

# UNIVERSIDADE DE SÃO PAULO

## Instituto de Ciências Matemáticas e de Computação

---

Predição espaço-temporal de eventos usando redes neurais  
convolucionais para grafos

*Pedro Henrique Rampim Natali*

---



São Carlos – SP



# Predição espaço-temporal de eventos usando redes neurais convolucionais para grafos

**Pedro Henrique Rampim Natali**

*Orientador: Prof. Dr. Ricardo Marcondes Marcacini*

Monografia final de conclusão de curso apresentada  
ao Instituto de Ciências Matemáticas e de  
Computação – ICMC-USP, como requisito parcial  
para obtenção do título de Bacharel em Engenharia  
de Computação.

**USP – São Carlos**  
**Junho de 2021**



*“Se eu vi mais longe,  
foi por estar sobre ombros de gigantes.”*  
*(Isaac Newton)*



# RESUMO

NATALI, P. H. R.. **Predição espaço-temporal de eventos usando redes neurais convolucionais para grafos**. 2021. 45 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Eventos observáveis no mundo real são muitas vezes formados por uma série de fatores distintos de áreas diferentes. Com a grande quantidade de dados disponíveis, uma das maneiras de realizar análises desses eventos é a partir de sensores textuais. Um exemplo recente são os sensores relacionados aos eventos de COVID-19. Nesse caso, cada sensor é responsável por classificar e monitorar eventos específicos dentro de um conjunto de dados. Um método para realização da análise temporal é a predição de tendências envolvendo a ocorrência de eventos, que pode ser realizada por meio de métodos de regressão, especialmente com uso de aprendizado de máquina para predição em séries temporais. No entanto, estudos recentes indicam que os eventos podem estar relacionados também espacialmente, ou seja, um ou mais sensores de eventos podem estar relacionados geograficamente. Como solução para desafios desse tipo, novos algoritmos foram propostos para lidar com predição espaço-temporal. Nesse sentido, um método investigado neste projeto é o STGCN (*Spatio-Temporal Graph Convolutional Networks*), que representa um conjunto de sensores por meio de grafos. Dessa forma, este trabalho visa a avaliação do STGCN para predição espaço-temporal em bases de eventos. Nessa avaliação é explorado um conjunto de eventos extraídos de notícias sobre COVID-19, organizados em mais de 200 sensores (subtemas). Para verificar o impacto da estrutura do grafo e informações geográficas, o STGCN foi comparado com o método *Prophet*, proposto pela equipe de pesquisadores do *Facebook* para previsões de séries temporais. O trabalho apresenta como conclusão a importância em considerar informações geográficas na análise de eventos, em especial, na predição espaço-temporal de eventos. Além da comparação entre os modelos, o projeto finaliza apresentando um repositório de código-fonte aberto para realizar análise de eventos via STGCN e *Prophet* de forma simplificada.

**Palavras-chave:** Predição Espaço-Temporal, Redes Neurais Convolucionais, STGCN, *Prophet*, COVID-19.



# ABSTRACT

NATALI, P. H. R.. **Predição espaço-temporal de eventos usando redes neurais convolucionais para grafos.** 2021. 45 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Observable events in the real world are often formed by a number of distinct factors from different fields. With the large amount of data available, one of the ways to carry out an analysis of these events is from textual sensors. A recent example is sensors related to COVID-19 events. In this case, each sensor is responsible for classifying and monitoring specific events within a dataset. One method for performing temporal analysis is the prediction of trends involving the occurrence of events, which can be performed using regression methods, especially with the use of machine learning for prediction in time series. However, recent studies indicate that events may also be spatially related, that is, one or more event sensors may be geographically related. As a solution to such challenges, new algorithms have been proposed to deal with spatiotemporal prediction. In this sense, a method investigated in this project is the STGCN (Spatio-Temporal Graph Convolutional Networks), which represents a set of sensors through graphs. Thus, this work aims to evaluate the STGCN for spatiotemporal prediction in event databases. This evaluation explores a set of events extracted from news about COVID-19, organized in more than 200 sensors (sub-themes). To verify the impact of graph structure and geographic information, STGCN was compared with the Prophet method, proposed by the *Facebook* research team for time series predictions. The work presents as a conclusion the importance of considering geographic information in the analysis of events, especially in the spatiotemporal prediction of events. In addition to comparing the models, the project ends up presenting an open source repository to perform event analysis via STGCN and Prophet in a simplified way.

**Key-words:** Spatiotemporal Forecast, Convolutional Neural Networks, STGCN, Prophet, COVID-19.



---

# LISTA DE ILUSTRAÇÕES

---

Figura 1 – Exemplo de eventos sobre Covid-19 extraídos da plataforma Websensors. . . . .	16
Figura 2 – Evolução temporal de eventos classificados no sensor “ <i>anti vaxxers</i> ”. Eventos coletados da base GDELT no período de Outubro de 2019 até Maio de 2021. . . . .	16
Figura 3 – Variação da População do Brasil de 1800 até 2000. . . . .	21
Figura 4 – Índice de interesse relativo de pesquisa sobre a Copa do Mundo no Brasil de 2004 até 2021. . . . .	21
Figura 5 – Índice de interesse relativo de pesquisa sobre Osama Bin Laden no Brasil de 2004 até 2021. . . . .	22
Figura 6 – Regressão linear aplicada ao conjunto de dados da população do Brasil. . . . .	23
Figura 7 – Ilustração da morfologia geral do neurônio e de seus componentes, apresentando o sentido de condução (impulso nervoso) da informação, da recepção (corpo celular e dendritos) a transmissão (ramificações do axônio). . . . .	24
Figura 8 – Ilustração evidenciando um contato entre neurônios (sinapse), no qual ocorre a transmissão da informação por meio de mediadores químicos (neurotransmissores). . . . .	25
Figura 9 – Modelo de um neurônio artificial . . . . .	25
Figura 10 – Modelo de MLP recorrente . . . . .	26
Figura 11 – Arquitetura proposta pelo autor. . . . .	32
Figura 12 – Total de eventos por período. . . . .	33
Figura 13 – Frequência relativa de eventos por país. . . . .	34
Figura 14 – Exemplo de série temporal para sensor sobre eventos de vacinas. . . . .	34
Figura 15 – Exemplo de série temporal para sensor sobre eventos de <i>lockdown</i> . . . . .	35
Figura 16 – Exemplo de série temporal para sensor sobre eventos de distanciamento social. . . . .	35
Figura 17 – Variação do MAE de acordo com o tamanho do lote. . . . .	37
Figura 18 – Variação do MAE de acordo com o número de histórico. . . . .	38
Figura 19 – Variação do MAE de acordo com a taxa de aprendizado. . . . .	38



## LISTA DE TABELAS

---

---

Tabela 1 – Lista de Parâmetros do STGCN e sua Descrição. . . . .	36
Tabela 2 – Melhores parâmetros obtidos através da iteração. . . . .	36
Tabela 3 – Resultados em função do MAE e RMSE para os métodos avaliados. . . . .	38



# SUMÁRIO

---

---

1	<b>INTRODUÇÃO</b>	15
1.1	<b>Motivação e Contextualização</b>	15
1.2	<b>Objetivo</b>	17
1.3	<b>Organização</b>	17
2	<b>MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS</b>	19
2.1	<b>Fundamentos de Séries Temporais</b>	19
2.1.1	<i>Decomposição de uma Série Temporal</i>	19
2.1.1.1	<i>Tendência</i>	20
2.1.1.2	<i>Sazonalidade</i>	20
2.1.1.3	<i>Resíduo</i>	22
2.2	<b>Análise e Predição em Séries Temporais</b>	22
2.3	<b>Spatio-Temporal Graph Convolutional Networks</b>	26
2.3.1	<i>Convolução em grafos</i>	26
2.3.2	<i>Modelo e Predição</i>	28
2.4	<b>Business Time Series Prediction</b>	29
3	<b>DESENVOLVIMENTO</b>	31
3.1	<b>Descrição do Problema</b>	31
3.2	<b>Atividades Realizadas</b>	32
3.3	<b>Resultados</b>	36
3.4	<b>Dificuldades e Limitações</b>	39
4	<b>CONCLUSÃO</b>	41
4.1	<b>Contribuições</b>	41
4.2	<b>Trabalhos Futuros</b>	41
	<b>REFERÊNCIAS</b>	43



# Capítulo 1

## INTRODUÇÃO

---



---

### 1.1 Motivação e Contextualização

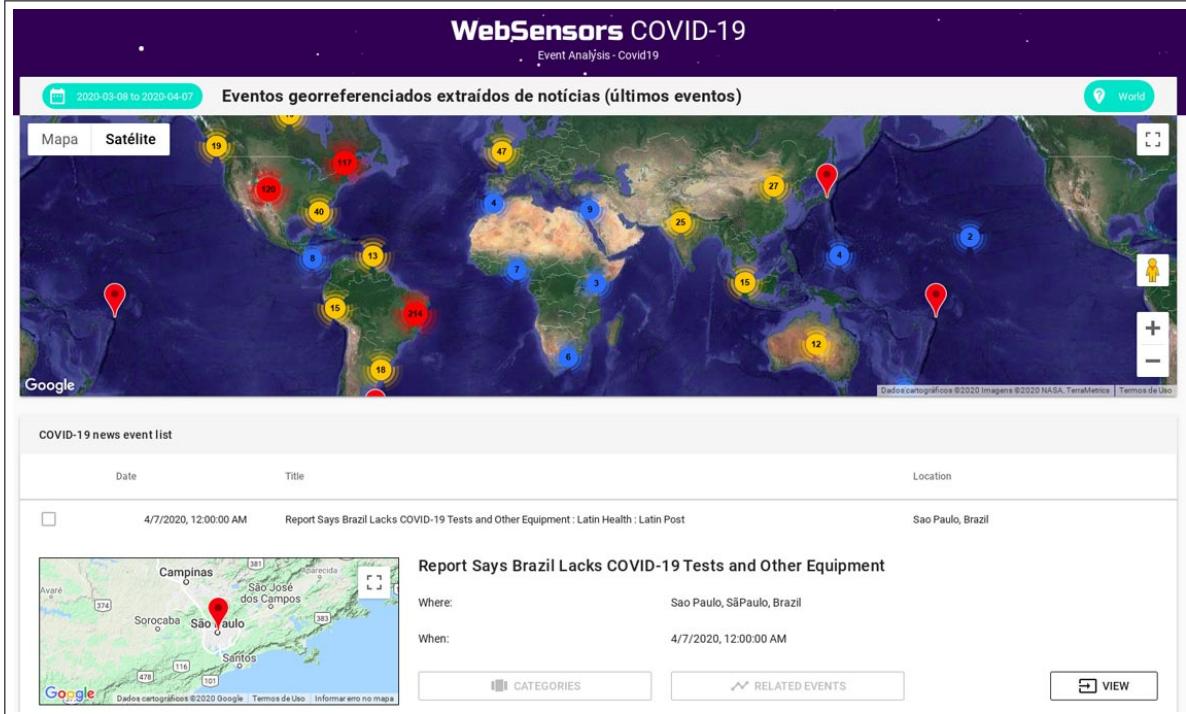
Eventos observáveis no mundo real são muitas vezes intercalados por uma série de fatores distintos de áreas diferentes (CHEN; LI, 2020). Uma crise financeira em um país, por exemplo, pode ocorrer devido a fatores relacionados com política, economia, sociedade dentre outras áreas (RADINSKY; HORVITZ, 2013). Com a grande quantidade de dados disponíveis, uma das maneiras de realizar uma análise de eventos é a partir de sensores textuais (HATTORI, 2012; MARCACINI *et al.*, 2017; MARCACINI; CARNEVALI; DOMINGOS, 2016; SANTOS *et al.*, 2020). Esses sensores são modelos que monitoram e classificam eventos extraídos de fontes públicas da *web*, incluindo também etapas de pré-processamento dos dados para identificar informações geográficas, temporais, pessoas e organizações. Assim, um conjunto de sensores pode mapear temas correlacionados. Cada sensor representa eventos de um subtema, mapeando assim as informações por meio de métricas específicas, como a frequência de eventos ao longo do tempo e locais de ocorrência (RADINSKY; DAVIDOVICH; MARKOVITCH, 2012; CHEN; LI, 2020).

Um exemplo recente são os sensores relacionados aos eventos de COVID-19. Nesse caso, cada sensor é responsável por classificar e monitorar eventos específicos dentro de um conjunto de dados. Exemplos de eventos são esforços pela aquisição e aplicação de vacina, distanciamento social, impactos econômicos, dentre outros. Na Figura 1 é ilustrado um exemplo de eventos extraídos de notícias obtido da plataforma Websensors-Covid19, em que os eventos são coletados, classificados em um dos sensores (de acordo com o tema), e organizados em um mapa (TUNES, 2020).

Além da organização do conjunto de eventos disponíveis em temas e subtemas, é importante analisar a evolução temporal desses subtemas para determinadas regiões. Com isso, sensores geram séries temporais que podem indicar tendências de ocorrência de um conjunto de eventos para um determinado subtema. Para exemplificar, na Figura 2 é ilustrado uma série temporal para um sensor que foi treinado com eventos sobre “*anti vaxxers*”. Para este sensor, por exemplo, outros eventos da coleção são identificados por meio de uma medida de similaridade (e.g. similaridade cosseno e estratégia de vizinho mais próximo), indicando a ocorrência desse subtema (similaridade média) ao longo do tempo<sup>1</sup>.

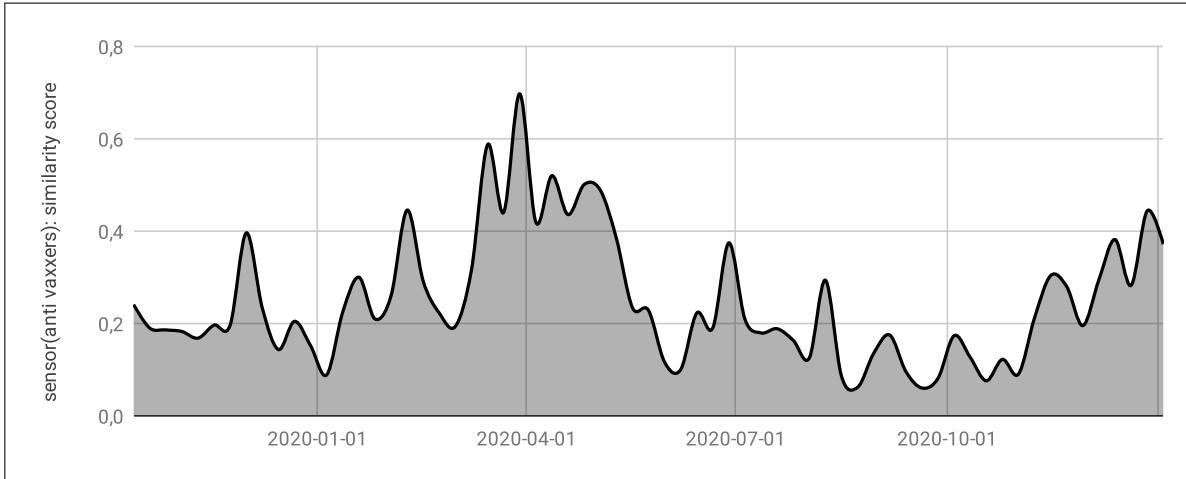
<sup>1</sup> Exemplo construído com eventos da base GDELT, no período de Outubro de 2019 até Maio de 2021

Figura 1 – Exemplo de eventos sobre Covid-19 extraídos da plataforma Websensors.



Fonte: [Tunes \(2020\)](#)

Figura 2 – Evolução temporal de eventos classificados no sensor “anti vaxxers”. Eventos coletados da base GDELT no período de Outubro de 2019 até Maio de 2021.



Fonte: Dados do projeto

A análise temporal de conjuntos de eventos é uma área de estudo que vêm recebendo destaque nos últimos anos ([ZHAO et al., 2015](#); [MARCACINI; CARNEVALI; DOMINGOS, 2016](#); [NING et al., 2019](#); [CHEN; LI, 2020](#)). A predição de tendências envolvendo a ocorrência de eventos pode ser realizada por meio de métodos de regressão, especialmente com uso de aprendizado de máquina para predição em séries temporais ([PARMEZAN; SOUZA; BATISTA, 2019](#)). No entanto, estudos recentes indicam que os eventos podem estar relacionados também espacialmente, ou seja, um ou mais sensores de eventos podem estar relacionados geograficamente ([NING et al., 2019](#)). Como solução para desafios desse tipo, novos algoritmos foram propostos

para lidar com predição espaço-temporal (YU; YIN; ZHU, 2018; NING *et al.*, 2019; MIN *et al.*, 2021). A ideia geral é explorar matrizes de distância espacial entre os sensores, conforme sua localização, junto com as respectivas séries temporais (e.g. provenientes de sensores).

Nesse sentido, um método investigado neste projeto é o STGCN (*Spatio-Temporal Graph Convolutional Networks*) (YU; YIN; ZHU, 2018), que representa um conjunto de sensores por meio de grafos. Cada vértice do grafo é um sensor. A série temporal do sensor é representada como um vetor de características associado ao vértice. A informação de distância geográfica, por sua vez, é representada por meio de pesos nas arestas entre vértices. A vantagem dessa modelagem é treinar modelos de predição usando *Graph Convolutional Networks* (GCNs) (SCHLICHTKRULL *et al.*, 2018), na qual são exploradas dependências espaciais do grafo, ou seja, as relações de vizinhança entre vértices, além dos vetores de características (séries temporais). Esse tipo de abordagem têm sido utilizado com sucesso para aplicações similares, por exemplo, envolvendo sensores na área de transporte público (YU; YIN; ZHU, 2018), análise de epidemias (KAPOOR *et al.*, 2020), e violência urbana (MIN *et al.*, 2021). Resultados promissores nesses domínios motivou a avaliação desse método para predição espaço-temporal envolvendo bases de eventos.

## 1.2 Objetivo

O objetivo principal deste trabalho é avaliar o método STGCN (*Spatio-Temporal Graph Convolutional Networks*) para predição espaço-temporal em bases de eventos. Nessa avaliação é explorado um conjunto de eventos extraídos de notícias sobre COVID-19, organizados em mais de 200 sensores (subtemas). Para verificar o impacto da estrutura do grafo e informações geográficas, o STGCN foi comparado com o método *Prophet* (TAYLOR; LETHAM, 2017), proposto pela equipe de pesquisadores do *Facebook* para predições de séries temporais. Para atingir esse objetivo principal, foram propostos os seguintes objetivos específicos:

- Avaliar os parâmetros do método STGCN proposto por Yu, Yin e Zhu (2018) no contexto de predição espaço-temporal de eventos;
- Avaliar o método Propheta (TAYLOR; LETHAM, 2017) para predição de séries temporais construídas a partir de bases de eventos;
- Desenvolver e disponibilizar um repositório de código aberto para uso do STGCN em análise de eventos.

## 1.3 Organização

No [Capítulo 2](#) é apresentada uma introdução aos fundamentos de séries temporais, delimitando seus principais componentes. Além disso, demonstra como é feita a análise e

predição nas séries. O capítulo apresenta conceitos de redes neurais, convolução em grafos, predição de séries temporais de negócios, métodos indicadores de erro e por fim demonstra os critérios de avaliação que serão utilizados para a comparação dos resultados.

O [Capítulo 3](#) é responsável pelo desenvolvimento do projeto, ou seja, é feita a descrição do problema proposto, demonstra as atividades realizadas, os resultados obtidos e por fim as dificuldades e limitações do projeto.

Finalmente, o [Capítulo 4](#) apresenta as conclusões sobre os resultados demonstrados, as contribuições que podem ser feitas em trabalhos futuros e uma análise sobre o curso de graduação em relação ao projeto.

## Capítulo 2

# MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS

---



---

## 2.1 Fundamentos de Séries Temporais

Séries temporais são compostas por um conjunto de dados igualmente (séries regulares) ou não (séries irregulares) espaçados no tempo (HAMILTON, 2020). Tratam-se de observações sequenciais onde a ordem faz diferença. Ou seja, em séries temporais há uma dependência na ordem explícita em que os eventos ocorrem, diferente de conjuntos de dados mais comuns utilizados para predição, em que cada evento é tratado igualmente perante os outros, sem um fator de dependência relacionado à ordem de ocorrência (PARMEZAN; SOUZA; BATISTA, 2019). As séries por sua vez podem ser uni-variadas - quando somente uma variável está presente - ou multivariada, no caso em que duas ou mais variáveis estão presentes no conjunto. Formalmente, uma série temporal  $Z$  de tamanho  $m$  é definida por  $Z = (z_1, z_2, \dots, z_m)$ , onde  $z_t \in \mathbb{R}$  representa uma observação  $z$  no instante de tempo  $t$  (BONTEMPI; TAIEB; BORGNE, 2012).

Os conjuntos de dados que compõem as séries temporais, por sua vez, podem ser determinísticos - quando há a possibilidade de determinação de uma função matemática em relação ao tempo. Ou ainda, estocástica (não-determinística), quando para que seja possível modelar a função, seja necessário incluir um termo aleatório (ruído) (PARMEZAN; SOUZA; BATISTA, 2019). Além disso, podemos também caracterizar tais conjuntos como discretos e contínuos, dependendo do espaçamento descrito pelas observações da série. Por fim, uma série temporal pode ser também classificada como estacionária, quando no tempo ocorre um desenvolvimento em torno de uma média constante. Para a realização de análises sobre os dados encontrados para uma série, utilizam-se os conceitos abordados até agora, porém com a ajuda dos componentes principais de uma série temporal, sendo eles a tendência, a sazonalidade e o resíduo (HAMILTON, 2020).

### 2.1.1 *Decomposição de uma Série Temporal*

Uma série temporal pode ser decomposta a fim do melhor entendimento de seus eventos em três principais componentes (note que não é obrigatório para uma série apresentar todos os três componentes). São eles a tendência (movimento desenvolvido ao longo da série), a sazonalidade (comportamento que se repete em períodos específicos de tempo) e o resíduo

(comportamentos gerados devido a eventos aleatórios). Sua decomposição pode ocorrer de modo aditivo, onde a série temporal é representada pela soma dos três termos, ou ainda de modo multiplicativo, onde a série é representada pela multiplicação dos termos de modo que a tendência apresenta a mesma unidade da série e os demais componentes apenas modificam a tendência.

### 2.1.1.1 *Tendência*

A tendência de uma série temporal pode ser definida como o padrão de crescimento da variável analisada em um período de tempo. Fatores que podem causar tendência são variados, dentre eles o aumento no uso de tecnologias, aumento da população (e por sua vez, no consumo), entre outros. Os tipos de crescimento que uma tendência pode apresentar são o crescimento linear, onde a taxa de crescimento é constante, o crescimento exponencial, onde as taxas de crescimento respeitam uma função exponencial, e por fim o crescimento amortecido, onde a taxa de crescimento é negativa. Com a tendência encontrada, torna-se possível retirá-la da série temporal para realçar os outros componentes, realizar a análise do seu comportamento para integrar um modelo preditivo, ou ainda determinar o nível (valor que a variável tende a assumir caso não ocorra uma tendência crescente ou decrescente) da série.

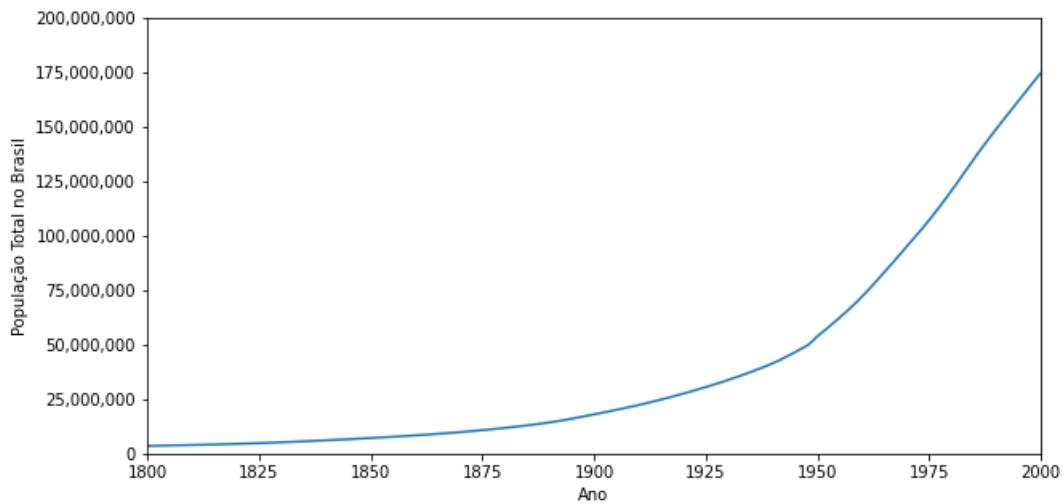
Um exemplo claro de uma tendência é o crescimento populacional. Se analisarmos o crescimento da população no Brasil de 1800 até 2000 na [Figura 3](#), fica claro que a tendência é uma exponencial crescente. A tendência assim é o comportamento constante que ocorre a longo prazo. Outros exemplos podem ser notados em investimentos, onde um ativo pode oscilar ao longo do tempo, porém apresenta um comportamento linear (ou exponencial) crescente ou decrescente. A identificação da componente de tendência em séries temporais depende de um período de observação. Por exemplo, embora seja possível identificar a tendência para um determinado período de tempo, essa componente pode mudar drasticamente ao se observar períodos de tempo maiores — o que torna desafiador o processo de análise de séries temporais.

### 2.1.1.2 *Sazonalidade*

A sazonalidade de uma série temporal pode ser definida como um comportamento que se repete durante diferentes períodos de tempo. Essas variações por sua vez são representadas por oscilações ao longo da componente de tendência. Exemplos práticos de sazonalidade podem se dar por conta das estações do ano, causas econômicas e políticas (eleições, por exemplo) ou ainda eventos que ocorrem a cada período de tempo. Para encontrar a sazonalidade, uma das técnicas é a análise gráfica, identificando picos e vales regularmente espaçados com magnitude aproximadamente igual. A componente em questão pode ser categorizada em aditiva, quando a sazonalidade apresenta uma flutuação estável, sem considerar o nível global da série, ou ainda em multiplicativa, quando a flutuação varia de acordo com o nível global da série.

Dois exemplos muito claros de sazonalidade podem ser eventos políticos (como eleições)

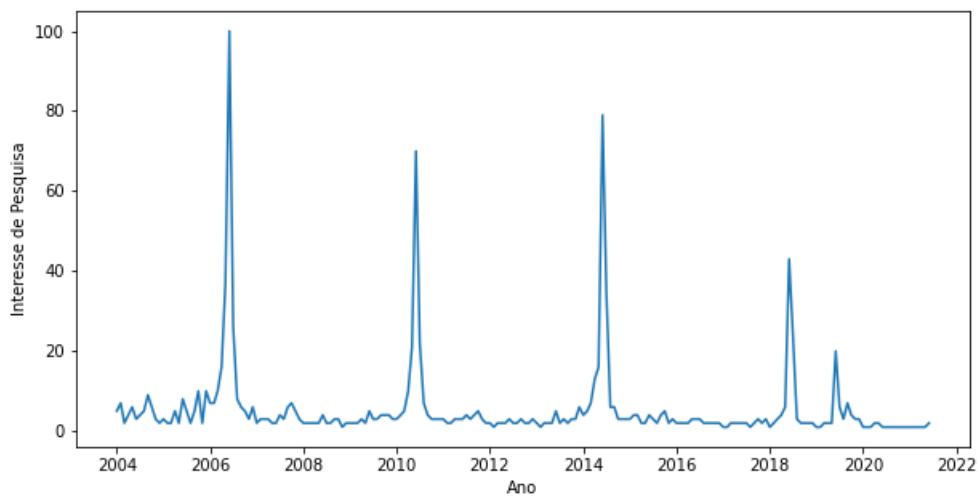
Figura 3 – Variação da População do Brasil de 1800 até 2000.



Fonte: [Roser e Ortiz-Ospina \(2013\)](#).

e eventos esportivos (Copa do Mundo ou Olimpíadas, por exemplo). Se analisarmos a [Figura 4](#), onde o eixo X representa o passar dos anos e o eixo Y representa um índice de interesse relativo da plataforma *Google* sobre a pesquisa de termos relacionados a Copa do Mundo no Brasil, é possível ver claramente a presença da sazonalidade no aumento do índice a cada quatro anos. Ainda, é possível analisar comportamentos relativamente semelhantes a um resíduo (próximo conceito a ser abordado) em 2020 ou em 2006 (cuja pesquisa obteve índices maiores que o normal no Brasil devido ao jogo icônico contra a seleção da França).

Figura 4 – Índice de interesse relativo de pesquisa sobre a Copa do Mundo no Brasil de 2004 até 2021.



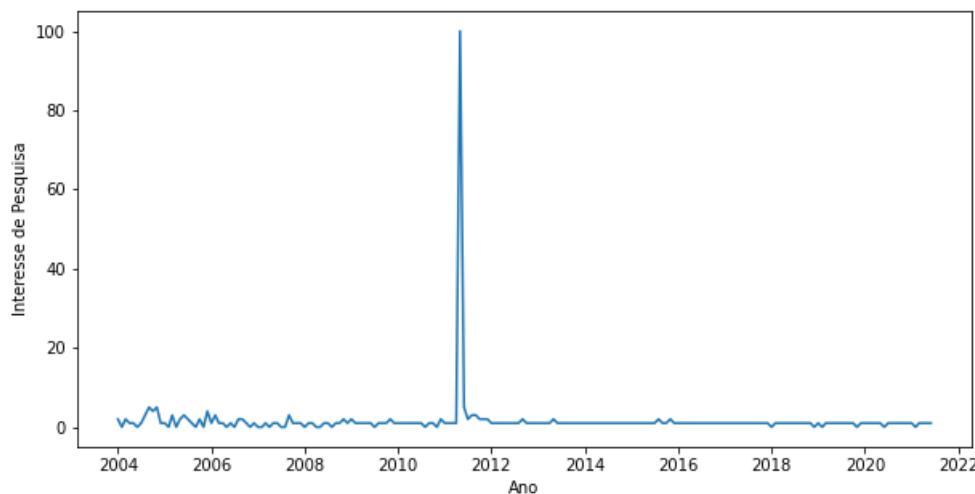
Fonte: Adaptada de [Google Trends](http://www.google.com/trends) ([www.google.com/trends](http://www.google.com/trends)).

### 2.1.1.3 Resíduo

O resíduo de uma série temporal pode ser definido como o restante da série após a remoção dos componentes sistemáticos, ou seja, aqueles que podem ser definidos por funções determinísticas no tempo - ou ainda - a sazonalidade e a tendência. Resíduos podem ocorrer por diversos fatores, como desastres naturais, acidentes, ou acontecimentos que não se repetem de maneira regular. E por sua vez, devem ser identificados para que seja possível remover sua componente da série temporal e realizar assim uma análise mais concisa.

Como explicado anteriormente, eventos incomuns podem causar resíduo na análise de séries temporais. Um exemplo relativamente simples é o disposto na [Figura 5](#), onde o eixo X representa o passar dos anos e o eixo Y representa um índice de interesse relativo da plataforma *Google* sobre a pesquisa de termos relacionados à Osama Bin Laden. Note que o índice apresenta valores baixos durante sua totalidade, porém em 2011 há um pico. Isso pois em 2 de maio de 2011 a missão nos arredores de Abbottabad (cidade no Paquistão) ocorreu, onde o governo americano divulgou que Bin Laden teria sido capturado e morto no seu esconderijo. Um outro ponto que seria considerado como resíduo, porém não há dados na plataforma analisada, é 2001, devido ao ocorrido nas Torres Gêmeas. Note assim que o resíduo é um comportamento incomum, e como citado, pode ser causado por eventos relacionados ao terrorismo, desastres naturais, ou os mais variados tipos de situações, positivas ou negativas.

Figura 5 – Índice de interesse relativo de pesquisa sobre Osama Bin Laden no Brasil de 2004 até 2021.



Fonte: Adaptada de Google Trends ([www.google.com/trends](http://www.google.com/trends)).

## 2.2 Análise e Predição em Séries Temporais

Para realizar a análise de uma série temporal, a decomposição descrita nos capítulos anteriores é muito útil. Isto porque permite observar comportamentos que podem ser independentes

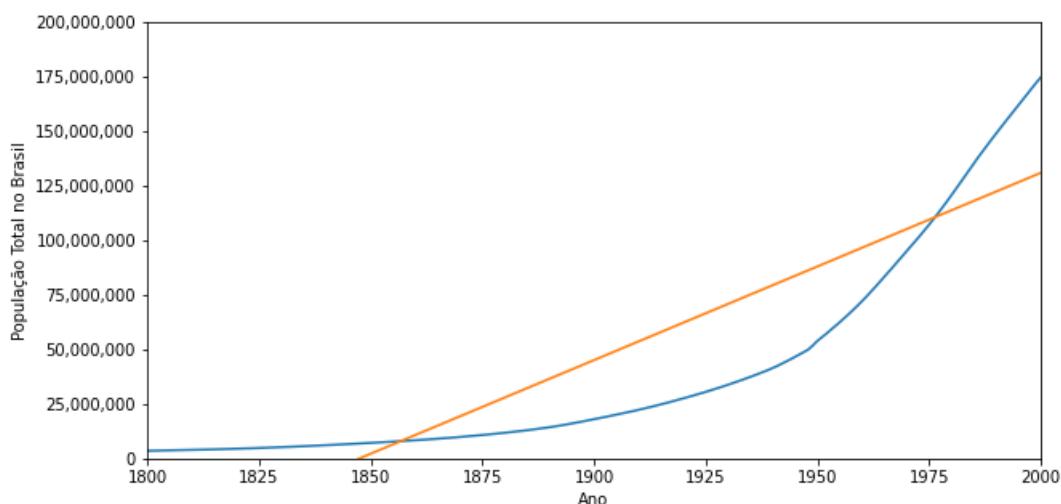
entre si de forma clara. Porém além da análise, séries temporais apresentam grande utilidade devido às técnicas estatísticas de predição.

Dentre as técnicas mais comuns utilizadas para predição de dados, pode ser citada a regressão. Trata-se de um processo matemático que relaciona variáveis dependentes (resposta) com variáveis independentes (explicativas) para encontrar um padrão - ou ainda, justificativa - para a variação da variável resposta dependendo do nível das variáveis explicativas. Sua fórmula está descrita na [Equação 2.1](#) onde  $\alpha$  é uma constante,  $\beta$  representa a inclinação em relação à variável independente,  $X_i$  é a variável independente,  $y_i$  é a variável explicada e  $\varepsilon_i$  representa o resíduo ([PARMEZAN; SOUZA; BATISTA, 2019](#)).

$$y_i = \alpha + \beta X_i + \varepsilon_i \quad (2.1)$$

Aplicando o método de regressão linear ao exemplo abordado na [Figura 3](#) é possível chegar ao resultado da [Figura 6](#), onde o resultado é traduzido de forma linear e aproximado na linha vermelha e a linha azul representa o crescimento populacional no Brasil do ano 1800 até 2000.

Figura 6 – Regressão linear aplicada ao conjunto de dados da população do Brasil.



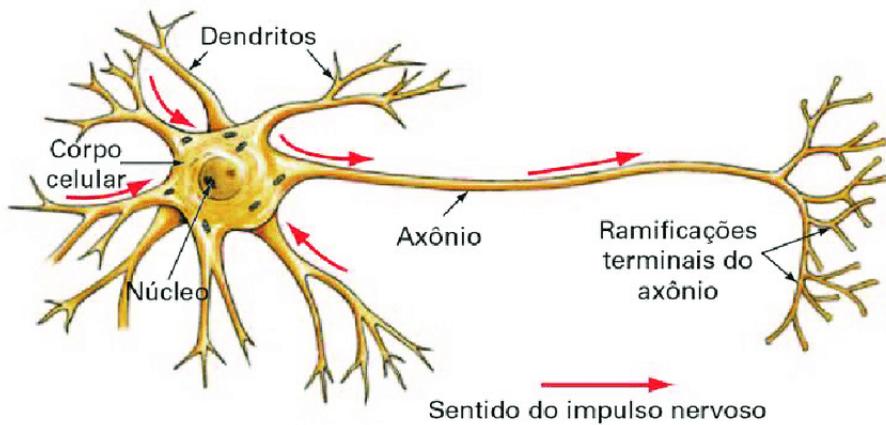
Fonte: Adaptada de [Roser e Ortiz-Ospina \(2013\)](#).

Contudo, a utilização de modelos relativamente mais simples como regressão não são uma boa escolha para a predição de qualquer tipo de dado. Dados com estruturas mais complexas pedem por algoritmos e métodos mais robustos. Com isso, por volta da década de 40, cientistas começaram a tentar replicar as ideias de aprendizado do cérebro humano, e com isso, os primeiros artigos sobre redes neurais artificiais começaram a surgir ([MCCULLOCH; PITTS, 1943](#)). Porém só em 1986, com a criação do algoritmo de *backpropagation*, as ideias ficaram

popularmente conhecidas e foram amplamente estudadas e desenvolvidas (RUMELHART; HINTON; WILLIAMS, 1986).

Uma Rede Neural Artificial (RNA) é uma abstração e simplificação dos conceitos envolvidos em neurônios biológicos. Partindo de sua unidade funcional, o cérebro é composto de células, chamadas de neurônios. Ela é composta por 3 estruturas principais: dendritos, corpo celular e axônios. Uma representação pode ser vista na [Figura 7](#).

Figura 7 – Ilustração da morfologia geral do neurônio e de seus componentes, apresentando o sentido de condução (impulso nervoso) da informação, da recepção (corpo celular e dendritos) a transmissão (ramificações do axônio).



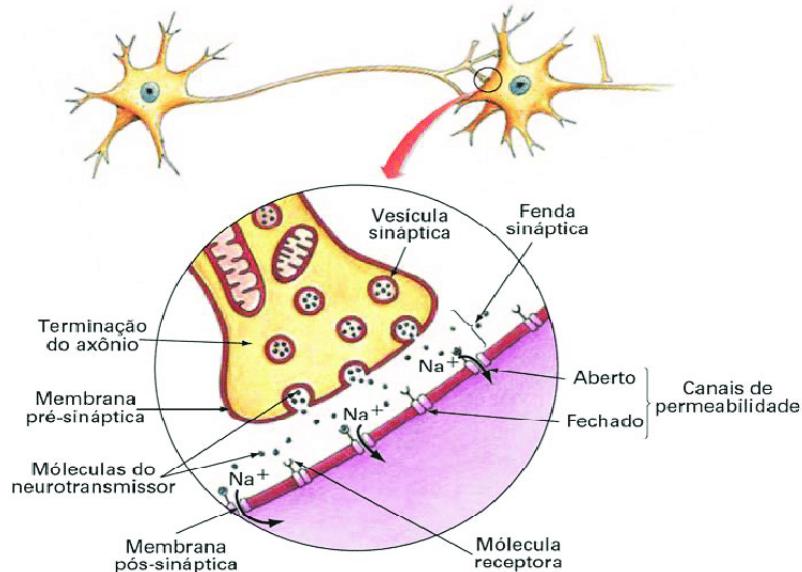
Fonte: Dechichi, Ferreira e Silva (2013).

A comunicação entre os neurônios ocorre por meio das sinapses, passando o impulso nervoso de uma célula para outra como visto na [Figura 8](#). Essa passagem de informações permite a ativação e desativação de regiões do cérebro, permitindo que seja possível realizar diversas ações, como aprender, falar e correr.

Em virtude disso, a RNA surge com o propósito de desenvolver um modelo matemático, baseado nas características do sistema nervoso, que possa aprender e resolver problemas por meio de reconhecimento de padrões. Assim como o cérebro humano, a RNA possui uma unidade funcional, o neurônio artificial. Segundo Haykin (2007), ele pode ser caracterizado por possuir a seguinte estrutura: entradas ( $x_i$ ), pesos ( $w_i$ ), bias ( $b$ ), função soma ( $\Sigma$ ), função de ativação ( $F$ ) e saída ( $y$ ), como pode ser visto na [Figura 9](#). Simulando o sentido do fluxo da informação, as entradas ao entrarem no neurônio são ponderadas com os seus respectivos pesos na função soma formando  $u = \sum x_i w_i$ . Após essa etapa é acrescentada a bias, tendo  $u + b$ . Essa sendo a entrada da função ativação, a qual, por meio de uma função linear ou não-linear calcula a saída  $y = F(u + b)$  do neurônio, sendo normalmente representada em um intervalo unitário fechado  $[0,1]$  ou  $[-1,1]$  (HAYKIN, 2007). Como exemplo de funções de ativação, temos a função degrau unitário e a função logística sigmoidal.

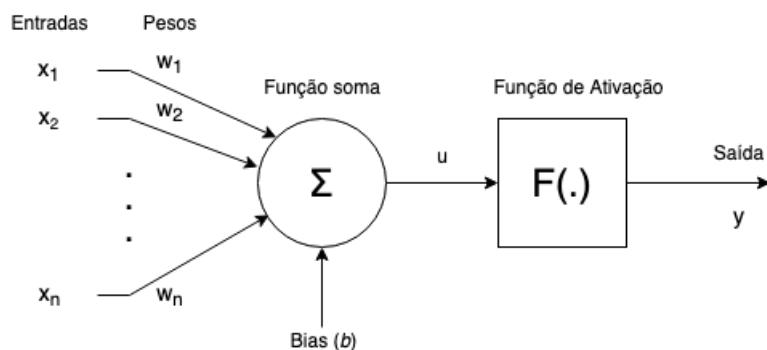
A primeira ideia de RNA, foi desenvolvida por Frank Rosenblatt em 1957, nomeada de perceptron, ela era formada por uma camada contendo um único neurônio e tinha o pretexto

Figura 8 – Ilustração evidenciando um contato entre neurônios (sinapse), no qual ocorre a transmissão da informação por meio de mediadores químicos (neurotransmissores).



Fonte: Dechichi, Ferreira e Silva (2013).

Figura 9 – Modelo de um neurônio artificial



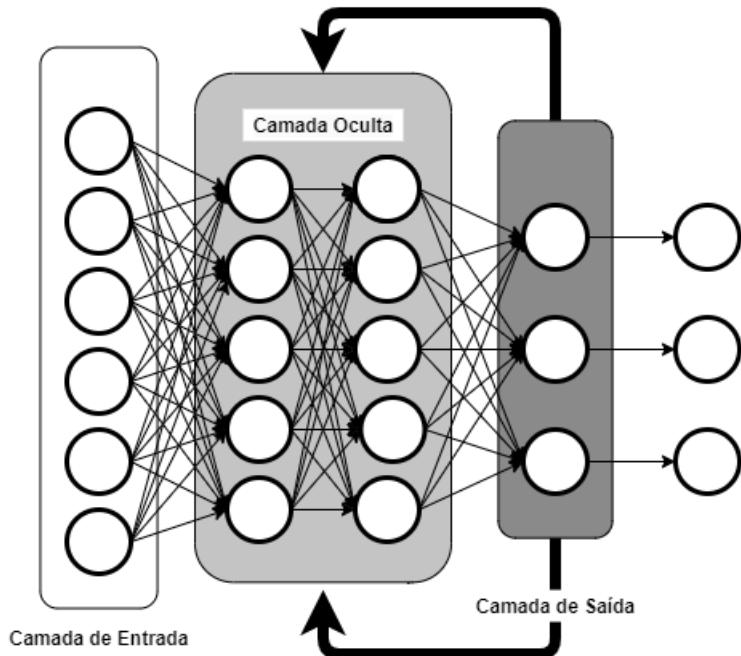
Fonte: Elaborada pelo autor.

de classificador linear. Devido às suas limitações, foram propostas as Rede Neurais *Multilayer Perceptron* (MLP). Para formá-la, os neurônios são ligados de forma a caracterizar a espécie de rede com mais de uma camada, podendo formar dois tipos de arquiteturas: as sem realimentação e as com realimentação.

Serão descritas as arquiteturas com realimentação, pois são utilizadas nesse projeto. A [Figura 10](#) demonstra uma rede neural recorrente (RNR), onde a saída de alguns neurônios podem alimentar neurônios anteriores ou o próprio. A estrutura de camada é dividida entre três subdivisões: camada de entrada, contendo os primeiro conjunto de neurônios, a camada saída, o último conjunto de neurônios responsáveis pelos resultados, e a camada oculta, que compreende todas as camadas entre a de entrada e de saída.

A capacidade de aprendizado do neurônio está na possibilidade de alteração dos pesos,

Figura 10 – Modelo de MLP recorrente



Fonte: Elaborada pelo autor.

os quais são modificados por meio do treinamento. Para realizar o treinamento é necessário um grande conjunto de dados com o propósito de representar as características do problema. Um algoritmo muito utilizado para o treinamento é o *backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986), o qual utiliza do aprendizado supervisionado, de forma a minimizar o erro entre o resultado real e esperado.

Com a evolução da pesquisa de redes neurais e sua utilização, novas técnicas mais robustas foram implementadas, pensando nos casos mais específicos do mundo real. Assim, os métodos que foram utilizados neste trabalho apresentam características em comum com as redes neurais propostas na seção atual, porém são robustos e apresentam o estado da arte em suas respectivas áreas de objetivo.

## 2.3 *Spatio-Temporal Graph Convolutional Networks*

### 2.3.1 *Convolução em grafos*

Grafos estão dentre as estruturas mais utilizadas em aprendizado de máquina, devido ao fato de que muitos problemas do mundo real são apresentados naturalmente por meio de grafos, como redes sociais. Muitos projetos vêm sendo desenvolvidos nessa área. Por exemplo, em Fan *et al.* (2019) os autores utilizam de redes neurais para grafos para predição de recomendações sociais. Li, Cai e He (2017) apresentam outro modelo baseado em grafos para predição de moléculas e análise em *datasets* ligados à componentes tóxicos e doenças. Também vale destacar

estudos que exploram aprendizado de representações para grafos (ZHANG *et al.*, 2018), em que são obtidas *embeddings*, ou vetores de características, para os vértices do grafo considerando a topologia da rede.

No escopo desse projeto, há interesse em redes neurais convolucionais baseadas em grafos (GCNNs), que visam aprender uma função baseado no sinal ou característica (no caso deste trabalho, as séries temporais) em um grafo  $G = (V, E)$  definido por uma entrada tal qual existe um atributo descrito (série)  $x_i$  para cada nó  $v_i \in V$ , formando assim uma matriz  $X$  de atributos com dimensão  $N \times D$ , onde  $N$  representa o número de nós (ou sensores) e  $D$  o número de atributos de entrada (observações das séries temporais). Além disso, outro componente de entrada é uma representação da estrutura de um grafo em forma de matriz, normalmente em forma de uma matriz  $A$  de adjacências (KIPF; WELLING, 2016).

O resultado do modelo descrito acima é um componente  $Z$ , que é uma matriz  $N \times F$ , onde  $F$  representa a dimensão do vetor de características aprendido pela rede neural para os vértices da rede. Dessa forma, cada camada da rede neural pode ser escrita de acordo com uma função não linear formulada na Equação 2.2, com  $H^{(0)} = X$  e  $H^{(L)} = Z$  e  $L$  representando o número de camadas. Com isso, modelos específicos diferem na forma em que essa equação é escolhida e parametrizada.

$$H^{(l+1)} = f(H^{(l)}, A) \quad (2.2)$$

Uma formulação proposta por Kipf e Welling (2016) é apresentada na Equação 2.3, onde  $W^{(l)}$  representa a matriz de pesos para a camada  $l$  da rede neural,  $A$  é a matriz de adjacências e  $\sigma$  é uma função não linear de ativação como uma ReLU - ou retificador.

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}) \quad (2.3)$$

Em geral, os dados presentes em grafos não são estruturados, nem euclidianos, dificultando assim a construção de modelos de aprendizado de máquina. *Graph Neural Networks* (GCNs) servem para generalizar estruturas clássicas de redes neurais convolucionais para o caso de dado estruturado por meio de grafos (CASELEGNO, 2021). Na prática, o objetivo é adicionada uma última camada de neurônios e uma função de ativação que permite ajuste de toda rede via *backpropagation*, como para tarefas de classificação ou regressão. A etapa de convolução da rede neural explora a estrutura do grafo e extrai características de alto nível que é utilizada como entrada para a etapa de classificação ou regressão (KIPF; WELLING, 2016).

Redes neurais convolucionais oferecem uma arquitetura eficiente para a extração de padrões estatísticos significantes de conjuntos de dados robustos. Isso devido a sua habilidade de aprendizado de propriedades locais da entrada dos dados, revelando padrões que são compartilhados pelo domínio dos dados em questão (DEFFERRARD; BRESSON; VANDERGHEYNST, 2017).

### 2.3.2 **Modelo e Predição**

O modelo presente em [Yu, Yin e Zhu \(2018\)](#), avaliado neste projeto, é denominado de STGCN (*Spatio-Temporal Graph Convolutional Networks*). Sua arquitetura é composta por blocos de convolução espaço-temporais. Cada um desses blocos apresenta uma estrutura onde dois blocos de convolução temporais envolvem um terceiro bloco de convolução espacial. Originalmente, o STGCN propõe o uso de redes espaço-temporais para analisar e prever dados de tráfego de rodovias em situações reais.

O uso de redes que dependem de grafos para o problema proposto no artigo é lógico, visto que redes de tráfego podem naturalmente ser modeladas por meio de grafos, na qual os vértices representam sensores que dão dados de movimentação dos veículos. O ponto chave demonstrado por [Yu, Yin e Zhu \(2018\)](#) foi a importância da análise espaço-temporal entre os sensores que em muitos trabalhos relacionados foi desconsiderada. Assim, as convoluções em grafos são utilizadas para extração de padrões significantes no domínio espacial. A utilização de uma rede neural convolucional permite vantagem quando comparado a redes neurais tradicionais, devido aos treinamentos que podem ser realizados mais rapidamente, com estrutura simples e sem dependência de passos anteriores.

Na prática, o STGCN pode ser considerado um *framework* universal para processar um conjunto de séries temporais representadas por sensores, em que há uma matriz de distâncias entre esses sensores.

O STGCN foi avaliado em dois conjuntos de dados, um de Beijing e outro da Califórnia, com dados de fluxo de tráfego junto com informações geográficas sobre sua localização. O intervalo utilizado para a observação dos dados foi de 5 minutos. Interpolação linear foi utilizada para preencher valores faltantes após limpar os dados. Por fim, os dados foram manipulados à partir da normalização pelo método *Z-score*.

O método *Z-score* é baseado em uma variável  $x$  que segue uma distribuição normal. Assim, é feita a relação entre um ponto, ou *score* com sua média e desvio padrão à partir da [Equação 2.4](#). Um ponto negativo desse método é exatamente o fato da influência da média e desvio padrão, assim, indiretamente é influenciável por *outliers*, ou seja, pontos com valores muito diferentes da média - ou ainda, ruído.

$$Z_{Score} = (x - \mu) / \sigma \quad (2.4)$$

O STGCN aprende os pesos da rede neural por meio de um problema de minimização de erro de valores futuros dos sensores de entrada, usando para isso o erro quadrático médio entre o valor predito e o valor real (*RMSE*). O *RMSE* é a raiz do erro quadrático médio, e sua fórmula é descrita em [Equação 2.5](#), onde  $A_t$  corresponde ao valor real,  $F_t$  é o valor predito e  $n$  representa a

quantidade de pontos que foram preditos.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (F_t - A_t)^2} \quad (2.5)$$

Além do *RMSE*, outras duas medidas podem ser utilizadas: *MAE* e *MAPE*. O *MAE* corresponde a média absoluta dos erros, assim, sua fórmula é descrita na [Equação 2.6](#), onde  $A_t$  corresponde ao valor real,  $F_t$  é o valor predito e  $n$  representa a quantidade de pontos que foram preditos.

$$MAE = \frac{1}{n} \sum_{t=1}^n |A_t - F_t| \quad (2.6)$$

Já o *MAPE* corresponde à média absoluta percentual dos erros. Sua fórmula pode ser vista em [Equação 2.7](#) onde  $A_t$  corresponde ao valor real,  $F_t$  é o valor predito e  $n$  representa a quantidade de pontos que foram preditos.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|A_t - F_t|}{A_t} \quad (2.7)$$

Discussões mais detalhadas sobre os métodos no contexto desse projeto para avaliação do *STGCN* são feitas nas seções do [Capítulo 3](#).

Por fim, vale destacar que como resultado obtiveram um treinamento até 14x mais rápido que métodos utilizando redes neurais recorrentes tradicionais. E quando comparado com diversos métodos de previsão apresentou a melhor performance (menor erro) nas métricas analisadas.

O trabalho proposto nesta monografia utiliza do *STGCN* e faz sua comparação com um modelo onde as distâncias entre sensores não são consideradas (métodos tradicionais). Esse modelo será apresentado na próxima seção e utiliza conceitos de previsão de séries temporais de negócios.

## 2.4 Business Time Series Prediction

O modelo Prophet proposto em [Taylor e Letham \(2017\)](#) é utilizado como um modelo de referência para comparação com o *STGCN*, visto que sua entrada de dados é somente a série temporal, não utilizando assim as matrizes de distância entre os sensores.

O Prophet foi proposto em 2017 pelo Facebook e baseia-se em ideias de previsão de séries temporais de negócios, levando em consideração assim que o conjunto de dados pode ser complexo, com componentes de sazonalidade presentes de diferentes maneiras e com a presença de ruído. Também teve seu *design* pensado para apresentar parâmetros intuitivos que podem ser ajustados sem muito conhecimento do modelo abordado. Na prática, incorpora no

modelo informações que podem afetar a análise de eventos, como finais de semana, feriados, entre outros.

O modelo constrói uma função em relação à tendência, sazonalidade, um termo especial para delimitar a presença de feriados e, por fim, ruído. A função que determina o modelo permite a acomodação de sazonalidade com múltiplos períodos. As observações das séries não precisam ser regularmente espaçadas e as interpolações não são necessárias para preencher valores vazios. Dentre suas vantagens, também apresenta uma etapa de treinamento rápida, permitindo a exploração de grandes bases de dados.

## Capítulo 3

# DESENVOLVIMENTO

---



---

### 3.1 Descrição do Problema

A importância dos algoritmos de predição para o mundo moderno é significativa. Nos últimos anos, uma série de modelos temporais, espaciais e causais foram desenvolvidos para lidar com os aspectos complexos de predição para eventos reais da sociedade baseados em estruturas espaço-temporais. Em [Kapoor et al. \(2020\)](#) é proposto um modelo chamado de *Spatio-Temporal Graph Neural Networks* (STGNNs) para entender a propagação e evolução do COVID-19. [Min et al. \(2021\)](#) propõe um modelo chamado Spatial-Temporal Graph Social Network (STGSN) que modela as redes sociais tanto na perspectiva temporal quanto espacial, e apresenta uma estrutura robusta que foi idealizada para análise de redes criminais. Por fim, outro modelo é proposto em [Jain et al. \(2016\)](#), onde é feita uma junção das características de uma Rede Neural Recorrente (RNR) adaptando assim para o uso de grafos, criando uma Rede Neural Recorrente Estruturada (SRNR).

A necessidade dessas estruturas se dá pelo fato da correlação existente entre um evento observado e uma série de parâmetros que os causam. Ou seja, em vários casos reais existem múltiplos sensores de eventos observáveis, e esses sensores influenciam na predição um do outro, necessitando assim de métodos mais robustos que tem a capacidade de lidar com essa correlação.

Um fator que permitiu esse avanço na pesquisa dessas estruturas de predição é a grande quantidade de dados - *Big Data* - de indicadores de código aberto, ou OSI (do inglês, *Open Source Indicators*), como notícias, indicadores econômicos, redes sociais e várias outras fontes. Assim, a predição de eventos sociais que dependem do espaço e do tempo conseguem prover observações e análises interessantes sobre doenças, economia, política, processos migratórios, etc.

Devido à grande presença de dados presentes na internet como um todo, uma série de algoritmos surgiram para utilização de técnicas de *web-crawling*. E com isso realizar o pré-tratamento dos dados através de modelos estatísticos e linguagem natural. Em [Hattori \(2013\)](#), por exemplo, é proposto um modelo que cria sensores *web* para coletar informações de fenômenos físicos a partir de valores numéricos espaço-temporais.

Este projeto utiliza dos sensores de texto presentes em [Marcacini et al. \(2017\)](#), onde é proposto um modelo baseado em *web-crawling* que utiliza os dados pré-processados por meio de

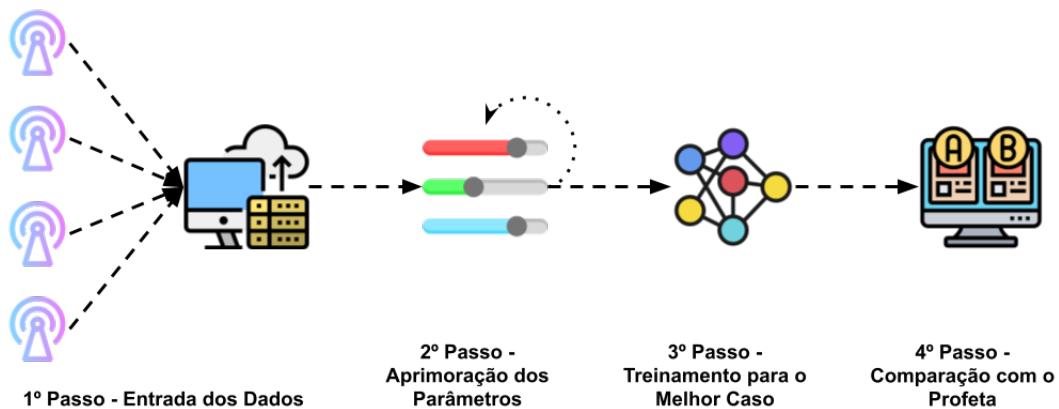
categorias ligadas à informações geográficas, temporais, de nome e organizações para realizar a divisão em temas. Esses temas são por sua vez correlacionados através das características citadas, em que eventos similares podem ser organizados em grupos. E assim é realizada a criação de alertas para os sensores, em que novos eventos similares a um grupo podem ativar um sensor.

A predição espaço-temporal de eventos utiliza de informações heterogêneas, como textos, datas e informação geográfica, o que torna um desafio (NING *et al.*, 2019). Para atender a esse desafio, neste trabalho é avaliado o STGCN (YU; YIN; ZHU, 2018) - e assim, um modelo espaço temporal baseado em grafos - e o compara com o Prophet, algoritmo proposto pela equipe do *Facebook* que não utiliza da correlação entre os eventos da série temporal para realizar a predição. Por fim, discorre sobre os aspectos positivos e negativos dos resultados, além de delimitar se o acréscimo da informação espaço-temporal traz melhorias dos resultados de predição.

## 3.2 Atividades Realizadas

Como método para encontrar uma solução do desafio descrito, uma arquitetura foi elaborada utilizando uma estratégia de divisão do problema em quatro grandes passos, assim como demonstra a Figura 11<sup>1</sup>. A primeira etapa é a entrada dos dados, composta por um conjunto de sensores representados por respectivas séries temporais, bem como uma matriz de adjacências que determina a distância entre esses sensores. Depois é feito o aprimoramento dos parâmetros do STGCN, na qual um conjunto de treinamento, validação e teste é utilizado. Com isso, treina-se novamente para os parâmetros com melhor desempenho. Finalmente, na última etapa é realizada a comparação dos resultados do STGCN com os resultados do Prophet.

Figura 11 – Arquitetura proposta pelo autor.



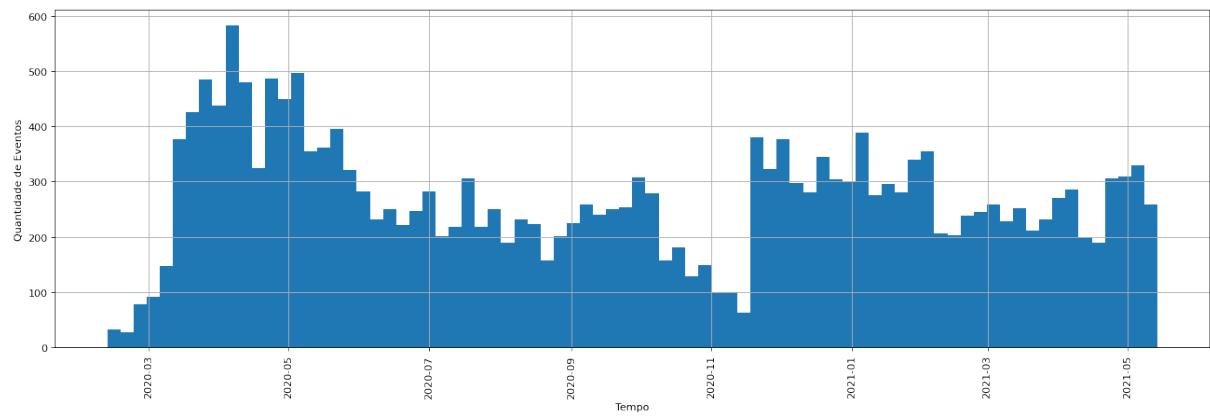
Fonte: Elaborada pelo autor.

No primeiro, os dados de entrada são formados por sensores, suas séries temporais e uma matriz de adjacências que indica a distância entre os sensores. Para esse projeto, foram avaliados

<sup>1</sup> Esta imagem foi criada usando recursos do Flaticon.com

sensores sobre Covid-19 coletados por meio do projeto Websensors, que analisa eventos da base GDEL (Global Database of Events, Language, and Tone)<sup>2,3</sup>. O conjunto de dados é formado por 21526 eventos georreferenciados por país entre Fevereiro de 2020 até Maio de 2021, organizados em 275 sensores, em que cada sensor é um subtema relacionado à Covid-19. Na Figura 12 é apresentada a quantidade de eventos em relação ao tempo.

Figura 12 – Total de eventos por período.



Fonte: Elaborada pelo autor.

Cada um dos 275 sensores representa um grupo de eventos, considerando similaridade de conteúdo e geográfica (país). Na Figura 13 é apresentada uma visão geral da frequência relativa (normalizada) de eventos distribuídos ao longo dos países. Observe que há maior concentração de eventos em poucos países, o que reflete a quantidade de notícias disponíveis e acessíveis pelo GDEL. Foram identificados um total de 16 países com mais de 200 eventos no período.

Para ilustrar, nas Figuras 14, 15 e 16 são apresentados três séries temporais de sensores, respectivamente sobre desenvolvimento de vacinas, lockdown e distanciamento social. Para cada sensor, foi selecionado um evento de cada grupo, considerando a similaridade média entre um evento e os eventos do seu grupo (centróide do grupo).

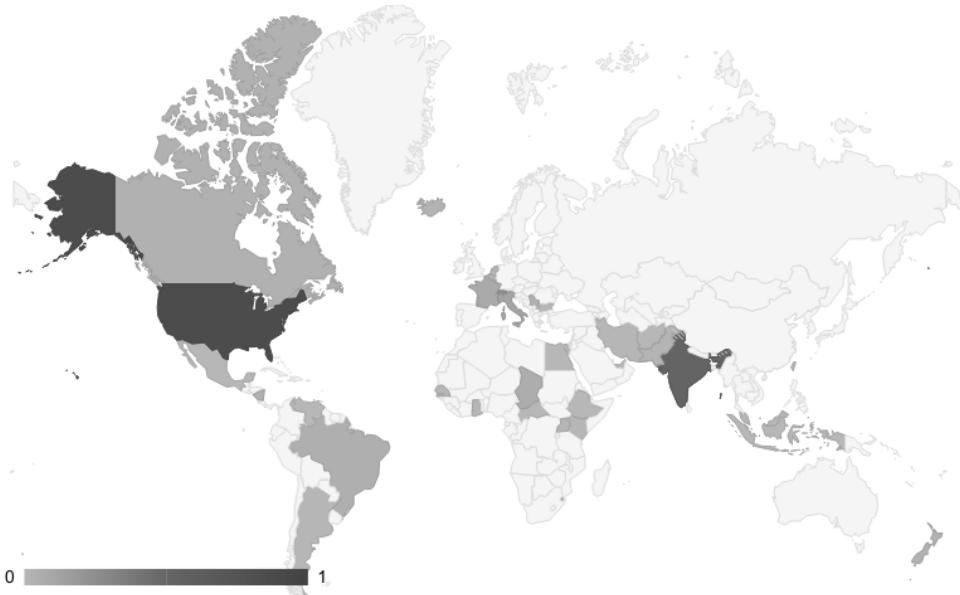
O segundo passo diz respeito aos parâmetros do algoritmo STGCN para predição espaço-temporal de eventos. Uma descrição dos parâmetros pode ser analisada na Tabela 1.

De todos os parâmetros descritos, os mais importantes no sentido de variação para o algoritmo em questão são a taxa de aprendizado, o número de histórico (o tamanho da janela deslizante de observações do passado para treinamento do modelo) e o tamanho do lote. Outros parâmetros não apresentaram espaço para variabilidade, ou apresentaram termos fixos. Um exemplo é o número de vértices, que deve ser igual ao número de vértices do grafo utilizado. Outro exemplo de parâmetro que não impactou profundamente nos resultados é o otimizador, onde a maioria dos modelos utilizados teve resultados semelhantes.

<sup>2</sup> <<https://www.gdelproject.org/>>

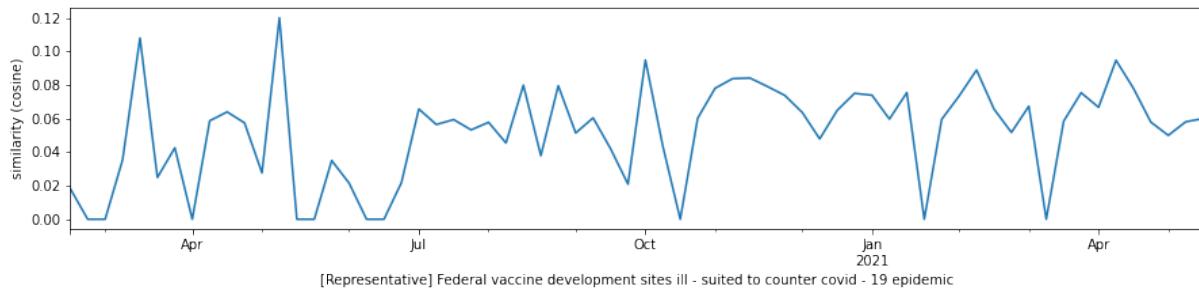
<sup>3</sup> <<https://blog.gdelproject.org/university-of-sao-paulo-covid-19-news-dashboard/>>

Figura 13 – Frequência relativa de eventos por país.



Fonte: Elaborada pelo autor.

Figura 14 – Exemplo de série temporal para sensor sobre eventos de vacinas.

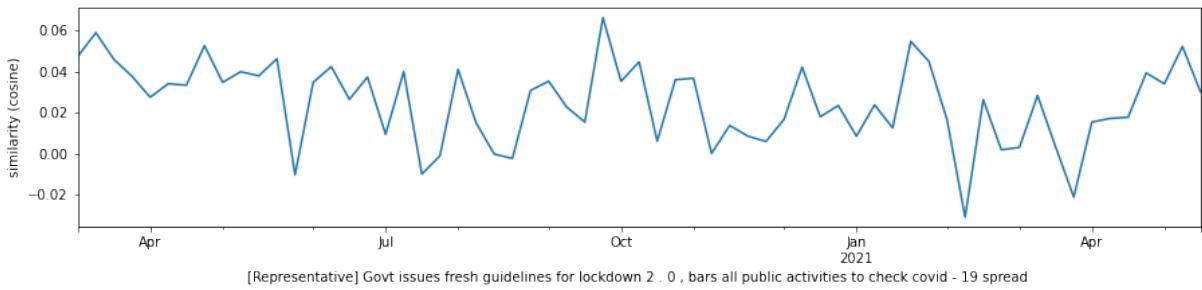


Fonte: Elaborada pelo autor.

O método de execução para aprimorar os parâmetros foi feito a partir do *MAE*, ou seja, sempre que uma execução com novos parâmetros consegue em algum dos passos um valor menor para a métrica analisada, esse novo parâmetro passa a ser o melhor parâmetro. E isso é feito em todas as iterações até que sejam encontrados os melhores parâmetros.

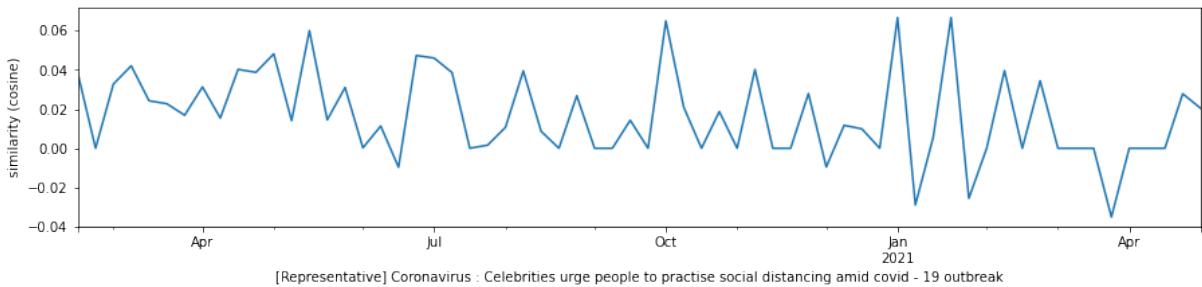
A escolha do indicador chave de precisão foi feita devido principalmente à escala do problema. Os dados preditos apresentam valores muito pequenos (escala de  $10^{-3}$ ). Valores muito próximos de 0 podem causar problemas no *MAPE* devido às suas divisões internas, assim, pode 'explodir' em um valor desproporcional. Algo similar ocorre no *RMSE*, que apresenta maior importância para erros mais significantes, ou seja, um único erro muito grande pode causar problemas na medida da métrica. O *MAE*, assim, dá igual importância para todo tipo de erro apresentado (VANDEPUT, 2021).

Com isso, o *dataset* de sensores que foi citado foi utilizado para prever o penúltimo

Figura 15 – Exemplo de série temporal para sensor sobre eventos de *lockdown*.

Fonte: Elaborada pelo autor.

Figura 16 – Exemplo de série temporal para sensor sobre eventos de distanciamento social.



Fonte: Elaborada pelo autor.

elemento do conjunto (o penúltimo valor real que sabemos), como em uma estratégia de organização por conjunto de treinamento e validação. Isso foi feito para estimar os melhores parâmetros. A estratégia utilizada de estimação de parâmetros seguiu focada na melhora do MAE.

A partir do momento que os melhores parâmetros foram encontrados, é escondido do algoritmo o último elemento para que seja possível realizar a comparação após a previsão, representado o conjunto de testes. Assim, utiliza-se da melhor configuração possível para realizar a execução que será testada contra o Prophet (que segue a mesma divisão de treinamento, validação e teste). O algoritmo é executado com esses parâmetros descritos e seus resultados são salvos em um arquivo *.csv*. As melhores execuções serão demonstradas no próximo capítulo.

O *Prophet* é executado a partir do *dataset* final, como a segunda execução do STGCN, e assim, prevendo o último elementos dos sensores (que é escondido do algoritmo para que ele não tente prever algo sabendo a resposta). O *Prophet* é um método competitivo para comparação por apresentar resultados eficientes para as três métricas apresentadas, além de fácil implementação e não utilização das matrizes de correlação, realizando a previsão a partir da série temporal unicamente.

Tabela 1 – Lista de Parâmetros do STGCN e sua Descrição.

Parâmetro	Descrição
Número de Vértices	Quantidade de vértices presentes no modelo de grafos.
Número de Histórico	Quantidade de eventos que serão considerados para prever o próximo.
Número de Predições	Tamanho da janela de observações utilizado para treinar a rede neural.
Tamanho do Lote	Quantidade de exemplos de treinamento que serão utilizadas em uma iteração.
Épocas	Quantidade de iterações que serão realizadas.
Ordem de Chebyshev	Número de componentes reativos necessários para a montagem do filtro
Kernel de Convolução Temporal	Tamanho do kernel de convolução.
Taxa de Aprendizado	Taxa em que o algoritmo irá aprender.
Otimizador	Nome do otimizador da biblioteca mxnet que será utilizado.

Fonte: Elaborada pelo autor.

### 3.3 Resultados

Nas iterações para encontrar os melhores parâmetros, foi possível delimitar a seguinte resposta demonstrada na [Tabela 2](#).

Tabela 2 – Melhores parâmetros obtidos através da iteração.

Parâmetro	Valor
Passo	6
Número de Vértices	275
Número de Histórico	9
Número de Predições	1
Tamanho do Lote	50
Épocas	50
Ordem de Chebyshev	1
Taxa de Aprendizado	$10^{-3}$
Otimizador	Adam

Fonte: Dados da pesquisa.

Em todas as execuções a taxa de aprendizado menor possível apresentou melhor MAE, isso já era esperado, visto que ela indica o ritmo em que os pesos são atualizados. Uma vantagem de utilizar taxas de aprendizados baixas é que aumenta o tempo para realização das execuções.

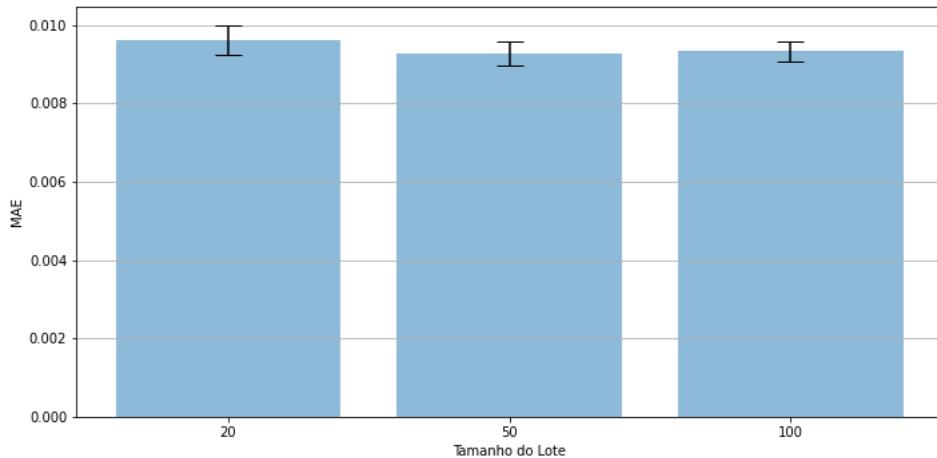
O Número de Histórico, ou seja, a quantidade de elementos que são analisados para realizar a predição do próximo foi a maior possível (9) em todos os melhores casos. Não é possível aumentar esse número devido ao conjunto de teste escolhido e às restrições do algoritmo.

Por fim, o *batch size*, ou ainda, tamanho do lote apresentou o comportamento mais variado, em alguns melhores casos com valor 20, outros com 50 e por fim com 100. No melhor caso descrito, seu valor foi o médio entre os três apresentados.

Para testar a influência direta que os parâmetros que foram iterados para otimização (tamanho do lote, taxa de aprendizado e número de histórico) apresentam no MAE calculado, foram realizados novos testes. Os melhores parâmetros foram fixados, e para cada variável de interesse foi executado dez vezes o STGCN para encontrar média e desvio padrão dos resultados.

Analizando o tamanho do lote, a [Figura 17](#) demonstrou o mesmo que o otimizador, mostrando assim que a variação não é significativa no resultado de acordo com a variação do tamanho do lote.

Figura 17 – Variação do MAE de acordo com o tamanho do lote.

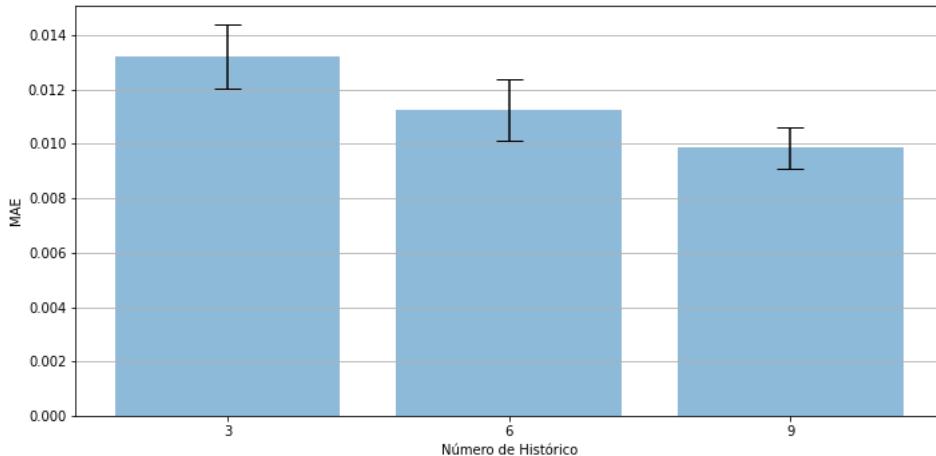


Fonte: Dados da pesquisa.

Contudo, as outras métricas apresentaram grande variação de acordo com a variável de interesse. A [Figura 18](#) demonstra o mesmo que o otimizador, e assim, confirmando que o MAE diminui conforme o número de eventos observados aumenta. Outra confirmação foi feita com a [Figura 19](#), onde é possível analisar a importância de taxas de aprendizado menores quando levado em conta o resultado do MAE.

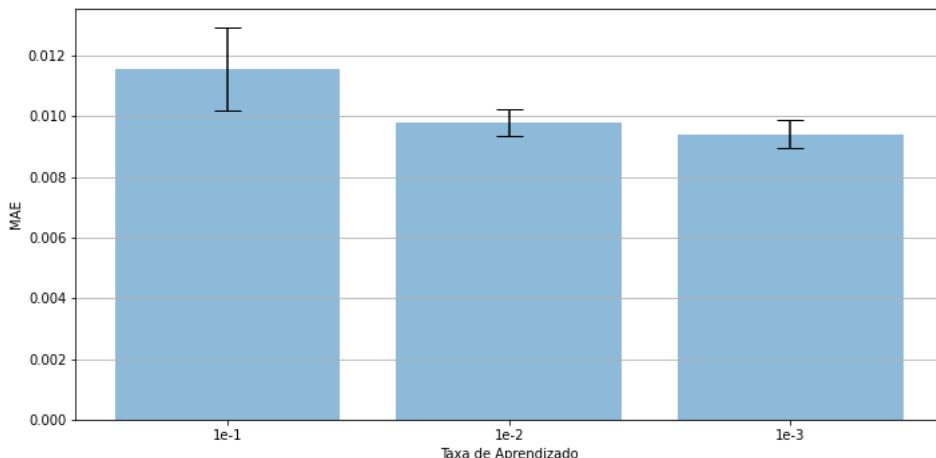
Portanto conclui-se que é necessário para um modelo como o descrito pelas redes convolucionais que sua execução se dê pelos melhores parâmetros, e o não cumprimento dessa

Figura 18 – Variação do MAE de acordo com o número de histórico.



Fonte: Dados da pesquisa.

Figura 19 – Variação do MAE de acordo com a taxa de aprendizado.



Fonte: Dados da pesquisa.

característica implica diretamente em resultados piores para a predição. Isso pois outros modelos completos como o *Prophet* utilizam de métricas semelhantes encapsuladas dentro de seus métodos.

Executando o STGCN para o melhor caso de parâmetros [Tabela 2](#) e o *Prophet* com os mesmos conjuntos de dados e condições semelhantes, chegou-se na resposta obtida na [Tabela 3](#).

Tabela 3 – Resultados em função do MAE e RMSE para os métodos avaliados.

Método	MAE	RMSE
<i>Spatio-Temporal Graph Convolutional Networks</i>	0,008871	0,020007
<i>Prophet</i>	0,017998	0,021134

Fonte: Dados da pesquisa.

Com isso, é possível notar que os resultados obtidos para o STGCN foram - quando comparados com o *Prophet* - duas vezes mais preciso. Isso devido à robustez do modelo que utiliza das matrizes de correlação para análise espaço-temporal. Já o RMSE, como apresenta uma maior interferência para erros significantes, resultou em valores semelhantes para ambos os métodos. O MAPE não foi analisado por apresentar valores muito grandes para ambos os algoritmos, devido a presença de valores muito próximos de zero que dificultam sua medição. Ainda assim, esses foram apenas os resultados obtidos para o melhor caso do STGCN. Sem uma prévia otimização dos parâmetros, em muitos casos o *Prophet* superou facilmente o algoritmo espaço temporal.

## 3.4 Dificuldades e Limitações

Alguns fatores podem ser notados quando se diz respeito às dificuldades e limitações ao realizar este trabalho. Primeiro, a construção do conjunto de sensores, realizado em colaboração com outros membros do laboratório, depende de coleta, monitoramento e pré-processamento de dados textuais dos eventos. É um processo computacionalmente custoso e que está fora do escopo deste projeto. Assim, esse projeto ficou limitado a avaliar e comparar métodos preditivos para análise de eventos já representados por meio de sensores.

Uma segunda dificuldade, um pouco mais técnica, foi a adaptação do STGCN. O método foi proposto inicialmente para análise de sensores em tráfego de rodovias e parte de sua estrutura interna dependia desse domínio. Foram realizadas adaptações e testes para torná-lo um framework mais geral. Ainda, também foram realizadas padronizações a respeito de restrições como a necessidade da igualdade entre os parâmetros de número de eventos históricos e preditos, ou ainda a utilização de métodos de comparação internos, que não apresentavam a mesma métrica utilizada no *Prophet*.

Por fim, houveram dificuldades técnicas em relação à plataforma utilizada para desenvolvimento, o *Google Colab*. Isso pois a versão gratuita é pensada para execuções *on demand* - interrompendo assim a execução caso leve muito tempo, impedindo assim a execução em muitas iterações.



## Capítulo 4

# CONCLUSÃO

---



---

## 4.1 Contribuições

O trabalho apresenta como conclusão a importância em considerar informações geográficas na análise de eventos, em especial, na predição espaço-temporal de eventos. Ainda que métodos baseados apenas nas séries temporais, como o *Prophet*, apresenta resultados promissores, incorporar uma matriz de adjacências com distâncias geográficas entre os sensores permite redução dos erros de predição.

No entanto, vale destacar que os resultados do *Prophet* são positivos, indicando que trata-se de uma ferramenta viável para a realização da predição. Outro fator que é levado em conta para o *Prophet* é sua fácil utilização e baixo custo computacional.

Além da comparação entre os modelos, o projeto finaliza apresentando um repositório de código-fonte aberto para realizar análise de eventos via STGCN e Prophet de forma simplificada ([NATALI, 2021](#)). Esse é um resultado relevante, facilitando que outros colaboradores e pesquisadores do o Laboratório de Inteligência Computacional (LABIC/USP) tenha uma ferramenta importante para a execução de predições em redes espaço-temporais baseadas (STGCN) ou não (*Prophet*) em grafos.

## 4.2 Trabalhos Futuros

Novas contribuições que poderiam ser realizadas são o estudo e criação de um novo modelo convolucional à partir das estruturas pré-existentes para a predição de dados espaço-temporais. Em especial, mecanismos de atenção ([LEE et al., 2019](#)) para *Graph Neural Networks* têm obtidos resultados promissores, uma vez que permitem identificar a relevância de estruturas locais do grafo durante o treinamento. Incorporar mecanismos de atenção no STGCN é uma direção para trabalhos futuros.

Além disso, é importante realizar experimentos com um maior número de bases de dados. Os conjuntos de eventos são volumosos e diferentes temas são relevantes, como economia, saúde, educação, violência pública, entre outros. A partir de um conjunto de dados maior, é possível análises mais robustas dos parâmetros do STGCN, de forma a melhorar mais ainda a performance do algoritmo.



## REFERÊNCIAS

---



---

- BONTEMPI, G.; TAIEB, S. B.; BORGNE, Y.-A. L. Machine learning strategies for time series forecasting. In: SPRINGER. **European business intelligence summer school**. [S.l.], 2012. p. 62–77. Citado na página 19.
- CASALEGNO, F. **Graph Convolutional Networks**. 2021. Disponível em: <<https://towardsdatascience.com/graph-convolutional-networks-deep-99d7fee5706f>>. Citado na página 27.
- CHEN, X.; LI, Q. Event modeling and mining: a long journey toward explainable events. **The VLDB Journal**, Springer, v. 29, n. 1, p. 459–482, 2020. Citado 2 vezes nas páginas 15 e 16.
- DECHICHI, C.; FERREIRA, J.; SILVA, L. **INCLUSÃO EDUCACIONAL E EDUCAÇÃO ESPECIAL: MÚLTIPLOS OLHARES E DIVERSAS CONTRIBUIÇÕES**. [S.l.: s.n.], 2013. ISBN 9788570783561. Citado 2 vezes nas páginas 24 e 25.
- DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. **Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering**. 2017. Citado na página 27.
- FAN, W.; MA, Y.; LI, Q.; HE, Y.; ZHAO, E.; TANG, J.; YIN, D. Graph neural networks for social recommendation. In: **The World Wide Web Conference**. New York, NY, USA: Association for Computing Machinery, 2019. (WWW '19), p. 417–426. ISBN 9781450366748. Disponível em: <<https://doi.org/10.1145/3308558.3313488>>. Citado na página 26.
- HAMILTON, J. D. **Time series analysis**. [S.l.]: Princeton university press, 2020. Citado na página 19.
- HATTORI, S. Spatio-temporal web sensors by social network analysis. In: IEEE. **2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining**. [S.l.], 2012. p. 988–995. Citado na página 15.
- \_\_\_\_\_. Spatio-temporal dependency analysis for temporally-shifted web sensors. In: **2013 Second International Conference on Informatics Applications (ICIA)**. [S.l.: s.n.], 2013. p. 30–35. Citado na página 31.
- HAYKIN, S. **Redes Neurais: Princípios e Prática**. [S.l.: s.n.], 2007. Citado na página 24.
- JAIN, A.; ZAMIR, A. R.; SAVARESE, S.; SAXENA, A. Structural-rnn: Deep learning on spatio-temporal graphs. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. Citado na página 31.
- KAPOOR, A.; BEN, X.; LIU, L.; PEROZZI, B.; BARNES, M.; BLAIS, M.; O'BANION, S. **Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks**. 2020. Citado 2 vezes nas páginas 17 e 31.
- KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. **arXiv preprint arXiv:1609.02907**, 2016. Citado na página 27.

LEE, J. B.; ROSSI, R. A.; KIM, S.; AHMED, N. K.; KOH, E. Attention models in graphs: A survey. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM New York, NY, USA, v. 13, n. 6, p. 1–25, 2019. Citado na página 41.

LI, J.; CAI, D.; HE, X. **Learning Graph-Level Representation for Drug Discovery**. 2017. Citado na página 26.

MARCACINI, R. M.; CARNEVALI, J. C.; DOMINGOS, J. On combining websensors and dtw distance for knn time series forecasting. In: **IEEE. 2016 23rd International Conference on Pattern Recognition (ICPR)**. [S.I.], 2016. p. 2521–2525. Citado 2 vezes nas páginas 15 e 16.

MARCACINI, R. M.; ROSSI, R. G.; NOGUEIRA, B. M.; MARTINS, L. V.; CHERMAN, E. A.; REZENDE, S. O. Websensors analytics: Learning to sense the real world using web news events. In: **Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web**. Porto Alegre, RS, Brasil: SBC, 2017. p. 169–173. ISSN 2596-1683. Disponível em: <[https://sol.sbc.org.br/index.php/webmedia\\_estendido/article/view/4857](https://sol.sbc.org.br/index.php/webmedia_estendido/article/view/4857)>. Citado 2 vezes nas páginas 15 e 31.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, Dec 1943. Citado na página 23.

MIN, S.; GAO, Z.; PENG, J.; WANG, L.; QIN, K.; FANG, B. Stgsn — a spatial–temporal graph neural network framework for time-evolving social networks. **Knowledge-Based Systems**, v. 214, p. 106746, 2021. ISSN 0950-7051. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705121000095>>. Citado 2 vezes nas páginas 17 e 31.

NATALI, P. **Spatio Temporal Forecasting of Events Using Graph Convolutional Neural Networks**. [S.I.]: GitHub, 2021. <<https://github.com/PedroNatali/Spatio-Temporal-Forecasting-of-Events-Using-Graph-Convolutional-Neural-Networks>>. Citado na página 41.

NING, Y.; ZHAO, L.; CHEN, F.; LU, C.-T.; RANGWALA, H. Spatio-temporal event forecasting and precursor identification. In: **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**. [S.I.: s.n.], 2019. p. 3237–3238. Citado 3 vezes nas páginas 16, 17 e 32.

PARMEZAN, A. R. S.; SOUZA, V. M.; BATISTA, G. E. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. **Information sciences**, Elsevier, v. 484, p. 302–337, 2019. Citado 3 vezes nas páginas 16, 19 e 23.

RADINSKY, K.; DAVIDOVICH, S.; MARKOVITCH, S. Learning causality for news events prediction. In: **Proceedings of the 21st international conference on World Wide Web**. [S.I.: s.n.], 2012. p. 909–918. Citado na página 15.

RADINSKY, K.; HORVITZ, E. Mining the web to predict future events. In: **Proceedings of the sixth ACM international conference on Web search and data mining**. [S.I.: s.n.], 2013. p. 255–264. Citado na página 15.

ROSER, H. R. M.; ORTIZ-OSPINAS, E. World population growth. **Our World in Data**, 2013. <Https://ourworldindata.org/world-population-growth>. Citado 2 vezes nas páginas 21 e 23.

- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533–536, Oct 1986. Disponível em: <[https://www.nature.com/articles/323533a0?error=cookies\\_not\\_supported&code=2926f83e-9c3a-46a7-9b28-b3d19d46768a](https://www.nature.com/articles/323533a0?error=cookies_not_supported&code=2926f83e-9c3a-46a7-9b28-b3d19d46768a)>. Citado 2 vezes nas páginas 24 e 26.
- SANTOS, B. N. dos; ROSSI, R. G.; REZENDE, S. O.; MARCACINI, R. M. A two-stage regularization framework for heterogeneous event networks. **Pattern Recognition Letters**, Elsevier, v. 138, p. 490–496, 2020. Citado na página 15.
- SCHLICHTKRULL, M.; KIPF, T. N.; BLOEM, P.; BERG, R. V. D.; TITOV, I.; WELLING, M. Modeling relational data with graph convolutional networks. In: SPRINGER. **European semantic web conference**. [S.l.], 2018. p. 593–607. Citado na página 17.
- TAYLOR, S.; LETHAM, B. Forecasting at scale. **PeerJ Prepr**, v. 5, p. e3190, 2017. Citado 2 vezes nas páginas 17 e 29.
- TUNES, S. **Inteligência artificial contra a Covid-19**. 2020. Revista Pesquisa FAPESP. <<https://revistapesquisa.fapesp.br/inteligencia-artificial-contra-a-covid-19/>>. Acessado em Junho de 2021. Citado 2 vezes nas páginas 15 e 16.
- VANDEPUT, N. **Data Science for Supply Chain Forecasting**. [S.l.: s.n.], 2021. ISBN 978-3-11-067110-0. Citado na página 34.
- YU, B.; YIN, H.; ZHU, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. **Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence**, International Joint Conferences on Artificial Intelligence Organization, Jul 2018. Disponível em: <<http://dx.doi.org/10.24963/ijcai.2018/505>>. Citado 3 vezes nas páginas 17, 28 e 32.
- ZHANG, D.; YIN, J.; ZHU, X.; ZHANG, C. Network representation learning: A survey. **IEEE transactions on Big Data**, IEEE, v. 6, n. 1, p. 3–28, 2018. Citado na página 27.
- ZHAO, L.; SUN, Q.; YE, J.; CHEN, F.; LU, C.-T.; RAMAKRISHNAN, N. Multi-task learning for spatio-temporal event forecasting. In: **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2015. p. 1503–1512. Citado na página 16.