

# Predição de Métricas em Grafos Temporais Utilizando Redes Neurais

Daiane M. Pereira<sup>1</sup>, Rodrigo S. Couto<sup>1</sup> \*

<sup>1</sup>Universidade Federal do Rio de Janeiro - PEE/COPPE/GTA

{daiane, rodrigo}@gta.ufrj.br

**Abstract.** *Several domains, such as characterization of traffic flow in transportation systems, use basic metrics for temporal graphs, such as the average degree and the clustering coefficient. Despite the applicability of these metrics, the literature does not address techniques for predicting their temporal evolution. To fill this gap, we use neural network models to forecast temporal graph metrics. Thus, we analyze the performance of a multilayer perceptron, a recurrent neural network, and a convolutional neural network. These models are compared with a simple base model, showing satisfactory performance, especially for recurrent and convolutional neural networks.*

**Resumo.** *Métricas básicas calculadas para grafos temporais, como o grau médio e o coeficiente de clusterização, são utilizadas em vários domínios, como na caracterização do fluxo de tráfego em sistemas de transporte. Apesar da aplicabilidade dessas métricas, a literatura não aborda técnicas para prever a sua evolução temporal. Para preencher essa lacuna, este trabalho aborda o uso de modelos de rede neural para prever métricas de grafos temporais. Assim, analisa-se o desempenho de uma perceptron multicamadas, de uma rede neural recorrente e de uma rede neural convolucional. Esses modelos são comparados com um modelo de base simples, mostrando desempenho satisfatório, principalmente para as redes neurais recorrentes e convolucionais.*

## 1. Introdução

A estrutura de redes complexas pode ser analisada por meio de métricas de grafos, como o coeficiente de clusterização e o grau médio. Essa análise é base de diversas aplicações, como a criação de sistemas de recomendação de amigos em redes sociais, envio de propagandas personalizadas e predição de falhas em enlaces de redes de computadores [Hegeman e Iosup, 2018]. Algumas análises de redes complexas envolvem grafos temporais, que são modificados ao longo do tempo. Em outras palavras, os nós e os enlaces formados no tempo  $t$  podem ser diferentes no tempo  $t + 1$ .

Os grafos temporais apresentam várias aplicações na literatura. Em [Charakopoulos et al., 2014] é utilizada a representação por grafos para modelar séries temporais. Investigam-se características dos grafos ao longo do tempo, para

---

\*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001. O trabalho também foi financiado pelo CNPq, pela FAPERJ com os auxílios E-26/203.211/2017 e E-26/211.144/2019, e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), auxílio no. 2015/24494-8.

analisar a trajetória de jatos turbulentos aquecidos. Métricas como o diâmetro da rede, o coeficiente de clusterização e a distribuição do grau são utilizadas para essa caracterização. Já em [Tang et al., 2016] e [Tang et al., 2014], a modelagem por grafos é utilizada para analisar o tráfego de veículos, por meio de métricas básicas como o coeficiente de clusterização e o grau da rede.

Dada a importância das métricas básicas, percebe-se que a capacidade de prever o seu comportamento ao longo do tempo pode auxiliar na tomada de decisões como, por exemplo, em melhorias de redes rodoviárias [Cui et al., 2019]. Entretanto, os trabalhos da literatura não abordam a predição desse tipo de métrica. Dessa forma, o objetivo deste trabalho é avaliar o comportamento de modelos de rede neural, amplamente utilizados na literatura, para predição de métricas básicas de grafos temporais.

Para obter séries temporais a partir de um grafo temporal, este trabalho utiliza a técnica de construção de *snapshots* [Nicosia et al., 2013]. Nessa técnica, realizam-se recortes do grafo no tempo, com tamanho fixo, e cada recorte temporal é tratado como um grafo estático. De forma mais precisa, para cada recorte temporal, medem-se métricas de grafos. Essas métricas são utilizadas como uma série temporal pelos modelos de rede neural, para aprender a evolução do grafo ao longo do tempo e prever os valores das métricas.

As técnicas deste trabalho objetivam prever o grau médio da rede e o coeficiente de clusterização médio. O grau médio fornece informação sobre a quantidade de enlaces existentes no grafo. Dessa forma, prever o aumento ou diminuição dessa métrica possibilita analisar se a rede será mais ou menos densa no futuro. Já o coeficiente de clusterização indica o quão próximo de um clique é a vizinhança de um nó. Um aumento ou diminuição dessa métrica fornece informação sobre a criação ou o desaparecimento de comunidades [Santoro et al., 2011].

A avaliação deste trabalho considera três tipos de redes neurais: uma perceptron multicamadas (*MultiLayer Perceptron* - MLP), uma memória longa de curto prazo (*Long Short Term Memory* - LSTM) e uma rede neural convolucional (*Convolutional Neural Network* - CNN). Essas redes são comparadas com um modelo base, que consiste em repetir o último valor de entrada como previsão do futuro. Os resultados deste trabalho mostram que é possível utilizar modelos de rede neural para prever as métricas consideradas, obtendo resultados superiores ao modelo base. Além disso, mostra-se que redes neurais mais complexas, como a LSTM e a CNN, obtêm os melhores resultados na maioria dos casos.

Este trabalho está dividido da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. Em seguida, a Seção 3 informa as bases de dados utilizadas e detalha o processo para a criação das séries temporais. A Seção 4 apresenta os modelos de aprendizado de máquinas utilizados, bem como o modelo de base. Finalmente, a Seção 5 fornece os resultados das análises e a Seção 6 apresenta conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

O problema de prever valores futuros em séries temporais encontra duas divisões na literatura. Na primeira, utilizam-se métodos lineares, como o modelo Auto Regressivo Integrado de Médias Móveis (*Auto-Regressive Integrated Moving Average* - ARIMA) [Nielsen, 2019]. Por muitos anos, esse método foi

considerado o mais apropriado para prever o comportamento de séries temporais [Siami-Namini et al., 2018], [De Gooijer e Hyndman, 2006]. Apesar de ainda ser muito utilizado e de apresentar bom desempenho em alguns domínios [Mbah et al., 2021, Makridakis et al., 2018, Yamak et al., 2019], o ARIMA apresenta limitações, como o fato de não apresentar um bom desempenho quando há uma relação não linear entre as variáveis. Para superar essa limitação, a segunda divisão utiliza métodos não lineares, como as redes neurais profundas [Selvin et al., 2017]. Em [Siami-Namini et al., 2018] e [Siami-Namini et al., 2019] são realizadas análises, comparando o ARIMA com métodos de aprendizado de máquinas. Nesses estudos, os modelos de aprendizado de máquina obtêm um menor erro na predição das séries temporais.

Na literatura de redes neurais, diversos métodos são utilizados para a predição de séries temporais. Em [Selvin et al., 2017] comparam-se três tipos de redes neurais na predição do preço de ações. As redes analisadas são as CNNs, Redes Neurais Recorrentes (*Recurrent Neural Network* - RNN) e a LSTM. Nesse trabalho, as CNNs obtiveram os melhores resultados.

O trabalho [Hua et al., 2019] realiza um estudo sobre o uso do modelo LSTM para a predição de séries temporais e propõe um novo modelo, baseado no LSTM e no uso de grafos esparsos. O objetivo do modelo proposto é reduzir a complexidade computacional do LSTM. O estudo mostra que apesar de o modelo proposto ter desempenho inferior ao LSTM, seu desempenho é ainda superior ao do ARIMA nas bases de dados analisadas. Outros estudos comparam o uso de modelos de Redes Neurais, como RNNs e CNNs, em diversos domínios de estudo. Por exemplo, [Kim e Kang, 2019] e [Borovykh et al., 2017] consideram séries temporais financeiras, enquanto [Binkowski et al., 2018] aborda a previsão do consumo de energia elétrica. Em [Lim e Zohren, 2021] é apresentado um *survey* sobre o uso de métodos de aprendizado de máquina em problemas de predição de séries temporais.

Há na literatura diversos trabalhos que realizam predição em grafos temporais. Por exemplo, [Yu et al., 2017], [Li et al., 2017] e [Cui et al., 2019] abordam grafos que modelam o tráfego de veículos em rodovias. Nesses trabalhos, utilizam-se modelos de rede neural, como o LSTM e o RNN, para realizar a previsão do tráfego em redes rodoviárias. Os trabalhos da literatura focam sempre um domínio específico, como é o caso do tráfego de veículos nos trabalhos citados anteriormente. No entanto, muitas vezes é importante prever métricas básicas de grafos temporais, como o grau médio e o coeficiente de clusterização. Isso justifica-se, pois, essas métricas básicas são utilizadas em diversas aplicações, como na construção de algoritmos para caracterização de tráfego em rodovias [Tang et al., 2014] e para previsão do comportamento de redes sociais [Cem e Sarac, 2016]. Dessa forma, este trabalho preenche uma lacuna da literatura, abordando a predição de métricas básicas de grafos temporais. Mais especificamente, realiza-se a predição do grau médio e do coeficiente de clusterização em grafos temporais, analisando o desempenho de diferentes modelos de redes neurais nessa tarefa.

### **3. Descrição das Bases de Dados e da Metodologia**

Nesta seção, descreve-se a metodologia utilizada para a criação dos grafos e o cálculo das métricas. Além disso, apresentam-se as bases de dados utilizadas na análise.

### 3.1. Criação dos Grafos

Este trabalho utiliza *snapshots* de grafos temporais, com período de um dia. Dessa forma, as transações de cada base de dados são agrupadas por dia e modeladas como grafos não direcionados com peso,  $G_d(V_d, E_d, \mathbf{w})$ , sendo  $V_d$  o conjunto dos nós que realizaram transações no dia  $d$  e  $E_d$  o conjunto dos enlaces. Cada enlace possui um peso  $w$  associado. O peso de cada enlace indica a quantidade de transações ocorridas entre o par de nós durante o dia  $d$ . Por exemplo, suponha que a base de dados possua registros de envio de e-mails. Nessa base, caso o usuário A e o usuário B enviem dois e-mails entre si no mesmo dia, o peso do enlace A-B será igual a dois. Note que o peso considera todas as transações entre A e B, independentemente de qual dos dois nós é a origem, visto que o modelo é um grafo não direcionado. Neste trabalho, o grau médio e o coeficiente de clusterização médio são calculados para cada grafo  $G_d(V_d, E_d, w)$ .

### 3.2. Aquisição dos Dados

Este trabalho utiliza três bases de dados distintas, de forma a analisar o modelo em diferentes domínios. Para cada base, utiliza-se o processo de construção de grafos descrito na Seção 3.1. A Tabela 1 apresenta um resumo das suas principais características dos grafos criados, como a quantidade de nós, quantidade de enlaces e a quantidade de dias existentes nas bases de dados. Esse resumo considera um grafo que utiliza todas as entradas existente nas bases de dados e utilizadas neste trabalho, sem a separação em *snapshots*. Vale notar que o resumo da Tabela 1 é apresentado apenas para fins de exemplo, já que a análise deste trabalho considera *snapshots* dos grafos temporais. A descrição de cada base é apresentada a seguir.

**Tabela 1. Características das Bases de Dados**

Nome da base	#Nós	#Arestas	Tempo (Dias)
Email-Eu-core	986	24929	803
CollegeMsg	1899	20296	193
Bitcoin	10964110	21827637	120

#### 3.2.1. Email-Eu-core

Esta base possui informações sobre a troca de e-mails entre membros de instituições de pesquisas europeias, coletadas durante 803 dias. O conteúdo e o endereço dos e-mails são anonimizados. A base possui três campos, a origem do e-mail (SRC), o destino do e-mail (TGT) e o horário (TS) em segundos desde o início da coleta. A existência de uma entrada na base com SRC=A, TGT=B e TS=10 significa que um e-mail foi enviado do usuário A para o usuário B e que a troca de e-mails ocorreu 10 segundos após o início da coleta. Na representação por grafos da Seção 3.1, SRC e TGT são nós, e a troca de e-mails é representada pela existência de um enlace unindo os dois nós. Na base Email-Eu-core, a transação é o envio de um e-mail. Assim, os pesos dos enlaces consistem na quantidade de e-mails enviados. A base de dados está disponível no site SNAP (*Stanford Network Analysis Project*)<sup>1</sup>.

<sup>1</sup><https://snap.stanford.edu/data/email-Eu-core-temporal.html>

### 3.2.2. CollegeMsg

Esta base possui informações sobre a troca de mensagens entre estudantes da UCI (*University of California, Irvine*) em uma rede social, durante 193 dias. A base possui os mesmos campos da Email-Eu-core e o seu modelo utiliza a mesma representação por grafos da Seção 3.2.1. A diferença é que a transação da CollegeMsg é o envio de uma mensagem na rede social. A base também faz parte do projeto SNAP<sup>2</sup>.

### 3.2.3. Bitcoin

Esta base possui transações do Bitcoin, sendo fornecida pela universidade ELTE (*Eötvös Loránd University*)<sup>3</sup>. A base possui todas as transações ocorridas desde o surgimento do Bitcoin até o dia 9 de fevereiro de 2018. Esta base possui diversos atributos que caracterizam uma transação de envio de bitcoins, divididos em diversos arquivos em formato de planilhas. Este trabalho utiliza os atributos `txID` e `addrID` das planilhas de `txin` e `txout`. O `txID` é utilizado como uma chave de ligação de cada transação, associando os `addrIDs` da planilha de origem (isto é, `txin`) com o `addrID` da planilha de destino (isto é, `txout`). Utilizam-se também os atributos `blockID` e `block_timestamp` para identificar a data e o horário de cada transação e agrupar as que ocorreram no mesmo dia.

Com o objetivo de diminuir o tempo de treinamento dos modelos e lidar com as limitações do *hardware* disponível, este trabalho usa apenas um subconjunto da base Bitcoin da ELTE. Assim, utiliza-se um período anterior a maio de 2013, já que o Bitcoin teve um aumento significativo no número de usuários e de transações após esse período. Dessa forma, o subconjunto utilizado possui todas as transações realizadas entre os dias 1 de fevereiro de 2012 e 30 de maio de 2013.

Este trabalho realiza também um pré-processamento da base Bitcoin da ELTE, visando remover inconsistências. Um exemplo é o envio de recompensa a um minerador, que é uma transação sem endereço de origem. Além disso, a análise deste trabalho agrega os endereços do Bitcoin utilizando a heurística apresentada em [Nakamoto, 2019]. Essa heurística objetiva agrupar endereços pertencentes a um mesmo usuário. Em linhas gerais, a heurística considera que se uma transação possui múltiplos endereços de origem, esses endereços são considerados pertencentes a um mesmo indivíduo.

## 4. Modelos de Redes Neurais

Esta seção apresenta os modelos utilizados para predição das séries temporais descritas na Seção 3. Utilizam-se três tipos de redes neurais: Perceptron Multicamadas, Redes Neurais Recorrentes e Redes Neurais Convolucionais. Além disso, utiliza-se um modelo base para analisar o desempenho das redes neurais.

Os modelos de redes neurais possuem hiperparâmetros utilizados no treinamento. Para a escolha dos hiperparâmetros deste trabalho, utiliza-se uma busca em grade (*Grid Search*) para cada um dos três tipos de rede neural considerados. Nesse

---

<sup>2</sup><https://snap.stanford.edu/data/CollegeMsg.html>

<sup>3</sup><https://senseable2015-6.mit.edu/bitcoin/>

método, é construída uma grade com os valores que serão testados para cada hiperparâmetro [Bergstra e Bengio, 2012]. Os modelos são então avaliados para todas as combinações dos valores listados e a combinação com o menor erro é selecionada para ser utilizada na análise da Seção 5. Neste trabalho seleciona-se, para cada tipo de rede neural, a combinação que apresentou o menor erro MAE (*Mean Absolut Error* - MAE) na predição. O MAE calcula a média do erro absoluto entre o valor real e o valor predito.

#### 4.1. Modelo Base

Esta solução é simples e implementada em diversos sistemas de previsão [Hyndman e Athanasopoulos, 2018]. Neste modelo, a previsão para o tempo  $t + 1$  consiste em repetir o valor da métrica observado na entrada atual (isto é, no tempo  $t$ ). Ou seja, a previsão  $\hat{y}(t + 1)$  do modelo é igual a  $y(t)$ . Essa solução é utilizada neste trabalho como base para comparar o desempenho dos diversos modelos de redes neurais.

#### 4.2. Perceptron Multicamadas

Uma MLP (*MultiLayer Perceptron* - *Perceptron* Multicamadas) consiste em uma rede neural com, no mínimo, três camadas totalmente conectadas [Frank et al., 2001]. Assim como o modelo base, a MLP realiza a previsão para o tempo  $t + 1$  com base no valor da métrica no tempo  $t$ . A diferença consiste no fato de que o modelo base apenas repete o valor observado, enquanto a MLP possui neurônios com função de ativação capazes de aprender relações não lineares entre os valores de entrada atual e a predição de saída.

Para o MLP, o método de busca em grade deste trabalho utiliza arquiteturas com diferentes números de camadas, quantidade de neurônios e funções de ativação. A arquitetura selecionada possui três camadas escondidas, contendo 128 unidades com função de ativação ReLU (*Rectified Linear Unit*). Já a de saída possui apenas uma unidade completamente conectado com função de ativação linear.

#### 4.3. Rede Neural Recorrente

Uma rede neural recorrente (*Recurrent Neural Network* - RNN) toma decisões baseadas não só na entrada atual, como também nas informações de entradas passadas. Este trabalho utiliza um tipo de RNN denominado Memória Longa de Curto Prazo (*Long Short Term Memory* - LSTM). Cada célula de uma LSTM tem seus estados internos divididos em dois,  $h_{(t)}$  e  $c_{(t)}$ . O  $h_{(t)}$  pode ser visto como uma memória de curto prazo, já que o peso associado as entradas mais antigas diminuem a cada nova iteração. Já o  $c_{(t)}$ , pode ser considerado uma memória de longo prazo, já que no processo de atualização entradas antigas podem continuar tendo peso relevante, dependendo a dinâmica do modelo [Yan, 2015]. O uso de informações do passado na tomada de decisões pode ser visto como um tipo de memória. Por esse motivo, é comum utilizar LSTM em tarefas como predição de preço de ações, reconhecimento de fala e outras envolvendo séries temporais [Hewamalage et al., 2020].

Após o ajuste dos hiperparâmetros da busca em grade, a LSTM utilizada neste trabalho possui três camadas, duas delas com 128 unidades com função de ativação tangente hiperbólica e *dropout* de 0,3. A última camada possui apenas uma unidade totalmente conectada. Diferentemente do modelo base e do modelo MLP, que realizam a predição utilizando apenas o valor da entrada atual, no modelo LSTM utilizam-se as últimas 14

entradas para prever a saída  $\hat{y}(t + 1)$ . A exceção é para a base de dados CollegeMsg, que apresentou menor erro para entrada com tamanho 7, durante a otimização dos hiperparâmetros.

#### 4.4. Rede Neural Convolutacional

Uma rede neural convolutacional (*Convolutional Neural Network* - CNN) é um tipo de rede neural muito utilizado em problemas de classificação e em problemas de visão computacional como, por exemplo, no reconhecimento de imagens. Apesar de serem mais comuns em tarefas de classificação, também podem ser utilizadas para previsão de séries temporais. Na classificação de imagens monocromáticas as camadas convolutacionais da CNN operam em duas dimensões, que representam a largura e altura da imagem. Ao lidar com séries temporais, a convolução é aplicada em apenas uma dimensão, que é o tempo. A ideia é aproveitar os filtros da CNN que identificam padrões nas sequências de dados e usar esses padrões para prever valores futuros das séries temporais [Borovykh et al., 2017].

Os principais componentes de uma CNN são a camada de convolução, a camada de agrupamento (*pooling*) e a camada *flatten*. A camada de convolução consiste em uma janela, chamada de *kernel* ou filtro, que é deslizada e multiplicada ponto a ponto com a série temporal. O tamanho e a quantidade de filtros aplicados são parâmetros importantes de uma CNN. O processo de convolução aumenta a dimensão dos dados. Por esse motivo, normalmente, uma camada de agrupamento é utilizada após uma camada de convolução, diminuindo assim a dimensão da saída. O tipo de camada de agrupamento utilizado neste trabalho é o *max-pooling*. Nesse agrupamento, gera-se uma janela que percorre a saída da camada convolutacional e, dentro de cada janela, o maior valor observado é retornado. Após a camada de agrupamento, é utilizada uma camada *flatten*, que simplesmente transforma a saída da camada de agrupamento em um vetor. Esse processo prepara a saída da camada de agrupamento para ser a entrada de uma camada totalmente conectada.

A CNN utilizada neste trabalho possui como entrada um vetor no qual cada elemento é o valor da métrica em um instante no passado. De forma similar ao modelo LSTM, na CNN também se utilizam as últimas 14 entradas, para as bases do Bitcoin e Email-Eu-core. Já a base College Msg foi treinada utilizando as últimas 7 entradas. Assim, o vetor de entrada possui 14 elementos. A CNN selecionada na busca em grade possui uma camada convolutacional com 128 filtros de tamanho dois e função de ativação ReLU. Essa camada é seguida por uma *max-pooling* com tamanho dois. Após a camada de *max-pooling*, é utilizada uma camada *flatten*. A saída da camada *flatten* é conectada a uma camada totalmente conectada, com 20 neurônios. Em seguida, há uma nova camada totalmente conectada, que possui apenas um neurônio, e calcula a saída  $\hat{y}(t + 1)$ .

### 5. Análises e Resultados

A análise deste trabalho consiste em empregar os modelos da Seção 3 na predição das séries temporais da Seção 3.1. Todos os modelos são implementados com as bibliotecas Keras e TensorFlow<sup>4</sup>, executando no *Google Colaboratory*<sup>5</sup>.

Primeiramente, aplica-se o método de janelas deslizantes com tamanho fixo, que é muito utilizado na previsão de séries temporais[Hua et al., 2019]. Esse método permite

---

<sup>4</sup><https://www.tensorflow.org/>

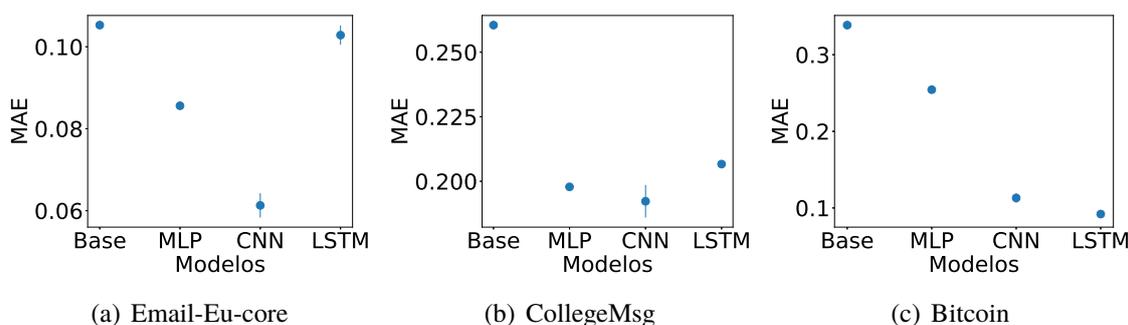
<sup>5</sup><https://colab.research.google.com>

tratar a previsão de séries temporais como um problema de aprendizado supervisionado, no qual os valores de entrada e de saída são conhecidos durante o treino. Para isso, a série temporal é dividida em intervalos com o mesmo tamanho, sendo o último valor de cada intervalo tratado como a saída que deve-se prever. Neste trabalho, a janela possui tamanho de 15 dias, sendo 14 dias utilizados como entrada do modelo e o último dia é o valor que o modelo deve prever. O método de janelas deslizantes é utilizado em todos os modelos deste trabalho. No entanto, para o modelo de base e para a MLP, realiza-se a previsão com um único valor de entrada por vez. Nesses casos, a janela deslizante tem tamanho dois, sendo o primeiro valor a entrada do modelo e o segundo o valor que o modelo deve prever.

As séries são divididas em duas partes, sendo 75% das janelas separadas para o treino e 25% para o teste. A base de treino é utilizada para a criação do modelo, treino e ajuste de parâmetros. Já a base de teste é utilizada para a avaliação do desempenho.

Utilizam-se duas métricas para avaliar o desempenho dos modelos. A primeira é o MAE, que também é utilizado na busca em grade descrita na Seção 4. Já a segunda é Raiz do Erro Quadrático Médio (*Root Mean Squared Error* - RMSE). O erro MAE atribui um peso igual a todos os erros, independentemente da sua amplitude. Já o RMSE tem como consequência a atribuição de um peso maior aos erros com maior magnitude, visto que o erro é elevado ao quadrado no seu cálculo.

Para cada série temporal, executam-se 30 medições dos erros MAE e RMSE. Em cada medição, a divisão da base de dados em treino e teste é refeita utilizando a função `train_test_split` do `scikit-learn`<sup>6</sup>. Todos os resultados são apresentados com intervalo de confiança de 95%. As Figuras 1 e 2 apresentam o MAE para o grau médio e a clusterização, respectivamente. Os resultados mostram que, para cada base de dados, sempre há um modelo de rede neural com desempenho superior ao método base.

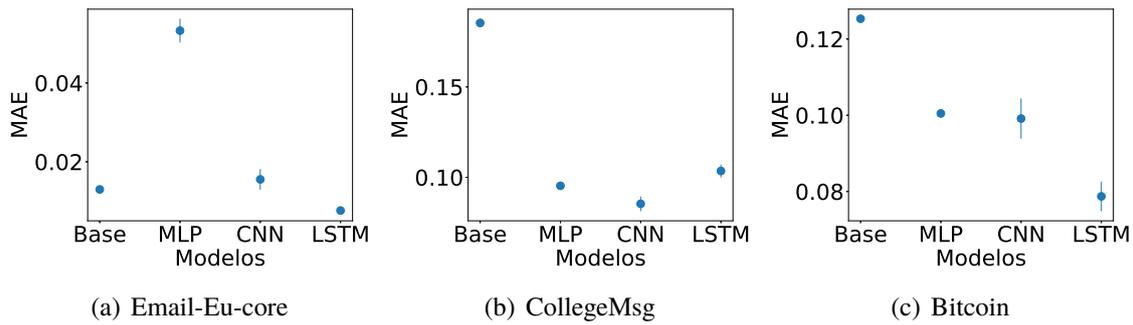


**Figura 1. MAE do Grau Médio**

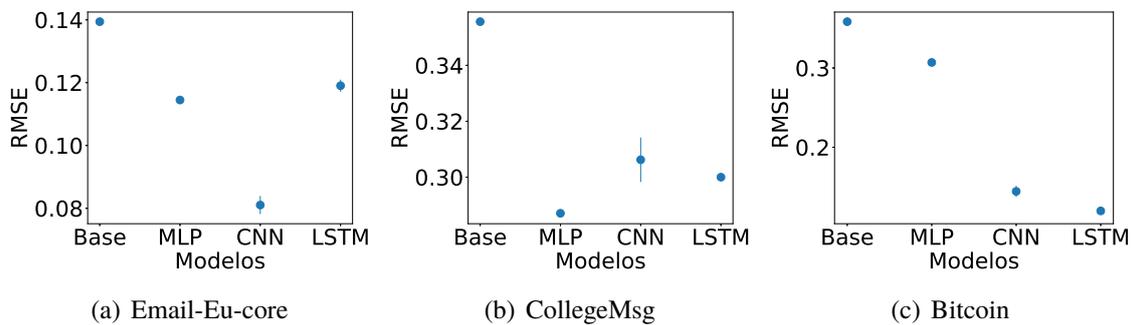
As Figuras 3 e 4 apresentam o RMSE dos modelos. Os melhores modelos considerando o MAE são também os melhores para o RMSE. Um caso particular ocorre para a série de grau médio da base CollegeMsg. Considerando o intervalo de confiança, o resultado do MAE da Figura 1 mostra que a MLP possui o mesmo desempenho da CNN. Entretanto, no RMSE da Figura 3, a MLP apresenta um melhor desempenho que a CNN. Um melhor desempenho no RMSE indica que os erros cometidos pela CNN possuem maior magnitude do que os erros da MLP.

Observa-se que, à exceção do MLP para o grau médio da base CollegeMsg, os

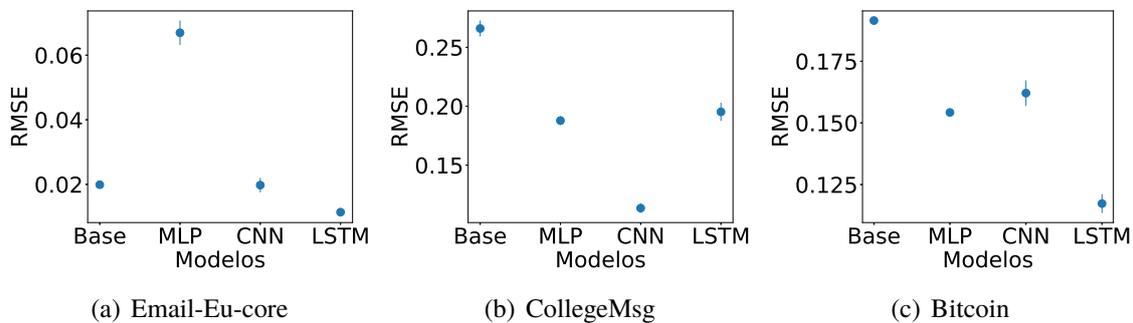
<sup>6</sup><https://scikit-learn.org>



**Figura 2. MAE do Coeficiente de Clusterização**



**Figura 3. RMSE do Grau Médio**



**Figura 4. RMSE do Coeficiente de Clusterização**

melhores modelos para cada série são a CNN ou a LSTM. Ou seja, a maior complexidades dessas arquiteturas resultou em um melhor desempenho na predição das métricas.

As Figuras 5, 6 e 7 mostram exemplos de uma janela de predição de grau médio, das bases de dados Email-Eu-core, CollegeMsg e Bitcoin, respectivamente. Para esses exemplos, utilizou-se o modelo que possui os melhores resultados da análise anterior. Nessas figuras, os pontos marcados como “Entrada” correspondem ao valor real da métrica, enquanto os pontos de “Predição” são as saídas do modelo. É possível notar que os modelos do Email-Eu-core (Figura 5) e do Bitcoin (Figura 7) conseguem rastrear a curva de valores reais. Entretanto, o modelo do CollegeMsg reage pouco às mudanças de comportamento da curva. Isso mostra a importância em utilizar, em um trabalho futuro, outras métricas de avaliação do modelo. Por exemplo, o coeficiente de determinação  $R^2$ , que pode quantificar a variação da saída capturada pelo modelo.

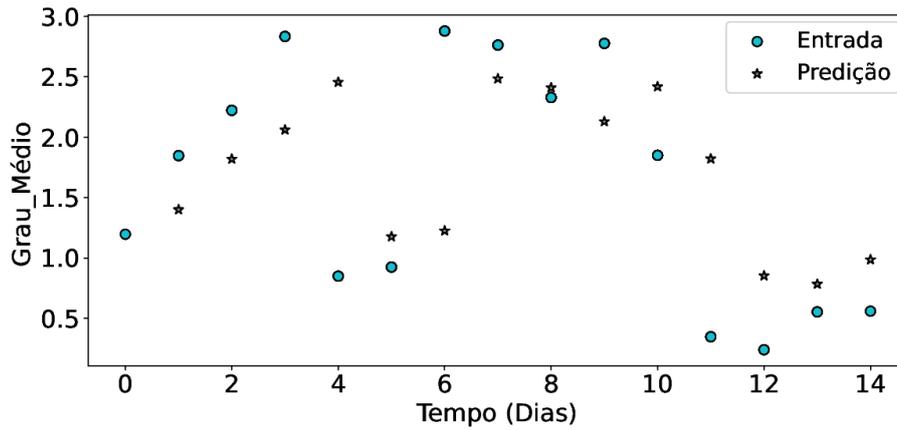


Figura 5. Grau Médio da base Email-Eu-core com a CNN

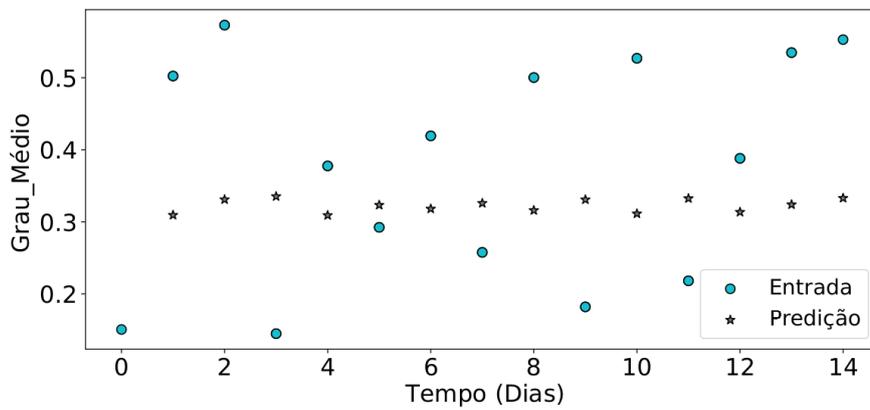


Figura 6. Grau Médio da base CollegeMsg com a MLP

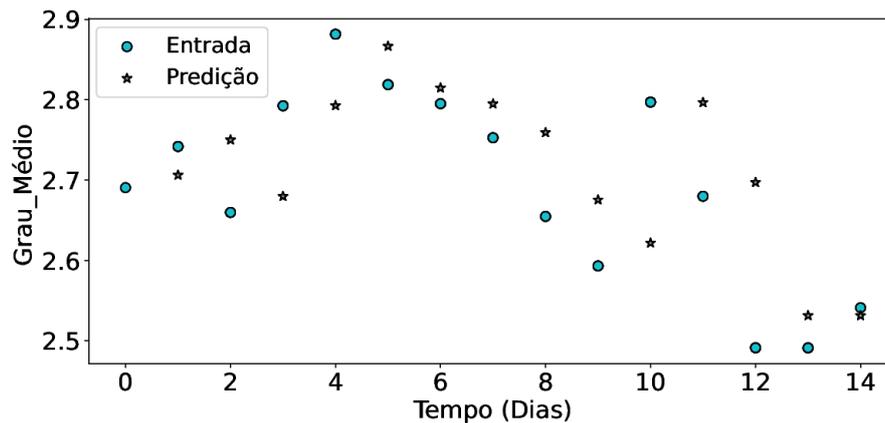


Figura 7. Grau Médio da base Bitcoin com a LSTM

## 6. Conclusões e Trabalhos Futuros

Este trabalho abordou o uso de modelos de redes neurais para predição de métricas em grafos temporais. Para tal, analisou-se o desempenho de uma CNN, de uma LSTM e de uma MLP. Os modelos CNN e LSTM obtiveram os melhores resultados, com exceção apenas de um caso particular, evidenciando a sua aplicabilidade em problemas desse tipo.

O caso particular se refere à previsão do grau médio da base de dados CollegeMsg, que possui erros de previsão, mesmo com resultados satisfatórios nos erros MAE e RMSE. A análise desse caso particular será abordada em trabalhos futuros, incluindo novas métricas de desempenho como o  $R^2$ . Um outro ponto futuro é incluir outros modelos de base para a comparação, como o ARIMA.

## Referências

- Bergstra, J. e Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).
- Binkowski, M., Marti, G. e Donnat, P. (2018). Autoregressive convolutional neural networks for asynchronous time series. Em *International Conference on Machine Learning*, p. 580–589. PMLR.
- Borovykh, A., Bohte, S. e Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*.
- Cem, E. e Sarac, K. (2016). Estimation of structural properties of online social networks at the extreme. *Computer Networks*, 108:323–344.
- Charakopoulos, A., Karakasidis, T., Papanicolaou, P. e Liakopoulos, A. (2014). The application of complex network time series analysis in turbulent heated jets. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 24(2):024408.
- Cui, Z., Henrickson, K., Ke, R. e Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894.
- De Gooijer, J. G. e Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473.
- Frank, R. J., Davey, N. e Hunt, S. P. (2001). Time series prediction and neural networks. *Journal of intelligent and robotic systems*, 31(1):91–103.
- Hegeman, T. e Iosup, A. (2018). Survey of graph analysis applications. *arXiv preprint arXiv:1807.00382*.
- Hewamalage, H., Bergmeir, C. e Bandara, K. (2020). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*.
- Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z. e Zhang, H. (2019). Deep learning with long short-term memory for time series prediction. *IEEE Communications Magazine*, 57(6):114–119.
- Hyndman, R. J. e Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- Kim, S. e Kang, M. (2019). Financial series prediction using attention lstm. *arXiv preprint arXiv:1902.10877*.
- Li, Y., Yu, R., Shahabi, C. e Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

- Lim, B. e Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.
- Makridakis, S., Spiliotis, E. e Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889.
- Mbah, T. J., Ye, H., Zhang, J. e Long, M. (2021). Using lstm and arima to simulate and predict limestone price variations. *Mining, Metallurgy & Exploration*, p. 1–14.
- Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. Relatório técnico, Manubot.
- Nicosia, V., Tang, J., Mascolo, C., Musolesi, M., Russo, G. e Latora, V. (2013). Graph metrics for temporal networks. Em *Temporal networks*, p. 15–40. Springer.
- Nielsen, A. (2019). *Practical time series analysis: prediction with statistics and machine learning*. "O'Reilly Media, Inc."
- Santoro, N., Quattrociochi, W., Flocchini, P., Casteigts, A. e Amblard, F. (2011). Time-varying graphs and social network analysis: Temporal indicators and metrics. *arXiv preprint arXiv:1102.0629*.
- Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K. e Soman, K. (2017). Stock price prediction using lstm, rnn and cnn-sliding window model. Em *2017 international conference on advances in computing, communications and informatics (icacci)*, p. 1643–1647. IEEE.
- Siami-Namini, S., Tavakoli, N. e Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. Em *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, p. 1394–1401. IEEE.
- Siami-Namini, S., Tavakoli, N. e Namin, A. S. (2019). A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. *arXiv preprint arXiv:1911.09512*.
- Tang, J., Liu, F., Zhang, W., Zhang, S. e Wang, Y. (2016). Exploring dynamic property of traffic flow time series in multi-states based on complex networks: Phase space reconstruction versus visibility graph. *Physica A: Statistical Mechanics and its Applications*, 450:635–648.
- Tang, J., Wang, Y., Wang, H., Zhang, S. e Liu, F. (2014). Dynamic analysis of traffic time series at different temporal scales: A complex networks approach. *Physica A: Statistical Mechanics and its Applications*, 405:303–315.
- Yamak, P. T., Yujian, L. e Gadosey, P. K. (2019). A comparison between arima, lstm, and gru for time series forecasting. Em *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, p. 49–55.
- Yan, S. (2015). Understanding lstm networks. *Online, acessado em Janeiro de 2021*, 11.
- Yu, B., Yin, H. e Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.