCINet conseguiu o melhor desempenho entre os modelos testados. Esse desempenho pode ser atribuído à capacidade de capturar melhor

a dinâmica temporal, as tendências e as dependências sazonais, devido ao uso de conexões residuais melhorando o fluxo de informações para camadas mais profundas. Comparado com os métodos baseados em *Transformer* Zhou et al. (2020); Kitaev et al. (2020), o DESCINet foi capaz de melhorar os resultados em até 63% e 91%, respectivamente. Em comparação com o SCINet e o DLinear, esses modelos obtiveram melhores resultados em alguns experimentos, alternando entre a segunda e terceira posição nos demais. Devido ao seu tamanho, foi necessário reduzir o número de camadas para realizar o treinamento, o que impactou no desempenho final do SCINet. É importante ressaltar que o consumo de energia é relativamente fácil de prever, pois o hábito de consumo se repete ao longo do tempo, e conseguir melhorias significativas neste conjunto de dados é uma tarefa desafiadora. Segundo Lai et al. (2017), no conjunto de dados sobre taxas de câmbio, é difícil identificar padrões repetitivos de longo prazo. Portanto, para métodos que modelam e utilizam com sucesso padrões repetitivos de curto e longo prazo nos dados, eles tendem a ter um bom desempenho quando os dados contêm tais padrões, o que não é o caso deste conjunto específico.

Tabela 5.6: Comparação do desempenho da previsão de longo prazo com modelos nos conjuntos de dados *Exchange Rate*, *ECL*, *Traffic*.

Métodos	Métricas	Exchange Rate				Electricity				Traffic			
		Horizontes de previsão				Horizontes de previsão				Horizontes de previsão			
		96	192	336	720	96	192	336	720	96	192	336	720
LSTMa	MSE	0,504	0,592	0,886	1,676	0,456	0,480	0,501	0,515	0,709	0,900	1,067	1,461
	MAE	0,590	0,610	0,795	1,095	0,480	0,497	0,581	0,595	0,401	0,523	0,599	0,787
Reformer	MSE	1,065	1,188	1,357	1,510	0,312	0,348	0,350	0,340	0,645	0,689	0,701	0,712
	MAE	0,829	0,906	0,976	1,016	0,402	0,433	0,433	0,420	0,425	0,429	0,459	0,472
LogTrans	MSE	0,968	1,040	1,659	1,941	0,258	0,266	0,280	0,283	0,584	0,585	0,633	0,677
	MAE	0,812	0,851	1,081	1,127	0,357	0,368	0,380	0,376	0,391	0,395	0,490	0,503
LSTNet	MSE	0,398	0,378	0,373	0,409	0,435	0,492	0,537	0,545	1,095	1,647	2,669	2,707
	MAE	0,460	0,451	0,439	0,454	0,471	0,512	0,589	0,605	0,154	0,354	0,364	0,386
Informer	MSE	0,847	1,204	1,672	2,478	0,274	0,296	0,300	0,373	0,354	0,419	0,583	0,916
	MAE	0,752	0,895	1,036	2,478	0,368	0,296	0,394	0,439	0,405	0,434	0,543	0,705
Autoformer	MSE	0,201	0,222	0,231	0,254	0,197	0,300	0,509	1,447	0,613	0,616	0,622	0,660
	MAE	0,317	0,334	0,338	0,361	0,323	0,369	0,524	0,941	0,388	0,382	0,337	0,408
Pyraformer	MSE	0,386	0,386	0,378	0,376	0,376	1,748	1,874	1,943	2,085	0,867	0,869	0,881
	MAE	0,449	0,443	0,443	0,445	1,105	1,151	1,172	1,206	0,468	0,467	0,469	0,473
FEDformer	MSE	0,193	0,201	0,214	0,246	0,148	0,271	0,460	1,195	0,587	0,604	0,621	0,626
	MAE	0,308	0,315	0,329	0,355	0,278	0,380	0,500	0,841	0,366	0,373	0,383	0,382
SCINet	MSE	0,061	0,106	0,181	0,525	0,168	0,175	0,189	0,231	0,613	0,635	0,640	0,642
	MAE	0,188	0,244	0,323	0,571	0,253	0,262	0,278	0,316	0,395	0,355	0,359	0,394
DLinear	MSE	0,140	0,153	**0,169**	**0,203**	**0,081**	0,157	0,305	0,643	**0,410**	**0,423**	**0,436**	**0,466**
	MAE	0,237	0,249	**0,267**	**0,301**	**0,203**	0,293	0,414	0,601	**0,282**	**0,287**	**0,296**	**0,315**
DESCINet	MSE	**0,055**	**0,102**	0,174	0,498	0,152	**0,151**	**0,173**	**0,201**	0,480	0,485	0,491	0,502
	MAE	**0,172**	**0,236**	**0,318**	0,506	0,234	**0,243**	**0,261**	**0,297**	0,295	0,302	0,305	0,311
Melhoria	MSE	**9,84** %	**3,77%**	-2,87%	-59,24%	-46,71%	**3,82%**	**8,47%**	**12,99%**	-14,58%	-12,78%	-11,20%	-7,17%
	MAE	**8,51%**	**3,28%**	-16,04%	-40,51%	-13,25%	**17,06%**	**6,12%**	**6,01%**	-4,41%	-4,97%	-2,95%	**1,29%**

Especificamente para a previsão de 96 intervalos, o algoritmo DLinear demonstrou superioridade, embora a diferença entre suas previsões e as dos outros três algoritmos fosse marginal, com todos apresentando desempenhos semelhantes e alcançando valores próximos aos reais em condições similares. Notavelmente, tanto o SCINet quanto o DESCINet exibiram previsões mais consistentes, com menos flutuações em comparação ao DLinear e, mais

acentuadamente, ao FEDformer.

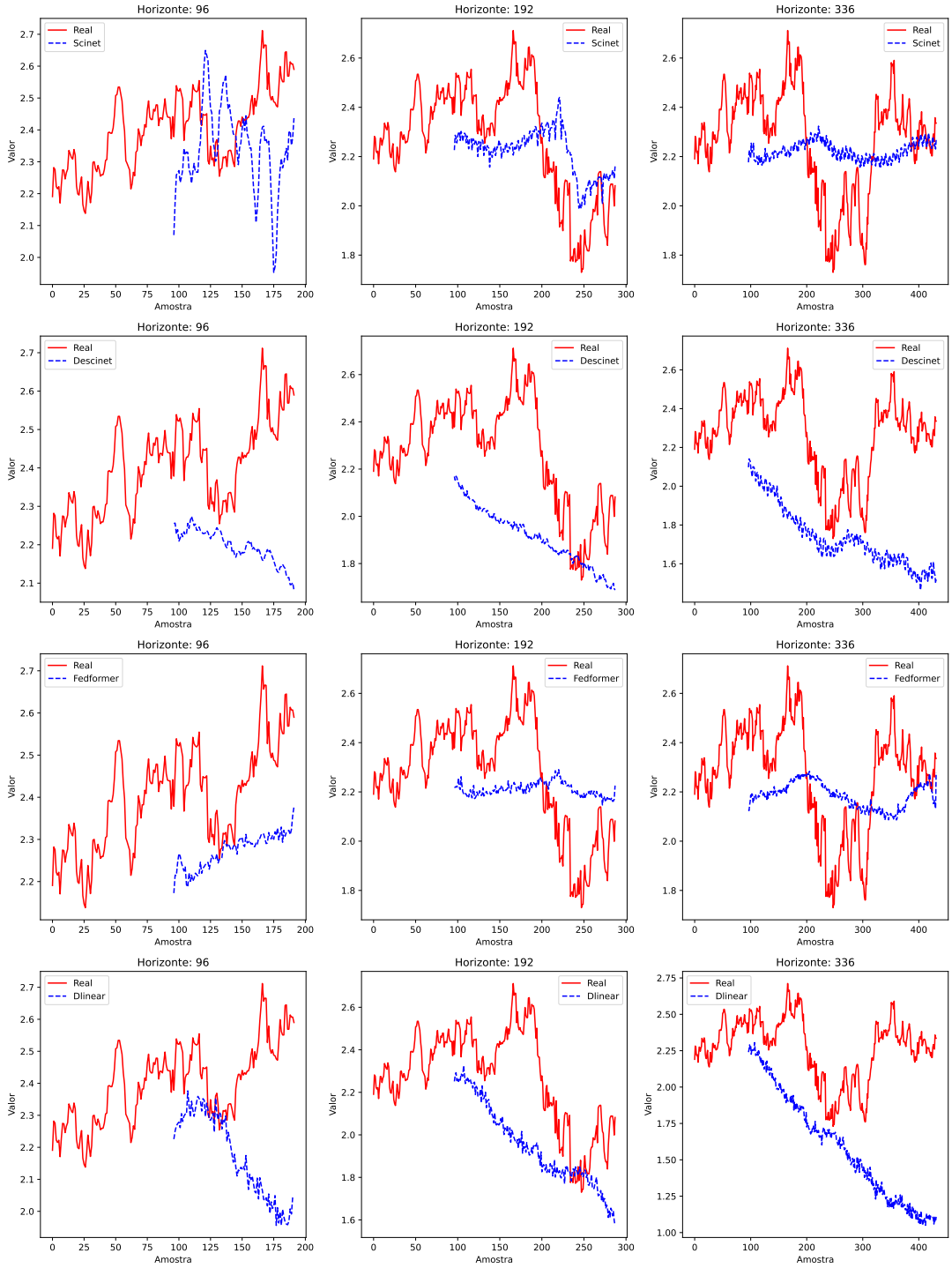


Figura 5.10: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Exchange Rate*.

Fonte: Elaborada pelo autor.

Para um horizonte temporal de 192 intervalos, o DLinear e o SCINet se destacaram, especialmente entre os pontos 50 e 100 da amostra, onde suas previsões se aproximaram significativamente dos valores reais. No entanto, após o ponto 100, em vez de manterem uma tendência linear, esses algoritmos

divergiram dos valores reais, seguindo uma trajetória descendente. Por outro lado, o DESCINet, apesar de seguir uma tendência de queda, o fez de forma mais moderada em comparação com os outros dois. Por fim, o FEDformer alcançou uma previsão razoável apenas nos últimos 50 pontos da amostra, permanecendo geralmente distante dos valores reais em comparação com os demais algoritmos.

No conjunto de dados *Electricity*, ao analisar as previsões dos modelos SCINet, DESCINet, FEDformer e DLinear em relação à série real para os horizontes de 96, 192 e 336 e ilustados na Figura 5.11, observa-se uma variedade de desempenhos que refletem as capacidades e limitações específicas de cada modelo em capturar as dinâmicas subjacentes da série temporal. Importante ressaltar que, para cada horizonte, os primeiros 96 dados da série real são utilizados para a previsão, e apenas os dados subsequentes são considerados na comparação com a série real, alinhando-se assim com a metodologia proposta.

Para o horizonte de 96, os modelos apresentam variações significativas em suas capacidades de previsão. O SCINet, por exemplo, demonstra uma habilidade moderada em seguir a série, embora com algumas imprecisões notáveis. O DESCINet, por sua vez, tende a subestimar ou superestimar a série em diversos pontos, refletindo uma sensibilidade variável às flutuações da série temporal. O FEDformer destaca-se novamente por sua capacidade de capturar tendências de médio prazo com maior precisão, embora ainda apresente desafios em ajustes finos para flutuações de curto prazo. O DLinear mostra-se como o modelo com maior dificuldade em replicar a série real, evidenciando limitações na captura das dinâmicas complexas da série temporal.

A análise para o horizonte de 192, as discrepâncias entre as previsões dos modelos e a série real tendem a aumentar, o que é esperado devido ao desafio crescente de prever com precisão em um horizonte temporal mais longo. Neste contexto, cada modelo revela diferentes graus de robustez e adaptabilidade. O SCINet e o DESCINet mostram melhorias em suas previsões, embora ainda enfrentem dificuldades em manter a precisão ao longo do horizonte. O FEDformer continua a demonstrar uma capacidade relativamente boa de acompanhar a série, apesar de algumas previsões desviarem significativamente dos valores reais. O DLinear, mantendo a tendência observada no horizonte de 96, luta para oferecer previsões alinhadas com a série real, destacando suas limitações em horizontes de previsão mais extensos.

No horizonte de 336, observa-se que os desafios enfrentados pelos modelos se intensificam, com todos eles apresentando dificuldades crescentes em replicar a série real com precisão. O SCINet e o DESCINet, apesar de algumas melhorias pontuais, não conseguem manter a consistência em suas previsões

ao longo de todo o horizonte. O FEDformer, embora mostre uma capacidade de capturar algumas tendências de longo prazo, enfrenta desafios significativos em manter a precisão. O DLinear, consistentemente com os horizontes anteriores, apresenta o desempenho mais fraco, com previsões frequentemente desalinhadas da série real, refletindo a dificuldade do modelo em adaptar-se às complexidades da série temporal em horizontes de previsão mais longos.

A Figura 5.12 apresenta os resultados dos testes realizados com o conjunto de dados *Traffic*. Ao analisar os algoritmos SCINet, DESCINet, FEDFormer e DLinear em relação à série real para os horizontes de previsão de 96, 192 e 336, observamos diferentes padrões de desempenho em termos de precisão e tendência na captura das dinâmicas subjacentes da série temporal. Importante ressaltar que a comparação é feita desconsiderando os primeiros 96 dados da série real para cada horizonte, focando assim na capacidade de previsão futura de cada modelo.

Para o horizonte de 96, o SCINet apresenta uma tendência geral de capturar os movimentos ascendentes e descendentes da série real, embora com algumas discrepâncias em magnitude, especialmente nos picos e vales. O DESCINet, por sua vez, mostra uma capacidade um pouco mais conservadora na previsão das variações, sugerindo uma menor sensibilidade às mudanças abruptas na série temporal. O FEDFormer destaca-se por sua tentativa de acompanhar as oscilações da série real, mas com evidentes desafios na precisão das amplitudes. O DLinear, embora apresente uma tentativa de seguir a tendência geral, parece ter dificuldades em capturar a variabilidade mais fina da série, resultando em previsões que podem parecer mais suavizadas em comparação com a série real.

Para o horizonte de 192, observa-se que o SCINet melhora sua capacidade de prever as tendências gerais, com uma aproximação mais precisa das variações da série real. O DESCINet continua a mostrar uma abordagem conservadora, com previsões que tendem a suavizar as flutuações reais. O FEDFormer, com sua abordagem, tenta capturar as tendências de longo prazo, mas ainda enfrenta desafios na correspondência exata das amplitudes. O DLinear mostra uma melhoria na captura das tendências gerais, embora ainda apresente limitações na precisão das flutuações menores.

Para o horizonte de 336, a complexidade aumenta devido ao maior alcance da previsão. O SCINet consegue razoavelmente capturar a direção da série, embora com variações na precisão. O DESCINet mantém sua característica de previsões mais suavizadas, potencialmente úteis para capturar tendências de longo prazo, mas menos eficazes para prever variações de curto prazo. O FEDFormer, interessantemente, mostra uma tentativa de adaptar suas previsões às mudanças de tendência da série real, embora com variações na precisão

das previsões. O DLinear, por sua vez, demonstra uma capacidade contínua de seguir a tendência geral da série, mas com limitações na captura de detalhes mais finos.

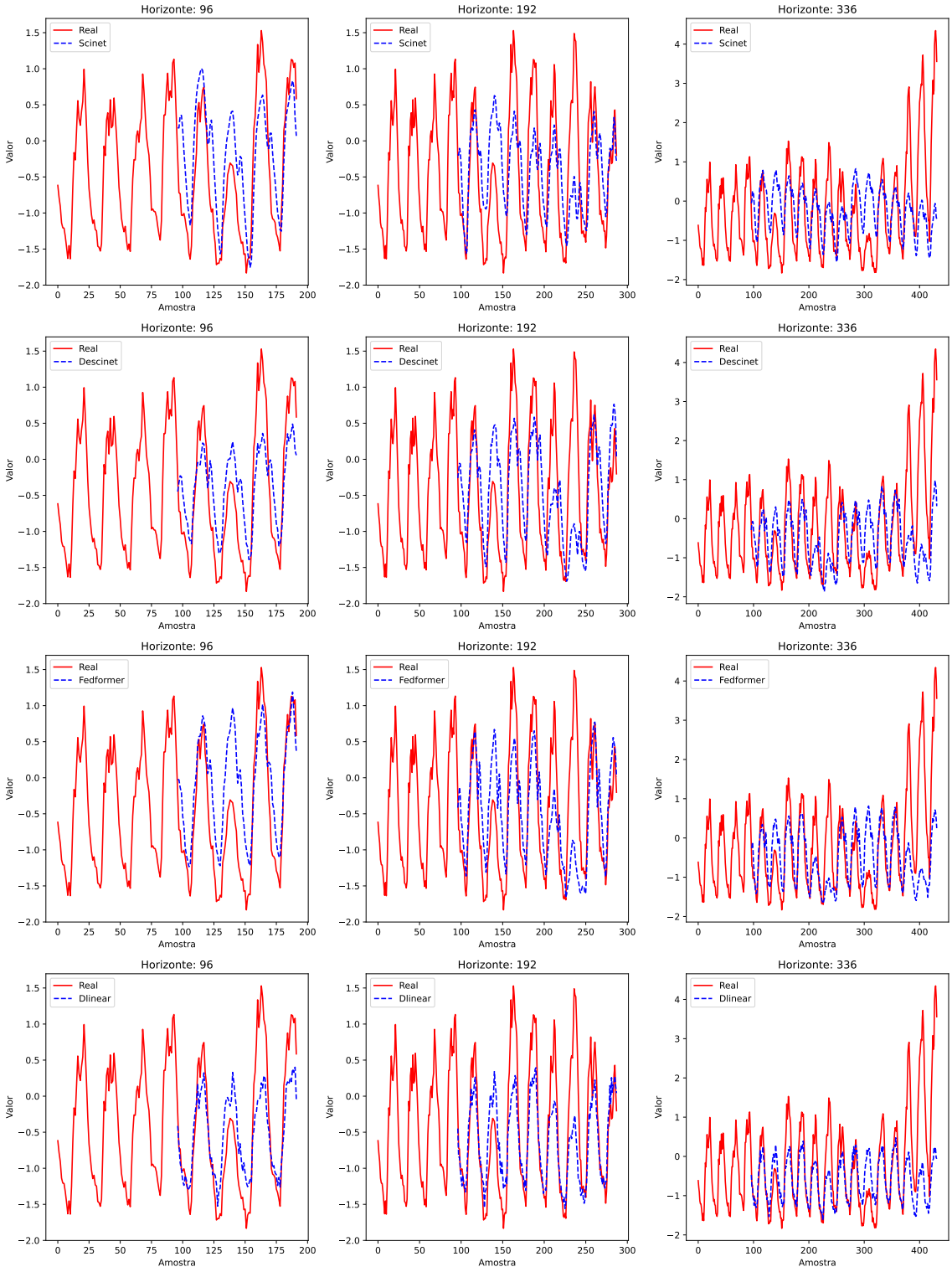


Figura 5.11: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Electricity*.

Fonte: Elaborada pelo autor.

Figura 5.12: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Traffic*.

Fonte: Elaborada pelo autor.

5.4.2 *Weather* e *Illness*

Nesta subseção, abordaremos os dois últimos conjuntos de dados testados: *Weather* e Illness. O conjunto de dados *Weather* representa um desafio maior

devido à sua complexidade, caracterizada por um número maior de variáveis. Esse conjunto engloba múltiplos fatores meteorológicos, como temperatura, umidade, pressão atmosférica, entre outros, tornando a tarefa de previsão significativamente mais desafiadora. A natureza multifacetada do clima, com suas inúmeras variáveis interdependentes, exige modelos capazes de capturar e interpretar a complexa dinâmica que define os padrões meteorológicos.

Por outro lado, o conjunto de dados Illness é considerado menos complexo em termos de volume de dados e variabilidade. Com menos dados disponíveis, o foco recai sobre a capacidade dos modelos de prever tendências em dados que podem ser mais limitados, mas não necessariamente menos complexos em termos de padrões subjacentes. A previsão de doenças, embora aparentemente mais direta devido ao menor número de variáveis envolvidas, ainda apresenta seus próprios desafios, especialmente no que diz respeito à precisão das previsões em face de surtos ou declínios na incidência de doenças.

A distinção entre esses dois conjuntos de dados ilustra a gama de desafios encontrados na modelagem e previsão de séries temporais. Enquanto o conjunto de dados *Weather* exige um entendimento profundo e uma modelagem precisa de múltiplas variáveis interconectadas, o conjunto de dados Illness demanda uma abordagem focada na captura de tendências em um conjunto de dados potencialmente mais esparso. Ambos os conjuntos de dados testam a flexibilidade e a robustez dos modelos de previsão, exigindo estratégias adaptativas para lidar com a complexidade variável e a disponibilidade de dados. A compilação desses dados estão na Tabela 5.7.

Tabela 5.7: Comparação do desempenho da previsão de longo prazo nos conjuntos de dados multivariado *Weather* e *ILI*.

Métodos	Métricas	Weather				ILI			
		Horizontes de previsão				Horizontes de previsão			
		96	192	336	720	24	36	48	60
Reformer	MSE	1,065	1,188	1,357	1,510	0,312	0,348	0,350	0,340
	MAE	3,829	4,906	5,276	5,516	1,402	1,433	1,463	1,520
Informer	MSE	0,300	0,598	0,578	1,059	5,764	4,755	4,763	5,264
	MAE	0,384	0,544	0,523	0,741	1,677	1,467	1,469	1,564
Autoformer	MSE	0,266	0,307	0,359	0,419	3,483	3,103	2,669	2,770
	MAE	0,336	0,367	0,395	0,428	1,238	1,270	1,203	1,225
Pyraformer	MSE	0,896	0,622	0,739	1,004	1,420	7,394	7,551	7,662
	MAE	0,556	0,624	0,753	0,934	2,012	2,031	2,057	2,100
FEDformer	MSE	0,217	0,276	0,339	0,403	2,687	2,887	2,797	**2,809**
	MAE	0,296	0,336	0,380	0,428	1,145	1,180	1,155	**1,163**
SCINet	MSE	0,261	0,306	0,381	0,375	3,168	3,175	3,189	3,231
	MAE	0,321	0,360	0,394	0,402	1,253	1,262	1,278	1,316
DLinear	MSE	**0,176**	**0,220**	**0,265**	**0,323**	2,940	2,826	**2,667**	3,011
	MAE	**0,237**	**0,282**	**0,319**	**0,362**	1,281	1,163	**1,154**	1,246
DESCINet	MSE	0,207	0,248	0,273	0,327	**2,612**	**2,819**	2,791	3,004
	MAE	0,242	0,299	0,278	0,397	**1,879**	**1,625**	2,5486	1,239
Melhoria	MSE	-14,98%	-11,29%	-2,93%	-1,22%	**11,16%**	**0,25%**	-4,44%	-6,49%
	MAE	-2,07%	-5,69%	-14,75%	-8,82%	**7,96%**	**0,39%**	-54,72%	-6,13%

A previsão meteorológica, bem como a modelagem de variáveis correlatas, constitui um desafio em virtude da complexidade intrínseca a esses fenômenos. Os resultados apresentados na Figura 5.13 mostram que nenhum dos modelos testados conseguiu realizar previsões acuradas para a amostra em estudo, independentemente dos horizontes de previsão considerados. Analisando de forma individual, temos, que o SCINet demonstrou uma boa capacidade de capturar tendências a longo prazo, mas somente em horizontes mais extensos. No entanto, sua precisão varia significativamente, com alguns períodos mostrando desvios consideráveis da série real. Este comportamento sugere que, enquanto o SCINet pode ser útil para previsões gerais de tendência.

DESCINet, por outro lado, apresentou uma relativa consistência em suas previsões, mantendo-se mais próximo da série real em vários pontos. O modelo consegue ter uma melhor capacidade de ajuste fino, possivelmente devido a abordagem mais detalhada na análise de padrões temporais com o uso de conexões densas. No entanto, ele também mostrou momentos de desalinhamento significativo, o que pode indicar limitações na captura de mudanças abruptas ou na previsão de eventos extremos.

Já o FEDFormer destacou-se por sua habilidade em capturar variações sutis e responder a mudanças na série temporal, sugerindo uma forte capacidade de modelagem de dependências de longo alcance. Este modelo parece ser particularmente eficaz em horizontes de previsão mais longos, onde a capacidade de antecipar e ajustar-se a tendências emergentes é crucial. No entanto, a variabilidade em suas previsões também indica uma sensibilidade a flutuações, que pode ser tanto uma força (em termos de responsividade) quanto uma fraqueza (em termos de estabilidade).

DLinear mostrou uma abordagem consistente e estável, com previsões que frequentemente seguem de perto a série real, indicando uma robustez notável em sua capacidade de previsão. Este modelo parece oferecer uma base sólida para previsões confiáveis, embora possa não capturar tão eficazmente as nuances ou mudanças repentinas como alguns dos outros modelos.

Essa performance limitada dos modelos reforça a complexidade da tarefa de previsão meteorológica e sinaliza a necessidade de desenvolvimento e aprimoramento de modelos mais sofisticados e adaptáveis, capazes de capturar as nuances e a volatilidade intrínsecas a séries temporais meteorológicas. Este cenário destaca a importância de continuar explorando e desenvolvendo novas abordagens e técnicas em modelagem de séries temporais, especialmente em contextos de alta complexidade e incerteza como é o caso da previsão meteorológica.

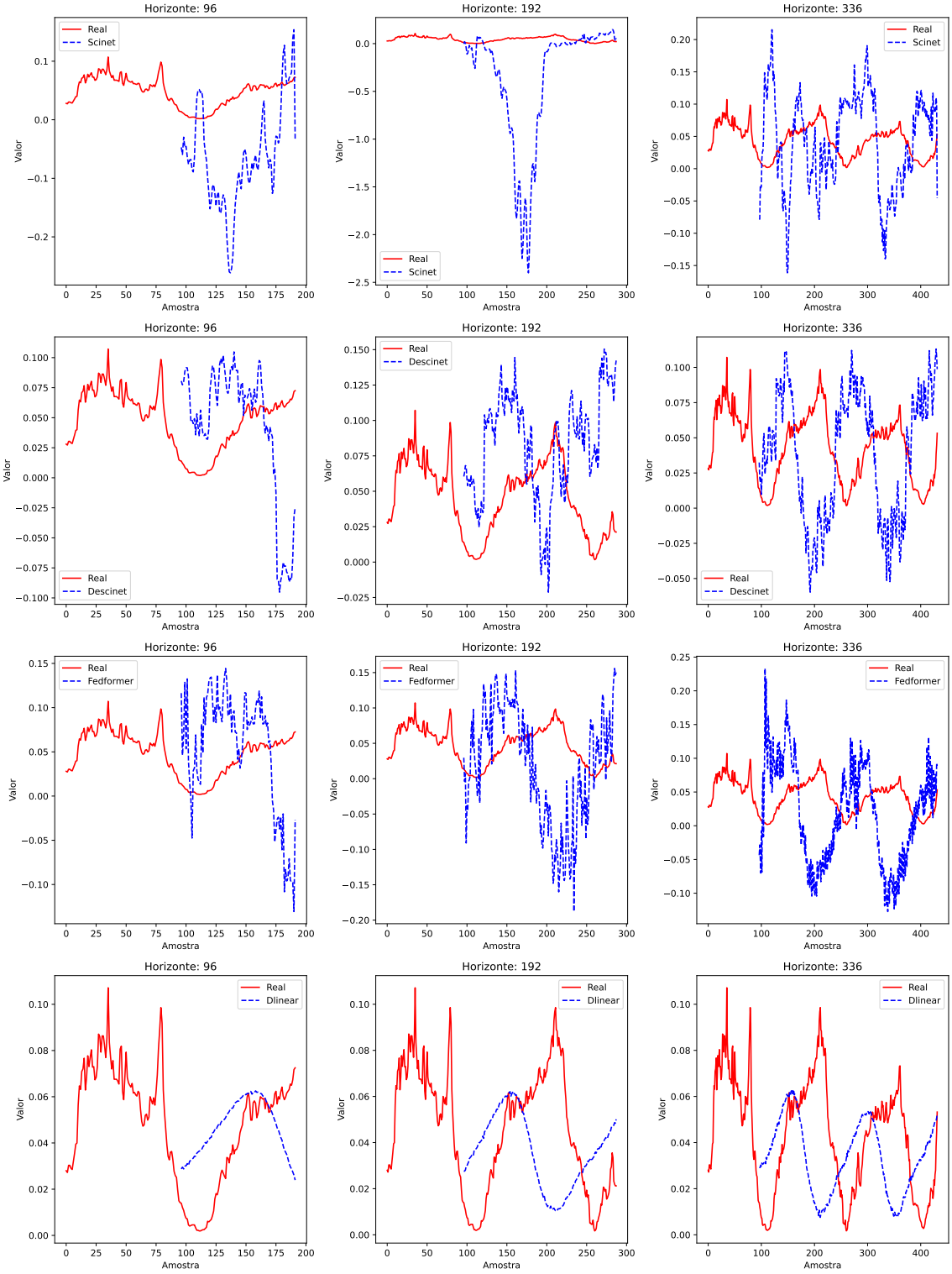


Figura 5.13: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *Weather*.

Fonte: Elaborada pelo autor.

O conjunto de dados *National Illness* possui um volume limitado de dados históricos, o que representa um desafio para a precisão das previsões de alguns modelos. A Figura 5.14 ilustra o desempenho de diferentes modelos na previsão desses dados.

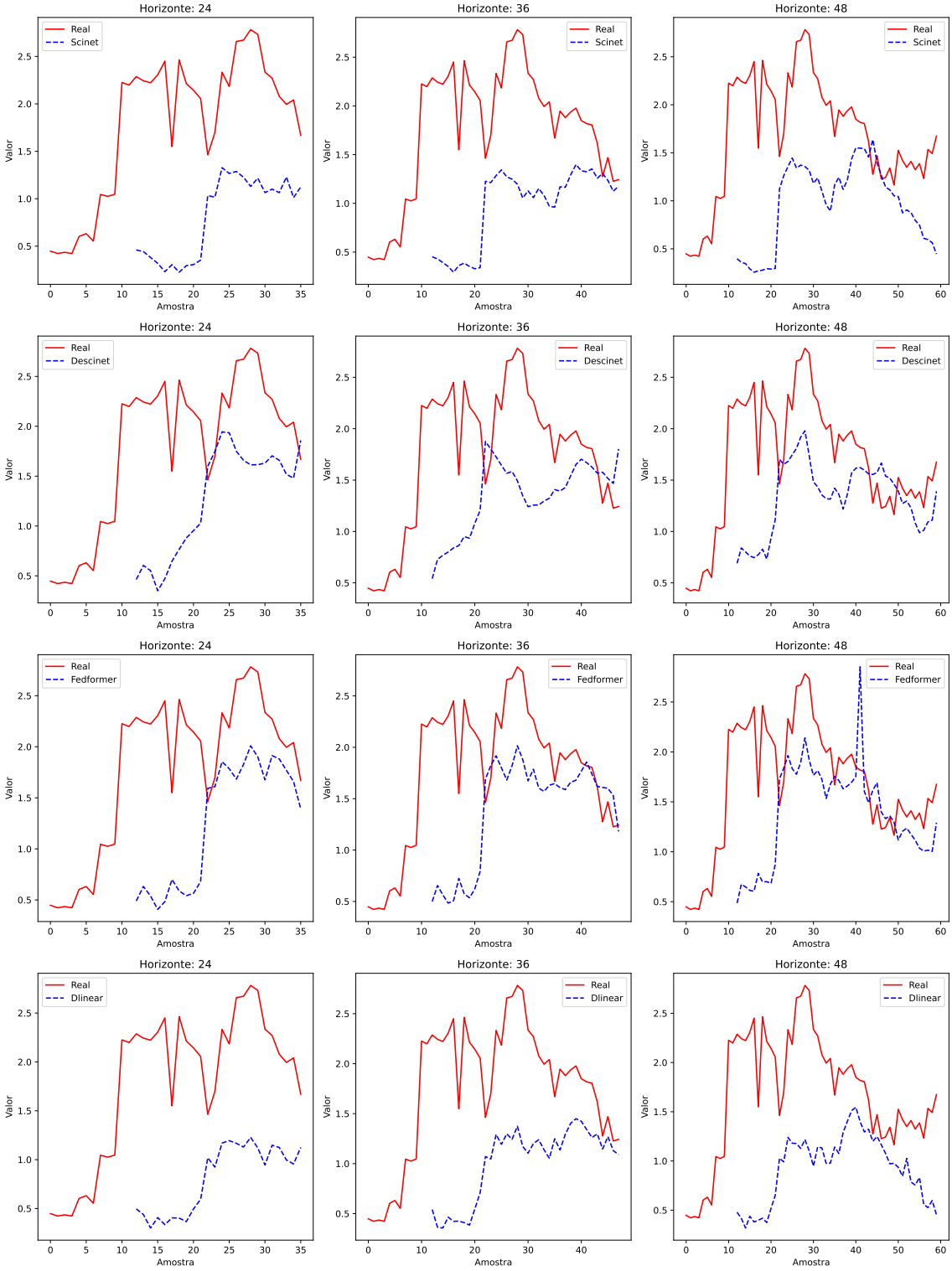


Figura 5.14: Comparativo de previsão para os horizontes de 96, 192 e 336 entre os algoritmos DESCINet, SCINet, FEDformer e DLinear no conjunto de dados multivariado *National Illness*.

Fonte: Elaborada pelo autor.

Para esse conjunto de dados o SCINet mostrou novamente uma habilidade em seguir a tendência geral da série real, embora seja melhor em horizontes mais curtos. Outro problema apresentado pelo modelo é que ele subestimar ou superestimar os picos e vales, indicando uma possível dificuldade em capturar

a volatilidade inerente aos dados de doença. Isso sugere que, enquanto o SCINet pode ser eficaz para previsões de tendências gerais, pode não ser o mais adequado para prever eventos extremos ou mudanças abruptas na série temporal.

O DESCINet apresentou uma capacidade melhor em capturar as flutuações da série real, com previsões que muitas vezes superestimam os valores reais. Este comportamento pode ser indicativo de uma sensibilidade maior do modelo às variações na série temporal, o que pode ser vantajoso em cenários onde a previsão de picos é crucial. No entanto, essa sensibilidade também pode levar a previsões menos precisas em períodos de menor volatilidade.

FEDFormer destacou-se pela sua capacidade de adaptar-se a mudanças na série temporal, oferecendo previsões que, em muitos casos, alinham-se bem com a série real. Este modelo parece ter uma forte capacidade de modelagem de dependências de longo alcance, o que é particularmente útil em dados de doença que podem ser influenciados por uma variedade de fatores externos e internos. Ainda assim, o modelo apresentou algumas dificuldades em prever com precisão os valores mais extremos, sugerindo limitações na captura de eventos atípicos.

DLinear forneceu uma abordagem consistente e estável, com previsões que frequentemente seguem de perto a série real. Este comportamento indica uma robustez notável na capacidade de previsão do modelo, embora possa não capturar tão eficazmente as nuances ou mudanças repentinas como alguns dos outros modelos. O DLinear pode ser preferível em situações que exigem previsões confiáveis e consistentes, mas pode não ser o ideal para capturar todos os aspectos da dinâmica da doença

5.5 Estudo da Loss landscape

A exploração da paisagem de perda em modelos de aprendizagem profunda constitui um campo de investigação essencial para compreender a otimização e a capacidade de generalização destes modelos. Ao analisar a paisagem de perda, procura-se visualizar e interpretar como a função de perda varia em resposta a mudanças nos parâmetros do modelo. Este entendimento é particularmente valioso para otimizar algoritmos de aprendizagem e para diagnosticar potenciais desafios em treinamentos de redes neurais profundas.

Dada a complexidade e a alta dimensionalidade dos espaços de parâmetros em modelos de aprendizagem profunda, a visualização direta da paisagem de perda torna-se impraticável. Portanto, abordagens simplificadas, mas informativas, são necessárias para mapear estas superfícies. A técnica escolhida envolve a projeção do espaço de parâmetros de alta dimensão em um espaço

de dimensão menor, através da perturbação dos parâmetros do modelo ao longo de direções aleatórias. Essa abordagem permite a análise da variação da função de perda em um subespaço representativo, revelando características da superfície de perda, como mínimos locais, sela e a rugosidade geral da paisagem.

A escolha de direções aleatórias para a análise da paisagem de perda baseia-se na premissa de que, mesmo em um espaço de parâmetros de alta dimensão, direções particulares podem revelar informações sobre a topologia da função de perda.

A análise da paisagem de perda dos modelos SCINet e DESCINet foi feita no conjunto de dados ETTh1 e pode ser visualizada na Figura 5.15.

A superfície inferior pertence ao modelo DESCINet, e pode-se observar que ela é notavelmente mais curva, o que é uma indicativa de um mínimo de perda com uma bacia de atração ampla. Esta característica da paisagem de perda é muitas vezes associada a uma maior capacidade de generalização. Minimizadores com essas propriedades são menos sensíveis a variações nos dados e perturbações nos parâmetros, evidenciando uma robustez desejável em modelos de aprendizado de máquina. Isso pode ser atribuído ao fato de que mínimos mais planos correspondem a uma maior região no espaço de parâmetros onde a função de custo é aproximadamente constante, sugerindo que o modelo pode manter um desempenho consistente mesmo na presença de ruído ou variações nos dados de entrada.

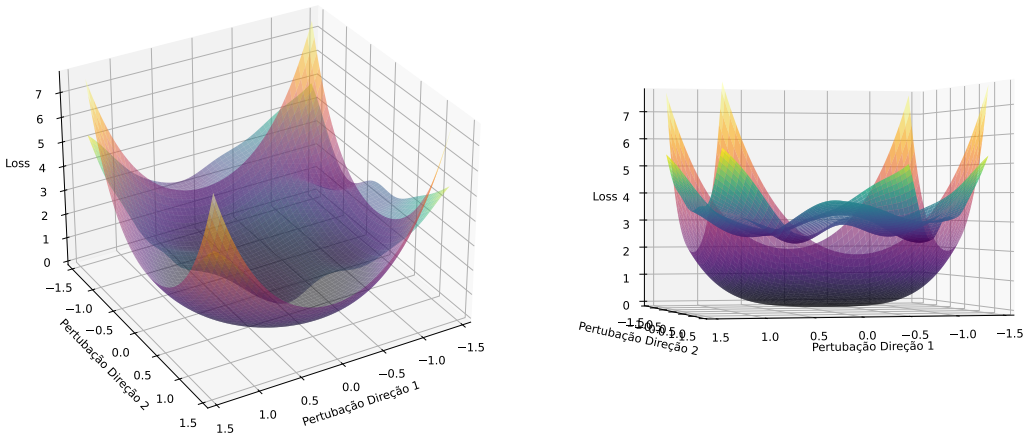


Figura 5.15: Comparativo de *loss landscape* entre SCINet e DESCINet.

Fonte: Elaborada pelo autor.

Por outro lado, a superfície superior, com uma forma quase plana, mas com mais rugosidade, isso pode ser interpretado como um sinal de que o modelo é capaz de ajustar-se de forma excessiva a detalhes específicos dos dados de treinamento, um fenômeno conhecido como sobreajuste. Em tal paisagem, pequenas alterações nos parâmetros do modelo podem levar a grandes varia-

ções na perda, refletindo uma sensibilidade aguda às características específicas do conjunto de dados. Como resultado, o modelo não apenas aprende as relações subjacentes entre os dados de entrada e saída, mas também captura ruído aleatório presente no conjunto de treinamento. Embora isso possa melhorar a performance do modelo nos dados de treinamento, geralmente resulta em uma generalização pobre em novos dados, uma vez que o ruído aprendido não se aplica a exemplos fora do conjunto de treinamento.

Embora os mínimos possam alcançar um erro de treinamento baixo, eles são potencialmente problemáticos do ponto de vista da generalização. Mínimos agudos são frequentemente associados a uma solução de sobreajuste, onde o modelo se ajusta excessivamente às idiossincrasias dos dados de treino, perdendo assim a flexibilidade para adaptar-se a novos dados.

Portanto, no contexto desta pesquisa e das metas estabelecidas na presente tese, a superfície de perda do DESCINet confirmou a hipótese lançada que as conexões densas dariam estabilidade de convergência ao modelo SCINet. Isso alinha-se com o objetivo central da aprendizagem profunda, que é desenvolver modelos que não apenas minimizem o erro empírico, mas que também exibam robustez e adaptabilidade frente à variabilidade inerente do mundo real.

Na questão de avaliar a *loss*, um outro estudo foi realizado para corroborar com a imagem anterior sobre convergência e estabilidade das conexões residuais densas adicionadas no algoritmo SCINet pode ser visto na Figura 5.16. A imagem apresenta uma comparação detalhada das curvas de *loss* de treino e validação para dois algoritmos de aprendizado de máquina, DESCINet e SCINet, em quatro horizontes de previsão: 96, 192, 336 e 720. Observa-se que ambos os algoritmos exibem um declínio rápido na *loss* de treino nas primeiras épocas, indicando um aprendizado efetivo inicial.

Para o DESCINet, a convergência entre a *loss* de treino e de validação é notável em todos os horizontes, com a *loss* de validação mantendo-se consistentemente inferior à de treino. Isso sugere que o DESCINet não apenas aprende bem, mas também generaliza eficientemente para dados não vistos, mantendo a *overfitting* controlado. O fenômeno é mais acentuado no horizonte de 96, onde a diferença entre as duas curvas é mais visível.

Por outro lado, o SCINet mostra uma maior discrepância entre as curvas de treino e validação, particularmente nos horizontes de 192 e 336, onde a *loss* de validação ultrapassa a de treino após um certo número de épocas. Isso pode indicar uma tendência ao *overfitting*, onde o modelo se ajusta demais aos dados de treino e perde sua capacidade de generalizar. No entanto, no horizonte de 720, o SCINet parece recuperar a generalização, como evidenciado pela proximidade das curvas de treino e validação.

Em termos de comparação direta entre os algoritmos, o DESCINet consis-

tentemente exibe uma menor *loss* de validação em comparação com o SCINet, sugerindo que ele pode ser o modelo mais robusto para a tarefa de previsão. Além disso, o DESCINet estabiliza mais rapidamente, atingindo o plateau de *loss* em menos épocas que o SCINet em todos os horizontes. Esta é uma indicação de que o DESCINet pode ser mais eficiente computacionalmente, além de ter uma performance superior em termos de generalização.

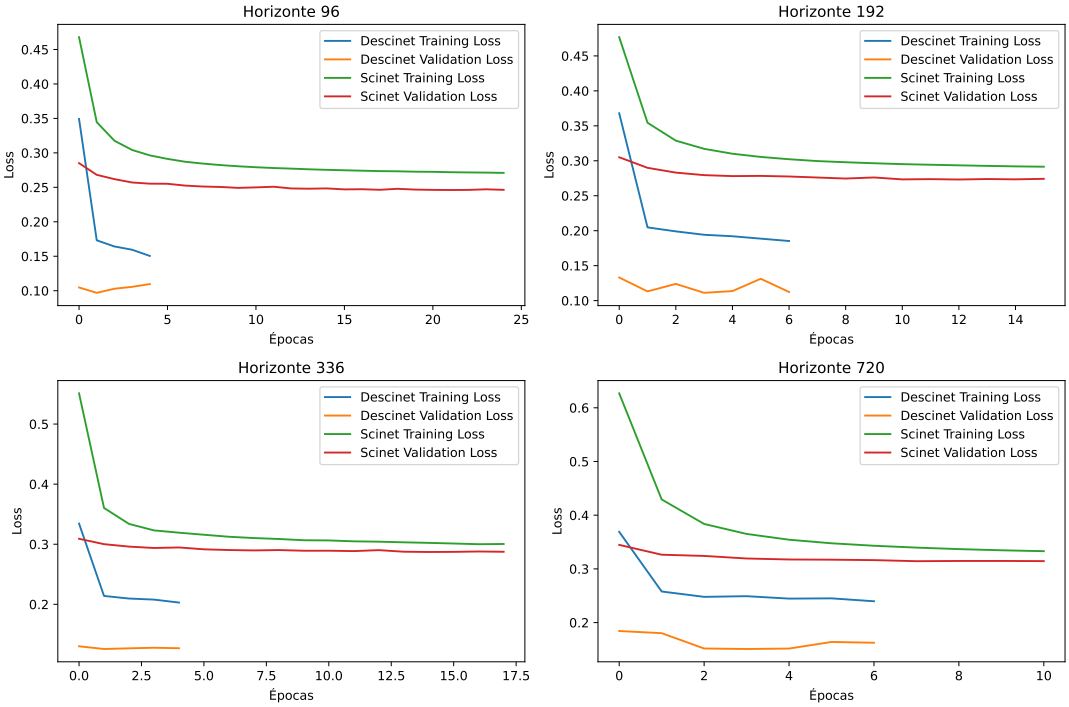


Figura 5.16: Comparativo de *loss* de treino e validação entre SCINet e DESCINet utilizando o conjunto de dados ETTh1.

Fonte: Elaborada pelo autor.

5.6 Estudo Tamanho da Sequência

No estudo do impacto do tamanho da sequência nas previsões de longo prazo para séries temporais, observa-se uma premissa comum de que campos receptivos amplos são benéficos para o desempenho. Esta noção é reforçada pela suposição de que quanto mais dados históricos um modelo pode processar, melhores serão suas previsões. No entanto, uma investigação mais aprofundada revela que essa não é sempre a realidade para muitos modelos baseados em *Transformer*, como discutido por (Zeng et al., 2022). A maioria desses modelos não demonstra melhora significativa no desempenho ao aumentar a janela de observação, sugerindo uma limitação na captura de informações temporais de longo prazo.

Em contraste, o modelo proposto se apoia numa arquitetura de redes neurais convolucionais com aplicação de conexões densas, e destaca-se por sua

capacidade de lidar de maneira robusta com diferentes tamanhos de sequência. Através da análise da Figura 5.17 fica evidente que o modelo DESCINet, diferentemente de linhas de base como o DLinear, não só mantém uma performance estável com uma menor quantidade de dados mas também se sobressai em relação aos outros modelos em conjuntos de dados reduzidos. Este fator é crítico, especialmente em domínios onde a aquisição de dados é limitada ou custosa.

A capacidade do modelo de aprender com janelas de observação mais longas também é comprovada, reduzindo consistentemente os escores de erro quadrático médio à medida que o campo receptivo aumenta. Isso não apenas valida a eficácia do modelo em capturar relações temporais profundas mas também indica sua superioridade em ajustar-se a diferentes contextos de entrada. Essa flexibilidade é um diferencial notável, considerando que muitos modelos exigem um grande volume de dados para generalizar bem.

Além disso, a abordagem utilizando CNN com conexões densas possibilita que o modelo capture padrões complexos nos dados sem a necessidade de um pré-processamento extensivo ou ajustes nos hiperparametros. A arquitetura densamente conectada permite que cada camada receba informações de todas as camadas anteriores, facilitando a propagação de características relevantes através da rede e aprimorando a capacidade de aprendizado.

A figura analisada ilustra que, mesmo quando o tamanho da sequência sofre uma diminuição, o modelo mantém um desempenho consistente. Isso sugere que o modelo tem uma habilidade inerente de extrair características, o que o torna particularmente valioso em cenários práticos. Esse aspecto é acentuado quando se compara com outros modelos, como o DLinear, que demonstram uma dependência crítica em grandes quantidades de dados para alcançar previsões confiáveis.

No contexto prático das previsões de séries temporais, a habilidade do modelo de entregar previsões precisas com uma quantidade variável de dados não é apenas uma vantagem técnica; é uma característica importante que potencializa sua aplicabilidade em uma vasta gama de situações. A eficácia em diferentes comprimentos de sequência e a menor dependência de grandes volumes de dados históricos conferem ao modelo uma versatilidade capaz de atender às necessidades de diversos domínios aplicados.

Portanto, o modelo desenvolvido não só aborda as limitações identificadas em modelos existentes mas também introduz uma robustez notável em seu desempenho, independentemente da quantidade de dados disponíveis. Tal característica o distingue e estabelece sua relevância em um campo em rápida evolução, onde a eficiência e a precisão são essenciais para a extração de informações a partir de séries temporais.

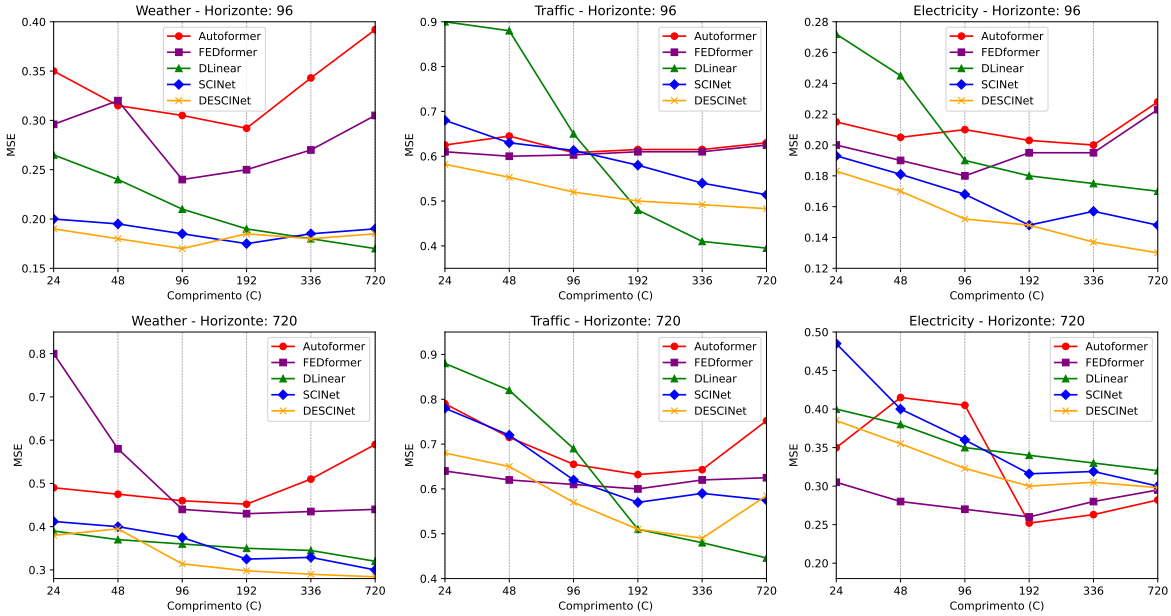


Figura 5.17: Desempenho de previsão (MSE) com janelas de observação variáveis nos 3 maiores conjuntos de dados: *Traffic*, *Electricity* e *Weather*. As janelas de observação são selecionadas para serem L = 24, 48, 96, 192, 336, 720, e os horizontes de previsão são T = 96, 720.

Fonte: Elaborada pelo autor.

5.7 Impacto das Conexões Residuais Densas

Nesta seção, avaliamos a eficácia preditiva do modelo proposto, que combina uma arquitetura convolucional com conexões residuais densas. O objetivo é verificar a eficiência preditiva do modelo em diferentes horizontes de previsão e entender o papel das conexões residuais densas no aprendizado da rede. Para isso, realizamos um estudo de ablação, consistindo em uma série de experimentos para analisar a contribuição dessas conexões no desempenho do modelo.

Os modelos avaliados variam conforme a quantidade de conexões residuais densas incorporadas, sendo classificados da seguinte maneira:

- DESC-0: Modelo sem conexões residuais (SCINet).
- DESC-2: Modelo incorporando 2 conexões residuais densas.
- DESC-4: Modelo com 4 conexões residuais densas.
- DESC-6: Modelo contendo 6 conexões residuais densas.
- DESC-8: Modelo equipado com 8 conexões residuais densas.
- DESC-10: Modelo completo, com todas as conexões residuais densas.

A Figura 5.18 apresenta um boxplot do erro quadrático médio (MSE) para cada configuração do modelo em três conjuntos de dados distintos. Essa representação visual permite uma análise clara da distribuição dos erros e facilita a identificação de outliers e a compreensão da variabilidade dos resultados.

Os dados indicam uma tendência de redução no MSE à medida que aumentamos o número de conexões residuais densas, evidenciando uma melhoria na precisão das previsões. Especificamente, observamos que o modelo DESC-10 (completo) apresenta os menores valores de MSE, sugerindo que a integração completa das conexões residuais densas oferece um aprimoramento significativo na capacidade preditiva do modelo para séries temporais extensas.

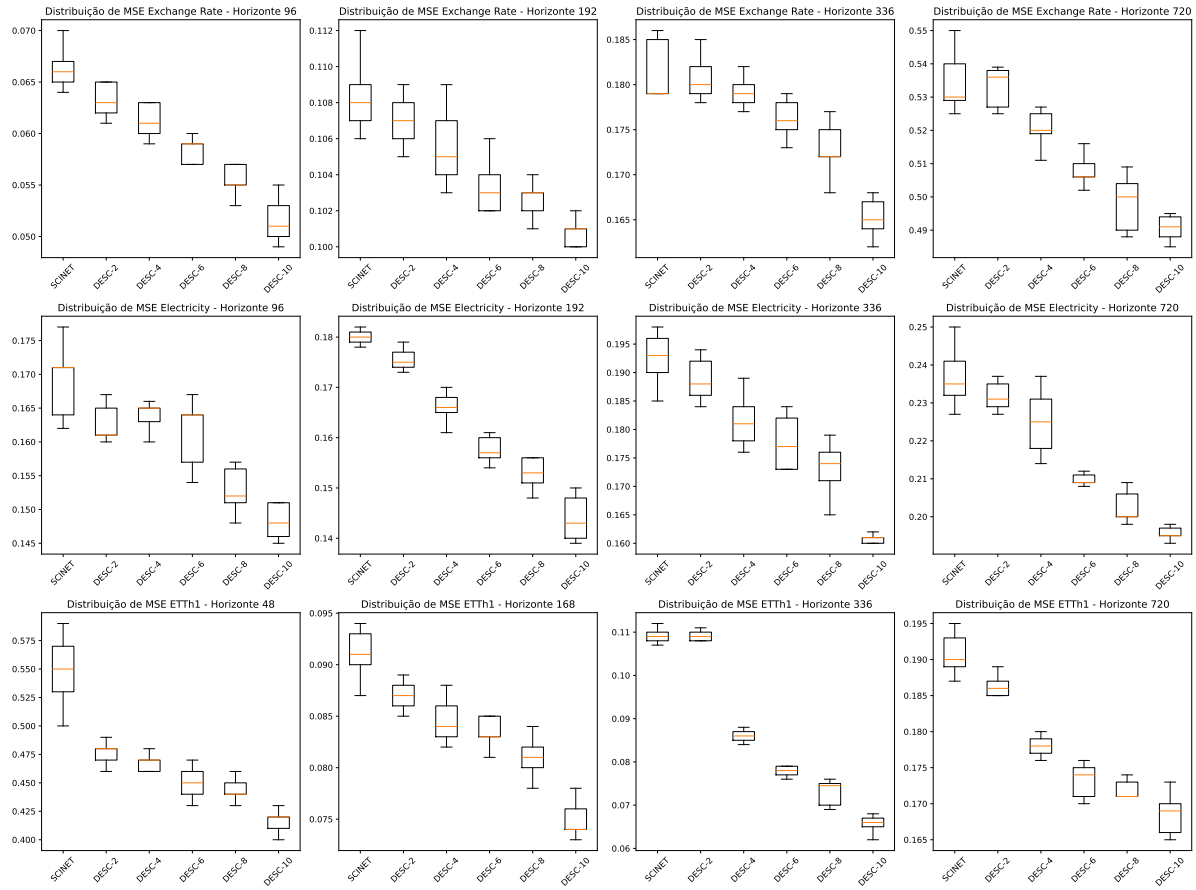


Figura 5.18: Boxplots comparando as estimativas médias do erro quadrático médio (MSE) para diferentes configurações do modelo DESCINet.

Na Tabela 5.8, apresentada compara o desempenho de diferentes configurações do modelo DESCNet, variando o número de conexões densas em três séries temporais multivariadas para diversos horizontes de previsão, utilizando as métricas de MSE e MAE. A organização da tabela em colunas que representam diferentes horizontes de previsão para cada um dos conjuntos de dados *Exchange Rate*, *Electricity* e *ETTh1* permite uma comparação direta entre as configurações com diferentes números de conexões densas.

A análise dos dados revela uma tendência geral de redução nos valores de MSE e MAE à medida que o número de conexões de salto aumenta, indicando uma melhoria na precisão das previsões. Esta tendência é consistente em todos os conjuntos de dados e horizontes de previsão, sugerindo que as conexões residuais densas contribuem positivamente para a capacidade do modelo de capturar a dinâmica subjacente das séries temporais. Além disso, a variação no desempenho entre os diferentes conjuntos de dados sugere que a eficácia das conexões de salto pode depender das características específicas de cada série temporal. Por exemplo, o conjunto de dados *Electricity* mostra uma redução mais acentuada no MSE e MAE com o aumento das conexões, em comparação com o conjunto de dados *ETTh1*.

Tabela 5.8: Simulação dos modelos DESCNet com diferentes números de conexões de salto em séries temporais (multivariadas) para diferentes horizontes.

Métodos	Métricas	*Exchange Rate*				*Electricity*				*ETTh1*			
		96	192	336	720	96	192	336	720	48	168	336	720
DESC-0	MSE	0,068	0,111	0,386	0,720	0,190	0,199	0,212	0,240	0,0467	0,095	0,115	0,195
	MAE	0,172	0,236	0,318	0,506	0,240	0,243	0,261	0,297	0,165	0,220	0,207	0,336
DESC-2	MSE	0,062	0,106	0,181	0,527	0,167	0,174	0,188	0,230	0,046	0,089	0,109	0,189
	MAE	0,185	0,244	0,330	0,576	0,263	0,269	0,290	0,316	0,165	0,232	0,238	0,356
DESC-4	MSE	0,062	0,107	0,180	0,526	0,166	0,174	0,185	0,230	0,047	0,087	0,094	0,181
	MAE	0,191	0,244	0,330	0,576	0,260	0,266	0,290	0,316	0,165	0,229	0,233	0,351
DESC-6	MSE	0,060	0,104	0,178	0,515	0,164	0,166	0,183	0,222	0,045	0,085	0,085	0,176
	MAE	0,180	0,214	0,330	0,516	0,253	0,259	0,290	0,316	0,165	0,224	0,220	0,340
DESC-8	MSE	0,059	0,105	0,179	0,506	0,159	0,154	0,175	0,208	0,046	0,083	0,077	0,175
	MAE	0,178	0,245	0,340	0,509	0,241	0,263	0,279	0,330	0,165	0,220	0,210	0,339
DESCNet	MSE	0,055	0,102	0,174	0,498	0,152	0,151	0,173	0,201	0,046	0,080	0,071	0,173
	MAE	0,172	0,236	0,318	0,506	0,24	0,243	0,261	0,297	0,165	0,220	0,207	0,306

O impacto das conexões de salto também varia de acordo com o horizonte de previsão. Em horizontes mais longos, por exemplo, 720, a melhoria no desempenho é mais pronunciada, o que pode indicar que as conexões densas são particularmente úteis para capturar dependências de longo prazo nas séries temporais.

A adição de conexões de salto densas melhora significativamente a precisão das previsões, especialmente para horizontes de previsão mais longos, evidenciado pela redução sistemática nos valores de MSE e MAE à medida que o número de conexões aumenta. O impacto das conexões de salto varia entre diferentes séries temporais, sugerindo que a configuração ótima do modelo pode precisar ser ajustada com base nas características específicas de cada conjunto de dados. A melhoria no desempenho em horizontes de previsão mais longos indica que as conexões residuais densas são eficazes em capturar dependências temporais complexas, o que é crucial para a previsão precisa de séries temporais multivariadas.

Esta análise técnica da tabela destaca a importância das conexões de salto densas na modelagem de séries temporais, oferecendo insights valiosos sobre

como essas conexões influenciam o desempenho preditivo do modelo DESCNet em diferentes contextos de dados e horizontes de previsão.

Para avaliar o impacto quantitativo das conexões residuais densas sobre o erro de previsão, aplicamos o teste ANOVA. Os resultados demonstram diferenças estatisticamente significativas entre os modelos, rejeitando a hipótese nula de equivalência. Notavelmente, o modelo DESC-4 mostrou uma melhoria significativa em relação ao DESC-0 (SCINet), e o DESC-10 superou o DESC-8, corroborando a hipótese de que o número de conexões residuais densas tem uma influência marcante no erro de previsão.

Esses achados reforçam a importância das conexões residuais densas na modelagem de séries temporais, destacando seu potencial para melhorar a precisão das previsões em diferentes horizontes temporais.

Para investigar o impacto quantitativo das conexões residuais densas no erro de previsão, aplicamos o teste ANOVA. Os resultados obtidos rejeitam a hipótese nula de equivalência entre os modelos, revelando diferenças estatisticamente significativas. Notavelmente, o modelo DESC-4 demonstrou uma melhoria significativa em comparação ao DESC-0, e o DESC-8 superou o DESC-4. Estes achados corroboram a hipótese de que o número de conexões residuais densas influencia de maneira significativa o erro de previsão.

Os experimentos de ablação foram realizados em quatro conjuntos de dados distintos, abrangendo três horizontes de previsão diferentes, para validar a eficácia dos módulos propostos. A eficácia de cada configuração foi avaliada através da variação no número de conexões residuais densas, com os resultados detalhados na Tabela 5.9.

Tabela 5.9: Statistic e P-Value.

	Exchange Rate	*Electricity*	*ETTh1*
Statistics	30,61	9,664	24,10
P-Value	2,857e-08	0,16e-18	2,081e-07

Em todos os experimentos, o DESCINet foi configurado com um nível máximo de $L = 3$, permitindo até 10 conexões residuais. Posteriormente, o número de conexões foi progressivamente reduzido até o mínimo de 2, iniciando a remoção a partir das camadas mais baixas e progredindo em direção à camada raiz. A análise dos dados revela que um aumento no número de conexões residuais densas está associado a uma redução no erro de previsão, ressaltando a importância deste componente arquitetural. De forma notável, a exclusão total das conexões residuais resultou em um incremento do erro de até 17,20% em alguns cenários, como observado no conjunto de dados *Electricity* para horizontes de previsão de 192 etapas.

5.8 Considerações Finais

Após uma análise detalhada dos resultados obtidos com a implementação do modelo DESCINet, este capítulo demonstrou os avanços significativos na previsão de séries temporais. O DESCINet, uma evolução do *framework* SCINet através da integração de conexões residuais densas, provou ser eficaz na captura de características temporais complexas, mantendo uma alta precisão preditiva em diversos contextos de séries temporais.

A inclusão das conexões residuais densas emergiu como uma ferramenta crucial para o desempenho preditivo do modelo, facilitando a comunicação entre os nós da árvore binária do modelo e permitindo uma distribuição mais eficaz das informações. Essa melhoria se traduziu em um desempenho superior quando comparado a algoritmos relevantes, particularmente em horizontes de previsão mais longos, onde o SCINet anteriormente mostrava limitações.

As análises comparativas entre o DESCINet e outros quinze modelos destacados na literatura sublinharam sua eficácia em uma variedade de contextos de séries temporais, tanto univariadas quanto multivariadas. Em séries univariadas, o DESCINet superou modelos como LSTMa, LSTNet, Reformer, FEDformer e DLinear, demonstrando uma capacidade notável de capturar características de curto e longo prazo. Em contextos multivariados, destacou-se pela habilidade de processar sequências temporais extensas de forma eficiente, superando desafios como previsões em séries com variações abruptas e complexidades sazonais.

A análise da paisagem de perda reforçou a robustez do DESCINet, revelando uma superfície de perda mais plana que está associada a uma maior generalização e estabilidade do modelo. O estudo detalhado do impacto das conexões residuais densas, realizado por meio de um experimento de ablação, confirmou a hipótese inicial de que um aumento no número dessas conexões leva a uma redução significativa no erro de previsão.

No próximo capítulo, será realizada uma análise crítica das hipóteses discutidas no capítulo introdutório, expondo quais foram confirmadas e os motivos por trás dessas confirmações. Além disso, serão discutidas as limitações encontradas durante a pesquisa e sugeridas direções futuras para a continuação deste trabalho, visando aprimorar ainda mais a modelagem de séries temporais com o uso de conexões residuais densas e outras técnicas avançadas.

CAPÍTULO

6

Conclusões

6.1 Conclusões

Nesta tese de doutorado, foi realizada uma investigação aprofundada sobre a previsão de séries temporais univariadas e multivariadas de longas sequências. Introduziu-se o modelo DESCINet como uma solução inovadora para previsões de longa duração. A análise criteriosa, suportada por robustas evidências empíricas, confirmou a eficácia do modelo DESCINet, demonstrando sua habilidade em identificar padrões de repetição de curto e longo prazo nos conjuntos de dados estudados. Nos experimentos com dados reais, o DESCINet mostrou-se superior, melhorando a capacidade de previsão para problemas de séries temporais longas.

Durante o desenvolvimento da pesquisa, foi dada especial atenção às conexões densas, cuja implementação resultou em melhorias significativas no modelo SCINet, culminando na criação do modelo DESCINet. A introdução dessas conexões densas visou fortalecer a capacidade do modelo de capturar dependências complexas e de longo alcance nos dados, o que se traduziu em um desempenho aprimorado. A análise comparativa entre o SCINet original e o DESCINet revelou que as conexões densas foram fundamentais para aumentar a estabilidade e a eficiência da convergência durante o treinamento, além de contribuir para uma melhor generalização do modelo em diferentes conjuntos de dados. Essa inovação não apenas reforçou a robustez do DESCINet como solução para previsão de séries temporais, mas também abriu caminhos para futuras investigações sobre a otimização de arquiteturas de redes neurais profundas para tarefas de previsão complexas.

Adicionalmente, ao comparar o desempenho do DESCINet com o do DLInear, seu principal competidor, observou-se que o DESCINet superou o DLInear em diversos conjuntos de dados multivariados. Esta superioridade evidencia a capacidade do DESCINet não apenas em lidar com a complexidade inerente aos dados multivariados, mas também em sua eficiência em prever valores com uma quantidade limitada de dados como entrada.

Apesar dos avanços significativos e da superioridade demonstrada em diversos aspectos, o modelo DESCINet apresenta limitações, particularmente em relação ao tempo de treinamento e eficiência computacional. Uma das principais limitações observadas é que o DESCINet demanda um tempo de treinamento mais longo em comparação ao SCINet e ao DLInear. Esta desvantagem é atribuída à complexidade adicional introduzida pelas conexões densas, que, embora melhorem a capacidade de previsão e a generalização do modelo, também aumentam a carga computacional durante o treinamento.

Diante desses desafios, a validação das capacidades do DESCINet e o impacto das conexões densas foram meticulosamente examinados através de uma série de experimentos rigorosos. Para este fim, sete bases de dados de séries temporais reais foram selecionadas: (i) ETT, (ii) *Weather*, (iii) *Exchange Rate*, (iv) *Electricity*, (v) *Traffic*, e (vi) *Illness*. O objetivo central desses experimentos era avaliar as métricas MAE e MSE em contextos de previsão, variando a quantidade de conexões densas no modelo. Os resultados desses experimentos permitiram confirmar as hipóteses lançadas inicialmente.

Conforme as hipóteses levantadas inicialmente:

- **Hipótese 1 (Estabilidade de convergência)**: Confirmada, visto que o modelo DESCINet apresentou uma *loss landscape* mais suave em comparação ao modelo original SCINet, refletindo em uma convergência mais estável e previsível. Esta hipótese pode ser confirmada pelas Figuras 5.15 e 5.16 que retratam essas informações.

- **Hipótese 2 (Robustez contra variações abruptas)**: Não podemos afirmar que o modelo reduziu significativamente o erro em séries temporais com variações abruptas, já que em alguns pontos o SCINet acabou se saindo melhor nesses casos do que o próprio DESCINet. Este resultado pode ser observado nas Figuras 5.5, 5.7, 5.8 e 5.10.

- **Hipótese 3 (Aprimoramento da acurácia com conexões densas)**: Confirmada, como evidenciado pelos experimentos de previsão, onde o DESCINet superou o modelo original, SCINet, em termos de acurácia. Esta hipótese pode ser avaliada nas Tabelas 5.4, 5.5, 5.6 e 5.7 mostram o desempenho do DESCINet através das métricas MSE e MAE.

Portanto, os resultados alcançados neste trabalho não apenas confirmam as hipóteses propostas, mas também estabelecem o DESCINet como um modelo avançado e eficaz para previsão de séries temporais.

6.2 Publicações e Colaborações

O resultado obtido ao longo do desenvolvimento desta tese de doutorado foi reportado no seguinte artigo:

- SILVA, André Quintiliano Bezerra; GONÇALVES, Wesley Nunes; MATSUBARA, Edson Takashi. DESCINet: A hierarchical deep convolutional neural network with skip connection for long time series forecasting. Expert Systems with Applications, p. 120246, 2023.

Outras publicações que foram realizadas de forma secundária durante o período do doutorado.

- SILVA, André Quintiliano Bezerra. Predicting cervical cancer with metaheuristic optimizers for training lstm. In: Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V 19. Springer International Publishing, 2019. p. 642-655.

- SAKIYAMA, Kenzo Miranda; SILVA, Andre Quintiliano Bezerra; MATSUBARA, Edson Takashi. Twitter breaking news detector in the 2018 Brazilian presidential election using word embeddings and convolutional neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019. p. 1-8.

6.3 Trabalhos Futuros

Como perspectivas para pesquisas futuras, diversas estratégias podem ser exploradas para aprimorar a eficácia e a eficiência do modelo DESCINet. Uma abordagem promissora envolve a otimização da rede por meio da redução do número de parâmetros, visando não apenas aprimorar a eficiência no tempo de treinamento, mas também minimizar o risco de sobreajuste, contribuindo para uma melhor generalização do modelo em diferentes conjuntos de dados. Além disso, a exploração de conexões densas em algoritmos alternativos apresenta um potencial significativo, explorando seus benefícios em contextos variados para melhorar a capacidade de aprendizado e a robustez do modelo.

Importante destacar que as séries temporais analisadas nesta tese possuem natureza determinística. No entanto, muitas aplicações práticas operam

com séries estocásticas, caracterizadas por sua imprevisibilidade e variações aleatórias. Portanto, existe um vasto campo de oportunidades para adaptar e evoluir o DESCINet para previsões em ambientes mais complexos e desafiadores, como o mercado financeiro, onde a capacidade de prever tendências e padrões em meio à volatilidade pode oferecer vantagens estratégicas significativas.

Adicionalmente, será conduzido um estudo detalhado sobre o modelo DLInear, com o objetivo de incorporar possíveis soluções e técnicas adotadas por esse modelo que possam contribuir para o aprimoramento do DESCINet. A análise comparativa entre os modelos permitirá identificar características e mecanismos específicos do DLInear que, se integrados ao DESCINet, podem potencializar sua performance, especialmente em termos de eficiência computacional e precisão de previsão. Essa abordagem sinérgica visa não apenas superar as limitações identificadas no DESCINet, mas também explorar novas fronteiras na modelagem e previsão de séries temporais, abrindo caminho para inovações que possam beneficiar uma ampla gama de aplicações práticas.

Essas direções para pesquisas futuras refletem o compromisso contínuo com a evolução da modelagem de séries temporais, destacando o potencial para novas descobertas e avanços significativos no domínio. Através desses esforços, espera-se não apenas expandir o entendimento das dinâmicas complexas inerentes às séries temporais, mas também contribuir para o desenvolvimento de soluções mais robustas, precisas e eficientes para desafios de previsão em diversos campos de aplicação.

Referências Bibliográficas

Al-Muzaini, H. A., Al-Yahya, T. N., e Benhidour, H. (2018). Automatic arabic image captioning using rnn-lstm-based language model and cnn. *International Journal of Advanced Computer Science and Applications*, 9(6). 29

Ali, S., Can, M., Shah, M. I., Jiang, J., Ahmed, Z., e Murshed, M. (2022). Exploring the linkage between export diversification and ecological footprint: evidence from advanced time series estimation techniques. *Environmental Science and Pollution Research*, 29(25):38395–38409. 3

Alligood, K. T., Sauer, T. D., e Yorke, J. A. (1996). *Chaos: An Introduction to Dynamical Systems*. Springer. 20

Amari, S.-i., Park, H., e Ozeki, T. (2006). Singularities affect dynamics of learning in neuromanifolds. *Neural computation*, 18(5):1007–1065. 64

Bahdanau, D., Cho, K., e Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. 46, 78

Bai, S., Kolter, J. Z., e Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. 4, 33, 48

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, 7700:437–478. 28

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., et al. (2022). Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, 55(6):1–36. 55, 56

Bertoin, D., Bolte, J., Gerchinovitz, S., e Pauwels, E. (2021). Numerical influence of relu'(0) on backpropagation. *Advances in Neural Information Processing Systems*, 34:468–479. 66

Box, G. E. et al. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons. 19, 78

Box, G. E. P. e Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day, revised edition edition. 19

Brockwell, P. J. e Davis, R. A. (1991). *Time Series: Theory and Methods*. Springer-Verlag New York, 2 edition. 12

Brockwell, P. J. e Davis, R. A. (2002). *Introduction to time series and forecasting*. Springer. 19

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., e Dubrawski, A. (2022). N-hits: Neural hierarchical interpolation for time series forecasting. 54

Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., Cui, X., Witbrock, M., Hasegawa-Johnson, M. A., e Huang, T. S. (2017). Dilated recurrent neural networks. *Advances in neural information processing systems*, 30. 46

Chatfield, C. (2004). *The Analysis of Time Series: An Introduction*. Chapman and Hall/CRC, 6 edition. 12

Chen, Z., Ma, M., Li, T., Wang, H., e Li, C. (2023). Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97:101819. 4, 21

Chen, Z., Sivaparthipan, C., e Muthu, B. (2022). Iot based smart and intelligent smart city energy optimization. *Sustainable Energy Technologies and Assessments*, 49:101724. 3

Cheng, D., Yang, F., Xiang, S., e Liu, J. (2022). Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218. 3

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., e Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. 4, 31

Cleveland, R. B., Cleveland, W. S., McRae, J. E., e Terpenning, I. (1990). Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73. 17

Cortes, C. e Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297. 4

Cui, Y., Xie, J., e Zheng, K. (2021). Historical inertia: A neglected but powerful baseline for long sequence time-series forecasting. 52

Devlin, J., Chang, M.-W., Lee, K., e Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 35

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211. 29

Fan, J., Wang, Z., Sun, D., e Wu, H. (2022). Sepformer-based models: More efficient models for long sequence time-series forecasting. *IEEE Transactions on Emerging Topics in Computing*. 51

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285. 4

Goodfellow, I., Bengio, Y., Courville, A., e Bengio, Y. (2016). *Deep learning*. MIT press Cambridge. 28

Graves, A. e Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, páginas 37–45. 4

Grosse, R. (2017). Lecture 15: Exploding and vanishing gradients. *University of Toronto Computer Science*. 60

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press. 12

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press. 20

Hanin, B. (2018). Which neural net architectures give rise to exploding and vanishing gradients? *Advances in neural information processing systems*, 31. 60

Harvey, A. C. (1993). *Time Series Models*. Harvester Wheatsheaf, 2 edition. 17

Harvey, A. C., Trimbur, T. M., e Van Dijk, H. K. (2007). Trends and cycles in economic time series: A bayesian approach. *Journal of Econometrics*, 140(2):618–649. 22

He, K., Zhang, X., Ren, S., e Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 770–778. xii, 5, 57, 58, 59, 60

Huang, G., Liu, Z., Van Der Maaten, L., e Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4700–4708. 5, 58, 61, 62

Huang, S., Wang, D., Wu, X., e Tang, A. (2019). Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM international conference on information and knowledge management*, páginas 2129–2132. 48

Hylleberg, S. (1992). *Modelling Seasonality*. Oxford University Press. 17

Hyndman, R. J. e Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts. 22

James, J., Markos, C., e Zhang, S. (2021). Long-term urban traffic speed prediction with deep learning on graphs. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7359–7370. 22

Kantz, H. e Schreiber, T. (2004). *Nonlinear Time Series Analysis*. Cambridge Nonlinear Science Series. Cambridge University Press. 20

Khan, K. S., Kunz, R., Kleijnen, J., e Antes, G. (2003). Five steps to conducting a systematic review. *Journal of the royal society of medicine*, 96(3):118–121. 45

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., e Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, páginas 1–24. 3

Khuntia, S. R., Rueda, J. L., e van Der Meijden, M. A. (2016). Forecasting the load of electrical power systems in mid-and long-term horizons: a review. *IET Generation, Transmission & Distribution*, 10(16):3971–3977. 22

Kim, G., Ahn, J.-B., Kryjov, V. N., Sohn, S.-J., Yun, W.-T., Graham, R., Kolli, R. K., Kumar, A., e Ceron, J.-P. (2016). Global and regional skill of the seasonal predictions by wmo lead centre for long-range forecast multi-model ensemble. *International Journal of Climatology*, 36(4):1657–1675. 22

Kitaev, N., Kaiser, L., e Levskaya, A. (2020). Reformer: The efficient transformer. 5, 50, 78, 91

Krizhevsky, A., Sutskever, I., e Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105. 33, 57

Lai, G., Chang, W.-C., Yang, Y., e Liu, H. (2017). Modeling long and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 78, 91

Lai, G., Chang, W.-C., Yang, Y., e Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks. 51

Lecun, Y., Bengio, Y., e Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444. 4

LeCun, Y., Bengio, Y., e Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444. 27

Lee, W. K. (2020). Partial correlation-based attention for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, páginas 13720–13721. 50

Li, C., Xu, H., Tian, J., Wang, W., Yan, M., Bi, B., Ye, J., Chen, H., Xu, G., Cao, Z., et al. (2022). mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*. 6

Li, H., Kadav, A., Durdanovic, I., Samet, H., e Graf, H. P. (2018). A note on the complexity of deep neural networks. *arXiv preprint arXiv:1802.08435*. 62

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., e Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. 36, 50, 78

Liu, F., Ren, X., Zhang, Z., Sun, X., e Zou, Y. (2020). Rethinking skip connection with layer normalization. In *Proceedings of the 28th international conference on computational linguistics*, páginas 3586–3598. xii, 58, 60

Liu, M., Zeng, A., Xu, Z., Lai, Q., e Xu, Q. (2021a). Time series is a special sequence: Forecasting with sample convolution and interaction. 4, 49, 75, 79

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., e Dustdar, S. (2021b). Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*. 5, 50, 79

Mahum, R., Munir, H., Mughal, Z.-U.-N., Awais, M., Sher Khan, F., Saqlain, M., Mahamad, S., e Tlili, I. (2023). A novel framework for potato leaf disease detection using an efficient deep learning model. *Human and Ecological Risk Assessment: An International Journal*, 29(2):303–326. 5

Manibardo, E. L., Laña, I., e Del Ser, J. (2021). Deep learning for road traffic forecasting: Does it make a difference? *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6164–6188. 22

Mehdizadeh, S. (2020). Using ar, ma, and arma time series models to improve the performance of mars and knn approaches in monthly precipitation modeling under limited climatic data. *Water Resources Management*, 34:263–282. 3

Montgomery, D. C., Jennings, C. L., e Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons. 19

Moon, J., Hossain, M. B., e Chon, K. H. (2021). Ar and arma model order selection for time-series modeling with imagenet classification. *Signal Processing*, 183:108026. 3

Morettin, P. A. e Toloi, C. (2006). *Análise de séries temporais*. ABE - Projeto Fisher. Edgard Blucher. 20

Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D., e De, S. (2023). Stock market prediction using deep learning algorithms. *CAAI Transactions on Intelligence Technology*, 8(1):82–94. 5

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., e Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. 48

Oreshkin, B. N., Carpov, D., Chapados, N., e Bengio, Y. (2019). N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*. 54, 78

Oreshkin, B. N., Carpov, D., Chapados, N., e Bengio, Y. (2021). Meta-learning framework with applications to zero-shot time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, páginas 9242–9250. 53

Orhan, A. E. e Pitkow, X. (2017). Skip connections eliminate singularities. *arXiv preprint arXiv:1701.09175*. 63, 64

Oyedotun, O. K. e Aouada, D. (2020). Why do deep neural networks with skip connections and concatenated hidden representations work? In *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part III 27*, páginas 380–392. Springer. 58

Oyedotun, O. K., Shabayek, A. E. R., Aouada, D., e Ottersten, B. (2020). Going deeper with neural networks without skip connections. In *2020 IEEE International Conference on Image Processing (ICIP)*, páginas 1756–1760. 57

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., e Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*. 47

Rios, R. A. (2013). *Improving time series modeling by decomposing and analysing stochastic and deterministic influences*. PhD thesis, University of São Paulo. Citado nas páginas. 20

Rumelhart, D. E., Hinton, G. E., e Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536. 4, 29

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252. 57

Saad, D. e Solla, S. A. (1995). On-line learning in soft committee machines. *Physical Review E*, 52(4):4225. 64

Salinas, D., Flunkert, V., Gasthaus, J., e Januschowski, T. (2020). Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191. 47, 78

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., e Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4510–4520. 63

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117. 28

Schneider, J. e Vlachos, M. (2023). A survey of deep learning: From activations to transformers. *arXiv preprint arXiv:2302.00722*. 61

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306. 4

Shewalkar, A., Nyavanandi, D., e Ludwig, S. A. (2019). Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, 9(4):235–245. 29

Shumway, R. H. e Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, 4 edition. 13

Siami-Namini, S., Tavakoli, N., e Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International conference on big data (Big Data)*, páginas 3285–3292. IEEE. 4

Simonyan, K. e Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 57

Song, X., Wen, Q., Li, Y., e Sun, L. (2022). Robust time series dissimilarity measure for outlier detection and periodicity detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, páginas 4510–4514. 51

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., e Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 1–9. 57

Taylor, S. J. e Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45. 78

Triebe, O., Laptev, N., e Rajagopal, R. (2019). Ar-net: A simple auto-regressive neural network for time-series. *arXiv preprint arXiv:1911.12436*. 3

Tuli, S., Casale, G., e Jennings, N. R. (2022). Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*. 36

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., e Polosukhin, I. (2017a). Attention is all you need. In *Advances in neural information processing systems*, páginas 5998–6008. xii, 5, 35, 50, 60, 61

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., e Polosukhin, I. (2017b). Attention is all you need. 37, 39, 58

Wang, H., Peng, J., Huang, F., Wang, J., Chen, J., e Xiao, Y. (2022a). Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*. 49

Wang, K. et al. (2019). Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing*, 360:107–119. 4

Wang, Y., Du, X., Lu, Z., Duan, Q., e Wu, J. (2022b). Improved lstm-based time-series anomaly detection in rail transit operation environments. *IEEE Transactions on Industrial Informatics*, 18(12):9027–9036. 3

Wei, H., Zhang, J., Cousseau, F., Ozeki, T., e Amari, S.-i. (2008). Dynamics of learning near singularities in layered networks. *Neural computation*, 20(3):813–843. 63, 64

Wen, Q., He, K., Sun, L., Zhang, Y., Ke, M., e Xu, H. (2021). Robustperiod: Robust time-frequency mining for multiple periodicity detection. In *Proceedings of the 2021 international conference on management of data*, páginas 2328–2337. 36

Wen, Q., Yang, L., Zhou, T., e Sun, L. (2022). Robust time series analysis and applications: An industrial perspective. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, páginas 4836–4837. 36

Wen, R., Torkkola, K., Narayanaswamy, B., e Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*. 46

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. In *Neural Networks*, páginas 339–356. Elsevier. 29

Wibawa, A. P. et al. (2022). Time-series analysis with smoothed convolutional neural network. *Journal of Big Data*, 9(1):44. 4

Wu, D., Wang, Y., Xia, S.-T., Bailey, J., e Ma, X. (2020a). Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*. 58

Wu, H., Xu, J., Wang, J., e Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430. 5, 36, 50, 51, 79

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., e Zhang, C. (2020b). Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, páginas 753–763. 78

Xu, J., Wu, H., Wang, J., e Long, M. (2021a). Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*. 36

Xu, Y., Zhou, Y., Sekula, P., e Ding, L. (2021b). Machine learning in construction: From shallow to deep learning. *Developments in the built environment*, 6:100045. 57

Yang, C.-H. H., Tsai, Y.-Y., e Chen, P.-Y. (2021). Voice2series: Reprogramming acoustic models for time series classification. In *International conference on machine learning*, páginas 11808–11819. PMLR. 36

Yang, H., Wang, Z., Liu, X., Li, C., Xin, J., e Wang, Z. (2023). Deep learning in medical image super resolution: a review. *Applied Intelligence*, páginas 1–26. 5

Yang, S., Yu, X., e Zhou, Y. (2020). Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, páginas 98–101. IEEE. 31, 32

Yao, L. e Guan, Y. (2018). An improved lstm structure for natural language processing. In *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, páginas 565–569. IEEE. 29

Yu, F. e Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*. 46, 48

Zagoruyko, S. e Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*. 5

Zarnowitz, V. (1992). Business cycles: Theory, history, indicators, and forecasting. *University of Chicago Press*. 18

Zeng, A., Chen, M., Zhang, L., e Xu, Q. (2022). Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504*. 53, 75, 79, 104

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., e Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, páginas 2114–2124. 36

Zhang, Q., Yang, L. T., Chen, Z., e Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42:146–157. 3

Zhao, B. et al. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169. 4

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., e Zhang, W. (2020). Informer: Beyond efficient transformer for long sequence time-series forecasting. 4, 5, 22, 23, 50, 79, 91

Zhou, T., Ma, Z., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R., et al. (2022a). Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems*, 35:12677–12690. 6, 52

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., e Jin, R. (2022b). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, páginas 27268–27286. PMLR. 4, 5, 36, 51, 75, 79

Zhou, T., Ye, X., Lu, H., Zheng, X., Qiu, S., Liu, Y., et al. (2022c). Dense convolutional network and its application in medical image analysis. *BioMed Research International*, 2022. 5

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., e Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, páginas 3–11. Springer. 58

Zhu, Y., Imamura, M., Nikovski, D., e Keogh, E. (2019). Introducing time series chains: a new primitive for time series data mining. *Knowledge and information systems*, 60(2):1135–1161. 3

Zhu, Y., Luo, S., Huang, D., Zheng, W., Su, F., e Hou, B. (2023). Drcnn: decomposing residual convolutional neural networks for time series forecasting. *Scientific Reports*, 13(1):15901. 5, 51

Zoph, B., Vasudevan, V., Shlens, J., e Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 8697–8710. 63