



PREDIÇÃO DE MÉTRICAS EM GRAFOS TEMPORAIS UTILIZANDO REDES NEURAIS

Daiane de Moura Pereira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Rodrigo de Souza Couto

Rio de Janeiro
Agosto de 2021

PREDIÇÃO DE MÉTRICAS EM GRAFOS TEMPORAIS UTILIZANDO REDES
NEURAIS

Daiane de Moura Pereira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Orientador: Rodrigo de Souza Couto

Aprovada por: Prof. Rodrigo de Souza Couto
Prof. Miguel Elias Mitre Campista
Prof^a. Dianne Scherly Varela de Medeiros

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2021

de Moura Pereira, Daiane

Predição de Métricas em Grafos Temporais Utilizando
Redes Neurais/Daiane de Moura Pereira. – Rio de Janeiro:
UFRJ/COPPE, 2021.

XIII, 46 p.: il.; 29, 7cm.

Orientador: Rodrigo de Souza Couto

Dissertação (mestrado) – UFRJ/COPPE/Programa de
Engenharia Elétrica, 2021.

Referências Bibliográficas: p. 42 – 46.

1. Redes Neurais. 2. Grafos Temporais. 3. Bitcoin. I.
de Souza Couto, Rodrigo. II. Universidade Federal do Rio de
Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Aos meus pais.

Agradecimentos

Primeiramente, quero agradecer aos meus pais, Elizabeth e Dorvalino, pelo apoio, amor e dedicação. Eu não teria conseguido sem a ajuda de vocês. Obrigada por terem acreditado em mim e por todo o investimento que fizeram na minha educação. Vocês sempre me apoiaram e me aceitaram da forma que eu sou, sem pedir nada em troca, apenas pelo prazer de me ver realizando os meus sonhos, eu não tenho palavras para expressar o quanto sou grata. Também, agradeço ao meu irmão, Douglas, pelo incentivo, amor e carinho que sempre teve por mim. Bem como, agradeço aos meus dois melhores amigos, Pericles e Lucas, pela parceria ao longo dos anos e força nos momentos que precisei. Sei que posso contar com vocês nos momentos felizes e nos momentos difíceis e isso faz uma diferença enorme na minha vida. Muito obrigada pelo amor e pela amizade de vocês.

Agradeço também aos amigos do Grupo de Teleinformática e Automação. Foi um prazer enorme esses anos de estudo ao lado de vocês que foram muito importantes na minha pesquisa.

Agradeço aos professores do Grupo de Teleinformática e Automação, Luís Henrique Maciel Kosmowski Costa, Miguel Elias Mitre Campista e Otto Carlos Duarte, muito obrigada pela contribuição na minha formação.

Em especial, agradeço ao meu orientador, Rodrigo de Souza Couto, por toda ajuda ao longo desses anos. Muitas vezes eu achei que não seria capaz e mesmo sem saber, você sempre soube me mostrar um caminho e me motivar. Muito obrigada por toda a ajuda e por todo o conhecimento compartilhado durante essa trajetória.

Aos membros da banca, obrigada por aceitarem analisar este trabalho.

Agradeço também à secretaria do Programa de Engenharia Elétrica.

À FAPERJ e CAPES pelo financiamento do projeto.

Por fim, agradeço a todos aqueles que, de alguma forma, contribuíram com este trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PREDIÇÃO DE MÉTRICAS EM GRAFOS TEMPORAIS UTILIZANDO REDES NEURAS

Daiane de Moura Pereira

Agosto/2021

Orientador: Rodrigo de Souza Couto

Programa: Engenharia Elétrica

Grafos temporais são amplamente utilizados na modelagem de sistemas nos quais o período de interação entre os nós é importante, como em redes de computadores, redes rodoviárias e redes sociais. Por esse motivo, 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. Visando preencher essa lacuna, esta dissertação aborda o uso de modelos de redes neurais para predição de 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, mostrando desempenho satisfatório, principalmente para as redes neurais recorrentes e convolucionais. Visando apresentar um exemplo real de aplicabilidade da predição de métricas de grafos temporais, é realizado um estudo de caso utilizando a *blockchain* do Bitcoin. Nesse estudo de caso, a *blockchain* é modelada como um grafo temporal e é proposta uma métrica baseada na centralidade de grau para identificar períodos de comportamento malicioso na rede. Utilizam-se os modelos de redes neurais propostos para realizar a predição dessa métrica. Como resultado, mostra-se que as redes neurais são capazes de prever o comportamento da métrica. Novamente, as redes neurais recorrentes e convolucionais apresentaram o melhor desempenho.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PREDICTION OF METRICS IN TEMPORAL GRAPHS USING NEURAL NETWORKS

Daiane de Moura Pereira

August/2021

Advisor: Rodrigo de Souza Couto

Department: Electrical Engineering

We use temporal graphs to model systems where the duration of the interaction between nodes is important, such as computer networks, road networks, and social networks. For this reason, basic metrics calculated for temporal graphs, such as the mean degree and the clustering coefficient, are used in several domains, such as in the characterization of the traffic flow in transport systems. Despite the applicability of these metrics, the literature does not address techniques to predict their temporal evolution. Aiming to fill this gap, we use neural network models to predict temporal graph metrics. We analyze the performance of a multilayer perceptron, a recurrent neural network, and a convolutional neural network. These models are compared with a baseline model, showing satisfactory performance, mainly for recurrent and convolutional neural networks. We also give a real example of the applicability of predicting temporal graphs metrics by analyzing a case study using the Bitcoin blockchain. In this case study, we model the blockchain as a temporal graph, and we propose a metric based on degree centrality to identify periods of malicious behavior in the network. The proposed neural network models are used to predict this metric. As a result, it is shown that neural networks can predict the behavior of the metric. Again, recurrent and convolutional neural networks performed best.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
1 Introdução	1
2 Redes Neurais	4
2.1 Modelo base e redes neurais	4
2.1.1 Modelo base	4
2.1.2 <i>Perceptron</i> multicamadas	5
2.1.3 Rede neural recorrente	6
2.1.4 Rede neural convolucional	8
3 Trabalhos Relacionados	10
3.1 Predição em séries temporais	10
3.2 Análise de grafos na <i>blockchain</i> do Bitcoin	12
4 Metodologia	15
4.1 Seleção das bases de dados	15
4.1.1 Bitcoin	16
4.1.2 Email-Eu-core	16
4.1.3 CollegeMsg	17
4.2 Criação dos grafos	17
4.3 Cálculo das métricas	18
4.4 Divisão em treino e teste	19
4.5 Tratamento dos dados	20
4.6 Treino e ajuste dos modelos	20
4.6.1 Ajuste de hiperparâmetros	20
5 Análises e Resultados da Predição dos Grafos Temporais	24
5.1 Bitcoin	25
5.2 Email-Eu-core	26

5.3	CollegeMsg	28
6	Estudo de Caso: Bitcoin	31
6.1	Análise do ranking	31
6.2	Análise de estabilidade dos nós	34
6.2.1	Método 1 para inserir novas transações	35
6.2.2	Método 2 para inserir novas transações	36
6.2.3	Método 3 para inserir novas transações	37
6.3	Predição da Repetição de Usuário	38
7	Conclusões e Trabalhos Futuros	40
	Referências Bibliográficas	42

Lista de Figuras

2.1	Modelo base.	5
2.2	Representação de uma MLP.	6
2.3	Célula LSTM.	7
2.4	Exemplo da arquitetura de uma LSTM.	8
2.5	Exemplo convolução 1 dimensão.	9
4.1	Metodologia do Estudo.	15
4.2	Exemplo de um <i>snapshot</i>	18
4.3	Exemplo de janelas deslizantes.	19
4.4	Arquitetura da MLP.	21
4.5	Arquitetura da LSTM.	22
4.6	Arquitetura da CNN.	23
5.1	MAE para as séries temporais do Bitcoin.	25
5.2	RMSE para as séries temporais do Bitcoin.	25
5.3	Grau médio da base Bitcoin com a LSTM.	26
5.4	Coefficiente de clusterização da base Bitcoin com a LSTM.	27
5.5	MAE para as séries temporais do Email-Eu-core.	27
5.6	RMSE para as séries temporais do Email-Eu-core.	27
5.7	Grau médio da base Email-Eu-core com a CNN.	28
5.8	Coefficiente de clusterização da base Email-Eu-core com a LSTM.	28
5.9	MAE para as séries temporais da base CollegeMsg.	29
5.10	RMSE para as séries temporais da base CollegeMsg.	29
5.11	Grau médio da base CollegeMsg com a MLP.	30
5.12	Coefficiente de clusterização da base CollegeMsg com a CNN.	30
6.1	Repetição de Usuário e variação no preço do Bitcoin.	33
6.2	Método 1 para inserir novas transações.	36
6.3	Método 2 para inserir novas transações.	37
6.4	Método 3 para inserir novas transações.	38
6.5	Predição da Repetição de Usuário utilizando LSTM.	39

Lista de Tabelas

4.1	Características das bases de dados.	17
4.2	Configuração de hiperparâmetros para a MLP e a LSTM.	21
4.3	Configuração de hiperparâmetros para a CNN.	22
6.1	Média da Repetição de Usuário.	34
6.2	Comparação entre as métricas dos modelos.	39

Lista de Abreviações

ARIMA *Auto-Regressive Integrated Moving Average* - Auto Regressivo Integrado de Médias Móveis

AR *Auto-Regressive* - Auto Regressivo

CNN *Convolutional Neural Network* - Redes Neurais Convolucionais

COPPE Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

ELTE *Eötvös Loránd University*

LSTM *Long Short Term Memory* - Memória Longa de Curto Prazo

MAE *Mean Absolute Error* - Erro Médio Absoluto

MAPE (*Mean Absolute Percentage Error* - Porcentagem do Erro Médio Absoluto

MASE (*Mean Absolute Scaled Error* - Erro Médio Absoluto Escalado

MA *Moving Average* - Médias Móveis

MLPs *MultiLayer Perceptrons* - Perceptron Multicamadas

MSE *Mean Squared Error* - Erro Quadrático Médio

ReLU *Rectified Linear Unit*

RMSE *Root Mean Squared Error* - Raiz Quadrada do Erro Médio Absoluto

sMAPE (*symmetric Mean Absolute Percentage Error* - Erro Médio Absoluto Percentual Simétrico

SNAP *Stanford Network Analysis Project*

UCI *University of California, Irvine*

WAPE (*Weighted Absolute Percentage Error* - Erro Percentual Absoluto Ponderado)

Capítulo 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 [1]. Algumas análises de redes complexas envolvem grafos temporais, que são modificados ao longo do tempo [2]. Em outras palavras, os nós e os enlaces existentes no tempo t podem ser diferentes no tempo $t + 1$.

Os grafos temporais apresentam várias aplicações na literatura. Em [3] é utilizada a representação por grafos para modelar séries temporais. Investigam-se características dos grafos ao longo do tempo, para 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 [4] e [5], 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. O objetivo desses trabalhos é verificar se a análise de redes complexas é eficiente para melhorar a acurácia de sistemas de predição e controle do tráfego. Os autores concluem que é possível aumentar a acurácia dos sistemas de predição com a abordagem de redes complexas. Em [6] os autores abordam o dilema do prisioneiro, que é um problema conhecido da teoria dos jogos. Os autores modelam o problema como um grafo e apenas com análises do grau médio e do coeficiente de clusterização conseguem medir a taxa de colaboração entre os agentes.

Apesar da sua importância, os trabalhos da literatura não abordam a predição de métricas básicas. Dessa forma, o objetivo desta dissertação 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* [7]. 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 redes neurais. Os modelos são então treinados com o histórico dessas séries temporais. Nesse treinamento, os modelos aprendem os padrões dessas séries com o objetivo de aplicar esse conhecimento na predição de seus valores futuros.

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. Em [8] é realizado um estudo sobre a evolução de métricas como a densidade e o coeficiente de clusterização em grafos temporais. Apesar de não realizar a predição de valores futuros, o estudo analisa a evolução dessas métricas ao longo do tempo e estuda se as curvas apresentam comportamento de subida ou descida. Os autores apontam que o conhecimento desse comportamento já é capaz de auxiliar na tomada de decisões. Dessa forma, modelos capazes de fornecer informações mais precisas do que comportamento de subida ou descida podem tornar a tomada de decisões mais precisa. Por esse motivo, esta dissertação emprega modelos de redes neurais para a predição dessas métricas.

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) [9]. Essas redes são comparadas com um modelo base, que consiste em repetir o último valor de entrada como previsão do futuro. As três redes neurais e o modelo base são aplicados em três bases de dados, Email-Eu-core e CollegeMsg do projeto SNAP (*Stanford Network Analysis Project*)¹ e no histórico de transações do Bitcoin. Os resultados deste trabalho mostram que é possível utilizar modelos de rede neural para prever as métricas consideradas, obtendo um erro de predição inferior ao obtido no 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.

O histórico de transações do Bitcoin também é utilizado em uma segunda etapa desta dissertação, quando é realizado um estudo de caso para identificação de atividade maliciosa no Bitcoin de manipulação de mercado. O estudo de caso tem foco nas suspeitas de atividades maliciosas da corretora de Bitcoin Mt.Gox. Uma corretora de Bitcoin é o local no qual as pessoas podem comprar, vender ou armazenar o seu dinheiro, Bitcoin ou demais criptomoedas. Até o início do ano de 2014, a Mt.Gox era a maior corretora de Bitcoin. No entanto, nesse mesmo ano, a Mt.Gox entrou com pedido de falência, gerando prejuízo para os seus clientes, que perderam os seus Bitcoins. Após esse pedido de falência, os dados internos da Mt.Gox, contendo o histórico de compras e vendas de Bitcoins,

¹<https://snap.stanford.edu/data>

foram invadidos e vazados. Esses dados foram então utilizados em alguns estudos que encontraram fortes indícios de que a Mt.Gox fazia manipulação de mercado. Ou seja, a Mt.Gox reportava um número excessivo de compras de Bitcoin para simular uma alta na demanda e aumentar o preço de mercado da criptomoeda [10].

Apesar dos estudos apontarem evidências de manipulação da Mt. Gox, o fato de serem feitos com uma base de dados interna vazada impossibilita que análises similares sejam realizadas em outros períodos de tempo ou com outras corretoras. Por esse motivo, no estudo de caso desta dissertação, analisa-se a *blockchain* do Bitcoin, durante o período no qual a corretora Mt.Gox foi acusada de realizar a maior parte das suas operações fraudulentas [10]. O objetivo é verificar se manipulações de mercado alteram a dinâmica da *blockchain*.

Para realizar o estudo de caso, realiza-se uma análise de grafos na rede do Bitcoin. Adota-se a hipótese de que a manipulação de mercado feita pela Mt.Gox muda o comportamento esperado dos nós da *blockchain*, gerando instabilidade nas métricas de grafo. Para investigar essa hipótese, é proposta uma métrica baseada na centralidade de grau. Essa métrica é simples e visa analisar como os nós com melhor classificação mudam ao longo do tempo e examinar se essas mudanças podem estar relacionadas à atividade maliciosa da Mt.Gox e com o aumento do preço do Bitcoin no período.

As análises mostram que fora do período no qual a Mt.Gox é suspeita de manipulação de mercado, os nós mais bem classificados do grafo do Bitcoin sofrem poucas alterações ao longo do tempo. Ou seja, os mesmos nós são responsáveis por um alto número de transações, mês a mês. No entanto, durante o período de suspeita de manipulação, a métrica apresenta alteração no seu comportamento, com menos nós sendo repetidos no ranking. Com base nesses resultados e no fato de as redes neurais serem capazes de prever o coeficiente de clusterização e o grau médio, empregam-se os modelos de redes neurais para prever os valores futuros da métrica proposta. Dessa forma, investiga-se a possibilidade de prever o início de um período de instabilidade na rede.

Esta dissertação está dividida da seguinte forma. O Capítulo 2 apresenta os conceitos básicos necessários para a compreensão do trabalho, que são os modelos de redes neurais utilizados. Em seguida, o Capítulo 3 apresenta os principais trabalhos relacionados. Ele é dividido em duas partes, a primeira é voltada aos trabalhos relacionados à predição de métricas em grafos temporais. Já a segunda parte tem foco no estudo de caso utilizando a rede do Bitcoin. Os dois capítulos seguintes focam apenas a predição de métricas em grafos temporais, sendo o Capítulo 4 voltado para a apresentação da metodologia adotada no estudo e o Capítulo 5 dedicado aos resultados e análises. O Capítulo 6 apresenta o estudo de caso do Bitcoin. Finalmente, o Capítulo 7 apresenta conclusões e trabalhos futuros.

Capítulo 2

Redes Neurais

Neste capítulo são apresentados os modelos de redes neurais utilizados para a predição das séries temporais, além do modelo empregado como base de comparação.

Para a predição das séries temporais, esta dissertação utiliza modelos de aprendizado supervisionado. Isso significa que, durante a etapa de treino do modelo, são fornecidos conjuntos de entrada e saída conhecidos. Esses conjuntos são utilizados pelo modelo para ajustar os parâmetros e aprender as relações entre as entradas e saídas da rede.

2.1 Modelo base e redes neurais

Esta seção apresenta os modelos utilizados para predição das séries temporais descritas. Utilizam-se três tipos de redes neurais: *Perceptron* Multicamadas (MLP - *MultiLayer Perceptron*), Memória Longa de Curto Prazo (LSTM - *Long Short Term Memory*) e Redes Convolucionais (CNN - *Convolutional Neural Network*). Além disso, utiliza-se um modelo base para analisar o desempenho das redes neurais.

2.1.1 Modelo base

Essa solução é simples e implementada em diversos sistemas de previsão [11]. Nesse 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 $y(t)$. A Figura 2.1 apresenta um exemplo de utilização do modelo base. Observa-se que a curva de predições é a própria curva de entrada deslocada para a direita. Essa solução é utilizada neste trabalho como base para comparar o desempenho dos diversos modelos de redes neurais.

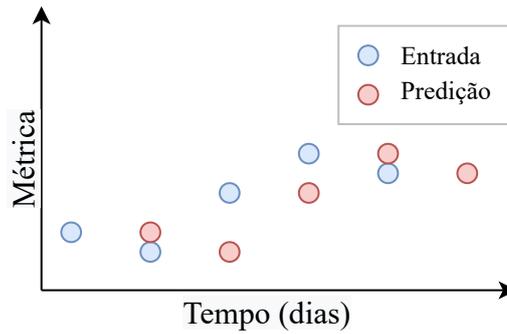


Figura 2.1: Modelo base.

2.1.2 Perceptron multicamadas

Uma MLP é uma rede neural baseada no modelo de representação de um neurônio natural proposto por Rosenblatt [12]. A Figura 2.2(a) mostra a representação de um neurônio artificial. Ele é composto por um vetor de entrada x , um vetor de pesos w , que ajusta os seus valores de acordo com a importância de cada entrada, e uma função de ativação $u(x)$. A constante um é utilizada para permitir operações matriciais com o peso w_i . A operação matemática realizada pelo neurônio é apenas uma soma ponderada das entradas e dos pesos e a aplicação de uma função de ativação. A função de ativação é responsável por mapear o valor da soma ponderada para os limites desejados, como por exemplo, entre zero e um, gerando assim a saída $\hat{y}(t + 1)$.

A estrutura de uma MLP é composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída [13]. Cada camada é composta por um determinado número de neurônios. Além disso, as camadas de uma MLP são totalmente conectadas. Ou seja, a saída de cada neurônio de uma camada é usada como a entrada de cada neurônio da camada seguinte.

A Figura 2.2(b) apresenta um exemplo de uma MLP. Esse exemplo é composto por duas camadas ocultas, contendo três neurônios cada, uma camada de entrada, com um neurônio para cada variável de entrada, e uma camada de saída, contendo um neurônio. A camada de entrada recebe o conjunto de dados. No exemplo da Figura 2.2(b), o valor de saída predito é determinado com base nas duas variáveis de entrada x_1 e x_2 . As camadas ocultas são as camadas que não possuem acesso direto à saída da MLP. Já a camada de saída é responsável por fornecer a saída da rede neural, ou seja, a predição $\hat{y}(t + 1)$.

As setas indicam o fluxo do sinal de entrada, que se propaga para frente, passando pelas camadas ocultas em direção à camada de saída. Durante a etapa de treinamento, esse fluxo do sinal de entrada até a saída é chamado de fase *forward*.

Após a fase *forward*, é calculado o erro entre a saída da última camada e a saída desejada. Esse erro é então retro propagado em direção a camada de entrada. Em cada camada são calculados os valores dos gradientes para cada peso. Cada neurônio então ajusta os

seus pesos w com base no gradiente, de modo a minimizar o erro na saída. Esse fluxo do erro em direção a entrada é chamado de fase *backward*.

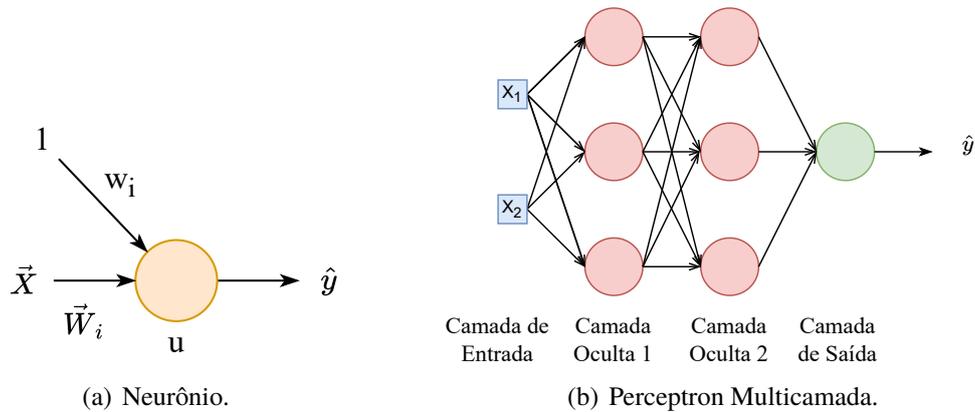


Figura 2.2: Representação de uma MLP.

Apesar de ser aplicada com sucesso em vários estudos para predição de séries temporais [14–16], a MLP possui como desvantagem o fato de não ser capaz de levar em consideração a ordem temporal das suas entradas [17]. Por exemplo, suponha uma MLP construída com o objetivo de prever a saída atual, com base em vetores de entrada compostos por quatro observações sequenciais da série temporal. Durante o processo de treinamento, a MLP não é capaz de ajustar os seus pesos levando em consideração a ordem das entradas. Isso significa que, para a MLP, não há distinção se cada vetor de entrada possui os seus valores na ordem da série, na ordem inversa, ou embaralhado. Esse comportamento das MLPs pode limitar o desempenho ao lidar com séries temporais [17].

Para lidar com a limitação exposta, outros tipos de redes neurais são analisados neste trabalho. São elas as redes neurais recorrentes (*Recurrent Neural Network* - RNN), que são um tipo de rede neural capaz de lidar com a relação temporal entre as entradas, e as redes convolucionais (*Convolutional Neural Network* - CNN), que utilizam a operação de convolução para capturar padrões repetidos nos dados e, por esse motivo, também são estudadas para predição de séries temporais.

2.1.3 Rede neural recorrente

Uma rede neural recorrente (*Recurrent Neural Network* - RNN) é um tipo de rede neural capaz de capturar a relação temporal entre as entradas. Isso ocorre devido ao fato de esse tipo de rede possuir uma realimentação, na qual as saídas no tempo $t - 1$ são alimentadas na rede para afetar o resultado no tempo t . Dessa forma, uma RNN toma decisões baseadas não só na entrada atual, como também nas informações de entradas passadas.

Apesar de possibilitar que a relação temporal seja levada em consideração nas predições, as RNNs são afetadas pelo problema de desaparecimento de gradiente. O desa-

parecimento de gradiente ocorre pois, quanto maior for a sequência, menor será o valor do gradiente para as entradas mais antigas. Dessa forma, na atualização dos parâmetros no treinamento, as entradas mais antigas tornam-se insignificantes. Assim, as RNNs não conseguem guardar as informações sobre entradas anteriores por longos períodos de tempo [9]. Em [18] é demonstrado que, para sequências superiores a 10 entradas, as RNNs não são mais capazes de considerar as entradas mais antigas.

Este trabalho utiliza um tipo de RNN denominado Memória Longa de Curto Prazo (*Long Short Term Memory* - LSTM). As LSTMs surgiram como uma alternativa ao problema de desaparecimento de gradiente [19]. A Figura 2.3 mostra a estrutura básica de uma célula LSTM. Cada célula de uma LSTM tem seus estados divididos em dois, $h_{(t)}$ (estado interno), que pode ser visto como a saída da célula, e $c_{(t)}$ (estado da célula), que controla a memória da célula.

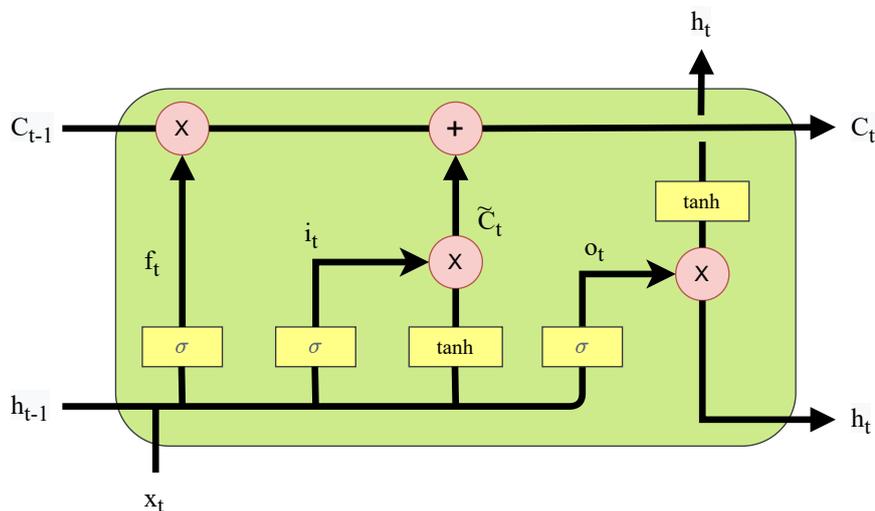


Figura 2.3: Célula LSTM.

Cada célula LSTM possui também quatro portas que controlam o fluxo de informação nas células. Três dessas portas estão relacionados ao estado da célula e controlam o quanto das informações antigas serão esquecidas (f_t), quando informações novas serão adicionadas (i_t) e quais informações serão escritas (g_t) no estado da célula. Já a o_t está relacionada com estado interno e controle do que será encaminhado como saída.

A Figura 2.4 mostra um exemplo de arquitetura de uma LSTM. Esse exemplo possui duas camadas LSTM, com três células cada e uma camada de saída com uma única unidade totalmente conectada. Nesse exemplo, são utilizados três valores de entrada x_1, x_2 e x_3 para prever um valor no futuro, $\hat{y}(t + 1)$.

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 previsão de preço de ações, reconhecimento de fala e outras envolvendo séries temporais [20].

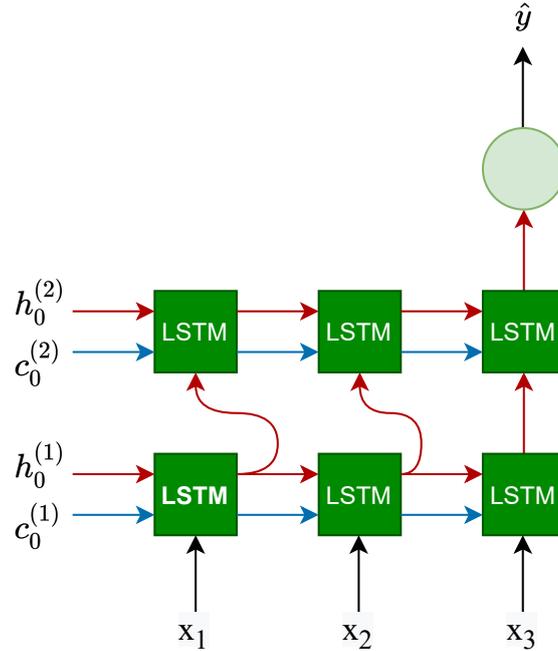


Figura 2.4: Exemplo da arquitetura de uma LSTM.

2.1.4 Rede neural convolucional

Uma rede neural convolucional (*Convolutional Neural Network* - CNN) é um tipo de rede neural muito utilizado em problemas de classificação e em problemas de visão computacional como o reconhecimento de imagens. Apesar de serem mais comuns em tarefas de classificação, as CNNs também podem ser utilizadas para predição de séries temporais. Na classificação de imagens monocromáticas, as camadas convolucionais 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 [21].

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 dos filtros e sua quantidade 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 convolucional 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.

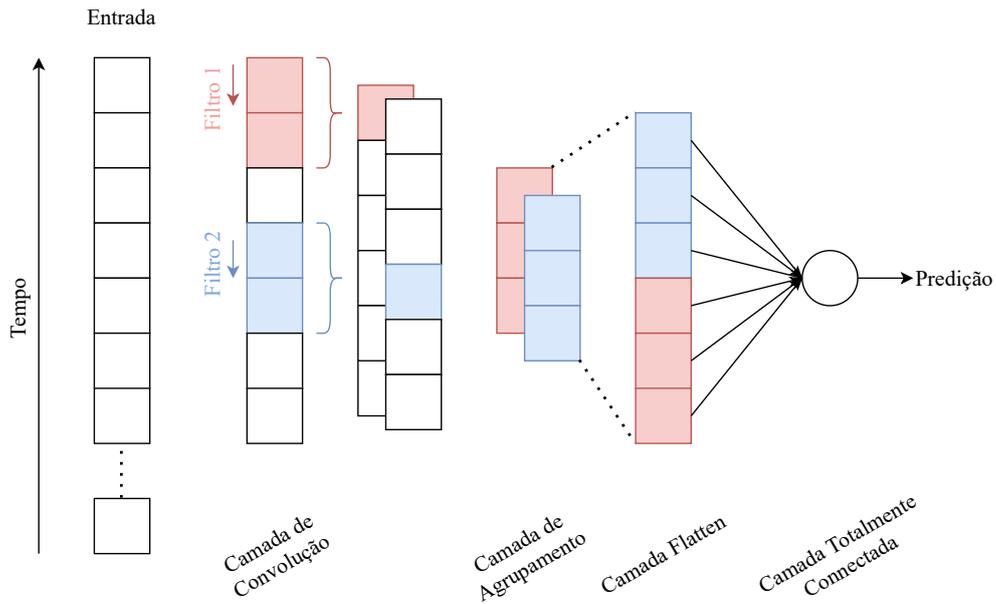


Figura 2.5: Exemplo convolução 1 dimensão.

A Figura 2.5 apresenta um exemplo simples de convolução aplicado a dados de uma dimensão. Nesse exemplo, as últimas sete medições da série temporal formam a entrada. São aplicados dois filtros de tamanho dois na camada de convolução. Em seguida, a camada de agrupamento diminui a dimensão dos dados e a camada *flatten* transforma a saída em um vetor que é a entrada para uma camada totalmente conectada, contendo apenas um neurônio. A saída dessa camada totalmente conectada fornece a saída $\hat{y}(t + 1)$.

Capítulo 3

Trabalhos Relacionados

Neste capítulo apresentam-se os principais trabalhos relacionados a esta dissertação. O capítulo é dividido em duas partes. Na primeira são apresentados os principais trabalhos sobre a predição de valores em séries temporais. Já a segunda aborda os principais trabalhos envolvendo a rede do Bitcoin, relacionados ao estudo de caso.

3.1 Predição em séries temporais

O problema de prever valores 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*) [22]. O ARIMA é um modelo que assume que os valores futuros de uma série temporal podem ser preditos com base em uma função linear dos seus valores passados e de erros aleatórios. Para o uso do ARIMA, é necessário que a série temporal seja estacionária, caso não seja, é preciso transformá-la. Ou seja, a série não pode possuir tendência ou sazonalidade. Esse modelo utiliza três componentes para realizar a predição, são eles: O componente auto regressivo (*Auto-Regressive - AR*), o componente de integração (*Integrated - I*) e o componente para as médias móveis (*Moving Average - MA*). O componente AR contribui com a predição fornecendo o cálculo de uma combinação linear dos valores passados da série. O componente I contribui realizando a diferenciação, quando necessário. A diferenciação é um processo que possibilita tornar uma série não estacionária em estacionária. Esse processo é baseado no cálculo da diferença entre uma observação e outra. Já o componente MA fornece a média móvel dos erros de predição passados. O ARIMA realiza a predição das séries temporais por meio da combinação desses três componentes.

Por muitos anos, esse método foi considerado o mais apropriado para prever o comportamento de séries temporais [23, 24]. Além disso, o ARIMA ainda é muito utilizado e apresenta bom desempenho em alguns estudos. Em [25] e [26], são propostas alternativas para o ARIMA. Em [25] compara-se o desempenho de uma rede neural, a LSTM,

e do ARIMA na predição do preço do calcário. A LSTM utilizada possui sete camadas, função de ativação ReLU, *dropout* de 0,2 e otimizador ADAM. Os métodos foram comparados com base no erro quadrático médio (*Mean Squared Error* - MSE) e o ARIMA apresentou o menor erro. Já em [26] o método ARIMA é comparado com diversos modelos de redes neurais, como a MLP, a RNN e a LSTM, na predição de bases de dados financeiras. As métricas de avaliação utilizadas foram o erro médio absoluto percentual simétrico (*symmetric Mean Absolute Percentage Error* - sMAPE) e o erro médio absoluto escalado (*Mean Absolute Scaled Error* - MASE). Novamente, o método ARIMA foi superior.

Apesar dos estudos citados demonstrarem que o ARIMA obteve melhor desempenho na predição, os autores apontam que alguns hiperparâmetros importantes das redes neurais foram selecionados manualmente, como o número de camadas e de neurônios. Já os parâmetros do ARIMA foram ajustados utilizando funções que realizam esse ajuste automaticamente. Isso pode influenciar negativamente o desempenho das redes neurais.

Embora seja muito utilizado, o ARIMA apresenta limitações, como o fato de não apresentar desempenho satisfatório quando há uma relação não linear entre as variáveis. Para superar essa limitação, a segunda divisão da literatura utiliza métodos não lineares, como as redes neurais profundas [23, 27–29].

Em [27] 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. Além disso, os resultados foram também comparados com o ARIMA. Foi utilizado o erro percentual para comparar o desempenho dos modelos. Nesse trabalho, as CNNs obtiveram os melhores resultados, ou seja, apresentaram o menor erro percentual na predição. Em [23] e [28] são realizadas análises comparando o ARIMA e a rede neural LSTM na predição do preço de ações. A métrica utilizada para a análise do desempenho é a raiz do erro quadrático médio (*Root-Mean-Square Error* - RMSE). Nesses estudos, os modelos de redes neurais obtêm um menor erro na predição das séries temporais.

Já em [29] é proposta uma arquitetura para a LSTM, no domínio de predição de produção de petróleo. No trabalho é utilizado um algoritmo genético para determinar a arquitetura da rede proposta. O modelo proposto é comparado com o ARIMA e outras arquiteturas da LSTM. A métrica utilizada para comparar o desempenho dos modelos é a RMSE. O modelo proposto apresenta melhor desempenho que os demais modelos, inclusive o ARIMA. O trabalho [30] realiza um estudo sobre o uso do modelo LSTM para a predição de séries temporais e propõe um novo modelo, baseado na LSTM e no uso de grafos esparsos. Esses grafos possuem poucos enlaces em relação à quantidade de nós. O objetivo do modelo proposto é reduzir a complexidade computacional da LSTM. O modelo proposto é testado na predição de tráfego em redes de telecomunicações. É utilizada a métrica RMSE para compará-lo com outros modelos de redes neurais, entre eles a LSTM e o método ARIMA. 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, [31] e [21] consideram séries temporais financeiras, enquanto [32] aborda a previsão do consumo de energia elétrica. Em [17] é apresentado um estudo sobre os principais trabalhos presentes na literatura sobre a previsão de séries temporais, utilizando rede neural. Além disso, é realizado um estudo experimental comparando 50 mil bases de dados, de diversos domínios, e várias arquiteturas de redes neurais. A métrica utilizada para a comparação foi o erro percentual absoluto ponderado (Weighted Absolute Percentage Error - WAPE). O estudo aponta que a LSTM e a CNN foram as redes neurais que tiveram os melhores resultados. Já em [33] é apresentado um *survey* sobre o uso de métodos de aprendizado de máquina em problemas de previsão de séries temporais.

Há na literatura diversos trabalhos que realizam previsão em grafos temporais. Por exemplo, [34] e [35] abordam grafos que modelam o tráfego de veículos em rodovias. Em [34] é proposto um novo modelo de rede neural para a previsão do tráfego em redes rodoviárias. O modelo é baseado nas redes neurais convolucionais. São utilizadas três métricas para comparar os modelos, o MAE, o MAPE e o RMSE. O modelo proposto obteve melhor desempenho ao ser comparado com o ARIMA e com modelos de rede neurais, como a MLP e a LSTM. Já em [35] é proposto um novo modelo de rede neural, baseado na convolução de grafos e na LSTM. O modelo proposto obteve resultado superior ao ser comparado com o ARIMA e com modelos de redes neurais, como a MLP e a LSTM, utilizando o MAE, o MAPE e o RMSE.

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 [5] e para previsão do comportamento de redes sociais [36]. Dessa forma, este trabalho preenche uma lacuna da literatura, abordando a previsão de métricas básicas de grafos temporais. Mais especificamente, realiza-se a previsão do grau médio e do coeficiente de clusterização em grafos temporais, analisando o desempenho de diferentes modelos de redes neurais nessa tarefa.

3.2 Análise de grafos na *blockchain* do Bitcoin

Analisar características de rede complexas do ambiente Bitcoin é um tópico de pesquisa bem explorado na literatura. Nesta dissertação é realizado um estudo de caso com a *blockchain* do Bitcoin. Para isso, as transações do Bitcoin são modeladas como um grafo

não direcionado e é estudado o comportamento desse grafo ao longo do tempo, com base em métricas básicas de grafo. O objetivo dessa análise é identificar períodos de comportamento malicioso na rede. Por fim, é proposta uma métrica para identificar esse comportamento malicioso e são utilizados modelos de redes neurais para prever o comportamento da métrica proposta ao longo do tempo. Os principais estudos que serviram de base para o estudo de caso são apresentados a seguir.

O estudo conduzido em [37] modela o histórico de transações do Bitcoin como um grafo não direcionado e explora as características da rede, como a distribuição de grau e o coeficiente de clusterização. Os autores estudaram também o fluxo de Bitcoins e concluem que a riqueza dos usuários com saldos maiores aumenta mais rápido do que a riqueza de usuários com baixo saldo.

Já em [38, 39] são estudados os primeiros anos de existência do Bitcoin. Os autores analisam diversas características do Bitcoin, como a sua distribuição entre países e como a rede evolui ao longo do tempo. Além disso, características do fenômeno de mundo pequeno são investigadas e confirmadas. Os autores também estudaram os *outliers* nas distribuições em grau e em grau externo. Maesa *et al.* consideram como *outliers* os nós com o grau de entrada ou de saída com pelo menos uma ordem de magnitude maior do que o grau médio. Ambos os comportamentos são associados à presença de transações artificiais na rede [40, 41]. Um exemplo do que os autores consideram como transação artificial são cadeias de transferências nas quais um nó transfere a quantidade exata de 0,00001 BTC para 101 outros nós, à exceção de um nó que recebe valor superior e repete o processo enviando 0,00001 BTC para outros 101 nós.

Um outro tópico fruto de interesse na rede Bitcoin é a análise de fraudes e manipulação de mercado. Em [10], os dados vazados da corretora Mt.Gox são usados para verificar a existência de manipulação de mercado na rede do Bitcoin. Os autores analisam os dados interno dessa corretora, vazados em 2014, e indicam suspeita de atividade maliciosa. Os autores identificaram dois robôs que realizavam um grande número de compra de Bitcoins com valores abaixo do valor de mercado. Esses robôs foram chamados Robôs Willy e Markus. O artigo aponta evidências de que o aumento do preço do Bitcoin no final de 2013 estava relacionado à atividade desses robôs. Em 2017, durante um julgamento no Japão, o CEO da Mt. Gox confirmou que a empresa era responsável pelas atividades do robô Willy [42].

Chen *et al.* também analisam a manipulação de mercado pela corretora Mt.Gox [43]. Nesse estudo, os autores realizam uma investigação das transações do Bitcoin por meio de análise de grafos. Eles também utilizam os dados vazados da Mt.Gox para buscar evidências de atividade maliciosa dessa corretora. Nessa análise, as contas dos clientes da Mt.Gox são classificadas em três categorias (extremamente alta, extremamente baixa e normal). Essas categorias são definidas com base no valor que cada conta pagou pelos Bitcoins e a referência é o histórico de preço diário do Bitcoin. Os autores consideram que

as contas classificadas como extremamente alta e extremamente baixa são contas anormais e verificam que elas possuem características diferentes quando comparadas com as contas normais. Depois disso, eles analisam o conjunto de dados procurando padrões nas transações feitas pelas contas anormais. Os autores descobrem a existência de muitos enlaces para si próprios, transações unidirecionais, triângulos e outras estruturas e concluem que esses padrões são fortes evidências de manipulação de preços. A maior parte dessas transações anormais ocorreu nos últimos 12 meses de operação da Mt.Gox, com presença mais relevante nos últimos seis meses de 2013.

Embora as investigações realizadas em [10] e [43] deem contribuições essenciais para os estudos sobre manipulação de mercado nas corretoras de Bitcoins, elas focam a análise na base de dados vazada da Mt.Gox. Como os detalhes sobre transações internas em outras corretoras não estão disponíveis (ou seja, transações que ocorrem na corretora e não aparecem no blockchain), sua análise não é extensível a outros períodos em que o preço do Bitcoin teve um grande aumento. Dessa forma, o estudo de caso realizado neste trabalho propõe uma métrica para verificar se os padrões de transação anormais identificados em [43] mudam o comportamento do grafo da *blockchain*. Além disso, a análise de predição de métrica em grafos temporais é utilizada como base para projetar mecanismos de predição de atividades maliciosas, independentemente dos dados vazados.

Capítulo 4

Metodologia

Neste capítulo é apresentada a metodologia adotada neste trabalho, desde a aquisição dos dados até a predição das séries temporais. A Figura 4.1 apresenta um fluxograma com todas as etapas seguidas para o estudo e descritas nas seções subsequentes deste capítulo.

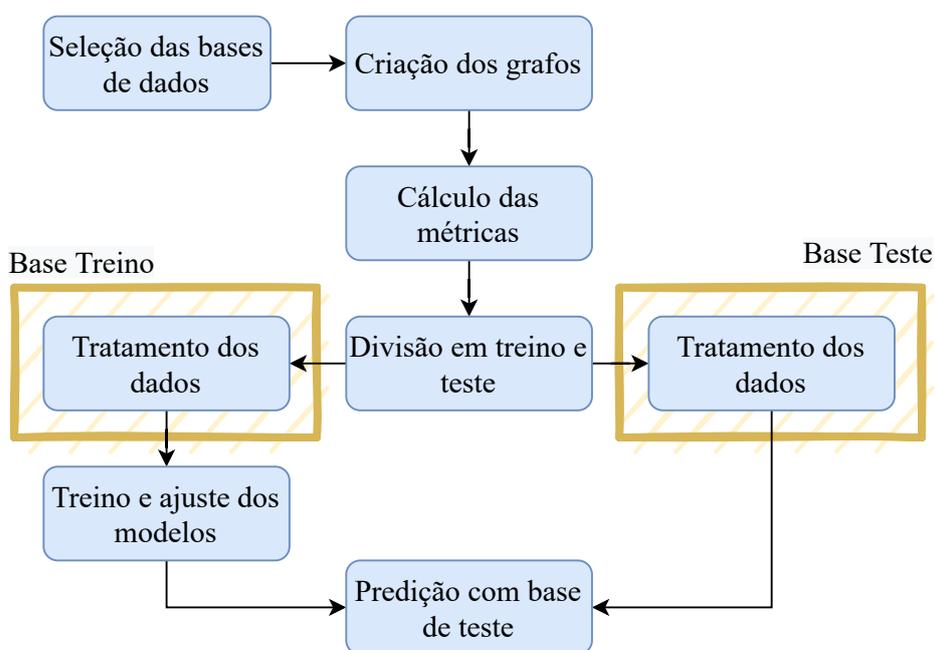


Figura 4.1: Metodologia do Estudo.

4.1 Seleção das bases de dados

Esta dissertação utiliza três bases de dados distintas, de forma a analisar os modelos em diferentes domínios. A descrição de cada base é apresentada a seguir.

4.1.1 Bitcoin

Esta base possui transações do Bitcoin, sendo fornecida pela universidade ELTE (*Eötvös Loránd University*)¹. 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 da transação (isto é, `txin`) com o `addrID` da planilha de destino (isto é, `txout`). Além desses, 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, esta dissertação 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 2012-02 e 2013-05. Uma exceção ocorre para o estudo de caso. Como nele o interesse é identificar comportamento malicioso na *blockchain* do Bitcoin, com foco no período de atividade da Mt.Gox, no estudo de caso são utilizadas todas as transações entre 2012-06 e 2015-06

Este trabalho realiza também um pré-processamento da base Bitcoin da ELTE, visando remover dados irrelevantes para as análises desta dissertação. 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 [44]. 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.

4.1.2 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 o evento ocorreu 10 segundos após o início da coleta. A base de dados está disponível no site do SNAP (*Stanford Network Analysis Project*)².

¹<https://senseable2015-6.mit.edu/bitcoin/>

²<https://snap.stanford.edu/data/email-Eu-core-temporal.html>

4.1.3 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. A diferença é que a transação da CollegeMsg é o envio de uma mensagem na rede social. A base também faz parte do projeto SNAP³.

4.2 Criação dos grafos

Um grafo $G(V, E)$ pode ser entendido como uma estrutura utilizada para representar a relação entre um conjunto de vértices V e um conjunto de enlaces E . No conjunto dos vértices, estão as entidades e o conjunto de enlaces indica a existência ou não de relação entre essas entidades. Por exemplo, caso essa estrutura seja utilizada para representar uma rede de roteadores e a existência de *links* entre eles, cada roteador poderia ser representado por um vértice e a existência de um *link* é representada pela existência de um enlace conectando esses dois vértices. Os grafos temporais adicionam a noção de tempo a essa representação, de forma que os enlaces passam a ter uma duração. A ideia por trás de grafos temporais é que a cada instante de tempo os vértices e os enlaces da rede podem ser modificados, com a inclusão ou remoção de vértices e enlaces.

A Tabela 4.1 apresenta um resumo das principais características dos grafos criados a partir das bases apresentadas na Seção 4.1, utilizando os dados de todo o período analisado. Essa tabela possui 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, sem a preocupação com a informação temporal das bases. Vale notar que o resumo da Tabela 4.1 é apresentado apenas para fins de exemplo, já que a análise deste trabalho considera a informação temporal das bases nas análises.

Tabela 4.1: 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

Para as análises que são realizadas nesta dissertação, é necessário o cálculo de métricas de redes complexas, a cada instante de tempo. Para isso, utiliza-se a representação de grafos temporais por meio de *snapshots*. Nessa representação, no lugar de construir um único grafo, contendo todas as transações realizadas durante todo o período de tempo analisado, são construídos grafos menores, contendo apenas as transações ocorridas em um período temporal menor. Dessa forma, a cada instante de tempo t é criado um novo

³<https://snap.stanford.edu/data/CollegeMsg.html>

grafo, contendo apenas os vértices e enlaces relacionados as transações ocorridas entre $t - 1$ e t . Essa metodologia permite que, em cada instante de tempo, o grafo seja tratado como um grafo estático, tirando assim a variável tempo das análises e simplificando o cálculo de métricas.

Define-se a utilização de períodos de um dia para cada *snapshot*. 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, para a construção dos grafos a partir da base Email-Eu-core, SRC e TGT são representados como nós no grafo, e a troca de e-mails é representada pela existência de um enlace unindo os dois nós. Caso tenha ocorrido duas trocas de e-mail entre dois nós no mesmo dia, o peso do enlace será igual a dois. Note que o peso considera todas as transações entre os nós, independentemente de qual dos dois é a origem, visto que o modelo é um grafo não direcionado.

A Figura 4.2 apresenta um exemplo da separação da base em *snapshots*. Nota-se que cada *snapshot* possui apenas os nós e enlaces que participam de transações no período definido pelo *snapshot*. Como já mencionado, nessa dissertação esse período é de um dia, à exceção do estudo de caso do Bitcoin descrito no Capítulo 6.

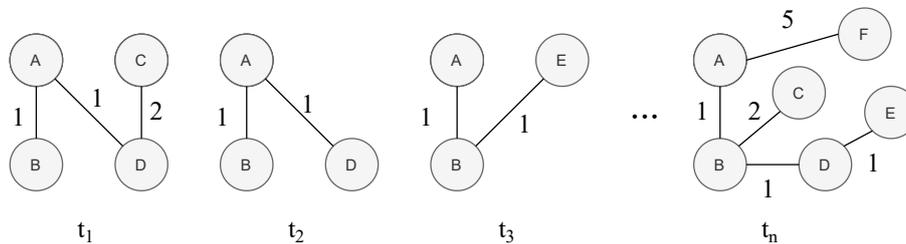


Figura 4.2: Exemplo de um *snapshot*.

4.3 Cálculo das métricas

Após a criação dos grafos contendo as transações ocorridas para cada dia, o próximo passo é calcular as métricas para cada grafo. Essas métricas são armazenadas com base na ordem cronológica da criação dos grafos. A sequência dessas métricas formam as séries temporais que serão previstas neste trabalho. Como o objetivo é prever os valores do grau médio e do coeficiente de clusterização, as duas métricas são calculadas para cada grafo. O grau médio representa a relação entre o total de enlaces e o total de nós de um grafo. A Equação 4.1 apresenta a fórmula para o cálculo do grau médio, sendo d_i o grau do nó i e N o total de nós do grafo.

$$\text{Grau Médio} = \frac{\sum_{i=1}^N d_i}{N}. \quad (4.1)$$

O coeficiente de clusterização é uma métrica que mensura a formação triângulos no grafo. A Equação 4.2 apresenta o cálculo para o coeficiente de clusterização do nó i . Sendo, E_i o número de enlaces entre os vizinhos de i e d_i o grau do nó i .

$$C_i = \frac{2 * E_i}{d_i(d_i - 1)}. \quad (4.2)$$

O coeficiente de clusterização médio do grafo é a média do coeficiente de clusterização de todos os nós. A Equação 4.3 apresenta o seu cálculo.

$$\text{Coef. de Clusterização Médio} = \frac{1}{N} \sum_{i=1}^N C_i. \quad (4.3)$$

Após o cálculo das métricas, a próxima etapa é a divisão da base em treino e teste, apresentada a seguir.

4.4 Divisão em treino e teste

As séries são divididas em duas partes, sendo os primeiros 75% das janelas separadas para o treino e os 25% restante são utilizadas 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. Além disso, é aplicado o método de janelas deslizantes nas bases de dados. Nesse método, a série temporal é separada em intervalos de tamanho fixo, que são então utilizadas como entrada dos modelos. O último valor da janela é utilizado como alvo, ou seja, é o valor que deve ser predito.

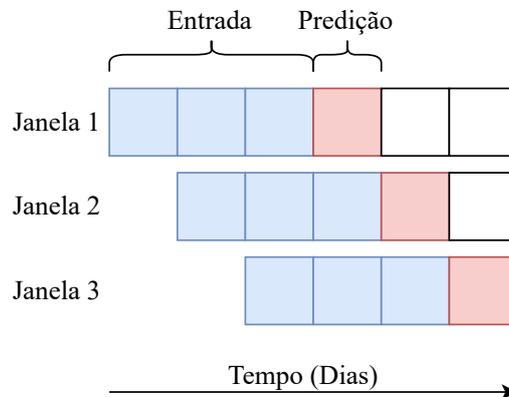


Figura 4.3: Exemplo de janelas deslizantes.

A Figura 4.3 mostra um exemplo da aplicação das janelas deslizantes em séries tempo-

rais. O tamanho da janela determina a quantidade de dados que serão utilizados na entrada dos modelos. Apesar de ser possível utilizar essa técnica para prever mais do que um dia no futuro, os experimentos realizados neste trabalho não abordam esse cenário. De forma preliminar, foram testados diferentes tamanhos de janela. A janela utilizando 14 dias do passado para prever um dia no futuro foi a que apresentou o menor erro nas redes neurais CNN e LSTM. Assim, quando não mencionado o contrário, este trabalho considera esse tamanho de janela.

4.5 Tratamento dos dados

Após a criação das séries temporais, o único tratamento aplicado nas bases de dados foi a normalização das séries temporais. A normalização é uma técnica na qual os dados são transformados e passam a ter a amplitude entre zero e um. Essa técnica é necessária pois os modelos de redes neurais possuem melhor desempenho quando as entradas possuem menor amplitude. A Equação 4.4 apresenta a fórmula para o cálculo do valor normalizado de X , sendo X o valor que será transformado, X_{min} o menor valor da série e X_{max} o maior valor. Para evitar vazamento de informação, os valores de X_{min} e X_{max} são calculados utilizando apenas a base de treino e aplicados na base de teste.

$$X_{normalizado} = \frac{X - X_{min}}{X_{max} - X_{min}}. \quad (4.4)$$

4.6 Treino e ajuste dos modelos

Os modelos de rede neurais possuem diversos hiperparâmetros que impactam de forma significativa os resultados das predições. Por esse motivo, para definir o modelo final utilizado para a predição das séries temporais, são testadas diversas combinações de arquiteturas e hiperparâmetros diferentes. A seguir é descrito o processo de ajuste de hiperparâmetros adotado nesta dissertação.

4.6.1 Ajuste de hiperparâmetros

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 redes neurais considerados. Nesse método, é construída uma grade com os valores que serão testados para cada hiperparâmetro [45]. 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. Neste trabalho seleciona-se, para cada tipo de rede neural, a combinação que apresentou o menor erro MAE na predição da base de teste. O MAE calcula a média do erro absoluto entre o valor real e o valor predito.

A Tabela 4.2 apresenta os hiperparâmetros utilizados na busca em grade para as redes MLP e LSTM. Nota-se que foram alterados apenas o número de camadas escondidas e o número de neurônios por camada. A função de ativação, o otimizador e a função erro foram configurados manualmente. O ajuste desses parâmetros também seguiu uma seleção baseada em tentativa e erro. No entanto, menos cenários foram analisados.

Tabela 4.2: Configuração de hiperparâmetros para a MLP e a LSTM.

Hiperparâmetros	MLP	LSTM
# Camadas escondidas	De 1 até 4	De 1 até 4
# Neurônios por camada	16, 32, 64, 128, 256	16, 32, 64, 128, 256
Função de Ativação	ReLU	Tanh
Otimizador	Adam [46]	Adam [46]
Erro	MAE	MAE

A Figura 4.4 apresenta a arquitetura da MLP, que apresentou o melhor desempenho na busca em grade. A arquitetura selecionada possui uma camada de entrada com 14 neurônios, duas camadas escondidas, contendo 128 neurônios, com função de ativação ReLU (*Rectified Linear Unit*). Já a camada de saída possui apenas uma unidade completamente conectada, com função de ativação linear.

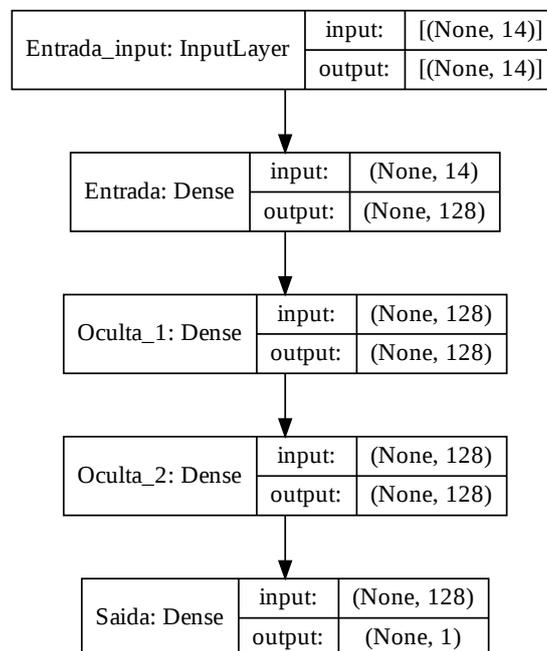


Figura 4.4: Arquitetura da MLP.

A Figura 4.5 mostra a arquitetura da LSTM selecionada. A rede neural possui três camadas, duas delas do tipo LSTM com 128 unidades, com função de ativação tangente

hiperbólica e *dropout* de 0,3. A camada de saída possui apenas uma unidade totalmente conectada, com um único neurônio. 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.

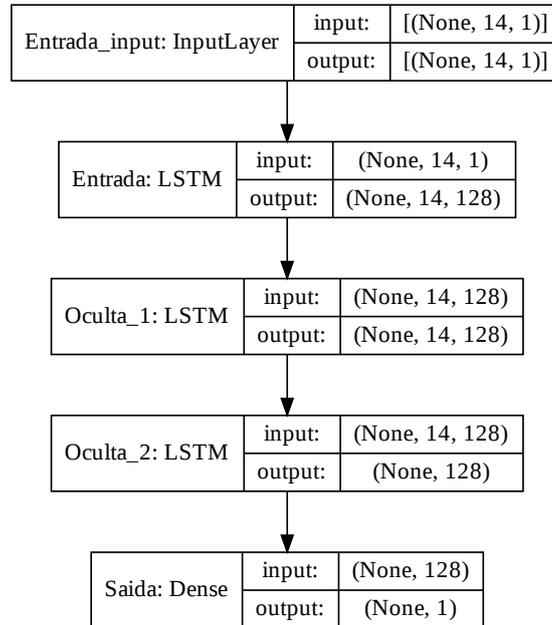


Figura 4.5: Arquitetura da LSTM.

A Tabela 4.3 apresenta os hiperparâmetros utilizados na busca em grade para o modelo CNN. Foram testados modelos com apenas uma camada de convolução, seguido por uma camada de *max pooling* e uma camada totalmente conectada. O número de neurônios presentes na tabela refere-se à camada totalmente conectada. A saída da CNN também possui uma camada totalmente conectada, porém com apenas um neurônio.

Tabela 4.3: Configuração de hiperparâmetros para a CNN.

Hiperparâmetros	CNN
# Filtros	8,16,32,64,128
Tamanho do Filtro	2,3,5
# Neurônios	[16,20,32,40,64]
Função de Ativação	ReLU
Otimizador	Adam
Função erro	MAE

A Figura 4.6 apresenta a arquitetura da CNN com melhor desempenho na busca em grade. O modelo selecionado possui uma camada convolucional 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 $\hat{y}(t + 1)$.

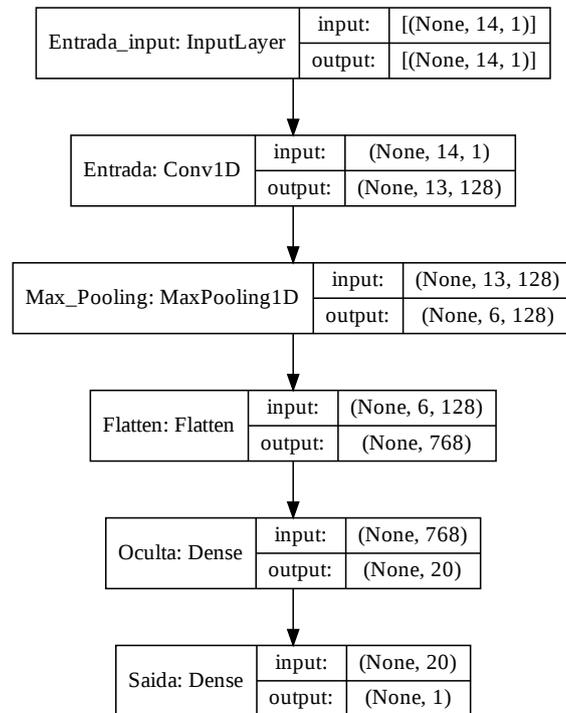


Figura 4.6: Arquitetura da CNN.

Capítulo 5

Análises e Resultados da Predição dos Grafos Temporais

Neste capítulo são analisados os erros dos modelos de redes neurais na predição das séries temporais, oriundas dos grafos temporais. Na análise, primeiramente, são realizadas as predições para cada série temporal, utilizando todos os modelos de rede neural apresentados e o modelo base. Nessa etapa, são analisados o erro médio absoluto (*Mean Absolute Error* - MAE) e raiz quadrada do erro médio (*Root Mean Squared Error* - RMSE). A Equação 5.1 apresenta a fórmula para o cálculo do MAE, onde y é o valor real, \hat{y} é o valor predito e n é o tamanho da amostra. Ao utilizar o erro MAE, é dada a mesma importância para os erros, independente da sua magnitude. Na Equação 5.2 é apresentado o cálculo do RMSE. Como no cálculo do RMSE a diferença entre o valor real e o valor predito é elevada ao quadrado, esse erro tem como consequência atribuir um peso maior para erros de maior magnitude.

$$MAE = \frac{1}{n} \sum_i^n |y_i - \hat{y}_i| \quad (5.1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2} \quad (5.2)$$

Posteriormente, são selecionados os melhores modelos para cada série temporal e realiza-se uma análise dos resíduos. Todos os modelos são implementados com as bibliotecas Keras e TensorFlow¹ e executados no *Google Colaboratory*². O capítulo é dividido em três seções, sendo uma para cada base de dados.

¹<https://www.tensorflow.org/>

²<https://colab.research.google.com>

5.1 Bitcoin

Após o ajuste de hiperparâmetros, a melhor configuração para cada modelo de rede neural é utilizada para a predição do grau médio e do coeficiente de clusterização da base do Bitcoin. Nessa etapa, para cada série temporal, executam-se 30 medições dos erros MAE e RMSE. Todos os resultados são apresentados com intervalo de confiança de 95%. Essa mesma metodologia é adotada para as demais bases, das Seções 5.2 e 5.3. As Figuras 5.1 e 5.2 apresentam o MAE e o RMSE, respectivamente. Os resultados mostram que, para a base de dados do Bitcoin, o modelo de rede neural LSTM apresenta o melhor desempenho nas duas séries temporais.

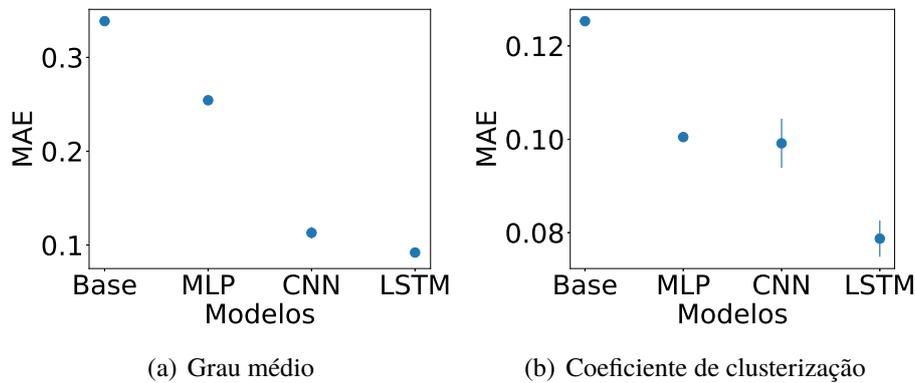


Figura 5.1: MAE para as séries temporais do Bitcoin.

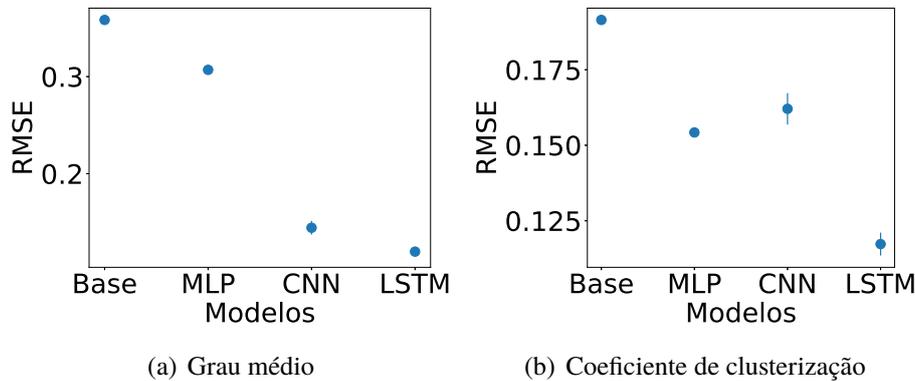


Figura 5.2: RMSE para as séries temporais do Bitcoin.

A partir dos resultados obtidos, seleciona-se o modelo LSTM para a predição das métricas da base de dados do Bitcoin. O modelo é treinado, utilizando a base de treino, e é realizada a predição da base de teste. A Figura 5.3 apresenta os resultados da LSTM para o grau médio. Para padronizar e facilitar a visualização das figuras, nas figuras contendo uma amostra da predição das métricas, foram selecionadas 30 observações da base de dados de treino de cada série temporal. Já para as figuras com análise de resíduos,

são levadas em consideração todo o intervalo de teste. O mesmo processo é adotado nas Seções 5.2 e 5.3.

A Figura 5.3(a) apresenta as 30 primeiras amostras de predição da base de teste do grau médio do Bitcoin. O eixo x da figura representa a contagem de dias da base de teste, sendo zero o primeiro dia e 29 o trigésimo dia. No eixo y estão os valores preditos e os valores alvo do grau médio. O valor alvo é o valor real da métrica, calculada a partir do grafo. Já o valor predito é a saída do modelo de rede neural. A Figura 5.3(b) apresenta a análise dos resíduos para toda a base de teste. Nela, o eixo x apresenta o valor predito e o eixo y a diferença entre o valor real e o valor predito.

As duas figuras mostram que o modelo obteve sucesso na predição para a base do Bitcoin, com as predições próximas aos valores alvo e com a distribuição dos resíduos sem indícios de não linearidade ou heterocedasticidade. A Figura 5.4 apresenta a mesma análise anterior, porém para a base de teste do coeficiente de clusterização do Bitcoin. Novamente, é observado que o modelo obteve sucesso nas predições, com as predições próximas aos valores alvo e com a distribuição dos resíduos sem indícios de não linearidade ou heterocedasticidade.

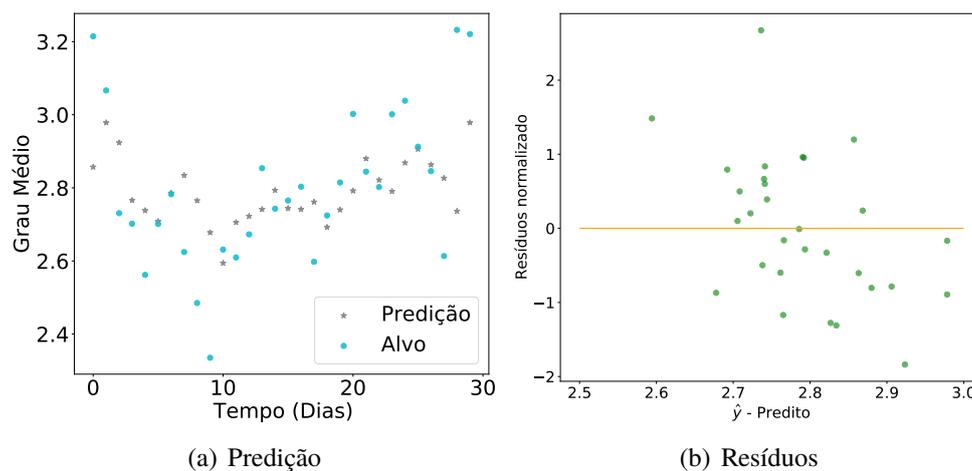


Figura 5.3: Grau médio da base Bitcoin com a LSTM.

5.2 Email-Eu-core

Nesta seção são analisados os resultados para a base Email-Eu-core. As Figuras 5.5 e 5.6 apresentam o MAE e o RMSE, respectivamente. Os resultados mostram que, para a série temporal com o coeficiente de clusterização, o modelo de rede neural LSTM apresenta novamente o melhor desempenho. Já para a série com o grau médio, o modelo CNN apresenta o melhor desempenho.

Com base nos resultados anteriores, utiliza-se uma CNN para apresentar na Figura 5.7 uma amostra com as primeiras 30 predições para o grau médio e a análise de resíduos,

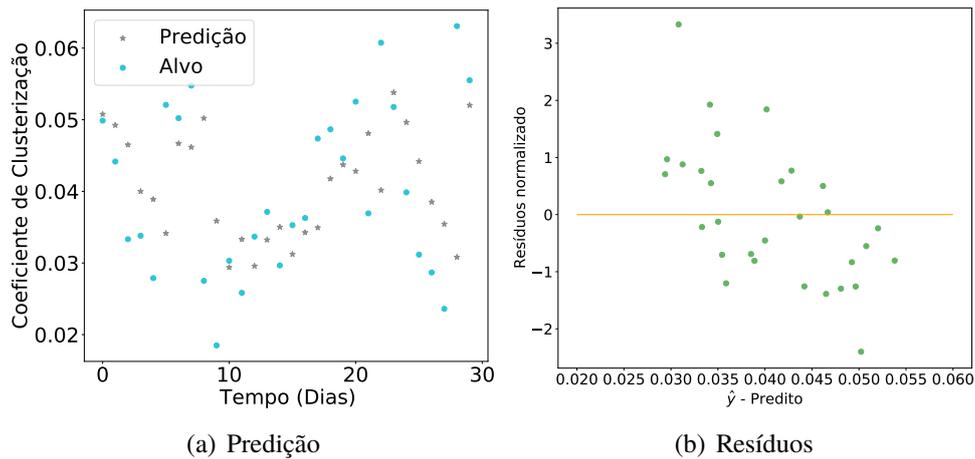


Figura 5.4: Coeficiente de clusterização da base Bitcoin com a LSTM.

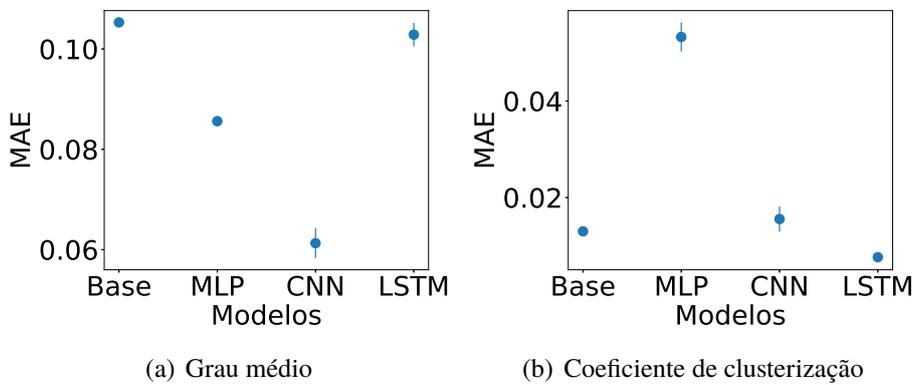


Figura 5.5: MAE para as séries temporais do Email-Eu-core.

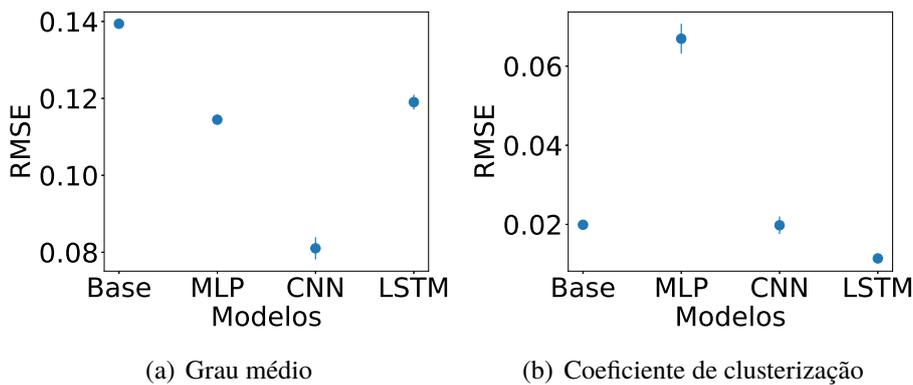


Figura 5.6: RMSE para as séries temporais do Email-Eu-core.

para toda a base de teste. Nessas imagens é possível observar que, apesar de possuir uma maior amplitude nos erros, novamente a distribuição dos resíduos não apresenta indícios de não linearidade ou heterocedasticidade. Na Figura 5.8 são repetidas as análises para o coeficiente de clusterização, porém, utiliza-se o modelo LSTM, já que ele apresenta o menor MAE para a base. Apesar de a amostra com 30 previsões apresentar previsões

próximas ao alvo, a análise dos erros apresenta característica de heterocedasticidade. Isso indica que o modelo, mesmo otimizado, não possui resultado satisfatório.

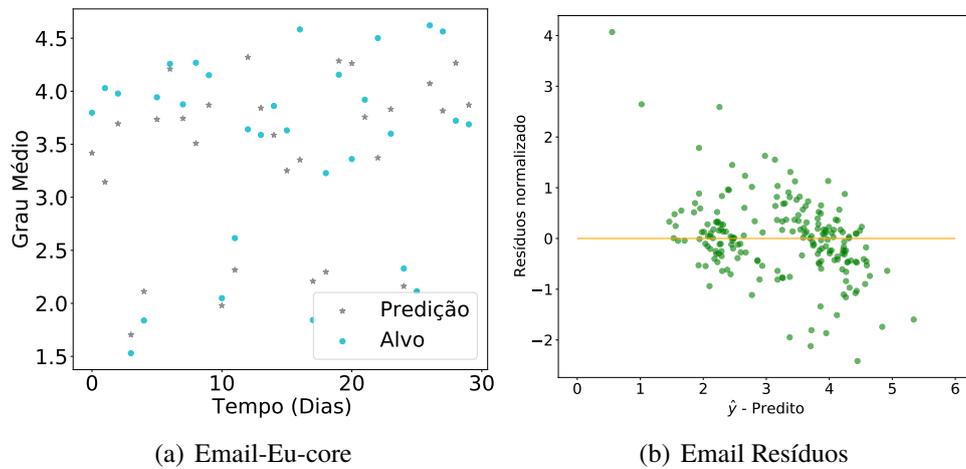


Figura 5.7: Grau médio da base Email-Eu-core com a CNN.

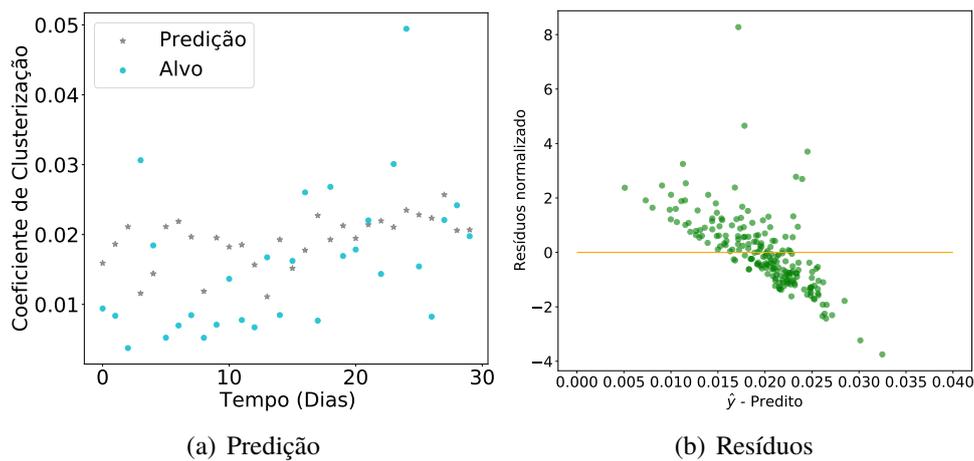


Figura 5.8: Coeficiente de clusterização da base Email-Eu-core com a LSTM.

5.3 CollegeMsg

As Figuras 5.9 e 5.10 apresentam o MAE e o RMSE dos modelos para a base CollegeMsg. Considerando o intervalo de confiança, o resultado do MAE mostra que a MLP possui o mesmo desempenho da CNN para a predição do grau médio. Entretanto, no RMSE a MLP apresenta um melhor desempenho do que a CNN. Um melhor desempenho no RMSE indica que os erros cometidos pela CNN possuem maior magnitude do que os erros da MLP. Por esse motivo, utiliza-se uma MLP para a predição da série temporal do grau médio. Para o coeficiente de clusterização, utiliza uma CNN pois essa rede obteve o melhor desempenho no MAE e no RMSE.

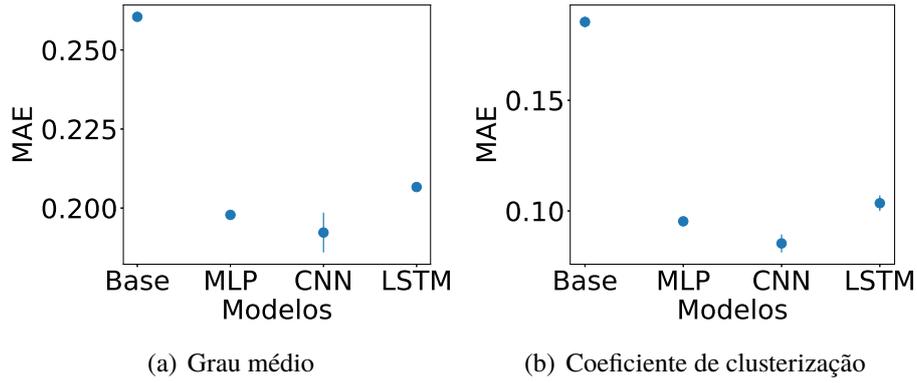


Figura 5.9: MAE para as séries temporais da base CollegeMsg.

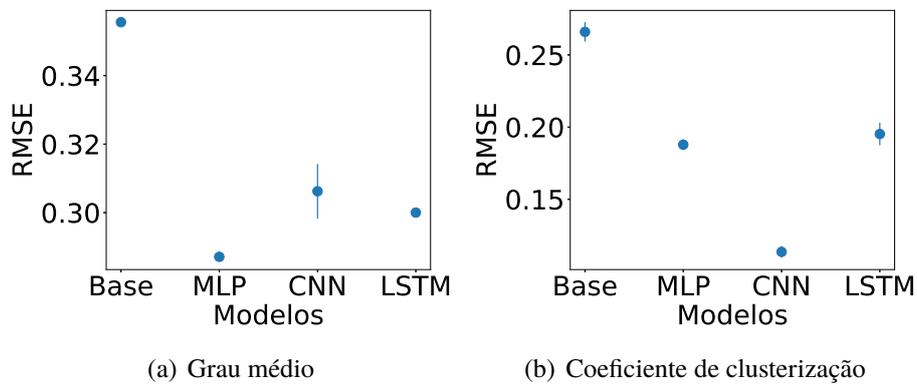


Figura 5.10: RMSE para as séries temporais da base CollegeMsg.

As Figuras 5.11 e 5.12 mostram, respectivamente, exemplos das 30 primeiras predições para o grau médio e o coeficiente de clusterização utilizando a base de teste, assim como as respectivas análises de resíduos. Observa-se que, entre as três bases analisadas, o desempenho das redes neurais na predição das métricas são piores para a base CollegeMsg. Isso pode ser consequência dos grafos formados para cada *snapshot* da CollegeMsg, já que a base apresenta dias sem ocorrência de medições, resultando em valores de grau médio e coeficiente de clusterização zerados. Não é possível saber se a ausência dessas medições é devido à ausência de troca de mensagens nos dias ou inconsistências da base. Na Figura 5.12, esse comportamento fica mais evidente. Como o coeficiente de clusterização possui muitos valores bem próximo de zero, as redes neurais tiveram dificuldade em aprender padrões nessa base, resultando em uma predição com análise de resíduos com sinais de heterocedasticidade. Apesar de esses resultados apresentarem uma dificuldade das redes neurais em realizarem a predição das métricas para essa base, mais estudos são necessários para determinar se isso ocorre devido a inconsistências na base ou características dessa série temporal.

Este capítulo abordou os resultados dos modelos de rede neural na predição das séries temporais. Os modelos CNN e LSTM obtiveram os melhores resultados, com exceção

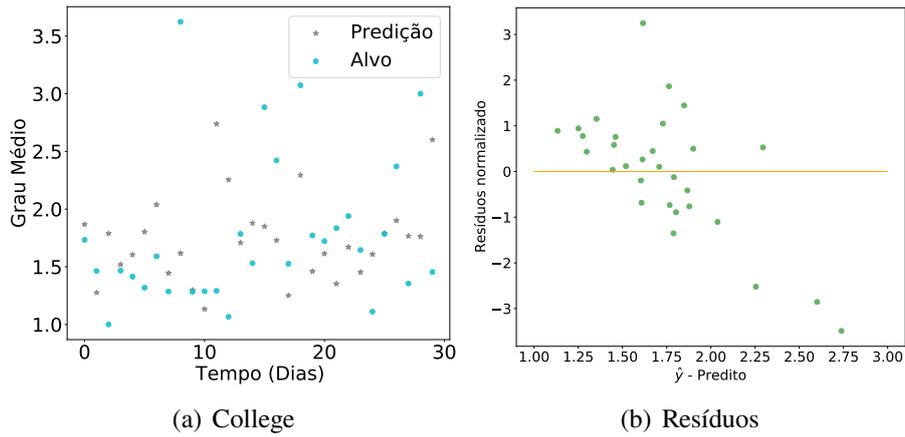


Figura 5.11: Grau médio da base CollegeMsg com a MLP.

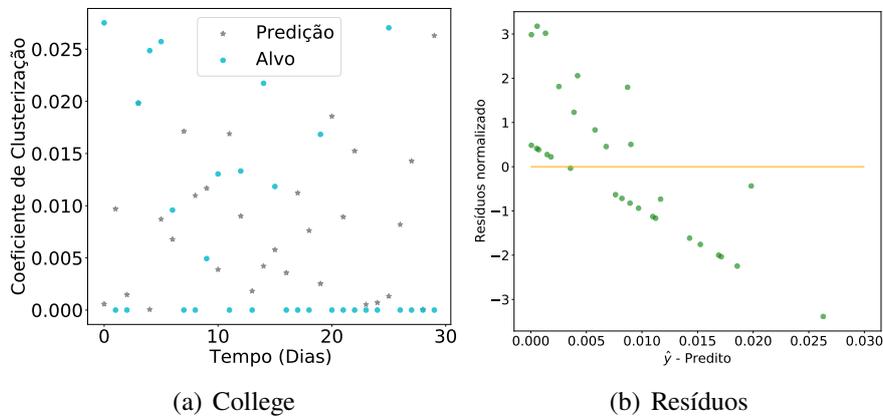


Figura 5.12: Coeficiente de clusterização da base CollegeMsg com a CNN.

apenas de um caso particular, evidenciando a sua aplicabilidade em problemas desse tipo. O caso particular se refere às predições da base de dados CollegeMsg, que possui erros de predição, mesmo com resultados satisfatórios nos erros MAE e RMSE.

Capítulo 6

Estudo de Caso: Bitcoin

De forma a exemplificar a utilidade de se prever métricas básicas de grafos, é realizado um estudo de caso com base na *blockchain* do Bitcoin. O estudo realizado na Seção 5 realiza a predição de métricas básicas de grafo, o grau médio e o coeficiente de clusterização. No entanto, como o objetivo do estudo de caso é realizar a detecção dos nós maliciosos, ou seja, cujo comportamento possa estar associado a manipulação de mercado, é utilizada uma métrica básica que permite ordenar os nós com base na quantidade de enlaces, chamada centralidade de grau. O estudo de caso é dividido em duas partes. Na primeira, verifica-se com análise de grafos se é possível identificar alterações na *blockchain* no período no qual a Mt.Gox estava ativa. Na segunda parte, é proposto utilizar um método, baseado em rede neurais, para prever essas alterações.

6.1 Análise do ranking

Como o interesse no estudo de caso é analisar a *blockchain* durante as atividades maliciosas relatadas da corretora Mt.Gox, que ocorrem no ano de 2013, a base de dados utilizada possui todas as transações do Bitcoin no intervalo entre 2012-06 até 2015-06. Foi selecionado um intervalo antes e depois do ano de 2013 para que fosse possível comparar o comportamento.

De forma similar à feita na Seção 4.2, os dados foram separados em *snapshots* e cada *snapshot* é modelado como um grafo não direcionado, $G_m(V_m, E_m, w)$. A diferença é que, em uma primeira análise, o *snapshot* é feito contendo um mês de transações e não apenas um dia. Dessa forma, V_m é o conjunto de nós que representam os usuários que fizeram transações durante o mês m . E_m é o conjunto de enlaces, representando as transações entre usuários que ocorrem em m , e w é um peso associado a cada enlace. Se dois usuários tiverem pelo menos uma transação entre eles durante o mês m , há um enlace conectando esses nós. O peso desse enlace é o número total de transações feitas entre eles durante o mês m .

Após a criação dos grafos, é calculada a centralidade de grau para cada grafo. Essa métrica mede a importância de um nó com base em seu número de enlaces. A Equação 6.1 mostra como calcular a centralidade de grau. Nessa equação, $N - 1$ é o número máximo de enlaces que um nó pode ter no mês m , e tem-se $e_{ij} = 1$ se houver um enlace entre o nó i e o nó j no mês m . Se um nó possuir $N - 1$ enlaces, ele terá um enlace com todos os outros nós.

$$C_D = \frac{\sum_{i=1}^N e_{ij}}{(N - 1)}. \quad (6.1)$$

Após o cálculo da centralidade de grau, os nós são ordenados, com base na centralidade de grau, em ordem decrescente e os k nós com a maior centralidade de grau em cada mês m são selecionados. Os *top-k* nós são utilizados para construir o conjunto \mathcal{N}_m^k . Esse conjunto possui os *top-k* nós da rede. Diferentes valores para k foram testados e o número de transações que esses nós realizam foi utilizado para escolher o tamanho desse conjunto. Para o caso de $k = 10$, verifica-se que esses 10 primeiros nós são responsáveis por quase 20% das transações no período. Para os cem primeiros, obtém-se 33%. Para os 150 primeiros, 34%. Aumentado para os primeiros mil, obtém-se aproximadamente 40% e 44% para os primeiros 2000. Dessa forma, os resultados usam apenas $k = 100$, para simplificar a análise e dado que esses nós são responsáveis por um número expressivo de transações.

Depois de selecionar os *top-k* nós, analisa-se como essa classificação muda ao longo do tempo. Para tanto, são comparados os nós de cada conjunto \mathcal{N}_m^k com o conjunto \mathcal{N}_{m-1}^k , isto é, os nós referentes ao mês anterior. Nesta análise, o objetivo é descobrir o número de nós repetidos. Ou seja, a cardinalidade da interseção entre \mathcal{N}_m^k e \mathcal{N}_{m-1}^k . Esse número é chamado de Repetição de Usuário no restante do trabalho e a Equação 6.2 formaliza a definição.

$$\text{Repetição de Usuário} = |\mathcal{N}_m^k \cap \mathcal{N}_{m-1}^k|. \quad (6.2)$$

A Figura 6.1 mostra como a Repetição de Usuário muda ao longo do tempo ¹. Esta figura também inclui a diferença de preço do Bitcoin durante o período analisado, comparando o preço do Bitcoin no mês m com o mês $m - 1$. Para tal, utiliza-se o preço de mercado histórico do Bitcoin, disponível em [47]. Observa-se que, no período entre 2013-02 e 2013-12, a Repetição de Usuário apresenta seus valores mais baixos. Isso significa que nesse período há menos nós repetidos na lista dos *top-k*. Esse comportamento é diferente dos meses anteriores e seguintes, nos quais a Repetição de Usuários é maior. Além disso, esse período entre 2013-02 e 2013-12 também coincide com o período de maior alta no preço do Bitcoin.

¹Para avaliar a Repetição de Usuário no primeiro mês (ou seja, $m = 1$), foi utilizado o conjunto \mathcal{N}_1^{100} e o conjunto \mathcal{N}_0^{100} , sendo $m = 0$ o mês 2012-05.

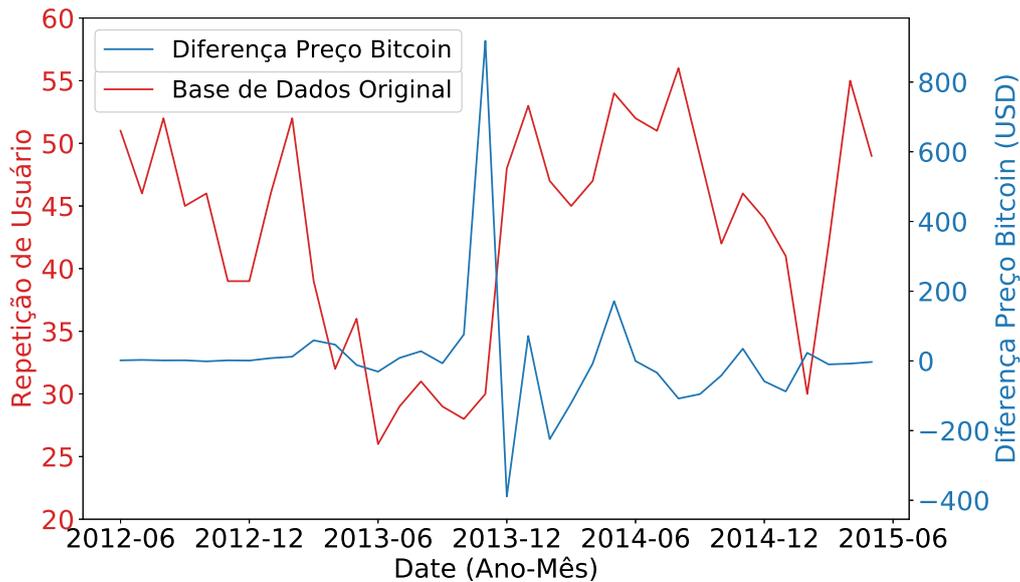


Figura 6.1: Repetição de Usuário e variação no preço do Bitcoin.

O fato de o valor da Repetição de Usuário possuir valores mais elevados nos demais meses mostram que os *top-k* mudam pouco, indicando estabilidade no comportamento desse ranking. Uma exceção ocorre entre 2014-12 e 2015-05. Nesse caso, a Repetição de Usuário permanece baixa por um curto período. Isso pode indicar que fatores externos (p.ex., escândalos políticos, ataques terroristas, decisões de governos etc.) podem influenciar a métrica de Repetição de Usuário. Essa análise não é abordada neste trabalho e será estudada em trabalhos futuros.

Calcula-se também a média da Repetição de Usuário em cada intervalo de seis meses do período considerado, mostrado na Tabela 6.1. Os nomes dos intervalos desta tabela são utilizados no restante deste capítulo. Os seis intervalos foram denominados de A0, A1, A2, B1, B2 e C1. Os intervalos A1 e A2 representam o período em que as evidências apontam que a corretora Mt.Gox realizou a maioria de suas transações suspeitas [43]. A1 e A2 são os intervalos com a menor Repetição de Usuário. No intervalo A0, já ocorriam atividades maliciosas da Mt.GOX, mas, com base na análise dos dados vazados, o número de transações suspeitas foi inferior ao número apresentado em A1 e A2. Observa-se também que o intervalo A2 (ou seja, entre 2013-06 e 2013-11) apresenta a menor média, conforme esperado com base na Figura 6.1. Este capítulo se concentra em compreender o comportamento da rede durante o intervalo A2 devido à sua baixa média de Repetição de Usuário, seu alto preço de mercado do Bitcoin e das atividades maliciosas relacionadas [10, 43].

Os resultados da Figura 6.1 e da Tabela 6.1 levam à hipótese de existir uma relação entre a métrica de Repetição de Usuário e a manipulação de mercado feita pela Mt.Gox, notada pela relação entre a métrica e o preço do Bitcoin. Visando analisar essa hipótese, realiza-se, a seguir, um estudo sobre a estabilidade dos nós. O objetivo do estudo é verificar como a inserção de transações artificiais tem impacto na métrica de Repetição de Usuário.

Tabela 6.1: Média da Repetição de Usuário.

Intervalo Data (De ~Até)	Nome do Intervalo	Diferença Média Repetição de Usuário
2012-06 ~2012-11	A0	46.5
2012-12 ~2013-05	A1	40.6
2013-06 ~2013-11	A2	28.8
2013-12 ~2014-05	B1	49.0
2014-06 ~2014-11	B2	49.3
2014-12 ~2015-05	C1	43.5

6.2 Análise de estabilidade dos nós

Para realizar a análise, seleciona-se um intervalo de seis meses para inserir novas transações. Para cada mês nesse intervalo, o grafo G_m é criado e novos enlaces são inseridos, de forma aleatória. Para inserir as novas transações, cria-se um subconjunto S_m com $k = 100$ nós de V_m , o processo para seleção dos nós que fazem parte de S_m é explicado mais à frente. Os nós em S_m são responsáveis pela entrada das novas transações. A saída de cada transação é selecionada de V_m ou S_m , dependendo da metodologia aplicada. A quantidade de novas transações inseridas em cada mês m é baseada em uma porcentagem do número total de transações que ocorreram nele.

Como a Mt.Gox suspendeu todas as operações em 2014-02, as novas transações foram inseridas no intervalo B2 (2014-06 ~ 2014-11), para ter certeza de que não há influência da Mt.Gox no período analisado. Portanto, no restante desta seção, a variável m refere-se aos meses incluídos no intervalo B2, sendo $m \in B2$.

A porcentagem do número total de transações artificiais inseridas começam sendo 2% do número total de transações que ocorreram no mês e aumenta em 2% a cada etapa, até o limite máximo de 16%. Isso significa que, a cada etapa, é gerado um novo grafo $G'_m(V'_m, E'_m, w')$, sendo V'_m , E'_m , e w' o conjunto de nós, arestas e pesos atualizados com as novas transações incluídas. Após cada etapa de inserção de transações artificiais a centralidade de grau é recalculada. O grupo com os $top-k$ nós é selecionado e a Repetição de Usuário é recalculada. O procedimento é repetido em todos os meses $m \in B2$.

Utilizam-se três métodos independentes para adicionar novas transações. No Método 1, o objetivo é investigar o comportamento da métrica Repetição de Usuário quando as transações de rede aumentam, mas como consequência de um grande número de nós sendo responsáveis por realizá-las (ou seja, os nós de S_m mudam com frequência). O Método 2 é similar ao 1, mas menos nós são responsáveis por fazer as novas transações. Tanto no Método 1 quanto no Método 2, escolhe-se um novo conjunto S_m a cada mês m . Já no Método 3, os mesmos nós são responsáveis por inserir as transações durante todos os meses. Cada um dos métodos é detalhado a seguir.

6.2.1 Método 1 para inserir novas transações

O objetivo do Método 1 é analisar o impacto das transações arbitrárias na Repetição de Usuário. Isso significa verificar o comportamento da Repetição de Usuário quando usuários aleatórios realizam muitas transações novas. Assume-se que as transações aleatórias não são organizadas e controladas por uma entidade. Este método representa nós que aumentam seu número de transações sem nenhum controle dos principais participantes da rede do Bitcoin. Ou seja, nesse método simula transações normais na rede do Bitcoin, mas com um grande aumento devido a algum fator externo e não uma manipulação do mercado.

No Método 1, para cada porcentagem, os seguintes procedimentos são realizados. Primeiro, escolhe-se aleatoriamente o subconjunto S_m a partir de V_m , usando uma distribuição uniforme. Em seguida, remove-se esses nós de V_m . Depois disso, novas transações são geradas usando distribuição uniforme para selecionar um nó de entrada de S_m . Em seguida, escolhe-se aleatoriamente o nó de saída de V_m ou S_m . Em cada transação, um dos conjuntos é utilizado na saída. A seleção de S_m aumenta o número de enlaces para si próprios e triângulos, pois esse grupo tem menos nós do que V_m . Quando o nó de saída é selecionado de V_m , aumenta-se a chance de que essa transação crie um novo enlace no grafo. Isso ocorre porque, como V_m possui um número maior de nós, há menos chance de selecionar um nó que já possui um enlace com a entrada. A seguir, criam-se os grafos G'_m e calcula-se C_D pela Equação 6.1. Então, selecionam-se os *top-k* nós e calcula-se a Repetição de Usuário pela Equação 6.2.

A Figura 6.2 mostra a Repetição de Usuário após a aplicação do Método 1 para cada porcentagem de novas transações. A marca no retângulo de cor rosa na figura o mostra o intervalo B2. Conforme descrito antes, este método muda S_m para cada porcentagem. Os resultados mostram que há uma diferença na Repetição de Usuário entre a curva do conjunto de dados original e a curva de 2%. No entanto, à medida que são adicionadas transações, essa diferença não tem uma mudança considerável, mesmo ao inserir 16% de novas transações. Isso acontece porque o subconjunto S_m muda constantemente ao longo dos meses e selecionam-se $|S_m| = 100$ novos nós em todo o G_m . Mesmo sem nenhuma regra que proíba um nó de ser selecionado novamente, a probabilidade de que isso aconteça é pequena. Isso implica que a chance de um nó de fazer mais de 2% das novas transações é pequena.

O Método 1 fornece informações sobre a concentração de transações na rede. Com 2% das transações criadas por usuários aleatórios, a Repetição de Usuário diminui em 15 posições. Como a Repetição de Usuário representa o número de nós repetidos na classificação dos *top-100* nós, essa diminuição significa que, apenas criando 2% das novas transações, inserem-se 15 novos nós na lista dos *top-100*. A Figura 6.2 mostra essa diminuição como a diferença máxima entre a curva do conjunto de dados original e a curva de

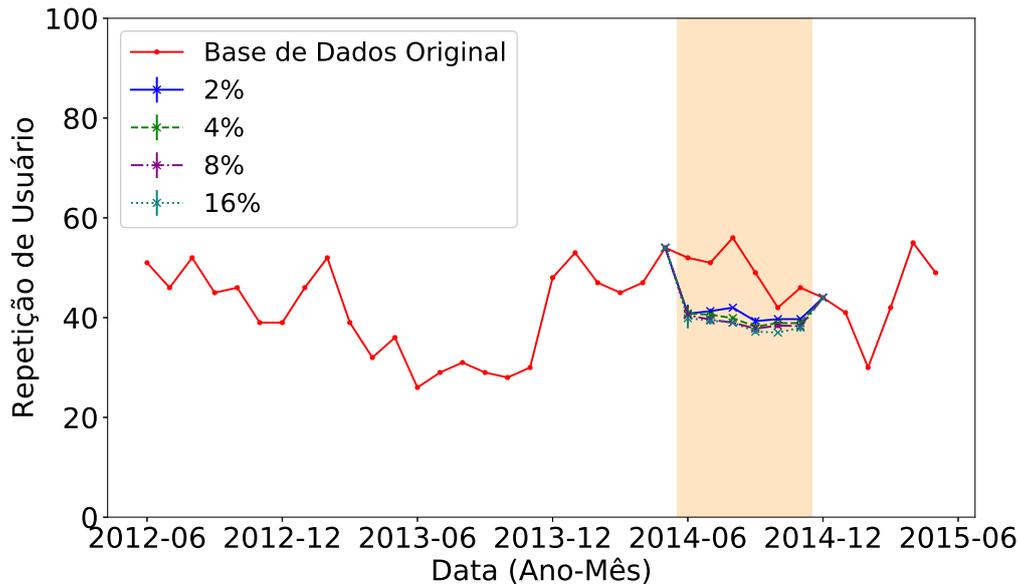


Figura 6.2: Método 1 para inserir novas transações.

2%. Conforme outras transações são inseridas, as outras curvas (ou seja, 4%, 8% e 16%) não mudam significativamente. Isso acontece por causa da aleatoriedade e frequência na seleção do subconjunto S_m . Os nós neste subconjunto são alterados com frequência (ou seja, a cada aumento de 2% no número de transações). Portanto, a cada mudança, um conjunto diferente de novos nós entra no $top-k$ e mantém a Repetição de Usuário com essa diminuição em quase 15 nós.

O Método 2 e o Método 3 têm muitas semelhanças com o Método 1. Por isso, são explicadas a seguir apenas as diferenças entre esses métodos e o Método 1. A diferença está em como S_m é escolhido.

6.2.2 Método 2 para inserir novas transações

No Método 2, seleciona-se apenas um subconjunto S_m , para cada m , e esse subconjunto é responsável por criar transações novas em G_m . Este processo é diferente do Método 1, no qual seleciona-se um subconjunto para cada porcentagem. O comportamento simulado no Método 2 é a inserção de um grande número de transações feitas por um mesmo grupo de nós. A consequência é que poucos usuários controlam a transação. Assim, esses usuários realizam uma quantidade considerável de interações (de 2% até 16% de todas as transações na rede).

A Figura 6.3 mostra a Repetição de Usuário após a aplicação do Método 2. Nesse método, o subconjunto S_m não se altera para cada porcentagem (ou seja, como no Método 1). Ao invés disso, muda-se o S_m apenas quando troca-se o mês m analisado. Por exemplo, em um determinado mês, para avaliar a Repetição de Usuário para 8% das novas transações, utiliza-se o mesmo subconjunto S_m empregado para 16% e assim por diante.

A Figura 6.3 mostra que, para 4% das novas transações, inclui-se quase 35 novos nós no $top-k$ (ou seja, a diminuição máxima na Repetição de Usuário para a curva de 4%). Além disso, com 16% de novas transações, a Repetição do Usuário é baixa. Isso significa que a cada mês em B2, insere-se a maioria dos nós S_m na lista $top-k$. Como essa lista muda uma vez por mês, a Repetição do usuário mantém valores baixos pois menos nós são repetidos. O comportamento da Repetição do Usuário no Método 2 é mais semelhante ao do intervalo A2 (ou seja, o período em que os preços do Bitcoin aumentaram devido à manipulação do mercado). Isso reforça a hipótese inicial desta dissertação de que a diminuição na Repetição do Usuário está relacionada à atividade da Mt. Gox.

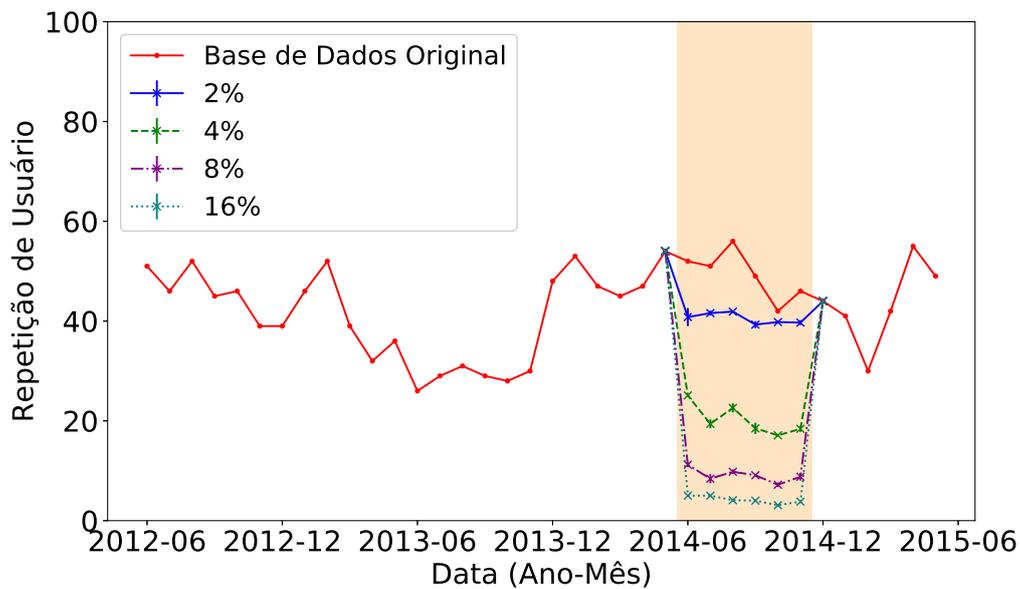


Figura 6.3: Método 2 para inserir novas transações.

6.2.3 Método 3 para inserir novas transações

No Método 3, ao invés de selecionar S_m de cada G_m , cria-se um grafo $G(V, E, w)$, usando todo o conjunto de dados B2 e então S_m é selecionado a partir deste grafo. Isso significa que em todos os meses m o mesmo subconjunto S_m é utilizado. Em outras palavras, existe um enlace entre dois nós em $G(V, E, w)$ se esses nós tiverem uma transação em qualquer mês do conjunto de dados B2. Da mesma forma que o Método 2, no Método 3 o subconjunto S_m é responsável pela entrada de todas as novas transações para cada m . Se em um mês m o nó selecionado em S_m não estiver em G_m , ele é adicionado ao grafo. O objetivo neste método é analisar a estabilidade do $top-k$ nós, conforme os nós são forçados a serem inseridos e repetidos a cada mês. Este método fornece informações sobre o comportamento nos períodos com alta Repetição de Usuário (por exemplo, A0, A1, B1 e B2).

A Figura 6.4 mostra a Repetição de Usuário após a aplicação do Método 3. Neste

método, o mesmo subconjunto S_m de nós é mantido durante todos os meses. Assim, verifica-se o comportamento da Repetição de Usuário quando esses nós inserem novas transações. Como esperado, quando o mesmo subconjunto de nós é responsável a cada mês por um número considerável de transações (isto é, de 2% até 16% das novas transações), há uma alta Repetição de Usuário. Esse resultado reproduz a estabilidade presente fora do intervalo A2. Também indica que um comportamento diferente ocorre na Repetição de Usuário durante o intervalo A2, pois os *top-k* nós mudam mais durante os meses neste intervalo.

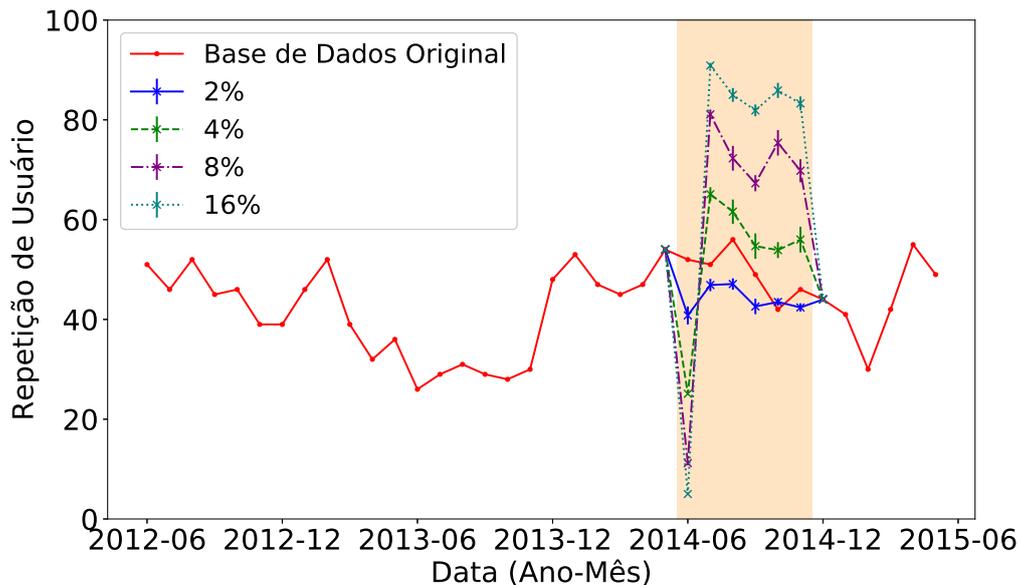


Figura 6.4: Método 3 para inserir novas transações.

6.3 Predição da Repetição de Usuário

Como durante a análise de estabilidade verificou-se que períodos com baixo valor da Repetição de Usuário possuem indícios de serem causados por anomalias na rede do Bitcoin e não um comportamento natural dos usuários, a capacidade de prever os valores futuros da Repetição de Usuário pode fornecer indícios de períodos com atividade anormal. Por exemplo, se a diferença entre a Repetição de usuário prevista e atual for alta, pode-se considerar que a rede está em um estado suspeito. Nesse caso, é possível acionar uma análise mais detalhada para detectar comportamento anômalo.

Por esse motivo, após as análises de redes complexas e da formulação da Repetição de Usuário, são utilizadas redes neurais para prever o comportamento da métrica. Por se tratar de uma análise de comportamento malicioso na rede do Bitcoin, para essa etapa foram utilizados *snapshots* de 1h. Isso significa que os grafos foram criados com todas as transações que ocorreram a cada 1h entre 2012-06 e 2015-05. A centralidade de grau, os

$top-k$ nós e a Repetição de Usuário foram calculados para cada grafo. A série temporal foi dividida em treino e teste. Porém, por possuir um número maior de pontos, quando comparada as bases utilizadas no Capítulo 4, foi utilizada a divisão de 90% treino e 10% teste. São então utilizadas as mesmas redes neurais apresentadas no Capítulo 2 para prever o valor da Repetição de Usuário para a série temporal criada.

A Tabela 6.2 apresenta a média e o intervalo de confiança de 95%, utilizando distribuição *T-Student* e uma amostra de 10 medições dos valores do MAE e do RMSE para o modelo base e para os modelos de rede neural MLP, CNN e LSTM. A média foi calculada com base em dez medições. A tabela mostra que o LSTM obteve os menores erros MAE e RMSE.

Tabela 6.2: Comparação entre as métricas dos modelos.

Métrica	Base	MLP	CNN	LSTM
MAE	$0,0609 \pm 0,0010$	$0,0671 \pm 0,0013$	$0,0544 \pm 0,0001$	$0,0529 \pm 0,0009$
RMSE	$0,0784 \pm 0,0013$	$0,0843 \pm 0,0015$	$0,0702 \pm 0,0002$	$0,0677 \pm 0,0013$

A Figura 6.5 mostra um exemplo de uma amostra com 30 predições da Repetição de Usuário utilizando a rede neural LSTM, assim como a respectiva análise de resíduos para toda a base de teste. É possível observar na Figura 6.5(a) que o modelo se aproxima do valor alvo na maioria dos pontos. Além disso, a análise de resíduos não apresenta indícios de não linearidade ou heterocedasticidade. Esses resultados indicam que o modelo conseguiu realizar a predição da Repetição de Usuário.

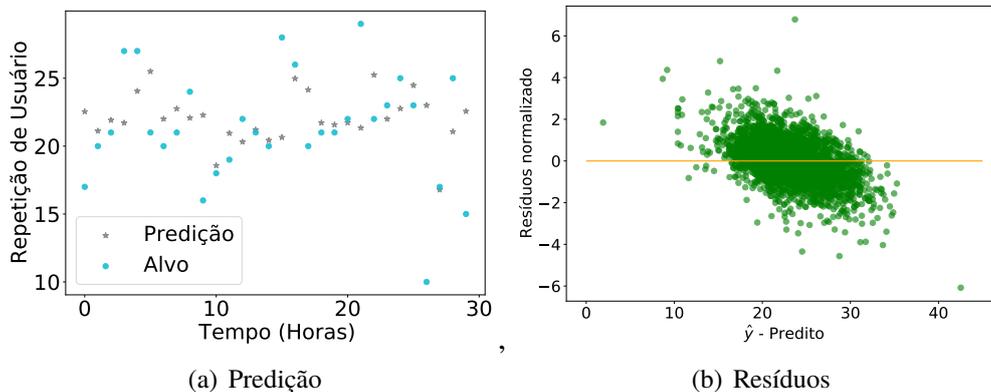


Figura 6.5: Predição da Repetição de Usuário utilizando LSTM.

Neste capítulo, foi apresentado um estudo de caso com a proposta de uma métrica para identificar períodos de atividade maliciosa no Bitcoin, chamada de Repetição de Usuário. Foram testados modelos de rede neural para prever a Repetição de Usuário e verifica-se que o modelo LSTM obtém os melhores resultados.

Capítulo 7

Conclusões e Trabalhos Futuros

Esta dissertação abordou o uso de modelos de redes neurais para a predição de métricas em grafos temporais. Grafos temporais são amplamente utilizados para a modelagem de sistemas nos quais os enlaces e os nós do grafo podem ser modificados ao longo do tempo. Ou seja, nós antigos podem sair ou novos nós podem entrar no grafo e enlaces podem surgir ou desaparecer da rede. Há trabalhos na literatura cujo foco é a predição de métricas em grafos temporais. No entanto, os trabalhos são concentrados em domínios específicos, como a predição do tráfego em redes rodoviárias ou na predição da existência de novos enlaces na rede. Assim, não há trabalhos cujo foco seja a predição de métricas básicas, como o coeficiente de clusterização e o grau médio. Ambas as métricas fornecem informações importantes sobre a estrutura das redes. Dessa forma, a capacidade de prever o valor futuro dessas métricas pode auxiliar na tomada de decisões.

Nesta dissertação, foram utilizados os modelos de rede neural CNN, LSTM e MLP na predição de métricas de grafos temporais. Além disso, foi utilizado um modelo de base para comparar o desempenho dos modelos de rede neural. Os modelos CNN e LSTM obtiveram os melhores resultados, com exceção apenas de um caso particular, evidenciando a sua aplicabilidade nos problemas tratados neste trabalho [48]. O caso particular se refere à previsão do grau médio para a base de dados CollegeMsg, que possui erros de predição, mesmo com resultados satisfatórios nos erros MAE e RMSE. Como essa base apresenta dias no qual não há troca de mensagens entre os nós, com grau médio e coeficiente de clusterização zerado, a suspeita é a de que a ausência de dados prejudicou a capacidade dos modelos e aprender. No entanto, mais análises precisam ser realizadas para validar essa hipótese e serão abordadas em trabalhos futuros.

O desempenho das redes neurais na predição das métricas motivou o desenvolvimento de um estudo de caso, com a base de dados do Bitcoin. Esse estudo investigou as suspeitas de atividades maliciosas da corretora de Bitcoin Mt.Gox. O objetivo foi investigar se o comportamento malicioso dessa corretora, reportado em diversos estudos feitos com a base nos dados vazada no ano de 2014, mudaram o comportamento esperado dos nós da *blockchain*, gerando instabilidade nas métricas de grafo. Para tal, a rede do Bitcoin foi

modelada como um grafo não direcionado e uma métrica baseada na centralidade de grau foi proposta. Essa métrica visa analisar como os nós com melhor classificação mudam ao longo do tempo e examinar se essas mudanças podem estar relacionadas à atividade maliciosa da Mt.Gox e com o aumento do preço do Bitcoin no período.

As análises mostram que, durante o período de suspeita de manipulação, a métrica teve alteração no seu comportamento, com menos nós sendo repetidos no ranking. Esses resultados levaram a uma análise de estabilidade dos nós, na qual foram simulados três cenários de inserção de novas transações na rede e analisado o impacto desses cenários na métrica. No cenário no qual poucos agentes eram responsáveis por um grande número de transações, a métrica apresentava o mesmo comportamento visto no período de suspeita de manipulação de mercado pela Mt.Gox. Esses resultados levaram à conclusão de que a métrica proposta é capaz de reconhecer períodos de Repetição de Usuário com indícios de serem causados por anomalias na rede do Bitcoin e não um comportamento natural dos usuários. Dessa forma, a capacidade de prever os valores futuros da Repetição de Usuário pode fornecer indícios de períodos com atividade anormal. Assim, adotou-se um estudo de caso, usando o mesmo procedimento empregado na predição do coeficiente de clusterização e do grau médio. Verificou-se que o modelo LSTM obteve o melhor desempenho na predição da Repetição de Usuário.

Apesar de os resultados mostrarem a capacidade das redes neurais de preverem as métricas do grafo, é necessário realizar uma comparação com métodos clássicos e lineares, como o ARIMA. Essa análise será realizada em trabalhos futuros. Além disso, os resultados mostraram que o melhor modelo de predição pode diferir dependendo da base considerada. Assim, um trabalho futuro é a ampliação do número de bases, a fim de investigar se há algum tipo de característica da base que beneficie algum modelo específico.

Referências Bibliográficas

- [1] HEGEMAN, T., IOSUP, A. “Survey of Graph Analysis Applications”, *arXiv preprint arXiv:1807.00382*, 2018.
- [2] KOSTAKOS, V. “Temporal graphs”, *Physica A: Statistical Mechanics and its Applications*, v. 388, n. 6, pp. 1007–1023, 2009.
- [3] CHARAKOPOULOS, A., KARAKASIDIS, T., PAPANICOLAOU, P., et al. “The application of complex network time series analysis in turbulent heated jets”, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, v. 24, n. 2, pp. 024408, 2014.
- [4] TANG, J., LIU, F., ZHANG, W., et al. “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*, v. 450, pp. 635–648, 2016.
- [5] TANG, J., WANG, Y., WANG, H., et al. “Dynamic analysis of traffic time series at different temporal scales: A complex networks approach”, *Physica A: Statistical Mechanics and its Applications*, v. 405, pp. 303–315, 2014.
- [6] LI, M., O’RIORDAN, C. “The effect of clustering coefficient and node degree on the robustness of cooperation”. In: *2013 IEEE Congress on Evolutionary Computation*, pp. 2833–2839. IEEE, 2013.
- [7] NICOSIA, V., TANG, J., MASCOLO, C., et al. “Graph metrics for temporal networks”. In: *Temporal networks*, Springer, pp. 15–40, 2013.
- [8] SANTORO, N., QUATTROCIOCCHI, W., FLOCCHINI, P., et al. “Time-varying graphs and social network analysis: Temporal indicators and metrics”, *arXiv preprint arXiv:1102.0629*, 2011.
- [9] BOCHIE, K., DA SILVA GILBERT, M., GANTERT, L., et al. “Aprendizado Profundo em Redes Desafiadoras: Conceitos e Aplicações”. In: *Minicursos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 1–50, 2020.

- [10] GANDAL, N., HAMRICK, J., MOORE, T., et al. “Price manipulation in the Bitcoin ecosystem”, *Journal of Monetary Economics*, v. 95, pp. 86–96, 2018.
- [11] HYNDMAN, R. J., ATHANASOPOULOS, G. *Forecasting: principles and practice*. "Monash University, Australia", OTexts, 2018.
- [12] ROSENBLATT, F. “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, v. 65, n. 6, pp. 386, 1958.
- [13] FRANK, R. J., DAVEY, N., HUNT, S. P. “Time series prediction and neural networks”, *Journal of intelligent and robotic systems*, v. 31, n. 1, pp. 91–103, 2001.
- [14] KOURENTZES, N., BARROW, D. K., CRONE, S. F. “Neural network ensemble operators for time series forecasting”, *Expert Systems with Applications*, v. 41, n. 9, pp. 4235–4244, 2014.
- [15] CLAVERIA, O., TORRA, S. “Forecasting tourism demand to Catalonia: Neural networks vs. time series models”, *Economic Modelling*, v. 36, pp. 220–228, 2014.
- [16] TORRES, J. F., GALICIA, A., TRONCOSO, A., et al. “A scalable approach based on deep learning for big data time series forecasting”, *Integrated Computer-Aided Engineering*, v. 25, n. 4, pp. 335–348, 2018.
- [17] LARA-BENÍTEZ, P., CARRANZA-GARCÍA, M., RIQUELME, J. C. “An Experimental Review on Deep Learning Architectures for Time Series Forecasting”, *arXiv preprint arXiv:2103.12057*, 2021.
- [18] HOCHREITER, S. “The vanishing gradient problem during learning recurrent neural nets and problem solutions”, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, v. 6, n. 02, pp. 107–116, 1998.
- [19] SCHMIDHUBER, J., HOCHREITER, S. “Long short-term memory”, *Neural Comput*, v. 9, n. 8, pp. 1735–1780, 1997.
- [20] HEWAMALAGE, H., BERGMEIR, C., BANDARA, K. “Recurrent neural networks for time series forecasting: Current status and future directions”, *International Journal of Forecasting*, 2020.
- [21] BOROVYKH, A., BOHTE, S., OOSTERLEE, C. W. “Conditional time series forecasting with convolutional neural networks”, *arXiv preprint arXiv:1703.04691*, 2017.

- [22] NIELSEN, A. *Practical time series analysis: prediction with statistics and machine learning*. New York, "O'Reilly Media, Inc.", 2019.
- [23] SIAMI-NAMINI, S., TAVAKOLI, N., NAMIN, A. S. "A comparison of ARIMA and LSTM in forecasting time series". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401. IEEE, 2018.
- [24] DE GOOIJER, J. G., HYNDMAN, R. J. "25 years of time series forecasting", *International journal of forecasting*, v. 22, n. 3, pp. 443–473, 2006.
- [25] MBAH, T. J., YE, H., ZHANG, J., et al. "Using LSTM and ARIMA to Simulate and Predict Limestone Price Variations", *Mining, Metallurgy & Exploration*, pp. 1–14, 2021.
- [26] MAKRIDAKIS, S., SPILLOTIS, E., ASSIMAKOPOULOS, V. "Statistical and Machine Learning forecasting methods: Concerns and ways forward", *PloS one*, v. 13, n. 3, pp. e0194889, 2018.
- [27] SELVIN, S., VINAYAKUMAR, R., GOPALAKRISHNAN, E., et al. "Stock price prediction using LSTM, RNN and CNN-sliding window model". In: *2017 international conference on advances in computing, communications and informatics (icacci)*, pp. 1643–1647. IEEE, 2017.
- [28] SIAMI-NAMINI, S., TAVAKOLI, N., NAMIN, A. S. "A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BILSTM", *arXiv preprint arXiv:1911.09512*, 2019.
- [29] SAGHEER, A., KOTB, M. "Time series forecasting of petroleum production using deep LSTM recurrent networks", *Neurocomputing*, v. 323, pp. 203–213, 2019.
- [30] HUA, Y., ZHAO, Z., LI, R., et al. "Deep learning with long short-term memory for time series prediction", *IEEE Communications Magazine*, v. 57, n. 6, pp. 114–119, 2019.
- [31] KIM, S., KANG, M. "Financial series prediction using Attention LSTM", *arXiv preprint arXiv:1902.10877*, 2019.
- [32] BINKOWSKI, M., MARTI, G., DONNAT, P. "Autoregressive convolutional neural networks for asynchronous time series". In: *International Conference on Machine Learning*, pp. 580–589. PMLR, 2018.
- [33] LIM, B., ZOHREN, S. "Time-series forecasting with deep learning: a survey", *Philosophical Transactions of the Royal Society A*, v. 379, n. 2194, pp. 20200209, 2021.

- [34] YU, B., YIN, H., ZHU, Z. “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting”, *arXiv preprint arXiv:1709.04875*, 2017.
- [35] CUI, Z., HENRICKSON, K., KE, R., et al. “Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting”, *IEEE Transactions on Intelligent Transportation Systems*, v. 21, n. 11, pp. 4883–4894, 2019.
- [36] CEM, E., SARAC, K. “Estimation of structural properties of online social networks at the extreme”, *Computer Networks*, v. 108, pp. 323–344, 2016.
- [37] KONDOR, D., PÓSFAL, M., CSABAI, I., et al. “Do the rich get richer? An empirical analysis of the Bitcoin transaction network”, *PloS one*, v. 9, n. 2, 2014.
- [38] LISCHKE, M., FABIAN, B. “Analyzing the bitcoin network: The first four years”, *Future Internet*, v. 8, n. 1, pp. 7, 2016.
- [39] MAESA, D. D. F., MARINO, A., RICCI, L. “Uncovering the bitcoin blockchain: an analysis of the full users graph”. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 537–546, 2016.
- [40] MAESA, D. D. F., MARINO, A., RICCI, L. “An analysis of the bitcoin users graph: inferring unusual behaviours”. In: *International Workshop on Complex Networks and their Applications*, pp. 749–760, 2016.
- [41] MAESA, D. D. F., MARINO, A., RICCI, L. “Detecting artificial behaviours in the bitcoin users graph”, *Online Social Networks and Media*, v. 3, pp. 63–74, 2017.
- [42] SUBERG, W. “Mt. Gox Trial Update: Karpeles Admits ‘Willy Bot’ Existence”. 2017. <https://cointelegraph.com/news/mt-gox-trial-update-karpeles-admits-willy-bot-existence>, accessed April 2020.
- [43] CHEN, W., WU, J., ZHENG, Z., et al. “Market manipulation of bitcoin: Evidence from mining the Mt. Gox transaction network”. In: *infocom*, pp. 964–972, 2019.
- [44] NAKAMOTO, S. *Bitcoin: A peer-to-peer electronic cash system*. Relatório técnico, Manubot, 2019.
- [45] BERGSTRA, J., BENGIO, Y. “Random search for hyper-parameter optimization”, *Journal of machine learning research*, v. 13, n. 2, 2012.
- [46] KINGMA, D. P., BA, J. “ADAM: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.

- [47] BLOCKCHAIN.INFO. “Bitcoin Market Price”. 2020.
<https://blockchain.info/charts/market-price>, accessed April 2020.
- [48] PEREIRA, D. M., COUTO, R. S. “Predição de Métricas em Grafos Temporais Utilizando Redes Neurais”. In: *Anais do XX Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, pp. 84–95. SBC, 2021.