Preprint de um artigo publicado no International Journal of Neural Systems, Vol. 31, No. 3 (2021) 2130001 (28 páginas) © World Scientific Publishing Company DOI: 10.1142/S0129065721300011

Uma revisão experimental sobre arquiteturas de aprendizado profundo para previsão de séries temporais*

Pedro Lara-Ben´ítez†
Manuel Carranza-Garc´ÿa Jos
´e C. Riquelme Divisão
de Informática, Universidade de Sevilha, ES-41012 Sevilha,
Espanha E-mail: plbenitez@us.es

www.us.es

Nos últimos anos, as técnicas de aprendizado profundo superaram os modelos tradicionais em muitas tarefas de aprendizado de máquina. As redes neurais profundas têm sido aplicadas com sucesso para resolver problemas de previsão de séries temporais, que é um tópico muito importante na mineração de dados. Eles provaram ser uma solução eficaz dada a sua capacidade de aprender automaticamente as dependências temporais presentes nas séries temporais. No entanto, selecionar o tipo mais conveniente de rede neural profunda e sua parametrização é uma tarefa complexa que requer considerável experiência. Portanto, há necessidade de estudos mais profundos sobre a adequação de todas as arquiteturas existentes para diferentes tarefas de previsão. Neste trabalho, enfrentamos dois desafios principais: uma revisão abrangente dos trabalhos mais recentes usando deep learning para previsão de séries temporais; e um estudo experimental comparando o desempenho das arquiteturas mais populares. A comparação envolve uma análise completa de sete tipos de modelos de aprendizado profundo em termos de precisão e eficiência. Avaliamos os rankings e a distribuição dos resultados obtidos com os modelos propostos sob diversas configurações de arquitetura e hiperparâmetros de treinamento. Os conjuntos de dados utilizados compreendem mais de 50.000 séries temporais divididas em 12 problemas de previsão diferentes. Ao treinar mais de 38.000 modelos nesses dados, fornecemos o mais extenso estudo de deep learning para previsão de séries temporais. Dentre todos os modelos estudados, os resultados mostram que a memória de longo prazo (LSTM) e as redes convolucionais (CNN) são as melhores alternativas, com os LSTMs obtendo as previsões mais precisas. As CNNs alcançam desempenho comparável com menor variabilidade de resultados sob diferentes configurações de parâmetros, além de serem mais eficientes.

Palavras-chave: aprendizagem profunda; previsão; séries temporais; Reveja.

1. Introdução

A previsão de séries temporais (TSF) desempenha um papel fundamental em uma ampla gama de problemas da vida real que possuem um componente temporal. Prever o futuro através da TSF é um tópico de pesquisa essencial em muitos campos, como clima,1 consumo de energia,2 índices financeiros,3 vendas no varejo,4 monitoramento médico,5 anomalia detecção,6 previsão de tráfego,7 etc. As características únicas dos dados de séries temporais, em que as observações têm uma ordem cronológica, muitas vezes tornam sua análise uma tarefa desafiadora. Dada a sua complexidade, o TSF é uma área de suma importância na mineração de dados. Os modelos TSF precisam levar em conta várias questões, como como tendências e variações sazonais das séries e as

ÿAo se referir a este artigo, favor citar a versão publicada: P. Lara-Ben´ÿtez, M. Carranza-Garc´ÿa e JC Riquelme, Uma revisão experimental sobre arquiteturas de aprendizado profundo para previsão de séries temporais. International Journal of Neural Systems 31(03), 2130001 (2021). https://doi.org/10.1142/S0129065721300011 †Autor correspondente

correlação entre os valores observados que estão próximos em tempo. Assim, nas últimas décadas, pesquisadores têm colocado seus esforços no desenvolvimento de modelos que podem capturar os padrões subjacentes de séries temporais, para que possam ser extrapoladas para o futuro de forma eficaz.

Nos últimos tempos, o uso de deep learning (DL) técnicas tornou-se a abordagem mais popular para muitos problemas de aprendizado de máquina, incluindo TSF.8 Ao contrário dos modelos clássicos baseados em estatísticas que só pode modelar relacionamentos lineares em dados, redes neurais têm mostrado um grande potencial para mapear interações de recursos não lineares complexos.9 Os sistemas neurais modernos baseiam seu sucesso em suas profundas estrutura, empilhando várias camadas e conectando densamente um grande número de neurônios. O aumento de capacidade computacional durante os últimos anos tem permitido criando modelos mais profundos, o que melhorou significativamente sua capacidade de aprendizado em comparação com redes. Portanto, técnicas de aprendizado profundo podem ser entendido como uma tarefa de otimização em larga escala: semelhante a um problema fácil em termos de formulação, mas complexo devido ao seu tamanho. Além disso, sua capacidade de adaptar-se diretamente aos dados sem nenhuma suposição prévia oferece vantagens significativas ao lidar com poucas informações sobre a série temporal.10 Com Com a crescente disponibilidade de dados, arquiteturas de aprendizado profundo mais sofisticadas foram propostas com melhorias substanciais nos desempenhos de previsão.11 No entanto, há a necessidade mais do que nunca para trabalhos que fornecem uma análise abrangente de a literatura da TSF, a fim de compreender melhor a avanços científicos na área.

Neste trabalho, uma revisão completa dos profundos técnicas de aprendizagem para TSF é fornecida. As revisões existentes se concentraram apenas em um tipo específico de arquitetura de aprendizado profundo12 ou em dados específicos cenário.13 Portanto, neste estudo, pretendemos preencher essa lacuna, fornecendo uma análise mais completa aplicações bem sucedidas de DL para TSF. A revisão da literatura inclui estudos de diferentes TSF domínios considerando todas as arquiteturas DL mais populares (perceptron multicamada, recorrente e convolucional). Além disso, também fornecemos uma comparação experimental minuciosa entre essas arquiteturas. Estudamos o desempenho de sete tipos de Modelos DL: perceptron multicamadas, Elman recorrente, memória de curto prazo longa, estado de eco, recorrente fechado rede convolucional unitária, convolucional e temporal funciona. Para avaliar esses modelos, usamos 12 conjuntos de dados publicamente disponíveis de diferentes campos, como como finanças, energia, tráfego ou turismo. Nós comparamos estes modelos em termos de precisão e eficiência, analisando a distribuição dos resultados obtidos com diferentes configurações de hiperparâmetros. Um total de 6432 modelos foram testados para cada conjunto de dados, cobrindo uma grande variedade de arquiteturas e treinamento possíveis parâmetros.

Uma vez que as novas abordagens de EAD na literatura são muitas vezes comparado a modelos clássicos em vez de outros técnicas DL, este estudo experimental visa fornecer um benchmark geral e confiável que pode ser usado para comparação em estudos futuros. Para o melhor de do nosso conhecimento, este trabalho é o primeiro a avaliar o desempenho de todos os tipos mais relevantes de redes neurais profundas em um grande número de problemas de TSF de domínios diferentes. Nosso objetivo neste trabalho é avaliar redes DL padrão que podem ser aplicável a tarefas gerais de previsão, sem considerar arquiteturas refinadas projetadas para um problema. As arquiteturas propostas usadas para comparação são independentes de domínio, com a intenção de fornecer diretrizes gerais para pesquisadores sobre como abordar problemas de previsão.

Em resumo, as principais contribuições deste artigo são as seguintes:

- Uma revisão exaustiva atualizada sobre os mais técnicas DL relevantes para TSF de acordo com estudos recentes.
- Uma análise comparativa que avalia a desempenho de várias arquiteturas DL em um grande número de conjuntos de dados de natureza diferente.
- Uma estrutura de aprendizado profundo de código aberto para TSF que implementa os modelos propostos.

O resto do artigo está estruturado da seguinte forma: A Seção 2 fornece uma revisão abrangente da literatura existente sobre aprendizagem profunda para TSF; na seção 3, os materiais utilizados e os métodos propostos para o estudo experimental são descritos; A seção 4 relata e discute os resultados obtidos; Seção 5 apresenta as conclusões e possíveis trabalhos futuros.

2. Arquiteturas de aprendizado profundo para o tempo previsão de série

Uma série temporal é uma sequência de observações em ordem cronológica que são registradas em intervalos fixos.

de tempo. O problema de previsão refere-se ao ajuste
um modelo para prever valores futuros da série considerando os
valores passados (o que é conhecido como defasagem).
Seja X = {x1, x2, ..., xT } os dados históricos de
uma série temporal e H o horizonte de previsão desejado, a tarefa
é prever os próximos valores do
{xT +1, ..., xˆT } xT + H}. Sendo Xˆ = {xˆ1, xˆ2, ..., série
o vetor de valores previstos, o objetivo é minimizar o erro de previsão
da seguinte forma:.

$$E = \sum_{i=1}^{h} | xT + i - xˆT + i | \quad (1)$$

As séries temporais podem ser divididas em univariadas ou
multivariada dependendo do número de variáveis
a cada passo de tempo. Neste trabalho, tratamos apenas de
análise de séries temporais univariadas, com uma única observação
registrada sequencialmente ao longo do tempo. Ao longo da última
décadas, as técnicas de inteligência artificial (IA)
aumentaram sua popularidade para abordar problemas de TSF,
com métodos estatísticos tradicionais sendo considerados como
linhas de base para comparação de desempenho em
novos estudos.

Antes do surgimento das técnicas de mineração de dados, o
métodos tradicionais usados para TSF foram baseados principalmente
em modelos estatísticos, como suavização exponencial
(ETS)14 e métodos Box-Jenkins como ARIMA.15
Esses modelos baseiam-se na construção de funções lineares a partir de
observações passadas recentes para fornecer previsões futuras, e
têm sido amplamente utilizados para a previsão
tarefas ao longo das últimas décadas.16 No entanto, esses métodos
estatísticos muitas vezes falham quando são aplicados diretamente,
sem considerar os requisitos teóricos de estacionariedade e
ergodicidade, bem como a
pré-processamento necessário. Por exemplo, em um ARIMA
modelo, a série temporal deve ser transformada em
estacionário (sem tendências ou sazonalidade) através
várias transformações diferenciais. Outro problema
é que a aplicabilidade desses modelos é limitada
a situações em que dados históricos suficientes, com
estrutura explicável, está disponível. Quando os dados
disponibilidade para uma única série é escassa, esses modelos
muitas vezes não conseguem extrair de forma eficaz as
características e padrões subjacentes. Além disso, como esses
métodos criam um modelo para cada série temporal individual, é
não é possível compartilhar o aprendizado entre instâncias
semelhantes. Portanto, não é viável usar essas técnicas para lidar
com grandes quantidades de dados, pois
seria computacionalmente proibitivo. Assim, artificio
técnicas de inteligência social (IA), que permitem construir
modelos globais para prever em várias séries relacionadas,
começou a ganhar popularidade. Enquanto isso, os métodos lineares
começaram a ser considerados como linhas de base para o desempenho
comparação em novos estudos.

Ref. 17 fornece uma pesquisa sobre mineração de dados precoce
técnicas para TSF e sua comparação com os clássicos
abordagens. Dentre todos os métodos de mineração de dados
propostos na literatura, os modelos que têm atraído
mais atenção têm sido aqueles baseados em
redes neurais (RNAs). Eles mostraram melhor
desempenho do que os métodos estatísticos em muitas situações,
especialmente devido à sua capacidade e flexibilidade para mapear
relações não lineares a partir de dados fornecidos
sua estrutura profunda.18 Outra vantagem das RNAs
é que eles podem extrair padrões temporais automaticamente sem
quaisquer suposições teóricas sobre
a distribuição de dados, reduzindo os esforços de pré-processamento.
Além disso, a capacidade de generalização das RNAs
permite explorar informações entre séries. Ao usar RNAs, um único
modelo global que aprende com
múltiplas séries temporais relacionadas podem ser construídas, o que pode
melhorar muito o desempenho da previsão.

No entanto, os pesquisadores têm se esforçado para desenvolver
topologias de rede ótimas e algoritmos de aprendizado,
devido às infinitas possibilidades de configurações de arquitetura
que as RNAs permitem. Partindo de propostas muito básicas do
Multi-Layer Perceptron (MLP), uma grande
vários estudos propuseram arquiteturas cada vez mais sofisticadas,
como redes recorrentes ou convolucionais, que melhoraram o
desempenho
para TSF. Neste trabalho, revisamos os mais relevantes
tipos de redes DL de acordo com a literatura existente, que podem
ser divididas em três categorias:

- Redes neurais totalmente conectadas.
  - MLP: perceptron multicamadas.
- Redes neurais recorrentes.
  - ERNN: Rede neural recorrente de Elman.
  - LSTM: Rede de memória de curto prazo longa.
  - ESN: Rede de estado de eco.
  - GRU: Rede de unidades recorrentes fechadas.
- Redes convolucionais.
  - CNN: Rede neural convolucional.
  - TCN: Rede convolucional temporal.

As variantes arquitetônicas e os campos em
quais essas redes DL foram aplicadas com sucesso serão discutidas
nas seções a seguir.

2.1. Perceptron de várias camadas

O conceito de RNAs foi introduzido pela primeira vez na Ref. 27, e foi inspirado no funcionamento do cérebro com neurônios atuando em paralelo para o processamento de dados. Multi Layer Perceptron (MLP) é o tipo mais básico de Rede neural artificial feed-forward. Sua arquitetura é composta por uma estrutura de três blocos: uma camada de entrada, camadas ocultas e uma camada de saída. Lá pode ser uma ou várias camadas ocultas, que determinam a profundidade da rede. É o aumento da profundidade, a inclusão de camadas mais escondidas, o que torna uma rede MLP um modelo de aprendizado profundo. Cada camada contém um conjunto definido de neurônios que têm conexões associadas a parâmetros treináveis. O algoritmo de aprendizado da rede neural atualiza os pesos dessas conexões para mapear o relação entrada/saída. As redes MLP só têm conexões diretas entre os neurônios, ao contrário de outros arquiteturas que têm loops de feedback. Os estudos mais relevantes utilizando MLPs são apresentados a seguir.

No início dos anos 90, pesquisadores começaram a RNAs como uma alternativa promissora em relação aos modelos estatísticos tradicionais. Esses estudos iniciais propunham o uso de redes MLP com um ou poucos camadas ocultas. Naquela época, devido à falta de metodologias sistemáticas para construir RNAs e sua difícil interpretação, os pesquisadores ainda estavam céticos sobre sua aplicabilidade à TSF.

No final da década, diversos trabalhos revisados a literatura existente com o objetivo de esclarecer a pontos fortes e deficiências das RNAs.28 Essas revisões concordaram sobre o potencial das RNAs para previsão devido às suas características únicas como aproximadores de funções universais e sua flexibilidade para se adaptar aos dados sem suposições prévias. No entanto, todos eles alegaram que uma validação mais rigorosa procedimentos foram necessários para concluir, em geral, que As RNAs melhoram o desempenho das alternativas clássicas. Além disso, outro raciocínio geral foi que determinar a estrutura ótima e os hiperparâmetros das RNAs foi um processo difícil que teve uma influência fundamental no desempenho. Daquele tempo em diante, muitas referências propondo RNAs como um poderoso ferramenta de previsão pode ser encontrada na literatura.

A Tabela 1 apresenta uma coleção de estudos usando MLPs para diferentes problemas de previsão de séries temporais. A maioria dos estudos se concentrou no desenho da rede, concluindo que o parâmetro de história pregressa tem uma influência maior na precisão do que o número de camadas ocultas. Além disso, esses trabalhos provaram a importância das etapas de pré-processamento, uma vez que modelos simples com variáveis de entrada cuidadosamente selecionadas tendem a obter melhor desempenho. No entanto, o estudos mais recentes propõem o uso de conjuntos de Redes MLP para obter maior precisão.

Embora as redes neurais feed-forward, como

Tabela 1. Estudos relevantes sobre previsão de séries temporais utilizando redes MLP.

Ref.	Ano	Técnica	Supera	Inscrição	Descrição/descobertas
16	2003	Híbrido ARIMA-MLP	ARIMA e MLP individualmente	Taxas de câmbio e dados ambientais	O componente ARIMA modela a correlação linear estruturas enquanto o MLP trabalha na parte não linear.
19	2006	MLP	Métodos estatísticos	Despesas de turismo	Etapas de pré-processamento, como eliminação de tendências e dessesonalização, foram essenciais. MLPs tiveram melhor desempenho medida que o horizonte de previsão aumenta.
20	2007	48 MLPs com entradas diferentes	Sem comparação	Série temporal trimestral competição M3	A preparação de dados foi mais importante do que otimizar o número de nós. Selecionou cuidadosamente a entrada variáveis e modelos simples projetados.
21	2009	MLP	ARIMA	Seis conjuntos de dados de domínios diferentes	Comparação entre previsão iterativa e direta métodos. Melhor desempenho com abordagem direta.
22	2012	MLP	Vencedor da competição	competição NN3	Esquema automático baseado em regressão generalizada redes neurais (GRNNs).
23	2014	MLP	ARIMA	Dados de turismo	Modelos ARIMA foram melhores para previsão de horizonte curto, enquanto RNAs foram melhores para previsões mais longas.
24	2014	Ensemble MLP	Suavização exponencial e previsão ingênua	Vendas no varejo	Avalia diferentes operadores de conjunto (média, mediana e moda). O operador de modo foi o melhor.
25	2016	Conjunto MLP	Métodos estatísticos	NN3 e NN5 competições	Conjunto de duas camadas. A primeira camada encontra um atraso apropriado, e a segunda camada emprega o atraso na previsão.
26	2018	Paralelizado MLPs	Regressão linear, Árvores Impulsionadas por Gradiente, Floresta Aleatória	Demanda de eletricidade	Divida o problema em vários subproblemas de previsão para prever muitas amostras simultaneamente.

MLP têm sido aplicados de forma eficaz em muitas circunstâncias, eles são incapazes de capturar o tempo ordem de uma série temporal, uma vez que tratam cada entrada independentemente. Ignorar a ordem temporal das janelas de entrada restringe o desempenho no TSF, especialmente ao lidar com instâncias de tamanhos diferentes que mudam dinamicamente. Portanto, mais especializado modelos, como redes neurais recorrentes (RNNs) ou redes neurais convolucionais (CNNs), iniciadas para aumentar o interesse. Com essas redes, o problema temporal se transforma em uma arquitetura espacial que pode codificar a dimensão do tempo, capturando assim mais efetivamente os padrões dinâmicos subjacentes de séries temporais.33

2.2. Redes Neurais Recorrentes

As Redes Neurais Recorrentes (RNN) foram introduzidas como uma variante de ANN para dados dependentes do tempo. Enquanto Os MLPs ignoram as relações de tempo nos dados de entrada, os RNNs se conectam a cada passo de tempo com o anteriores para modelar a dependência temporal de os dados, fornecendo suporte nativo RNN para sequência dados.34 A rede vê uma observação de cada vez e pode aprender informações sobre as observações anteriores e quão relevante a observação é para previsão. Por meio desse processo, a rede não só aprende padrões entre entrada e saída, mas também aprende padrões internos entre observações de a sequência. Essa característica torna as RNNs das redes neurais mais comuns usadas para o tempo dados da série. Eles foram implementados com sucesso para aplicações de previsão em diferentes campos, como previsão de preço do mercado de ações,35 previsão de velocidade do vento36 ou previsão de radiação solar.37 Além disso, as RNNs alcançaram os melhores resultados em previsão competições, como a recente competição M4.38

No final dos anos 80, diversos estudos trabalhados em diferentes abordagens para fornecer memória a uma rede neural.39 Essa memória ajudaria o modelo a aprender a partir de dados de séries temporais. Um dos mais promissores propostas foi o Jordan RNN.40 Foi a principal inspiração para criar o Elman RNN, que é conhecido como a base do moderno RNN.34

2.2.1. Redes Neurais Recorrentes Elman

A ERNN visava resolver o problema de lidar com com padrões de tempo nos dados. A ERNN alterou a camada oculta de feed forward para uma camada recorrente, que conecta a saída da camada oculta à entrada de a camada oculta para o próximo passo de tempo. Essa conexão permite que a rede aprenda uma representação compacta da sequência de entrada. Para implementar o algoritmo de otimização, as redes são geralmente desdobrado e o método de retropropagação é modificado, resultando no algoritmo Truncated Backpropagation Through Time (TBPTT). mesa 2 apresenta alguns estudos onde ERNNs foram aplicadas com sucesso para TSF em diferentes campos. De modo geral, os estudos demonstram a melhoria das redes recorrentes em relação às MLPs e modelos lineares como o ARIMA.

Apesar do sucesso da aproximação de Elman, a aplicação do TBTT enfrenta dois problemas principais: os pesos podem começar a oscilar (problema de gradiente explosivo), ou um tempo computacional excessivo para aprender padrões de longo prazo (problema de gradiente de fuga) 41

2.2.2. Redes de memória de longo prazo

Em 1997, redes de Long Short-Term Memory (LSTM)56 foram introduzidas como uma solução para ERNN's problemas. LSTMs são capazes de modelar dependências temporais em horizontes maiores sem esquecer a

Tabela 2. Estudos relevantes sobre previsão de séries temporais usando ERNNs.

Ref.	Ano	Técnica	Supera	Inscrição	Descrição/descobertas
29	2000	ERNN	MLP, RBF, ARIMA	Radiação solar	Apesar dos bons resultados, as ERNNs não conseguiram modelar as descontinuidades das séries temporais. ERNNs tinham taxa de convergência mais lenta do que os modelos não recorrentes.
30	2012	ERNN RBFNN, ARMA-ANN, PREÇO		Série temporal caótica	Compara diferentes algoritmos de treinamento, concluindo que os algoritmos evolucionários levam significativamente mais tempo para convergir do que métodos baseados em gradiente.
31	2018	ERNN	ARIMA, SVR, BPNN, RBFNN	Dados de eletricidade	Desempenho aprimorado com etapas de pré-processamento, como como decomposição de sinal e seleção de características.
32	2018	ERNN	NAR, NAR	Consumo de energia	Relatou uma melhora importante em relação à redes autorregressivas não lineares.

padrões de curto prazo. As redes LSTM diferem de
ERNN na camada oculta, também conhecida como LSTM
célula de memória. As células LSTM usam uma entrada multiplicativa
gate para controlar as unidades de memória, impedindo-as
de ser modificado por perturbações irrelevantes.
Da mesma forma, uma saída multiplicativa protege outras células
de perturbações armazenadas na memória atual.
Mais tarde, outro trabalho adicionou um portão de esquecimento ao LSTM
célula de memória.57 Essa porta permite que o LSTM aprenda a
redefinir o conteúdo da memória quando ele se tornar irrelevante.

Uma lista de estudos relevantes que abordam problemas de
TSF com redes LSTM pode ser encontrada na Tabela
3. No geral, esses estudos comprovam as vantagens de
LSTM sobre MLP tradicional e ERNN para extração
obter informações significativas de séries temporais. Pelagem
Além disso, alguns estudos recentes propuseram soluções mais
inovadoras, como modelos híbridos, algoritmos genéticos para
otimizar a arquitetura da rede, ou
adicionando uma etapa de clustering para treinar redes LSTM em
várias séries temporais relacionadas.

2.2.3. Redes de estado de eco

Echo State Networks (ESNs) foram introduzidos pela Ref.
62. Eles são baseados em Computação de Reservatório (RC),
que simplifica o procedimento de treinamento de
RNNS. RNNs anteriores, como ERNN ou LSTM,
tem que encontrar os melhores valores para todos os neurônios do
rede. Em contraste, o ESN ajusta apenas os pesos
dos neurônios de saída, o que torna o treinamento
problema uma tarefa de regressão linear simples.63

Tabela 3. Estudos relevantes sobre previsão de séries temporais utilizando redes LSTM.

Ref.	Ano	Técnica	Supera	Inscrição	Descrição/descobertas
42	2015	LSTM	MLP, PREÇO, SVM	Velocidade do trânsito	LSTM provou ser eficaz para previsão de curto prazo sem informação prévia de desfasamento de tempo.
43	2015	LSTM	MLP, Autoencoders	Fluxo de tráfego	O LSTM determina os intervalos de tempo ideais dinamicamente, mostrando maior capacidade de generalização.
44	2018	LSTM	Regressão logística, MLP, RF	Índice S&P 500	Desembaralhe a caixa preta LSTM para encontrar padrões comuns de ações em dados financeiros barulhentos.
45	2018	LSTM	Regressão linear, kNN, RF, MLP	Carga elétrica	Seleção de recursos e algoritmo genético para encontrar atrasos de tempo ideais e número de camadas para o modelo LSTM.
46	2019	LSTM	ARIMA, ERNN, GRU	Petróleo Produção	Deep LSTM usando algoritmos genéticos para otimizar configurar a arquitetura.
47	2019	LSTM + Atenção	LSTM, MLP	Geração solar	Mecanismo de atenção temporal para melhorar o desempenho em relação ao LSTM padrão, também usando correlação au parcial para determinar o atraso de entrada.
48	2020	Híbrido ETS-LSTM	Vencedor da competição	O ETS da competição M4	captura os principais componentes, como a sazonalidade, enquanto as redes LSTM permitem tendências não lineares e aprendizagem cruzada de várias séries relacionadas.
49	2020	Clustering + LSTM	LSTM, ARIMA, ETS	CIF2016 e NN5 competições	Construindo um modelo para cada cluster de séries temporais relacionadas pode melhorar a precisão da previsão, ao mesmo tempo em que reduz tempo de treino.

Tabela 4. Estudos relevantes sobre previsão de séries temporais usando ESNs.

Ref.	Ano	Técnica	Supera	Inscrição	Descrição/descobertas
50	2009	ESN	MLP, RBFNN	Preço do mercado de ações	A aplicação do PCA para filtrar ruídos melhorou o desempenho do ESN em alguns índices.
51	2011	ESN	Outros tipos de reservatórios	7 séries temporais de campos diferentes	Um ESN com uma topologia de reservatório de ciclo simples alcançou alto desempenho para problemas de TSF.
52	2012	ESN com Função Laplace	Vetor de suporte, Gaussiano, e Bayesian ESN	Conjuntos de dados simulados	ESN aplicado em uma estrutura de aprendizagem Bayesiana. o A função de Laplace provou ser mais robusta para outliers que a distribuição gaussiana.
53	2012	ESN	Sem comparação	Carga de eletricidade previsão	A ESN provou sua capacidade de aprender dinâmicas complexas de carga elétrica, obtendo resultados de alta precisão sem insumos adicionais.
54	2015	Otimizado para GA ESN	ARIMA, Generalizado Regressão NN	Velocidade do vento	ESN para prever várias velocidades do vento simultaneamente. PCA e clustering espectral para pré-processar dados e algoritmo genético para buscar parâmetros ótimos.
55	2020	ESN ensacado	BPNN, RNN, LSTM	Consumo de energia	Combina ESN, ensacamento e algoritmo de evolução diferencial para reduzir o erro de previsão e melhorar a capacidade de generalização.

Um ESN é uma rede neural com um RNN aleatório chamado de reservatório como a camada oculta. O reservatório geralmente possui um número elevado de neurônios e interconectividade (cerca de 1%) entre os neurônios. Essas características tornam o reservatório um conjunto de subsistemas que funcionam como funções de eco, sendo capaz de reproduzir padrões temporais específicos.64

Na literatura, muitas aplicações ESN para TSF pode ser encontrado. Especialmente, as redes ESN têm provou superar MLPs e métodos estatísticos ao modelar dados caóticos de séries temporais. Além disso, vale ressaltar que os não treináveis os neurônios do reservatório tornam esta rede um modelo muito eficiente em comparação com outras RNNs. Tabela 4 apresenta diversos estudos utilizando ESNs e seus descobertas interessantes.

2.2.4. Unidades recorrentes fechadas

As Unidades Recorrentes Fechadas (GRU) foram introduzidas por Ref. 65 como outra solução para o gradiente explosivo e problema de gradiente de fuga de ERNNs. No entanto, também pode ser visto como uma simplificação das unidades LSTM. Em uma unidade GRU, o esquecimento e a entrada são combinados em um único portão de atualização. Esta modificação reduz os parâmetros treináveis que fornecem redes GRU com um melhor desempenho em termos de computação tempo enquanto alcança resultados semelhantes.66

A célula GRU usa um portão de atualização e um portão de redefinição que aprenderá a decidir quais informações deve ser mantido sem esvaziá-lo ao longo do tempo e quais informações são irrelevantes para o problema. o portão de atualização é responsável por decidir quanto do informações passadas devem ser passadas para o futuro, enquanto o portão de reinicialização decide quanto do informações passadas para esquecer.

A Tabela 5 apresenta vários estudos que utilizam GRU redes para problemas de TSF. Esses trabalhos muitas vezes propõem modificações no modelo GRU padrão em para melhorar o desempenho em relação a outros redes. No entanto, o número de estudos existentes propor modelos GRU é muito menor do que aqueles que utilizam redes LSTM.

2.3. Redes Neurais Convolucionais

CNNs são uma família de arquiteturas profundas que foram originalmente projetado para tarefas de visão computacional. Elas são considerados de última geração para muitas classificações tarefas como reconhecimento de objetos,78 reconhecimento de fala79 e processamento de linguagem natural.80 CNNs podem extrair automaticamente recursos de alta dimensão dados brutos com uma topologia de grade, como os pixels de uma imagem, sem a necessidade de qualquer engenharia de recursos. O modelo aprende a extrair recursos significativos dos dados brutos usando a operação convolucional, que é um filtro deslizante que cria mapas de recursos e visa capturar padrões repetidos em diferentes regiões dos dados. Este processo de extração de recursos fornece CNNs com uma característica importante chamada invariância de distorção, o que significa que os recursos são extraídos independentemente de onde eles estão nos dados. Essas características tornam as CNNs adequadas para lidar com com dados unidimensionais, como séries temporais. Se os dados de sequência podem ser vistos como uma imagem unidimensional de onde a operação convolucional pode extrair recursos.

Uma arquitetura CNN é geralmente composta por camadas de convolução, camadas de agrupamento (para reduzir o espaço dimensão dos mapas de recursos) e camadas totalmente conectadas (para combinar recursos locais em recursos globais). Essas redes são baseadas em três princípios: conectividade, pesos compartilhados e equivalência de tradução. Ao contrário das redes MLP padrão, cada nó

Tabela 5. Estudos relevantes sobre previsão de séries temporais utilizando redes GRU.

Ref.	Ano	Técnica	Supera	Inscrição	Descrição/descobertas
58	2014	GRU	ERNN	Música e fala modelagem de sinal	Os resultados comprovaram as vantagens do GRU e do LSTM redes sobre ERNN, mas não mostrou um diferença entre as duas unidades fechadas.
59	2017	GRU	LSTM, GRU	Carga de eletricidade	Unidades lineares exponenciais escaladas propostas para superar gradientes de fuga, mostrando uma melhoria significativa em relação aos modelos LSTM e GRU padrão.
60	2018	GRU	LSTM, SVM, ARIMA	O coeficiente de Pearson de energia fotovoltaica é usado para extrair as principais	características e K-means para agrupar grupos semelhantes que treinam o modelo GRU.
61	2018	GRU	MLP, CNN, LSTM	Preço da eletricidade	GRU teve melhor desempenho e treinou mais rápido que LSTM redes. A utilização de um número maior de valores defasados melhorou os resultados.

Tabela 6. Estudos relevantes sobre previsão de séries temporais usando CNNs.

Ref.	Ano	Técnica	Supera	Inscrição	Descrição/descobertas
67	2017	CNN	SVM, MLP	Preço das ações	Os dados foram pré-processados usando normalização. O modelo obteve melhor desempenho nas previsões de curto prazo.
68	2018	CNN	LSTM, MLP	Carga de energia	O modelo usou convolução e agrupamento para prever a carga para os próximos três dias. Mostrou-se útil para reduzindo despesas em futuras redes inteligentes.
69	2018	CNN	LSTM	Energia solar e eletricidade	A CNN foi significativamente mais rápida do que a recorrente abordagem com precisão semelhante, portanto, mais adequado para aplicações práticas.
70	2018	Híbrido CNN-LSTM	RNN e LSTM individualmente	Eletricidade	O LSTM é capaz de extrair as dependências de longo prazo e a CNN captura padrões de tendências locais.
71	2018	Conjunto CNN-LSTM	ARIMA, ERNN, RBF	Velocidade do vento	Os dados são decompostos usando o pacote wavelet, com uma CNN prevendo os dados de alta frequência e um CNN-LSTM para os dados de baixa frequência.
72	2019	CNN	RN, ARIMAX	Carga de construção	O modelo CNN com abordagem direta proporcionou a melhores resultados, melhorando significativamente a previsão precisão do modelo ARIMAX sazonal.
73	2019	Híbrido CNN-LSTM	LSTM, CNN	Dados financeiros	O LSTM aprende características e reduz a dimensionalidade, e a CNN dilatada aprende diferentes intervalos de tempo.

está conectado apenas a uma região da entrada, que é conhecido como campo receptivo. Além disso, os neurônios nas mesmas camadas compartilham a mesma matriz de peso para a convolução, que é um filtro com um tamanho de kernel definido. Essas propriedades especiais permitem que as CNNs tenham um número muito menor de parâmetros treináveis em comparação com um RNN, portanto, o processo de aprendizado é mais eficiente em termos de tempo.74 Outro aspecto fundamental do sucesso da CNNs é a possibilidade de empilhar diferentes camadas convolucionais para que o modelo de deep learning possa transformar os dados brutos em uma representação efetiva da série temporal em diferentes escalas.81

Modelos baseados em CNN não foram extensivamente usado na literatura TSF desde que os RNNs foram dado muito mais importância. Apesar disso, vários trabalhos propuseram CNNs como extratores de recursos sozinho ou em conjunto com blocos recorrentes para fornecer previsões. Os trabalhos mais relevantes envolvendo esses propostas estão detalhadas na Tabela 6.

2.3.1. Rede Convolucional Temporal

Estudos recentes propuseram uma abordagem mais especializada Arquitetura CNN conhecida como Convolucional Temporal Rede (TCN). Essa arquitetura é inspirada o modelo autoregressivo Wavenet, que foi projetado para problemas de geração de áudio.82 O termo O TCN foi apresentado pela primeira vez na Ref. 83 para se referir a um tipo de CNN com características especiais: as convoluções são causais para evitar a perda de informação, e a arquitetura pode processar uma sequência de qualquer comprimento e mapeá-lo para uma saída do mesmo comprimento. Para permitir que a rede aprenda as dependências de longo prazo presente em séries temporais, a arquitetura TCN torna uso de convoluções causais dilatadas. Esta convolução aumenta o campo receptivo da rede (neurônios que são convolvidos com o filtro) sem perder a resolução já que a operação de pooling não é necessária.84 Além disso, o TCN emprega conexões residuais para permitir aumentar a profundidade da rede, de modo que

Tabela 7. Estudos relevantes sobre previsão de séries temporais usando TCNs.

Ref.	Ano	Técnica	Supera	Dados	Descrição/descobertas
74	2019	TCN	LSTM	financeiros do aplicativo	O TCN pode aprender efetivamente as dependências e interpolar séries sem a necessidade de dados históricos longos.
75	2019	Codificador-decodificador TCN	RN	Vendas no varejo	Combinado com o aprendizado de representação, o TCN pode aprender padrões complexos, como sazonalidade. Os TCNs são eficiente devido à paralelização das circunvoluções.
76	2019	TCN	LSTM, ConvLSTM	Meteorologia	O TCN apresentou maior eficiência e capacidade de generalização, tendo melhor desempenho em previsões mais longas.
77	2020	TCN	LSTM	Os modelos TCN de demanda de energia	mostraram uma maior capacidade de lidar com sequências de entrada mais longas. Os TCNs foram menos sensíveis a a parametrização do que as redes LSTM.

ele pode lidar efetivamente com um grande tamanho de histórico. Dentro Ref. 83 os autores apresentam uma comparação experimental entre RNNs genéricos (LSTM e GRU)
e TCNs em várias tarefas de modelagem de sequência. Esse trabalho enfatiza as vantagens dos TCNs como: seus baixos requisitos de memória para treinamento devido a os filtros convolucionais compartilhados; longas sequências de entrada pode ser processado com circunvoluções paralelas em vez de sequencialmente como em RNNs; e um treinamento mais estável esquema, evitando assim problemas de gradiente de fuga.

Os TCNs estão adquirindo popularidade crescente nos últimos anos devido à sua adequação para lidar com dados temporais. A Tabela 7 apresenta alguns estudos onde As TCNs se inscreveram com sucesso para a previsão de séries temporais.

3. Materiais e métodos

Nesta seção, apresentamos os conjuntos de dados usados para o estudo experimental e o detalhamento das arquiteturas e configurações dos hiperparâmetros estudados. O código fonte dos experimentos pode ser encontrado na Ref. 85. Este repositório fornece um aprendizado profundo framework baseado em TensorFlow para TSF, permitindo reprodutibilidade total dos experimentos.

Tabela 8. Conjuntos de dados utilizados para o estudo experimental. As colunas N, FH, M e m referem-se ao número de séries temporais, horizonte de previsão, comprimento máximo e comprimento mínimo respectivamente.

#	Conjuntos de dados	N	FH	M	m
1	CIF2016o12	57	12	108	48
2	CIF2016o6	15	6	69	22
3	Taxa de câmbio	8	6	7588	7588
4	M3	1428	18	126	48
5	M4	48000	18	2794	111
6	NN5	735	735	56	42
7	Energia solar	137	6	52560	52560
8	Turismo	336	24	309	67
9	Tráfego	862	24	17544	17544
10	Tráfego-metr-la	207	12	34272	34272
11	Baia de permissão de tráfego	325	12	52116	52116
12	WikiWebTraffic	997	59	550	550

3.1. Conjuntos de dados

Neste estudo, selecionamos 12 disponíveis publicamente conjuntos de dados de natureza diferente, cada um deles com várias séries temporais relacionadas. Esta coleção representa uma grande variedade de problemas de previsão de séries temporais, cobrindo diferentes tamanhos, domínios, comprimentos de séries temporais e horizonte de previsão. A Tabela 8 apresenta as características de cada conjunto de dados em detalhes. No total, o estudo experimental envolve mais de 50.000 séries temporais entre todos os conjuntos de dados selecionados. Além disso, a Figura 1 ilustra alguns exemplos de instâncias de cada conjunto de dados. Como pode ser visto, os conjuntos de dados selecionados apresentam uma ampla diversidade de características em termos de escala e sazonalidade. A maioria desses conjuntos de dados foi usada em competições de previsão e outras revisões da TSF, como como Ref. 12.

O conjunto de dados da competição CIF 201686 contém um total de 72 séries mensais, das quais 12 eles têm um horizonte de previsão de 6 meses, enquanto o as 57 séries restantes têm um horizonte de previsão de 12 meses. Algumas dessas séries temporais são análises reais de risco bancário indicadores ysis enquanto outros são gerados artificialmente.

O conjunto de dados ExchangeRate87 registra o taxas de câmbio de 8 países, incluindo Austrália, Britânico, Canadá, Suíça, China, Japão, Novo Zelândia e Cingapura de 1990 a 2016 (um total de 7588 dias para cada país). O objetivo deste conjunto de dados é prever valores para os próximos 6 dias.

Ambas as competições M3 e M488, 89 conjuntos de dados incluir séries temporais de diferentes domínios e observar frequências de vação. No entanto, devido à computação restrições de tempo, limitamos a experimentação às séries mensais, pois elas são mais longas do que as séries temporais anuais, trimestrais, semanais e diárias. Ambas as competições pedem uma previsão de 18 meses e contêm séries temporais de diferentes categorias como indústria, finanças ou demografia. O número de instâncias pertencentes a cada categoria são uniformemente distribuídos de acordo com sua presença no mundo, levando novos estudos a conclusões representativas. Considerando apenas a série temporal mensal, o conjunto de dados M4 tem 48.000 séries temporais, enquanto o M3 é consideravelmente menor com 1428 séries temporais.

O conjunto de dados de competição NN590 é formado por 111 séries temporais com um comprimento de 735 valores. Esse conjunto de dados representa 2 anos de saques diários em dinheiro em caixas automáticos (ATMs) da Inglaterra. A competição estabeleceu um horizonte de previsão de 56 dias à frente.

O conjunto de dados de Turismo91 é composto por 336 séries mensais de duração diferente, exigindo um previsão de 24 meses. Os dados foram fornecidos por órgãos de turismo da Austrália, Hong Kong e Nova Zelândia, representando números totais de turismo em um nível do país.

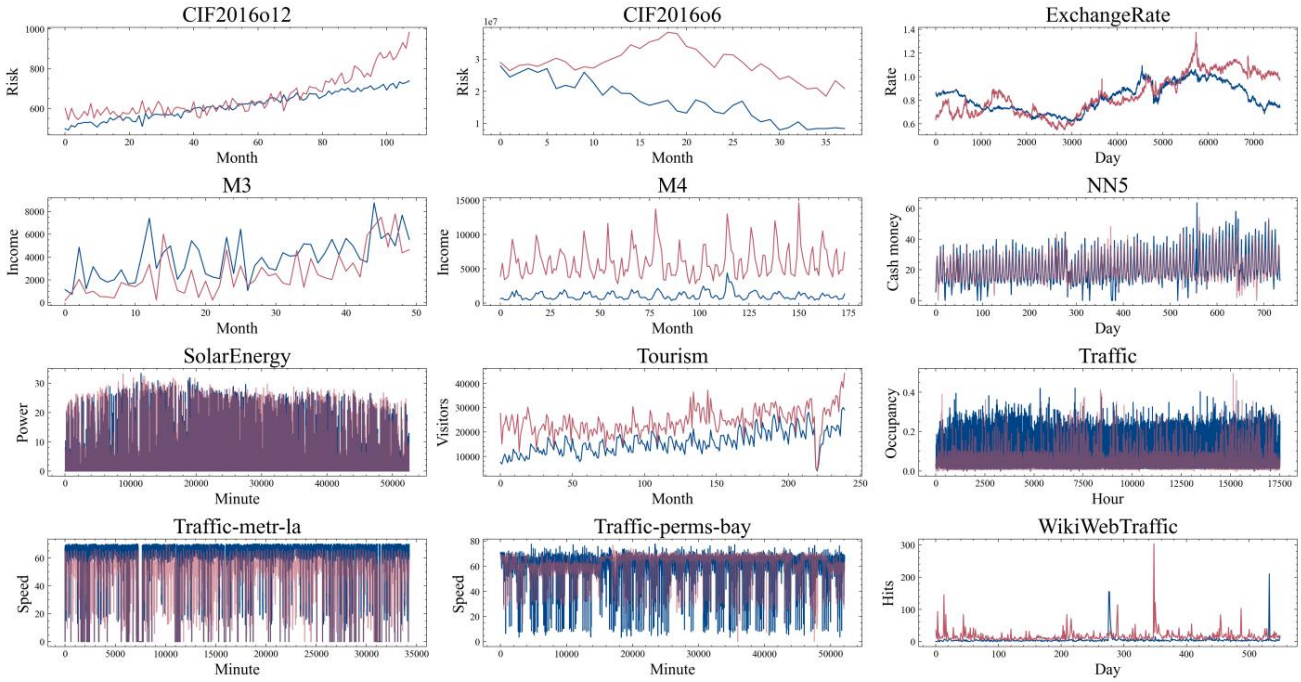

Figura 1. Dois exemplos de instâncias de séries temporais de cada conjunto de dados. As cores diferenciam as duas instâncias. O eixo y representa o valor que a série temporal assume para cada passo de tempo ao longo do eixo x.

O conjunto de dados SolarEnergy92 contém a energia solar recordes de produção de energia no ano de 2006, que é amostrado a cada 10 minutos de 137 usinas de energia solar fotovoltaica no estado do Alabama. O horizonte de previsão foi estabelecido para uma hora, o que compreende 6 observações.

Traffic-metr-la e Traffic-perms-bay são dois conjuntos de dados de rede de tráfego público.93 Traffic-metr-la contém informações de velocidade de tráfego de 207 sensores no rodovias de Los Angeles por quatro meses. De forma similar, Traffic-perms-bay registra seis meses de estatísticas sobre velocidade do tráfego de 325 sensores na área da baía. Por Em ambos os conjuntos de dados, as leituras dos sensores são agregadas em janelas de cinco minutos.

O conjunto de dados de tráfego é uma coleção de horas série do Departamento de Transporte da Califórnia coletada durante 2015 e 2016. Os dados descrevem as taxas de ocupação das estradas (entre 0 e 1) medido por diferentes sensores na Baía de São Francisco rodovias da área.

O conjunto de dados WikiWebTraffic pertence a uma competição Kag gle94 com o objetivo de prever tráfego da web de um determinado conjunto de páginas da Wikipedia. o o tráfego é medido em número diário de acessos e o horizonte de previsão é de 59 dias.

3.2. Configuração experimental

Esta subseção apresenta o estudo comparativo realizado para avaliar o desempenho de sete arquiteturas de aprendizado profundo de última geração para séries temporais previsão. Explicamos as arquiteturas e configurações de hiperparâmetros que exploramos, juntamente com os detalhes do procedimento de avaliação.

O estudo experimental é baseado em uma análise estatística com os resultados obtidos com 6432 configurações de arquitetura diferentes em 12 conjuntos de dados, resultando em mais de 38.000 modelos testados.

3.2.1. Arquiteturas de aprendizado profundo

Sete tipos diferentes de modelos de aprendizado profundo para TSF são comparados neste estudo experimental: perceptron multicamada, Elman recorrente, longo curto prazo memória, estado de eco, unidade recorrente fechada, redes convolucionais e convolucionais temporais. Para todos desses modelos, o número de hiperparâmetros que tem que ser configurado é alto comparado ao tradicional técnicas de aprendizado de máquina. Portanto, o adequado o ajuste desses parâmetros é uma tarefa complexa que requer uma experiência considerável e geralmente é conduzido pela intuição. Neste trabalho, realizamos uma pesquisa exaustiva em grade sobre a configuração de cada tipo

de arquitetura para encontrar o valor mais adequado
s. A Tabela 9 apresenta a busca realizada ao longo do
principais parâmetros dos modelos de aprendizado profundo, como
o número de camadas ou unidades. A grade de possibilidades
foi decidido com base em valores típicos encontrados no
literatura. Por exemplo, é uma prática comum
selecione potências de dois para o número de neurônios, unidades,
ou filtros. Tentámos estabelecer uma comparação justa entre
arquitecturas, mantendo os valores
consistente entre os modelos o maior tempo possível. Por
Por exemplo, em todos os casos, exploramos modelos de camada única
até redes mais profundas com quatro camadas empilhadas.

Tabela 9. Grade de parâmetros das configurações de arquitetura para os sete tipos de modelos de deep learning.

Parâmetros de modelos		Valores
Camadas ocultas de MLP		[8], [8, 16], [16, 8], [8, 16, 32], [32, 16, 8], [8, 16, 32, 16, 8], [32], [32, 64], [64, 32], [32, 64, 128], [128, 64, 32], [32, 64, 128, 64, 32]
ERNN	Camadas	1, 2, 4
	Unidades	32, 64, 128
	Sequência de retorno	Verdadeiro falso
LSTM	Camadas	1, 2, 4
	Unidades	32, 64, 128
	Sequência de retorno	Verdadeiro falso
GRU	Camadas	1, 2, 4
	Unidades	32, 64, 128
	Sequência de retorno	Verdadeiro falso
ESN	Camadas	1, 2, 4
	Unidades	32, 64, 128
	Sequência de retorno	Verdadeiro falso
CNN	Camadas	1, 2, 4
	Filtros	16, 32, 64
	Tamanho da piscina	0, 2
TCN	Camadas	1, 3
	Filtros	32, 64
	Dilatações	[1, 2, 4, 8], [1, 2, 4, 8, 16]
	Tamanho do kernel	3, 6
	Sequência de retorno	Verdadeiro falso

Em primeiro lugar, experimentamos a arquitetura de rede
neural mais simples, o perceptron multicamada.
Os modelos MLP podem ser usados como linha de base para mais
arquiteturas complexas que obtêm um melhor desempenho. Em
particular, implementamos 12 MLP
modelos, que diferem no número de camadas ocultas e no número
de unidades em cada camada. As configurações selecionadas podem
ser vistas na Tabela 9, onde
os parâmetros das camadas ocultas são definidos como uma lista
$[v1, v2, ..., v_{nós}, ..., v]$. Um valor vi da lista repre
envia o número de unidades na i-ésima camada oculta.
O número de neurônios em cada camada varia de
8 a 128, que visa atender cada vez mais
sequências de entrada de histórico passado. Além disso, o projeto
considera estruturas de codificador e decodificador, com mais
neurônios nas camadas iniciais e menos nas camadas iniciais.
o fim e vice-versa.

No que diz respeito às arquiteturas recorrentes, quatro tipos
diferentes de modelos foram implementados neste
estudo: rede neural recorrente de Elman (ERNN),
memória de longo prazo (LSTM), gated recorrente
unidade (GRU) e rede de estado de eco (ESN). Uma grade
pesquisa nos três parâmetros principais é realizada,
resultando em 18 modelos para cada arquitetura. Esses
parâmetros são o número de camadas recorrentes empilhadas, o
número de unidades recorrentes em cada camada e
se a última camada retorna a sequência de estado ou
apenas o estado final. Se a sequência de retorno for False, o
última camada retorna um valor para cada unidade recorrente.
Em contraste, se a sequência de retorno for True, o
camada retorna o estado de cada unidade para cada passo de tempo,
resultando em uma matriz de forma (passos de tempo de entrada ×
número de unidades). Finalmente, como estamos trabalhando com
um problema de previsão multi-passo à frente, a saída
do bloco recorrente está conectado a uma camada densa
com um neurônio para cada previsão. A Tabela 9 mostra
todos os valores possíveis para cada parâmetro. O número de
unidades varia de 32 a 128, visando explorar
a conveniência de ter mais ou menos células de aprendizagem
em paralelo.

Além disso, 18 modelos convolucionais foram
implementado (CNN). Esses modelos são formados por
blocos convolucionais empilhados, que são compostos de
uma camada convolucional unidimensional seguida por uma
camada de pool máximo. Na Tabela 9, apresentamos o valor
procure cada parâmetro. Os blocos convolucionais
são implementados com tamanhos de kernel decrescentes, pois é
comum na literatura. Os modelos de camada única têm um
kernel de tamanho 3, modelos de duas camadas têm tamanhos de kernel
de 5 e 3, e os modelos de quatro camadas têm núcleos de
tamanho 7-5-3 e 3. Como as sequências de entrada no
conjuntos de dados estudados não são excessivamente longos
(variando aproximadamente de 20 a 300 passos de tempo), temos apenas
considerado um fator de agrupamento de 2.

Por fim, também experimentamos o
arquitetura de rede convolucional temporal (TCN).
Os modelos TCN são definidos principalmente por cinco parâmetros:
número de camadas TCN, número de filtros convolucionais, fatores
de dilatação, tamanho do kernel convolucional e
se deve retornar a última saída ou a sequência completa. Uma
pesquisa de grade sobre esses parâmetros foi
feito com os valores especificados na Tabela 9, resultando

em 32 arquiteturas TCN diferentes. O número de
dilatações e os tamanhos do kernel foram selecionados ac
de acordo com o campo receptivo do TCN, que segue a fórmula
(número de camadas×tamanho do kernel×
última dilatação). Com a grade selecionada cobrimos campos
receptivos que variam de 24 a 288, o que se adequa ao
comprimento variável das sequências de entrada do estudo
conjuntos de dados.

3.3. Procedimento de avaliação

Para fins de avaliação, dividimos o
conjuntos de dados em conjuntos de treinamento e teste. Seguimos
um esquema de teste de origem fixa, que é o mesmo
procedimento de divisão como em trabalhos recentes que tratam
conjuntos de dados contendo várias séries temporais relacionadas.12
O conjunto de teste consiste na última parte de cada série temporal
individual dentro do conjunto de dados, daí o comprimento
é igual ao horizonte de previsão. Consequentemente, o
parte restante da série temporal é usada como treinamento
dados. Esta divisão foi aplicada igualmente a todos os
conjuntos de dados, obtendo um total de mais de um milhão
passos de tempo para testes e mais de 210 milhões para
Treinamento. Neste estudo, usamos milhares de séries temporais
com uma ampla variedade de horizontes de previsão ao longo de
mais de 38.000 modelos. Portanto, este procedimento de validação
de origem fixa pode levar a
resultados e descobertas.

Depois de dividir a série temporal em treinamento
e teste, realizamos uma série de etapas de pré-processamento.
Em primeiro lugar, um método de normalização é usado para dimensionar cada
série temporal de dados de treinamento entre 0 e 1, que
ajuda a melhorar a convergência de redes neurais profundas. Em
seguida, transformamos as séries temporais em instâncias de
treinamento para alimentar os modelos. Existem várias técnicas
disponíveis para esse fim:95 a estratégia recursiva, que realiza
previsões de uma etapa
e alimenta o resultado como a última entrada para a próxima
previsão; a estratégia direta, que constrói um modelo
para cada passo de tempo; e a abordagem de múltiplas saídas,
que gera o vetor de horizonte de previsão completo usando apenas
um modelo. Neste estudo, selecionamos a estratégia Multi-Input
Multi-Output (MIMO). Estudos recentes mostram que o MIMO supera

abordagens de saída única porque, ao contrário da estratégia
recursiva, ela não acumula o erro sobre
as previsões. Também é mais eficiente em termos de
tempo de computação do que a estratégia direta, uma vez que
usa um único modelo para cada previsão.95

Seguindo esta abordagem, usamos um movimento
esquema de janela para transformar a série temporal em
instâncias de treinamento que podem alimentar os modelos. Esse
processo desliza uma janela de tamanho fixo, resultando em
uma instância de entrada-saída em cada posição. O profundo
modelos de aprendizagem recebem uma janela de comprimento fixo
(histórico passado) como entrada e tem uma camada densa como
saída com tantos neurônios quanto o horizonte de previsão definido
pelo problema. O comprimento da entrada
depende do parâmetro da história passada que deve
ser decidido. Para a experimentação, estudamos três
valores diferentes do histórico passado, dependendo do horizonte
de previsão (1,25, 2 ou 3 vezes o horizonte de previsão).
A Figura 2 ilustra um exemplo desta técnica com
7 observações como história passada e um horizonte de previsão
de 3 valores.

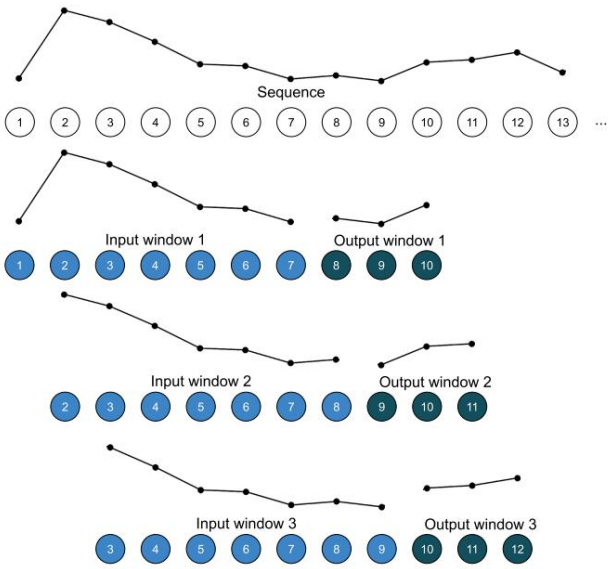

Figura 2. Exemplo de procedimento de janela móvel que
obtém as instâncias de entrada-saída para treinar os modelos.
Neste exemplo, os modelos recebem uma instância com janela de
histórico passado de comprimento 7 como valores de entrada e saída 3
como horizonte de previsão.

Junto com o parâmetro de história passada, temos
também realizou uma pesquisa em grade sobre a configuração de
treinamento ideal dos modelos de aprendizado profundo. Nós
experimentaram várias possibilidades de
tamanho do lote, a taxa de aprendizado e a normalização
método a ser utilizado. Selecionamos comumente usados
valores para o tamanho do lote (32 e 64) e aprendizado
taxa (0,001 e 0,01). O otimizador Adam foi
escolhido, que implementa um método de otimização estocástica
adaptativa que provou ser robusto

e adequado para uma ampla gama de aprendizado de máquina problemas.96 Além disso, os dois mais comuns nem funções de malização na literatura foram selecionadas para pré-processar os dados: min-max scaler (Eq. 2) e normalização média, também conhecida como z-score (Eq. 3). A grade completa de parâmetros de treinamento pode ser encontrado na Tabela 10.

$$\text{min-max}(x) = \frac{x \ddot{y} \min(x)}{\max(x) \ddot{y} \min(x)} \quad (2)$$

$$\text{z-score}(x) = \frac{x \ddot{y} \text{média}(x)}{\max(x) \ddot{y} \min(x)} \quad (3)$$

Tabela 10. Grade de parâmetros de treinamento para o Deep modelos de aprendizagem.

Parâmetros	Valores
Histórico passado (1,25, 2,0, 3,0) × Horizonte de previsão	
Tamanho do lote	32, 64
Nº de épocas	5
Otimizador	Adam
Normalização da taxa de aprendizado	0,001, 0,01
	minmax, zscore

3.3.1. Métricas de avaliação

O desempenho dos modelos propostos é avaliado em termos de precisão e eficiência. Analisamos os melhores resultados obtidos com cada tipo de arquitetura, bem como a distribuição dos resultados obtidos com as diferentes configurações de hiperparâmetros. Para avaliar o desempenho preditivo de

todos os modelos usamos a porcentagem absoluta ponderada erro (WAPE). Essa métrica é uma variação da média erro percentual absoluto (MAPE), que é um dos as medidas de precisão de previsão mais usadas devido às suas vantagens de independência de escala e interpretabilidade.97 WAPE é uma alternativa mais adequada para dados intermitentes e de baixo volume. Ele redimensiona o erro dividindo pela média para torná-lo comparável ao longo de séries temporais de escalas variadas.

O WAPE pode ser definido da seguinte forma:

$$WAPE(y, o) = \frac{QUE(o, o)}{\text{média}(y)} = \frac{\text{média}(|y \ddot{y} o|)}{\text{significa (s)}}, \quad (4)$$

onde y e o são dois vetores com os valores real e previsto, respectivamente, que têm comprimento igual para o horizonte de previsão

Além disso, desenvolver modelos eficientes é essencial em TSF, pois muitas aplicações requerem respostas em tempo real. Por isso, estudamos também a média tempos de treinamento e inferência de cada modelo.

3.3.2. Análise estatística

Uma vez que os valores de erro para cada método em cada conjunto de dados foram registrados, uma análise estatística será tratado. Esta análise permite comparar corretamente o desempenho das diferentes arquiteturas de deep learning. Como estamos estudando vários modelos em vários conjuntos de dados, o teste de Friedman é o método recomendado.98 Este método não paramétrico test permite detectar diferenças globais e fornece uma classificação dos diferentes métodos. No caso de obter um valor p abaixo da significância nível (0,05), a hipótese nula (todos os algoritmos são equivalente) pode ser rejeitado, e podemos prosseguir com a análise post hoc. Para isso, usamos O procedimento de Holm-Bonferroni, que realiza comparações em pares entre os modelos. Com isso procedimento n×n, podemos detectar diferenças significativas entre cada par de modelos, o que permite estabelecer um ranking estatístico.

Neste estudo, realizamos a análise estatística sobre diferentes métricas de desempenho para avaliar os sete tipos de modelos de aprendizado profundo de todos perspectivas. Classificamos os modelos de acordo com a melhores resultados em termos de precisão de previsão (WAPE) e eficiência (tempos de treinamento e inferência). Além disso, avaliamos as diferenças estatísticas de o pior desempenho e a média e padrão desvio dos resultados obtidos com cada tipo de arquitetura. Por fim, realizamos um ranking de classificações médias para ver quais modelos são melhores no geral.

Uma análise estatística adicional é realizada usando um teste de postos sinalizados de Wilcoxon pareado, a fim de estudar a diferença estatística entre as configurações de arquitetura estudadas de cada tipo de modelo. Observação que neste caso, comparamos modelos dentro do mesmo tipo de rede de aprendizado profundo, então vamos descobrir quais configurações são ideais para cada um deles. A hipótese nula é que modelos com duas valores para uma configuração específica não são estatisticamente diferente ao nível de significância de $\ddot{y} = 0,05$. Um valor de $p \ddot{y} 0,05$ rejeita a hipótese nula, indicando uma diferença significativa entre os modelos.99

4. Resultados e discussão

Esta seção relata e discute os resultados obtidos dos experimentos realizados. Ele fornece uma análise dos resultados em termos de precisão de previsão e tempo computacional. Os experimentos levaram cerca de quatro meses, usando cinco GPUs NVIDIA TI TAN Xp 12GB em computadores com CPU Intel i7-8700 e uma GPU adicional no serviço de nuvem da Amazon.

4.1. Precisão da previsão

A primeira parte da análise é focada na precisão de previsão obtida para cada modelo, utilizando a métrica WAPE. Na Figura 3, apresentamos uma comparação entre a distribuição dos resultados obtidos com todas as diferentes arquiteturas de modelo para cada conjunto de dados. Em alguns gráficos, a distribuição ERNN foi cortada para permitir uma melhor visualização do restante dos modelos. Isto pode ser visto à primeira vista, que algumas arquiteturas, como o ERNN ou o TCN, são mais sensíveis a

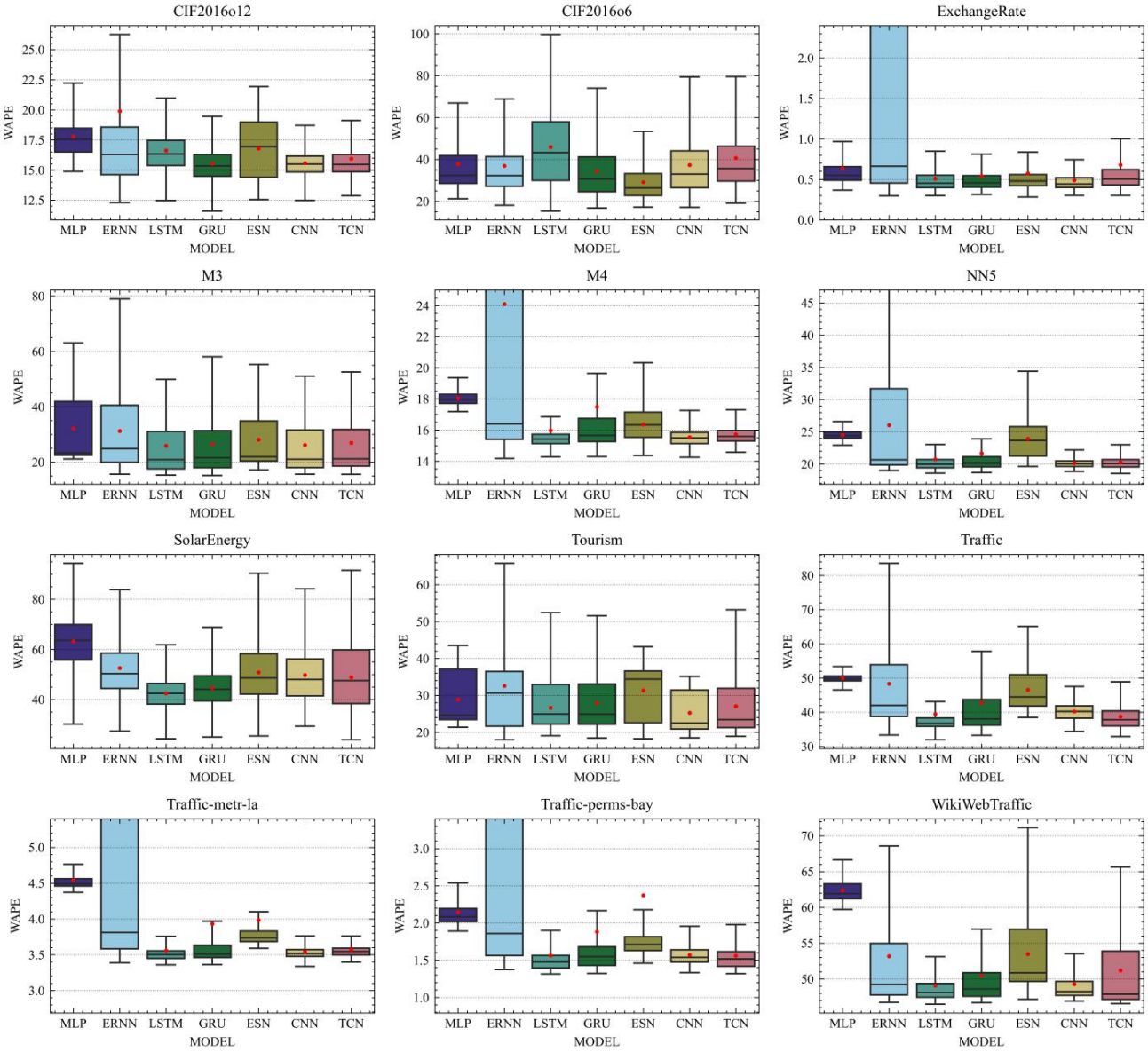

Figura 3. Boxplots mostrando a distribuição dos resultados de precisão WAPE para cada tipo de modelo em cada conjunto de dados. O ponto vermelho indica o WAPE médio, a caixa mostra os quartis dos resultados e os bigodes se estendem para mostrar o resto da distribuição.

Tabela 11. Melhores resultados WAPE obtidos com cada tipo de arquitetura para todos os conjuntos de dados.

Modelo \ Conjunto de dados	1	2	3	4	5	6	7	8	9	10	11	12
MLP	14.886	21.245	0,367	21.114	17.191	22.930	30.193	21.365	45.515	4.373	12.303	18.101
ERNN	27.407	17.958	33.354	3.388	12.475	15.352	0,300	15.282	14.281			
LSTM							18,589	24,333	19,081	31,960	3,359	1,314
GRU	11,596	16,772	0,314	15,182	14,298	18,682	25,054	18,443	33,306	3,363	1,322	46,682
ESN	12,552	17,227	0,283	17,184	14,366	19,626	25,490	18,232	38,488	3,590	1,459	47,149
CNN	12,479	17,143	0,303	15,612	14,256	18,852	29,350	18,497	34,406	3,337	1,333	46,914
TCN	12,866	19,091	0,303	15,587	14,575	18,528	23,930	18,893	32,927	3,398	1,320	46,556

MLP	0,298	15,621	14.178	18.997							1.891	59.710
ERNN											1,374	46,739
LSTM												
GRU												
ESN												
CNN												
TCN												

a parametrização, pois apresentam um WAPE mais amplo distribuição em comparação com outros como CNN ou MLP. Em geral, exceto para o MLP que não é especificamente projetado para lidar com séries temporais, o restante do arquiteturas podem obter precisão de previsão semelhante ao melhor modelo em quase todos os casos. Isso pode ser visto observando os valores mínimos de WAPE do modelos próximos uns dos outros. No entanto, em este enredo, é mais importante analisar o quão difícil é é alcançar tal desempenho. Distribuições mais amplas indicam uma maior dificuldade para encontrar a configuração ideal do hiperparâmetro. Nesse sentido, como será visto mais tarde na análise estatística, CNN e LSTM são as alternativas mais adequadas para um projeto rápido de modelos com bom desempenho.

Além disso, a Tabela 11 apresenta uma visualização dos melhores resultados obtidos para cada arquitetura em cada conjunto de dados. Como esperado, os modelos MLP executam o pior geral. As redes MLP são modelos simples que podem servir como uma linha de base de comparação útil com o resto das arquiteturas. Podemos notar que Os modelos LSTM alcançam os melhores resultados em quatro dos doze conjuntos de dados e está entre as três principais arquiteturas para os conjuntos de dados restantes, exceto Turismo para qual LSTM está em sexto lugar apenas sobre MLP. Além disso, podemos destacar também que o GRU parece ser uma técnica muito consistente, pois obtém previsões para a maioria dos conjuntos de dados, estando dentro do três primeiras arquiteturas para dez dos doze conjuntos de dados. Os modelos TCN e ERNN obtêm os melhores resultados em dois conjuntos de dados, semelhante ao GRU. No entanto, esses dois arquiteturas são muito mais instáveis, observando resultados diferentes dependendo do conjunto de dados. A CNN apresenta um comportamento ligeiramente pior que o TCN em termos de melhores resultados. Ainda assim, como se viu no boxplot, ele supera os modelos TCN quando comparando o WAPE médio para todas as configurações de hiperparâmetros. Finalmente, a ESN é uma das piores modelos, ocupando o sexto lugar na maioria dos conjuntos de dados, apenas superando o MLP.

Devido a limitações de espaço, o relatório completo dos resultados é fornecido em um apêndice online que pode ser encontrada na Ref. 85. Este apêndice contém os resultados para cada configuração de arquitetura em várias planilhas. Além disso, possui um resumo arquivo com o melhor, média, desvio padrão e piores resultados agrupados por tipo de modelo. Vale a pena mencionando que a CNN supera o resto do modelos em termos de média e desvio padrão ção de WAPE em muitos conjuntos de dados. Isso indica que é mais fácil encontrar um conjunto de parâmetros de alto desempenho para modelos convolucionais do que para os recorrentes. No entanto, os TCNs têm um desvio padrão mais alto de WAPE, dado que são arquiteturas mais complexas e com mais parâmetros a ter em conta.

Os LSTMs também alcançam um bom desempenho em termos de WAPE médio, o que reforça ainda que é a melhor alternativa entre as redes recorrentes estudadas. GRU e ERNN são os modelos entre todos que sofrem um desvio padrão maior de resultados, o que demonstra a dificuldade de ajuste de seus hiperparâmetros. Os modelos ESN também têm maior variabilidade de precisão e não têm um bom desempenho em média.

4.2. Tempo de computação

O segundo aspecto em que as arquiteturas de deep learning são avaliadas é a eficiência computacional.

A Figura 4 representa a distribuição de treinamento e tempo de inferência para cada arquitetura. Em geral, nós pode ver que o MLP é o modelo mais rápido, seguido de perto pela CNN. A diferença entre CNN e o resto dos modelos é altamente significativo. Dentro de as redes neurais recorrentes, LSTM e GRU funcionam de forma semelhante, enquanto ERNN é a arquitetura mais lenta em geral. Em geral, a análise para todas as arquiteturas é análoga ao comparar os tempos de treinamento e de referência. No entanto, os modelos ESN não atendem

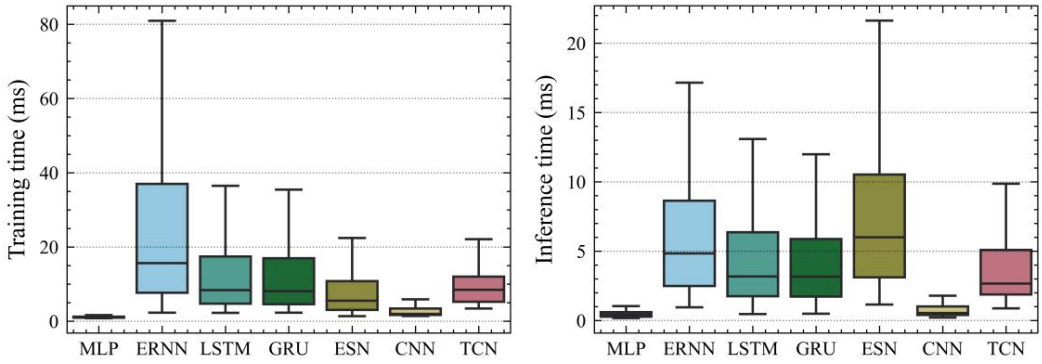

Figura 4. Distribuição dos resultados de treinamento e tempo de inferência para todas as arquiteturas.

este padrão como a maioria dos pesos da rede
não treináveis, o que resulta em treinamento eficiente
e tempo de inferência lento. Por fim, a arquitetura TCN apresenta
resultados comparáveis em termos de treinamento médio e tempo
de inferência com GRU e LSTM
modelos. No entanto, pode-se observar que os modelos TCN
têm menos variabilidade na distribuição do tempo de computação.
Isso indica que uma arquitetura mais profunda
com maior número de camadas tem um efeito menor
modelos convolucionais do que em arquiteturas recorrentes.
Portanto, projetar modelos recorrentes muito profundos pode
não ser adequado sob restrições de tempo difíceis.

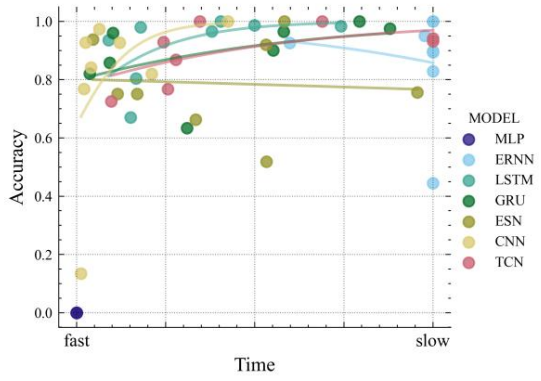

Figura 5. Precisão normalizada versus tempo de treinamento
normalizado dos melhores modelos de cada arquitetura de
aprendizado profundo para cada conjunto de dados. As linhas representam uma logística
regressão para cada modelo.

A Figura 5 visa ilustrar a velocidade/precisão
trade-off neste estudo experimental. O enredo
apresenta precisão em relação ao tempo de computação para cada
modelo de aprendizado profundo sobre cada conjunto de dados,
o que confirma as descobertas discutidas anteriormente. A figura
mostra que as redes MLP são o modelo mais rápido, mas
fornecer os piores resultados de precisão. Também pode ser
visto que o ERNN obtém previsões precisas, mas
requer altos recursos de computação. Entre as redes recorrentes,
a LSTM é a melhor rede com muito
alto desempenho preditivo e tempo de computação adequado.
CNN atinge o melhor equilíbrio de precisão de tempo entre todos
os modelos, com TCN sendo significativamente
Mais devagar.

4.3. Análise estatística

Realizamos a análise estatística, agrupando os resultados obtidos
com cada tipo de arquitetura, com relação a onze classificações
diferentes: melhor WAPE, média
WAPE, desvio padrão WAPE, pior WAPE,
tempo de treinamento e inferência do melhor modelo, tempo médio
de treinamento e inferência, desvio padrão de
tempo de treinamento e inferência, e um ranking global de
as classificações médias de todas essas comparações. Em tudo
Nos casos, o valor-p obtido no teste de Friedman indicou que as
diferenças globais nos rankings entre arquiteturas foram
significativas. Com estes resultados, podemos prosseguir com a
pesquisa de Holm-Bonferroni
análise post-hoc para realizar uma comparação de pares
entre os modelos. A Figura 6 apresenta uma visualização compacta
dos resultados da análise estatística
realizado. Ele exibe a classificação dos modelos de
da esquerda para a direita para todas as métricas consideradas.
Além disso, também permite visualizar o significado da
observaram diferenças emparelhadas com as linhas horizontais
traçadas. Neste diagrama de diferenças críticas (CD),
modelos estão ligados quando a hipótese nula de seus
equivalência não é rejeitada pelo teste. As parcelas em
que existem vários grupos sobrepostos indicam

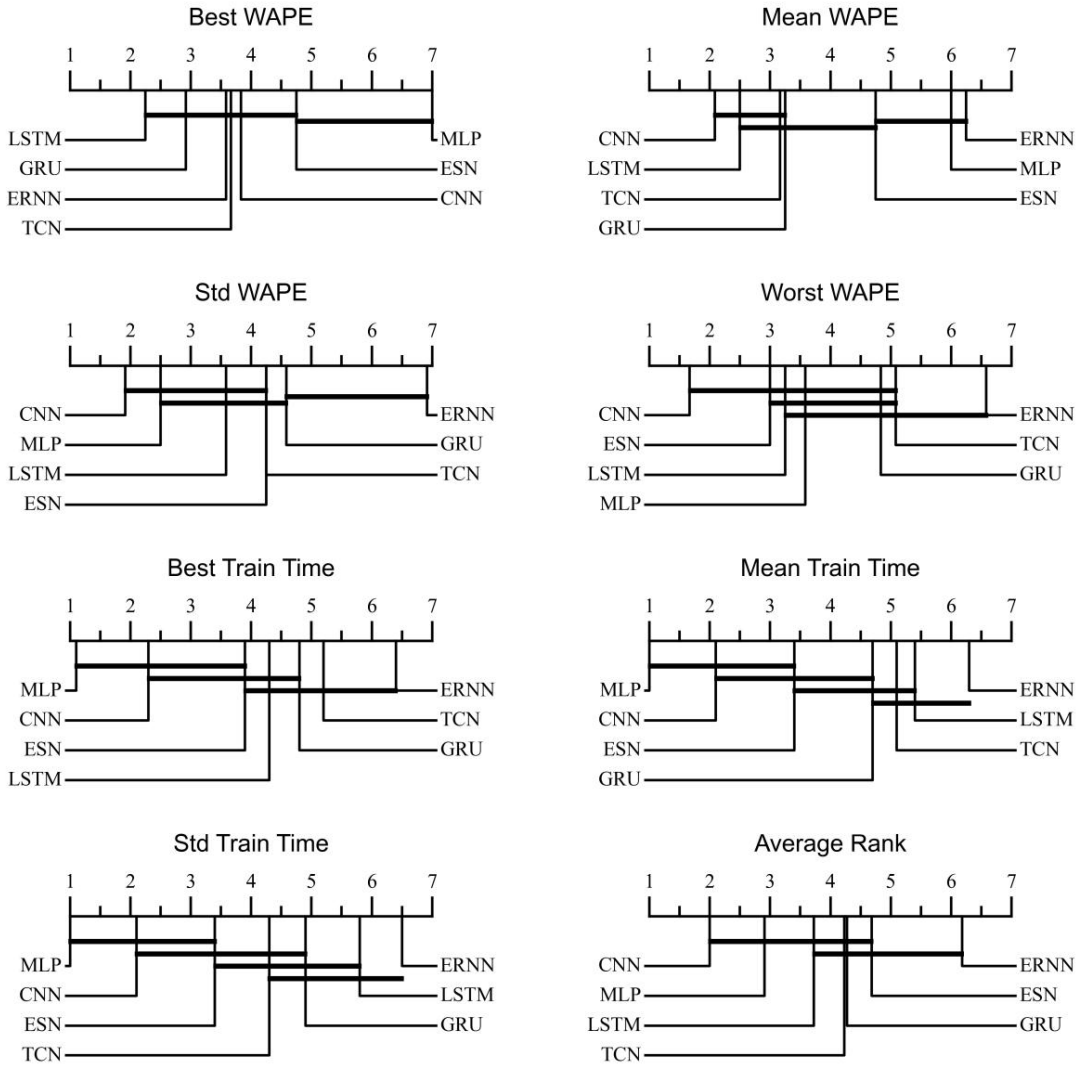

Figura 6. Diagrama de classificações e diferenças críticas (usando o procedimento post-hoc de Holm-Bonferroni) das diferentes arquiteturas de deep learning de acordo com várias métricas de desempenho

que é difícil detectar diferenças entre os modelos com desempenho semelhante.

Em termos de melhor precisão de previsão do WAPE, as redes recorrentes lideram o ranking. O LSTM ocupa o primeiro lugar, fornecendo a melhor precisão para a maioria dos conjuntos de dados, seguido de perto pelo GRU. Os modelos convolucionais, TCN e CNN, obtêm classificação quatro e cinco respectivamente. No entanto, o diagrama CD informa que as diferenças estatísticas entre os melhores modelos obtidos para cada tipo de arquitetura não são significativas, exceto para o MLP. Este fato indica que, quando a configuração ótima do hiperparâmetro é encontrada, todas essas arquiteturas podem atingir desempenho semelhante para TSF.

Ao analisar o ranking WAPE médio, os resultados são completamente diferentes. Nesse caso, a CNN ocupa o primeiro lugar, com a LSTM em segundo lugar. Isso indica que esses dois modelos são menos sensíveis à parametrização, sendo, portanto, as melhores alternativas para reduzir o alto custo de realizar uma extensa busca na grade. Este fato é ainda corroborado pelo diagrama de desvio padrão WAPE, no qual CNN e LSTM também ocupam as primeiras posições. Os modelos MLP têm logicamente um baixo desvio padrão dada a sua simplicidade. A diferença de classificação entre CNN e LSTM é maior no desvio padrão, confirmando que a CNN tem um bom desempenho em uma ampla gama de configurações. Também notamos que ERNN

apresenta o pior desempenho em termos de distribuição de resultados, mesmo abaixo do MLP. Isso sugere que a parametrização de ERNNs é complexa, sendo, portanto, o método menos recomendado entre todos os estudados.

No que diz respeito à eficiência, as conclusões obtidos analisando os tempos de treinamento e inferência foram semelhantes em ambos os casos. Portanto, para simplificar a visualização, exibimos apenas os rankings referentes ao tempo de treino. Apesar de sua previsão mais pobre precisão, o MLP ocupa o primeiro lugar em eficiência computacional. As redes MLP são os modelos mais rápidos e também apresentam um desvio padrão de tempo de treinamento muito baixo. Modelos CNN aparecem como o DL mais eficiente modelo em todos os rankings, com a ESN em terceiro lugar. Pode ver também que existem diferenças entre os diagrama do primeiro tempo (tempo de treinamento do modelo com configuração de hiperparâmetros de melhor desempenho) e a segunda (tempo médio de treinamento entre todas as configurações). O LSTM tem uma classificação muito mais baixa em termos de tempo médio de cálculo e desvio padrão, o que indica que projetar modelos LSTM muito profundos é muito caro e não conveniente para aplicações em tempo real. Além disso, vale mencionar que a CNN parece uma opção melhor do que TCN para TSF univariada. Embora os TCNs sejam mais bem projetados para séries temporais, Os modelos CNN mostraram um desempenho semelhante em termos de melhores resultados WAPE. Além disso, As CNNs têm se mostrado mais eficientes computacionalmente e mais simples em termos de encontrar uma boa configuração de hiperparâmetros.

Com todas essas análises em mente, pode-se concluir que CNN e LSTM são os mais adequados alternativas para abordar esses problemas de TSF, uma vez que é exibido no último diagrama (classificação da média classificações). Enquanto o LSTM fornece a melhor precisão de previsão, as CNNs são mais eficientes e sofrem menos variabilidade de resultados. Esta precisão versus eficiência trade-off implica que o modelo mais conveniente ser dependente dos requisitos do problema, tempo restrições e o objetivo do pesquisador.

4.4. Configuração da arquitetura do modelo

Projetar a arquitetura mais adequada para o modelos de aprendizado profundo estudados é uma tarefa muito complexa que exige uma perícia considerável. Portanto, para cada arquitetura, analisamos os resultados obtidos com os diferentes parâmetros em termos de número de camadas e neurônios, as características de con unidades voluntárias e recorrentes, etc. Apresentamos as tabelas de resultados de precisão com as configurações de arquitetura ordenadas pelas classificações médias do WAPE. Observação que, nesta análise, os rankings são independentes para cada um dos sete tipos de redes. Cada modelo configuração é treinada várias vezes com diferentes hiperparâmetros de treinamento. Dado o grande número de possibilidades, por simplicidade, relatamos apenas o melhor Resultados WAPE para cada configuração. Entre estes melhores resultados, mostramos nas tabelas as quatro primeiras e as três piores configurações (entre o melhor WAPE) para cada tipo de modelo. As tabelas de resultados completas são fornecidas no apêndice online.85

4.4.1. Multi-Layer Perceptron (MLP)

A Tabela 12 apresenta os melhores resultados para cada configuração das redes MLP. O MLP mais bem classificado modelo é composto por 5 camadas ocultas, com um total de 80 neurônios. Os resultados da tabela confirmam os achados que podem ser lidos na literatura, visto que o design de MLP é um campo bem estudado. Eles mostram que o aumento de neurônios ocultos não implica em melhor desempenho. Na verdade, o modelo mais complexo, um MLP com 320 neurônios distribuídos em 5 camadas, é posicionado penúltimo, obtendo piores previsões do que uma rede simples de uma camada de 8 neurônios.

Tabela 12. Resultados WAPE dos melhores Configurações da arquitetura MLP.

#	Camadas ocultas	Classificações médias
1	[8, 16, 32, 16, 8]	5.417
2	[32, 64]	5.917
3	[128, 64, 32]	6.167
…	…	….
10	[32, 64, 128, 8]	7.000
11	[32, 64, 128, 64, 32]	7.083
12	[32]	7.583

4.4.2. Redes Neurais Recorrentes

Entre as arquiteturas recorrentes, o melhor desempenho foi obtido com os modelos LSTM. Tabela 13 apresenta as métricas WAPE obtidas para os diferentes Configurações do modelo LSTM. Os melhores resultados foram obtido com duas camadas empilhadas de 32 unidades que retornam a sequência completa antes da saída densa camada.

Como todas as redes recorrentes possuem os mesmos parâmetros, apresentamos uma comparação global na Figura

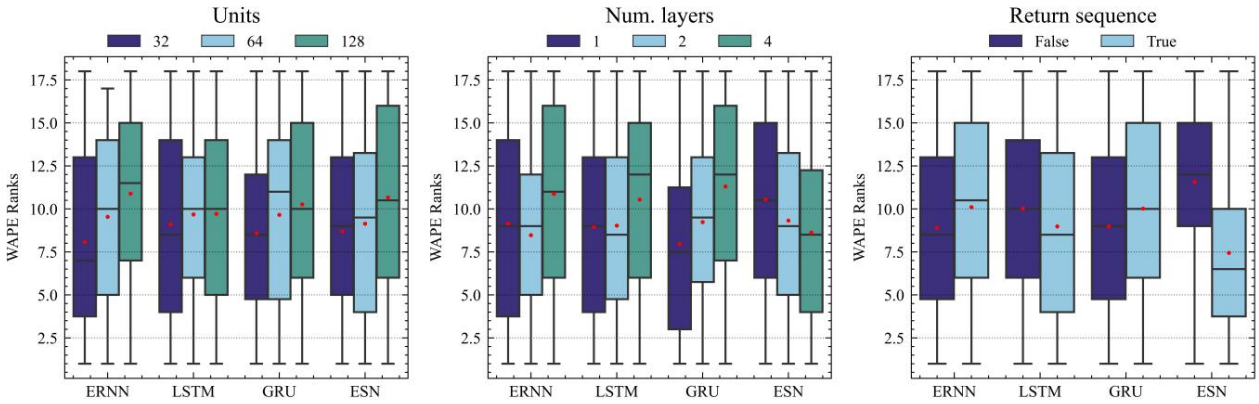

Figura 7. Distribuição dos ranks WAPE obtidos para cada arquitetura recorrente com as diferentes configurações estudadas. O ponto vermelho representa a média média.

7. Este gráfico mostra os resultados das diferentes RNN arquiteturas dependendo dos três parâmetros principais: número de camadas, unidades e se as sequências são devolvidos. Em média, podemos observar que todos modelos recorrentes tendem a obter melhores previsões com poucas unidades. Ao analisar o parâmetro do número de camadas empilhadas, podemos observar que LSTM, ERNN e GRU têm comportamento semelhante, obtendo piores resultados médios quando ela é aumentada. No entanto, os modelos ESN apresentam o padrão oposto, com os melhores resultados obtidos com 4 camadas. No que diz respeito a retornar a sequência completa ou apenas a última saída, pode-se ver claramente que os modelos ESN obtêm melhor desempenho ao retornar o sequência inteira.

Tabela 13. Resultados WAPE das melhores configurações de arquitetura LSTM.

#	Modelo			Classificações médias
	Número Sequência de retorno de unidades de camadas			
1	2	32	Verdadeiro	6.667
2		64	Verdadeiro	7.500
3	2 1	32	Falso	8.083
…	…	…	…	…
16	4	128	Falso	11.250
17	2	128	Falso	11.500
18	4	64	Falso	11.917

4.4.3. Redes Neurais Convolucionais

A Tabela 14 relata o ranking do melhor modelo CNN configurações. Pode-se notar que a previsão a precisão é proporcional ao número de camadas convolucionais. Os melhores resultados foram obtidos com modelos com quatro camadas enquanto o de camada única modelos estão na parte inferior do ranking. No entanto, em relação ao número de filtros, não há significância não pode fazer diferença à primeira vista. Também vale a pena mencionar que as melhores previsões foram obtidas de modelos sem max-pooling, sugerindo que esta operação de processamento de imagem popular não é adequada para previsão de séries temporais.

Tabela 14. Resultados WAPE das melhores configurações de arquitetura CNN.

#	Modelo			Classificações médias
	N. Camadas N. Filtros Fator de pool			
1	4	16	0	6.000
2	4	64	0	6.750
3	4	64	2	6.917
…		…		…
16	1	32	2	11.917
17	1	64	2	12.000
18	1	16	0	13.000

Tabela 15. Resultados WAPE da melhor arquitetura TCN configurações.

#	Modelo					Classificações médias
	N. Camadas N. Filtros 1 64		Dilações Kernel Return seq.			
1			[1, 2, 4, 8] 6 [1, 2, 4,		Falso	11.083
2	1	32	8] [1, 2, 4, 8, 16]	6	Falso	11.500
3	4	32		6	Falso	11.500
…		…				…
30	4	64	[1, 2, 4, 8]	3	Falso	21.250
31	4	64	[1, 2, 4, 8, 16]	3	Verdadeiro	21.500
32	4	64	[1, 2, 4, 8, 16]	3	Falso	21.667

A Tabela 15 mostra os resultados obtidos com as diferenças ent configurações de modelos TCN. A melhor rede geral é projetado com uma única camada TCN com 4 camadas convolucionais dilatadas, um kernel de comprimento 6, e 64 filtros que não retornam a sequência completa antes da camada de saída. Esses resultados indicam uma clara padrão para projetar modelos TCN. Em primeiro lugar, deve notar que os modelos de camada única com poucas dilatações camadas superam os modelos mais complexos. Relativo

Tabela 16. Exemplo de como os resultados do LSTM são coletados para comparar o parâmetro do número de camadas com o teste de classificação sinalizada de Wilcoxon.

#	Parâmetro fixo / Parâmetros			Camadas								
	Unidades	Retorno Seq.	Conjunto de dados	1	vs	2	1	vs	4	2	vs	4
	32	Falso	Falso Falso	14.554 -		15.224	14.554 -		15.352 14.747	15.224	-	14.747
1 2	32		1 2	16.212 -		16.212		-	22.489	15.352	-	22.489
3	32		3	0,312	-	0,349	0,312	-	0,337	0,349	-	0,337
…		…			…			…			…	
12	32	Falso	12	46.477 -		46.714	46.477 -		47.020 46.714	12.749	13.372 -	47.020
13	32	Verdadeiro		13.372 -		13.672	12.749		20.193 18.161	28.026	20.193 -	13.672
14	32	Verdadeiro	1 2	18.161	-			-				28.026
…		…										
24	32	Verdadeiro	12	46.608 -		46.758	46.608 -		47.067 46.758	14.150	14.259 -	47.067
25	64	Falso		14.259 -		14.419	14.150		18.131 21.601	22.234	18.131 -	14.419
26	64	Falso	1 2	21.601	-			-				22.234
…		…										
48	64	Verdadeiro	12	46.863 -		47,108	46.863 -		47,167 47,108	14,549	14,825 -	47.167
49	128	Falso	1	14.825 -		15,479	14,549		-		-	15.479
…		…										
71	128	Verdadeiro	11	1.314	-	1.340	1.314	-	1.350	1.340		1.350
72	128	Verdadeiro	12	46.822 -		46.858	46.822		- 46.994	46.858		46.994

o tamanho do kernel, os modelos com um tamanho de kernel maior fornecer as melhores previsões. Além disso, de acordo com os resultados, revela-se que a devolução do sequência completa antes que a última camada densa possa aumenta muito a complexidade do modelo, resultando em uma queda no desempenho.

4.5. Comparação estatística de configurações de arquitetura

Utilizando os resultados apresentados na subseção anterior, apresentamos uma análise estatística comparando os estudou configurações de arquitetura para cada tipo de modelo. Este estudo tem como objetivo confirmar os achados que foram discutidos anteriormente sobre as melhores escolhas de parâmetros. O método usado para esta comparação é o teste de postos sinalizados de Wilcoxon pareado. A Tabela 16 ilustra um exemplo de como os resultados são coletados para a comparação. Dado um determinado parâmetro, comparamos os resultados entre cada par de valores possíveis, mantendo fixos os demais parâmetros. Isso é vale ressaltar que, dada a grande distribuição amostral de possíveis configurações de arquitetura, o os resultados aqui apresentados são confiáveis e significativos. Por exemplo, no caso RNN, um total de 108 as amostras são comparadas para parâmetros com 2 possíveis valores, enquanto 72 amostras são comparadas para parâmetros com 3 valores.

A Tabela 17 apresenta os achados obtidos a partir de o teste de Wilcoxon, fornecendo as melhores configurações encontrados para cada tipo de arquitetura. Os números com ** indicam que é o melhor valor com um diferença estatística significativa em relação ao resto ($p < 0,05$). Indicamos com * que existe uma certa tendência sugerindo que é melhor escolher que parâmetro ($p < 0,2$), e com = quando não houver diferenças significativas entre a escolha de qualquer um dos parâmetros possíveis.

Tabela 17. Melhor configuração de arquitetura para cada tipo de modelo. Valores com ** são significativamente melhores (p-value ÿ 0,05), * sugere que é uma escolha melhor embora não significativo (p-valor ÿ 0,2), e = média que não foram encontradas diferenças entre os valores possíveis s.

ERNN		LSTM	
Camadas	1**, 2*	Camadas	1*, 2**
Unidades	32**, 64*	Unidades	=
Sequência de retorno	Falso*	Sequência de retorno	=
GRU		**ESN**	
Camadas	1**, 2**	Camadas	2*, 4**
Unidades	32*	Unidades	32**, 64**
Sequência de retorno	=	Sequência de retorno	Verdadeiro**
CNN		**TCN**	
Camadas	4**	Camadas	1**
Filtros	=	Filtros	32*
		Dilatações	=
Fator de agrupamento	0**	Núcleo	6**
		Sequência de retorno	Verdadeiro*

Para as redes recorrentes, pode-se observar que um menor número de camadas empilhadas (uma ou duas) fornece melhor performance. A única exceção é o ESN, para o qual o valor ideal é de quatro camadas. Em relação ao número de unidades, os valores 32 ou 64 são melhor que 128 para ERNN, GRU e ESN, enquanto

Tabela 18. Melhores valores de hiperparâmetros de treinamento para cada arquitetura. Valores com ** são significativamente melhores (p-valor ÿ 0,05), * sugere que é uma escolha melhor, embora não estatisticamente significativa (p-valor ÿ 0,2), e = significa que não foram encontradas diferenças entre os valores possíveis.

Parâmetro \ Arquitetura	MLP	ERNN	LSTM	GRU	ESN	CNN	TCN
Tamanho do batch	=	64**	64**	64*	32**	64**	=
Fator de histórico passado	1,25**	1,25**	1,25**	1,25**	1,25**	1,25**	
Taxa de Aprendizagem	0,001**	0,001**	0,001**	0,001**	0,001**	0,001**	
Método de normalização	zscore**	zscore**	minmax**	minmax*	zscore**	zscore**	zscore**

nenhuma diferença é encontrada para o LSTM. Além disso, o teste estatístico confirma que a ESN atinge um desempenho signif icantemente melhor ao devolver a totalidade seqüência. Também sugere que ERNN funciona melhor sem retornar as sequências, enquanto o desempenho de LSTM e GRU não é afetado por este parâmetro. No caso dos modelos LSTM, pode-se ver que a escolha do parâmetro tem um impacto mínimo sobre os resultados de precisão. Essa descoberta corrobora ainda o fato de que as LSTMs são o tipo mais robusto de rede recorrente. Como foi visto na Seção 4.3, LSTM os modelos foram os melhores em termos de WAPE médio, apresentando menor variabilidade de resultados. Como não foi encontrada diferença importante no desempenho entre os parâmetros, encontrar um projeto de arquitetura de alto desempenho é mais fácil.

Para modelos CNN, o teste Wilcoxon confirma que empilhar quatro camadas é significativamente melhor do que usando apenas um e dois. O número de filtros não não afeta o desempenho, embora esteja claro que o pooling máximo não é recomendado para esses problemas de TSF. No caso de TCNs, o teste indica que os modelos de camada única superam os modelos mais complexos. Adicionalmente, confirma-se também que os modelos com tamanho de kernel maior fornece previsões significativamente melhores. Quanto ao número de filtros e dilatação camadas, nenhuma diferença estatística foi encontrada. Além disso, os resultados sugerem que o retorno do sequência antes da última camada densa pode aumentar a complexidade do modelo excessivamente, resultando em degradação de desempenho.

4.6. Análise do treinamento parâmetros

A pesquisa em grade realizada no estudo experimental também envolveu vários hiperparâmetros de treinamento. Nós use o teste de postos sinalizados de Wilcoxon pareado para poder para comparar os resultados. A Tabela 18 resume o melhores valores de parâmetro de treinamento para cada arquitetura de aprendizado profundo. Em primeiro lugar, um padrão claro que é comum a todas as arquiteturas pode ser notado. o melhores resultados são alcançados com uma baixa taxa de aprendizado e um pequeno fator de história passada. Em relação ao método de normalização, podemos observar que o método de normalização min max tem melhor desempenho para GRU e LSTM enquanto o z-score supera significativamente o método min max nas outras arquiteturas. Além disso, resultados também mostram que a maioria das arquiteturas (ERNN, GRU, CNN e TCN) têm melhor desempenho com tamanhos de lote. Apenas para LSTM, o valor preferencial é 32, enquanto nenhuma diferença foi encontrada para MLP e ESN.

5. Conclusões

Neste artigo, realizamos uma revisão experimental sobre o uso de modelos de aprendizado profundo para séries temporais previsão. Revisamos as aplicações mais bem sucedidas de redes neurais profundas nos últimos anos, observando que as redes recorrentes têm sido as mais abordagem estendida na literatura. No entanto, as redes convolucionais estão ganhando cada vez mais popularidade devido à sua eficiência, principalmente com o desenvolvimento de redes convolucionais temporais.

Além disso, realizamos um extenso estudo experimental usando sete métodos populares de aprendizado profundo. arquiteturas: multilayer perceptron (MLP), Elman rede neural recorrente ERNN, longo prazo memória (LSTM), unidade recorrente fechada (GRU), eco rede de estado (ESN), rede neural convolucional (CNN) e rede convolucional temporal (TCN). Avaliamos o desempenho desses modelos, em termos de precisão e eficiência, mais de 12 diferentes problemas de previsão com mais de 50.000 séries temporais no total. Fizemos uma pesquisa exaustiva de configuração de arquitetura e hiperparâmetros de treinamento, construindo mais de 38.000 modelos diferentes. Além disso, realizamos uma análise estatística completa sobre várias métricas para avaliar as diferenças de o desempenho dos modelos.

As conclusões obtidas a partir deste estudo experimental estão resumidas abaixo:

- Exceto MLP, todos os modelos estudados obtêm previsões precisas quando parametrizadas corretamente. No entanto, a distribuição dos resultados de os modelos apresentaram diferenças significativas. Isso mostra a importância de encontrar um configuração de arquitetura ideal.
- Independentemente da profundidade de seus blocos, as redes MLP são incapazes de modelar a ordem temporal dos dados da série temporal, fornecendo desempenho preditivo ruim.
- LSTM obteve os melhores resultados WAPE seguindo baixado pelo GRU. No entanto, a CNN supera eles na média e desvio padrão de WAPE. Isso indicou que a convolução arquiteturas são mais fáceis de parametrizar do que os modelos recorrentes.
- As CNNs atingem a melhor relação velocidade/precisão, o que as torna mais adequadas para aplicações em tempo real do que abordagens recorrentes.
- As redes LSTM obtêm melhores resultados com um menor número de camadas empilhadas, ao contrário a arquitetura GRU. Dentro da rede recorrente funciona, o número de unidades não era importante e retornando as sequências completas só se mostrou útil nos modelos ESN.
- As CNNs exigem mais camadas empilhadas para melhorar a precisão, mas sem usar operações de pooling máximo. Para TCNs, um bloco único com um tamanho de kernel maior é recomendado.
- Com relação aos hiperparâmetros de treinamento, descobrimos que valores mais baixos de a história passada e as taxas de aprendizado são melhores escolha para esses modelos de aprendizado profundo. CNNs melhor desempenho com a normalização do z-score, enquanto min-max é sugerido para LSTM.

Em trabalhos futuros, pretendemos estudar a aplicação dessas técnicas de deep learning para séries temporais previsão em streaming. Neste cenário em tempo real, um análise mais aprofundada da velocidade versus precisão troca será necessária. Outro trabalho relevante seria estudar os melhores modelos de aprendizado profundo dependendo da natureza dos dados de séries temporais e outras características, como o comprimento ou o horizonte de projeção. Também seria importante focar futuros experimentos de ajuste dos hiperparâmetros de treinamento, uma vez que as arquiteturas de modelo mais adequadas tenham sido encontradas. Outras extensões possíveis deste estudo são a análise de tecnologia de regularização técnicas em redes profundas e realizando o estudo experimental em séries temporais multivariadas. Além disso, pesquisas futuras devem abordar as tendências atuais na literatura, como modelos de conjunto para melhorar precisão ou transferência de aprendizado para reduzir a carga de altos tempos de treinamento. Os esforços futuros também devem ser focado na construção de uma previsão de alta qualidade maior banco de dados que poderia servir como referência geral para validar novas propostas.

Financiamento

Esta pesquisa foi financiada pelo FEDER/Ministerio de Ciência, Inovação e Universidades – Agência Estadual de Pesquisa/Projeto TIN2017-88209-C2 e pelo Governo Regional da Andaluzia sob os projetos: BIDASGRI: Tecnologias de Big Data para Smart Grids (US-1263341), modelos híbridos adaptativos prever a produção de energia renovável solar e eólica (P18-RT-2778).

Agradecimentos

Somos gratos à NVIDIA por seu GPU Grant Program que nos forneceu dispositivos GPU de alta qualidade para a realização do estudo.

Bibliografia

1. Y. Zhao, L. Ye, Z. Li, X. Song, Y. Lang e J. Su, A novo mecanismo bidirecional baseado em séries temporais modelo para previsão de energia eólica, energia aplicada 177 (2016) 793-803.
2. C. Deb, F. Zhang, J. Yang, SE Lee e KW Shah, Uma revisão sobre técnicas de previsão de séries temporais para consumo de energia em edifícios, Renew. e Sust. Energ. Rev. 74 (2017) 902 - 924.
3. M. Chen e B. Chen, Uma série temporal difusa híbrida modelo baseado em computação granular para preço de ações previsão, Inf. Ciências 294 (2015) 227 – 241.
4. MH Rafiei e H. Adeli, Um novo modelo de aprendizado de máquina para estimativa de preços de venda de unidades imobiliárias, Journal of Construction Engineering and Gestão 142 (08 2015) p. 04015066.
5. S. Tuarob, CS Tucker, S. Kumara, CL Giles, AL Pincus, DE Conroy e N. Ram, como você está? sentimento?: Uma metodologia personalizada para prever estados mentais de físicos temporalmente observáveis e informação comportamental, J. Biomed. Informar. 68 (2017) 1 - 19.
6. M. Paolanti, D. Liciotti, R. Pietrini, A. Mancini e E. Frontoni, Detecção online de dados falsos furtivos ataques de injeção na estimativa de estado do sistema de energia, Trans. IEEE em Smart Grid 9(3) (2018) 1636–1646.

7. Y. Zhang, T. Cheng e Y. Ren, Um gráfico profundo método de aprendizagem para previsão de tráfego de curto prazo em grandes redes rodoviárias, engenharia civil e de infraestrutura assistida por computador 34(10) (2019) 877–896.
8. J. Schmidhuber, Aprendizado profundo em redes neurais: Uma visão geral, Redes Neurais 61 (2015) 85 – 117.
9. O. Reyes e S. Ventura, Realizando multi-alvo regressão por meio de uma rede profunda baseada em compartilhamento de parâmetros, International Journal of Neural Systems 29(09) (2019) pág. 1950014.
10. P. Lara-Ben´ytez, M. Carranza-Garc´ya, J. Garc´ya Guti´errez e J. Riquelme, Asynchronous dual pipeline deep learning framework for online data classificação de fluxo, auxiliado por computador integrado Engenharia 27(2) (2020) 101–119.
11. JCB Gamboa, Deep learning para análise de séries temporais, CoRR abs/1701.01887 (2017).
12. H. Hewamalage, C. Bergmeir e K. Bandara, Redes neurais atuais para previsão de séries temporais: Situação atual e direções futuras, Int. Diário de Previsão (2020).
13. OB Sezer, MU Gudelek e AM Ozbayoglu, Previsão de séries temporais financeiras com aprendizado profundo : Uma revisão sistemática da literatura: 2005–2019, Appl. Soft Comp. 90 (2020) pág. 106181.
14. R. Hyndman, A. Koehler, K. Ord e R. Snyder, Previsão com suavização exponencial (Springer Berlim Heidelberg, 2008).
15. GEP Box e GM Jenkins, Time Series Analysis: Forecasting and Control (Prentice Hall, 1994).
16. G. Zhang, Previsão de séries temporais usando um híbrido ARIMA e modelo de rede neural, Neurocomputing 50 (2003) 159 - 175.
17. F. Mart´ynez-Alvarez, A. Troncoso, G. Asencio e J. Riquelme, Uma Pesquisa sobre Técnicas de Mineração de Dados Aplicado à Previsão de Séries Temporais Relacionadas à Eletricidade, Energies 8(11) (2015) 13162–13193.
18. MQ Raza e A. Khosravi, Uma revisão sobre técnicas de previsão de demanda de carga baseadas em inteligência artificial para redes inteligentes e edifícios, Renew. e Sust. Energ. Rev. 50 (2015) 1352 - 1372.
19. A. Palmer, J. Montaño e A. Ses´e, Designing an rede neural artificial para previsão do tempo de turismo série, Gestão de Turismo 27(5) (2006) 781 – 790.
20. GP Zhang e DM Kline, série temporal trimestral previsão com redes neurais, transações IEEE em Redes Neurais 18(6) (2007) 1800–1814.
21. C. Hamza¸cebi, D. Akay e F. Kutay, Comparação de previsão de rede neural artificial direta e iterativa abordagens na previsão de séries temporais multiperiódicas, Sistema especialista. Aplic. 36(2) (2009) 3839 – 3844.
22. W. Yan, Rumo à previsão automática de séries temporais usando redes neurais, IEEE Trans. Rede Neural. Aprender. Sistema 23(7) (2012) 1028-1039.
23. O. Claveria e S. Torra, Previsão da demanda turística para a Catalunha: redes neurais versus séries temporais modelos, Modelagem Econômica 36 (2014) 220 – 228.
24. N. Kourentzes, DK Barrow e SF Crone, Neural operadores de conjunto de rede para previsão de séries temporais, Expert Syst. Aplic. 41(9) (2014) 4235 – 4244.
25. MM Rahman, MM Islam, K. Murase e X. Yao, Arquitetura de conjunto em camadas para previsão de séries temporais, IEEE Transactions on Cybernetics 46(1) (2016) 270-283.
26. J. Torres, A. Galicia de Castro, A. Troncoso e F. Mart´ynez-Alvarez, Uma abordagem escalável baseada em aprendizado profundo para previsão de séries temporais de big data, Engenharia Assistida por Computador Integrada 25 (08 2018) 1-14.
27. WS McCulloch e W. Pitts, Um cálculo lógico de as idéias imanentes na atividade nervosa, O boletim de biofísica matemática 5(4) (1943) 115-133.
28. HS Hippert, CE Pedreira e RC Souza, Redes neurais para previsão de carga de curto prazo: uma revisão e avaliação, IEEE Trans. em Sistemas de Energia 16(1) (2001) 44-55.
29. A. Sfetsos e A. Coonick, Previsão univariada e multivariada da radiação solar horária com técnicas de inteligência artificial, Energia Solar 68(2) (2000) 169-178.
30. R. Chandra e M. Zhang, Coevolução cooperativa de redes neurais recorrentes de Elman para tempo caótico predição de série, Neurocomputing 86 (2012) 116 - 123.
31. M. Mohammadi, F. Talebpour, E. Safaee, N. Ghadimi e O. Abedinia, Edifício de pequena escala previsão de carga com base no mecanismo de previsão híbrido, Cartas de Processamento Neural 48(1) (2018) 329–351.
32. L. Ruiz, R. Rueda, M. Cu´ellar e M. Pegalajar, Previsão de consumo de energia baseada em Elman neural redes com otimização evolutiva, Expert Syst. Aplic. 92 (2018) 380 - 389.
33. AM Schäfer e HG Zimmermann, recorrente redes neurais são aproximadores universais, Proc. 16º ICANN, (Springer-Verlag, 2006), pp. 632–640.
34. JL Elman, Encontrando estrutura no tempo, Cognitivo Ciência 14(2) (1990) 179-211.
35. H. Kim e C. Won, Prevendo a volatilidade do índice de preços de ações: um modelo híbrido integrando LSTM com vários modelos do tipo GARCH, Expert Syst. Aplic. 103 (2018) 25–37.
36. C. Yu, Y. Li, Y. Bao, H. Tang e G. Zhai, Um romance estrutura para previsão de velocidade do vento com base em redes neurais de aluguel recorrente e máquina de vetor de suporte, Energia Convers. Gerenciar 178 (2018) 137-145.
37. Wang Y, Shen Y, Mao S, Chen X e Zou H. Modelo temporal integrado LASSO e LSTM para previsão de intensidade solar de curto prazo, IEEE Inter net Things 6(2) (2019) 2933–2944.
38. S. Makridakis, E. Spiliotis e V. Assimakopoulos, A competição M4: Resultados, conclusões, conclusão e caminho a seguir, International Journal of Forecasting 34(4) (2018) 802 – 808.
39. K. Doya e S. Yoshizawa, Adaptive neural oscil lator using continuous-time back-propagation learning, Neural Networks 2(5) (1989) 375-385.

40. MI Jordan, Capítulo 25 - Ordem Serial: Um Paralelo
Abordagem de Processamento Distribuído, Institute for Cognitive
Science Technical Report 8604, 1986.
41. Y. Bengio, P. Simard e P. Frasconi, Aprender dependências de
longo prazo com gradiente descendente é difícil,
Trans. IEEE Rede Neural. 5(2) (1994) 157-166.
42. X. Ma, Z. Tao, Y. Wang, H. Yu e Y. Wang,
Rede neural de memória de longo prazo para previsão de
velocidade de tráfego usando dados de sensores remotos de
micro-ondas, Pesquisa em Transporte Parte C: Emergentes
Tecnologias 54 (2015) 187–197.
43. Y. Tian e L. Pan, Previsão de tráfego de curto prazo
fluxo pela memória de longo prazo recorrente neural
rede. Anais IEEE ISC2 2015, pp. 153–158.
44. T. Fischer e C. Krauss, Deep learning com longa
redes de memória de curto prazo para o mercado financeiro
previsões, European Journal of Operational Re search 270(2)
(2018) 654–669.
45. S. Bouktif, A. Fiaz, A. Ouni e M. Serhani, modelo LSTM de
aprendizado profundo otimizado para previsão de carga elétrica
usando seleção de recursos e algoritmo genético:
Comparação com abordagens de aprendizado de máquina,
Energias 11(7) (2018).
46. A. Sagheer e M. Kotb, Previsão de séries temporais de
produção de petróleo usando LSTM profundo
redes, Neurocomputação 323 (2019) 203 – 213.
47. C. Pan, J. Tan, D. Feng e Y. Li, Previsão de geração solar de
muito curto prazo com base no LSTM
com mecanismo de atenção temporal, 2019 IEEE 5th
ICCC 267-271.
48. S. Smyl, Um método híbrido de suavização exponencial
e redes neurais recorrentes para previsão de séries temporais,
Int. J. Previsão. 36 (1) (2020) 75 - 85.
49. K. Bandara, C. Bergmeir e S. Smyl, Previsão
em bancos de dados de séries temporais usando
redes em grupos de séries semelhantes: Um agrupamento
abordagem, Expert Syst. Aplic. 140 (2020) pág. 112896.
50. X. Lin, Z. Yang e Y. Song, preço das ações de curto prazo
previsão baseada em redes de estado de eco, Expert Syst.
Aplic. 36(3) (2009) 7313-7317.
51. A. Rodan e P. Tiÿno, eco de complexidade mínima
rede estadual, IEEE Trans. Rede Neural. 22(1)
(2011) 131-144.
52. D. Li, M. Han e J. Wang, previsão de séries temporais caóticas
com base em uma nova rede robusta de estados de eco,
Trans. IEEE Rede Neural. Aprender. Sistema 23(5) (2012)
787-797.
53. A. Deihimi e H. Showkati, Aplicação de eco
redes estaduais em previsão de carga elétrica de curto prazo,
Energia 39(1) (2012) 327–340.
54. D. Liu, J. Wang e H. Wang, vento de curto prazo
previsão de velocidade com base em agrupamento espectral e
redes otimizadas de estado de eco, Renovar. Energ. 78
(2015) 599-608.
55. H. Hu, L. Wang, L. Peng e Y. Zeng, Previsão de consumo de
energia efetiva usando ensacados aprimorados
rede estadual de eco, Energia 193 (2020).
56. S. Hochreiter e J. Schmidhuber, Long short term
memória, computação neural 9 (1997) 1735-1780.
57. FA Gers, J. Schmidhuber e F. Cummins, Aprendendo a esquecer:
Previsão contínua com LSTM, Computação neural 12(10)
(2000) 2451–2471.
58. J. Chung, C¸. G¨ul¸cehre, K. Cho e Y. Bengio, Avaliação empírica
de redes neurais recorrentes fechadas
na modelagem de sequência, CoRR abs/1412.3555
(2014).
59. L. Kuan, Z. Yan, W. Xin, C. Yan, P. Xiangkun,
S. Wenxue, J. Zhe, Z. Yong, X. Nan e Z. Xin,
Método de previsão de carga de eletricidade de curto prazo baseado
na rede GRU de auto-normalização multicamadas. 2017
Conferência IEEE EI2, pp. 1–5.
60. Y. Wang, W. Liao e Y. Chang, Gated recorrente
previsão fotovoltaica de curto prazo baseada em rede de
unidades, Energias 11(8) (2018).
61. U. Ugurlu, I. Oksuz e O. Tas, Preço da eletricidade
previsão usando redes neurais recorrentes, Energias 11(5)
(2018).
62. H. Jaeger, A abordagem do estado de eco para analisar e
treinando redes neurais recorrentes - com uma errata
nota, Centro Nacional Alemão de Pesquisa para a Tecnologia
da Informação Relatório 148 (2001).
63. H. Ismail Fawaz, G. Forestier, J. Weber,
L. Idoumghar e P.-A. Muller, Aprendizado profundo para
classificação de séries temporais: uma revisão, Data Mining e
Descoberta de Conhecimento 33(4) (2019) 917–963.
64. H. Jaeger e H. Haas, Aproveitando a não linearidade: Prevendo
sistemas caóticos e economizando energia em redes sem fio
comunicação, Ciência 304(5667) (2004) 78–80.
65. K. Cho, B. van Merrienboer, C. Gulcehre,
F. Bougares, H. Schwenk e Y. Bengio, Aprendizagem
representações de frases usando codificador-decodificador RNN
para tradução automática estatística, Proc. EMNLP
2014, pp. 1724-1734.
66. M. Ravanelli, P. Brakel, M. Omologo e Y. Bengio,
Unidades recorrentes com portas de luz para reconhecimento de fala,
IEEE T. Emerg. Topo. Com. 2(2) (2018) 92-102.
67. A. Tsantekidis, N. Passalis, A. T, J. K, M. Gab bouj e A. Iosifidis,
Previsão de preços de ações de
o livro de ordens de limite usando redes neurais convolucionais,
2017 IEEE CBI, 01, pp. 7–12.
68. P.-H. Kuo e C.ÿJ. Huang, um modelo de redes neurais artificiais
de alta precisão para energia de curto prazo
previsão de carga, Energias 11(1) (2018).
69. I. Koprinska, D. Wu e Z. Wang, Convolucional
redes neurais para previsão de séries temporais de energia,
IJCNN 2018, pp. 1–8.
70. C. Tian, J. Ma, C. Zhang e P. Zhan, A deep neural
modelo de rede para previsão de carga de curto prazo com base em
rede de memória de curto prazo longa e convolucional
rede neural, Energias 11(12) (2018).
71. H. Liu e Y. Li, vento baseado em aprendizado profundo inteligente
modelo de previsão de velocidade usando posição de
decomposição de pacotes wavelet, rede neural convolucional
e rede de memória convolucional de longo prazo, Energia
Conversão e Gestão 166 (2018) 120 – 131.
72. M. Cai, M. Pipattanasomporn e S. Rahman, Day-

previsões de carga no nível do edifício usando deep learning versus técnicas tradicionais de séries temporais, Applied Energia 236 (2019) 1078 – 1088.
73. Z. Shen, Y. Zhang, J. Lu, J. Xu e G. Xiao, A novo modelo de previsão de séries temporais com aprendizado profundo, Neurocomputing 396 (2020) 302 – 313.
74. A. Borovykh, S. Bohte e C. Oosterlee, dilatados redes neurais convolucionais para previsão de séries temporais, Journal of Computational Finance 22(4) (2019) 73-101.
75. Y. Chen, Y. Kang, Y. Chen e Z. Wang, Proba bilistic forecasting with temporal convolutional neural network, arXiv:1906.04397 (2019).
76. R. Wan, S. Mei, J. Wang, M. Liu e F. Yang, Rede convolucional temporal multivariada: Uma abordagem de redes neurais profundas para séries temporais multivariadas previsão, Eletrônica 8(8) (2019).
77. P. Lara-Ben´ytez, M. Carranza-Garc´ÍA, J. Luna Romera e J. Riquelme, Temporal Convolutional Redes Aplicadas a Séries Temporais Relacionadas à Energia Previsão, Ciências Aplicadas 10(7) (2020) p. 2322.
78. D. Kang e Y.ÿJ. Cha, UAVs autônomos para monitoramento de integridade estrutural usando aprendizado profundo e um sistema de sinalização ultrassônica com geo-tagging, Engenharia Civil e de Infraestrutura Assistida por Computador 33(10) (2018) 885–902.
79. G. Hinton, L. Deng, D. Yu, GE Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, TN Sainath e B. Kingsbury, redes neurais profundas para modelagem acústica no reconhecimento de fala: o pontos de vista de quatro grupos de pesquisa, IEEE Signal Processing Magazine 29(6) (2012) 82–97.
80. Y. Kim, Redes neurais convolucionais para sentença classificação, Anais da Conferência EMNLP 2014, págs. 1746-1751.
81. J.ÿB. Yang, N. Nhut, P. San, X. li e P. Shonali, Redes neurais convolucionais profundas em multicanal série temporal para reconhecimento de atividade humana, IJCAI (07 2015).
82. A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior e K. Kavukcuoglu, Wavenet: Um modelo generativo para áudio bruto, arXiv:11609.03499 (2016).
83. S. Bai, J. Kolter e V. Koltun, Uma avaliação empírica de redes convolucionais e recorrentes genéricas para modelagem de sequência, arXiv:1803.01271 (2018).
84. F. Yu e V. Koltun, agregação de contexto multi-escala por convoluções dilatadas, ICLR 2016 ,
85. P. Lara-Ben´ytez e M. Carranza-Garc´ya, Time Series Forecasting - Deep Learning https://github.c om/pedrolarben/TimeSeriesForecasting-DeepLea rning, (2020).
86. M. Stÿepniÿcka e M. Burda, Inteligência Computacional em Previsão (CIF) https://irafm.osu.cz /cif, (2016).
87. G. Lai, W.ÿC. Chang, Y. Yang e H. Liu, Modelagem padrões temporais de longo e curto prazo com redes neurais, arXiv:1703.07015 (2017).
88. S. Makridakis e M. Hibon, A competição M3: resultados, conclusões e implicações, International Journal of Forecasting 16(4) (2000) 451 – 476.
89. S. Makridakis, E. Spiliotis e V. Assimakopoulos, A competição M4: 100.000 séries temporais e 61 métodos de previsão, International Journal of Forecasting 36(1) (2020) 54 – 74.
90. Competição de previsão de séries temporais NNGC, NN5 para redes neurais http://www.neural-forecasting -competition.com/NN5, (2008).
91. G. Athanasopoulos, RJ Hyndman, H. Song e DC Wu, previsão de turismo parte dois www.kaggle .com/c/tourism2, (2010).
92. NREL, Dados de energia solar para estudos de integração ww w.nrel.gov/grid/solar-power-data.html, (2007).
93. Y. Li, R. Yu, C. Shahabi e Y. Liu, Rede neural recorrente convolucional de difusão: Tráfego orientado a dados previsão, arXiv:1707.01926 (2017).
94. Google, competição de previsão de séries temporais de tráfego da Web www.kaggle.com/c/web-traffic-time-series -previsão, (2017).
95. S. Ben Taieb, G. Bontempi, AF Atiya e A. Sor jamaa, Uma revisão e comparação de estratégias para previsão de séries temporais com vários passos à frente com base em a competição de previsão NN5, Expert Syst. Aplic. 39(8) (2012) 7067-7083.
96. D. Kingma e J. Ba, Adam: Um método para otimização estocástica, arXiv:1412.6980 (2014).
97. S. Kim e H. Kim, Uma nova métrica de erro percentual absoluto para previsões de demanda intermitente, Int. J. Previsão 32(3) (2016) 669 - 679.
98. S. García e F. Herrera, Uma extensão sobre “comparações estatísticas de classificadores sobre múltiplos dados sets” para todas as comparações de pares, Journal of Machine Learning Research 9 (2008) 2677–2694.
99. F. Wilcoxon, Comparações Individuais por Classificação Métodos, Avanços em Estatística: Metodologia e Distribuição (Springer, 1992), pp. 196-202.