MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2022/07.04.11.00-TDI

# PREDIÇÃO LOCAL DA CINTILAÇÃO IONOSFÉRICA EM BAIXA LATITUDE MAGNÉTICA UTILIZANDO APRENDIZADO DE MÁQUINA

Pedro Alexandre dos Santos

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Stephan Stephany, aprovada em 24 de junho de 2022.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34T/477R4DL>

INPE
São José dos Campos
2022

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2022/07.04.11.00-TDI

# PREDIÇÃO LOCAL DA CINTILAÇÃO IONOSFÉRICA EM BAIXA LATITUDE MAGNÉTICA UTILIZANDO APRENDIZADO DE MÁQUINA

Pedro Alexandre dos Santos

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Stephan Stephany, aprovada em 24 de junho de 2022.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34T/477R4DL>

INPE
São José dos Campos
2022

Dados Internacionais de Catalogação na Publicação (CIP)

Santos, Pedro Alexandre dos.

Sa59p Predição local da cintilação ionosférica em baixa latitude magnética utilizando aprendizado de máquina / Pedro Alexandre dos Santos. – São José dos Campos : INPE, 2022.

xxiv + 174 p. ; (sid.inpe.br/mtc-m21d/2022/07.04.11.00-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2022.

Orientador : Dr. Stephan Stephany.

1. Cintilação ionosférica. 2. Bolhas de plasma. 3. Aprendizado de máquina. 4. Predição de cintilação. 5. Gradient Boosting Tree. I.Título.

CDU 52-658:004.85

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**DEFESA FINAL DE TESE DE PEDRO ALEXANDRE DOS SANTOS**
**BANCA Nº184/2022 REG 141380/2017**

No dia 24 de junho de 2022, as 14h00min, por teleconferência, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora, por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Doutor em Ciência do Sistema Terrestre. O trabalho deve incorporar as correções sugeridas pela Banca Examinadora, a critério e sob revisão final do(s) orientador(es).

**Novo título: "PREDIÇÃO LOCAL DA CINTILAÇÃO IONOSFÉRICA EM BAIXA LATITUDE MAGNÉTICA UTILIZANDO APRENDIZADO DE MÁQUINA"**

Dr. Leonardo Bacelar Lima Santos – Presidente – CEMADEN
Dr. Stephan Stephany – Orientador – INPE
Dr. Eurico Rodrigues de Paula – Membro Interno – INPE
Dr. Marcio Tadeu de Assis Honorato Muella – Membro Externo – UNIVAP
Dr. João Victor Cal Garcia – Membro Externo – CEMADEN

Documento assinado eletronicamente por **João Victor Cal Garcia**, **Tecnologista**, em 28/06/2022, às 10:47 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Leonardo Bacelar Lima Santos**, **Pesquisador**, em 28/06/2022, às 10:48 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Stephan Stephany**, **Pesquisador Titular**, em 28/06/2022, às 14:23 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Eurico Rodrigues de Paula**, **Pesquisador**, em 28/06/2022, às 15:54 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **MARCIO TADEU DE ASSIS HONORATO MUELLA (E)**, **Usuário Externo**, em 30/06/2022, às 18:23 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site http://sei.mctic.gov.br/verifica.html, informando o código verificador **10044746** e o código CRC **9BBB38D3**.

**Referência:** Processo nº 01340.004759/2022-97 SEI nº 10044746

*"Toda grande caminhada começa com um simples passo".*

Buda

## AGRADECIMENTOS

À Deus, o principio, meio e fim.

Ao meu orientador Stephan por todo suporte, apoio e paciência.

Aos membros da banca pelo trabalho dispendido na leitura deste texto.

À minha família como um todo, com atenção especial para os meus pais José Domingues e Maria Aparecida por todo o amor que me dedicam, e também às minhas irmãs Jane Aparecida, Giovana Aparecida e Silvana Aparecida, e ao meu irmão Paulo Henrique.

Ao meu amigo Gabriel Augusto Lins Leal Pinheiro pela amizade, pelo companheirismo e pela ajuda nas horas de provas, listas de exercícios, trabalhos, revisões de artigos.

À minha querida amiga Maria Aparecida de Camargo por me ouvir, por me ajudar, pela paciência, pela sabedoria e por todo o carinho.

Ao apoio do Laboratório Nacional de Computação Científica (MCTIC/LNCC)[1] pelo uso do supercomputador Santos Dumont, como parte do projeto "Implementação de abordagens de aprendizado de máquina que demandam supercomputação para a predição de cintilação e previsão meteorológica de eventos convectivos severos".

À Divisão de Heliofísica, Ciências Planetárias e Aeronomia (DIHPA/INPE) pelo apoio e disponibilização dos dados utilizados neste trabalho.

Ao Dr. Gerald Jean Francis Banon pelo desenvolvimento e disponibilização deste estilo de editoração para publicação na biblioteca do INPE.

Ao CNPq pela bolsa de doutorado 167950/2017-7.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código financeiro 001.

À CAP/INPE e ao próprio INPE que permitiram o desenvolvimento deste trabalho.

E aos contribuintes brasileiros, que de forma direta contribuem para a manutenção das instituições públicas de ensino e pesquisa e das organizações de fomento.

[1] https://sdumont.lncc.br/

## RESUMO

A ionosfera é composta basicamente por uma camada de gás em estado de plasma cujo processo de ionização tem a radiação solar como seu principal agente. A distribuição do plasma ionosférico não é uniforme no espaço e no tempo, sendo a transição entre dia e noite um elemento importante na geração de irregularidades. Essa transição causa a ressurgência da Anomalia de Ionização Equatorial, a qual acoplada ao mecanismo de instabilidade do plasma ocasiona a formação de depleções, isto é, regiões com baixa densidade de íons e elétrons. Essas estruturas são conhecidas como bolhas ionosféricas, e são geradas no equador magnético após o pôr do sol para em seguida ascender a altitudes maiores e migrar para baixas latitudes ao longo do campo magnético da Terra. Os sinais de radiofrequência de sistemas de telecomunicações ou de navegação global por satélites sofrem flutuações em fase e amplitude devido às irregularidades presentes em bolhas ionosféricas. Essas perturbações de sinal são chamadas de cintilação ionosférica, medida pelo desvio padrão normalizado da intensidade do sinal, no caso da amplitude. Este trabalho aborda o uso de técnicas de aprendizado de máquina para predição local de cintilação ionosférica em baixa latitude magnética, especificamente em São José dos Campos. Utilizaram-se dados históricos de cintilação ionosférica e outros dados de atividade solar, geomagnéticos e ionosféricos para o período 2011-2018, o qual abrange o último ciclo solar. Os algoritmos testados para predição incluem dois algoritmos de *Gradient Boosting Tree* e uma rede neural convolucional, todos disponíveis no ambiente de programação Python. O desempenho de predição foi avaliado por métricas padrão de predição/classificação, sendo obtidos resultados promissores em todas as abordagens, mas que foram limitados pela qualidade dos dados disponíveis para esse período.



Palavras-chave: Cintilação Ionosférica. Bolhas de Plasma. Aprendizado de Máquina. Predição de Cintilação. Gradient Boosting Tree. Redes Neurais Convolucionais.

# LOCAL PREDICTION OF IONOSPHERIC SCINTILLATION IN A LOW MAGNETIC LATITUDE USING MACHINE LEARNING


## ABSTRACT

The ionosphere is a layer of gas in the state of plasma that was mainly ionized by the effect of solar radiation. Ionospheric plasma distribution is not uniform in space and time, being the generation of plasma irregularities triggered by the day-to-night transition. Such transition generates the resurgence of the Equatorial Ionization Anomaly, which coupled with the plasma instability mechanism cause depletions, i.e. regions with low density of ions and electrons. Such structures are known as ionospheric bubbles and are generated at the magnetic equator just after sunset. They ascent to higher altitudes and migrate to low latitudes along the Earth's magnetic field. Radiofrequency signals of telecommunication or global navigation satellite systems may experience phase and amplitude fluctuations due to irregularities present in the ionospheric bubbles. These perturbations are known as ionospheric scintillation, which is measured by the normalized standard deviation of the signal intensity, in the case of amplitude. This work addresses the use of machine learning techniques for the local prediction of ionospheric scintillation at low latitudes, specifically over São José dos Campos, Brazil. It employed historical data of ionospheric scintillation and other ionospheric, geomagnetic and solar-activity data for the period 2011-2018 that covers the last Solar Cycle. The proposed prediction algorithms include two Gradient Boosting Tree algorithms, and a convolutional neural network, all available in the Python programming environment. The prediction performances were evaluated by standard prediction/classification metrics, showing promising results for all techniques, which were limited by the quality of the data available for the considered period.



Keywords: Ionospheric Scintillation. Plasma Bubbles. Machine Learning. Scintillation Prediction. Gradient Boosting Tree. Convolutional Neural Network.

## LISTA DE FIGURAS

**Pág.**

## LISTA DE TABELAS

**Pág.**

## LISTA DE ABREVIATURAS E SIGLAS

| | | |
|---|---|---|
| GPS | – | Sistema de Posicionamento Global |
| OACI | – | Organização da Aviação Civil Internacional |
| GNSS | – | Sistema de Navegação Global por Satélite |
| EIA | – | Anomalia da Ionização Equatorial |
| TEC | – | *Total Eletronic Content* |
| VTEC | – | *Vertial Eletronic Content* |
| KDD | – | Knowledge Discovery in Databases |
| CART | – | Classification and Regression Trees |
| CNN | – | Convolution Neural Networks |
| MCTIC | – | Ministério da Ciência, Tecnologia e Inovações |
| LNCC | – | Laboratório Nacional de Computação Científica |
| DIHPA | – | Divisão de Heliofísica, Ciências Planetárias e Aeronomia |
| INPE | – | Instituto Nacional de Pesquisas Espaciais |
| CNPq | – | Conselho Nacional de Desenvolvimento Científico e Tecnológico |
| CAPES | – | Coordenação de Aperfeiçoamento de Pessoal de Nível Superior |
| CAP | – | Pós-graduação em Computação Aplicada |
| GNSS | – | Global Navigation Satellite System |
| EMBRACE | – | Estudo e Monitoramento Brasileiro do Clima Espacial |
| XGBoost | – | Extreme Gradient Boosting Tree |
| CatBoost | – | Categorical Boosting |
| UV | – | Ultravioleta |
| MHD | – | Magneto-hidrodinâmica |
| AE | – | Auroral Electrojet |
| VP | – | Verdadeiro Positivo |
| FP | – | Falso Positivo |
| VN | – | Verdadeiro Negativo |
| FN | – | Falso Negativo |
| ACC | – | Acurácia |
| ECB | – | Entropia Cruzada Binária |
| MSE | – | Erro quadrático médio |
| MAE | – | Erro absoluto máximo |
| GTSH | – | Gap Time Series Holdout |
| TSCV-GKF | – | Time Series Cross Validation Gap K-Fold |
| TSCV-GWF | – | Time Series Cross Validation Gap Walk Forward |
| GBT | – | Gradient Boosting Tree |
| ANN | – | Rede neural artificial |
| MLP | – | Multi Layer Perceptron |
| CNN | – | Convolutional Neural Network |
| OC | – | Ocorrência |
| N-OC | – | Não Ocorrência |
| SJC | – | São José dos Campos |

| | | |
|---|---|---|
| SMOTE | – | Synthetic Minority Over-sampling Technique |
| CIGALA | – | Concept for Ionospheric Scintillation Mitigation for Professional GNSS in Latin America |
| CALIBRA | – | Countering GNSS high Accuracy applications Limitations due to Ionospheric disturbances in Brazil |
| ICEA | – | Instituto de Controle de Espaço Aéreo |
| LISN | – | Low-Latitude Scintilation Network |
| RBMC | – | Rede Brasileira de Monitoramento Contínuo dos Sistemas GNSS |
| NSA | – | Sem seleção de atributos |
| SSA | – | Com seleção de atributos |
| PRED | – | Predito |
| OBSV | – | Observado |

## SUMÁRIO

**Pág.**

# 1 INTRODUÇÃO

A necessidade de localização espacial levou a humanidade ao desenvolvimento de diversas ferramentas, tais como os sistemas de coordenadas, a bússola, os mapas e, mais recentemente, o Sistema de Posicionamento Global (GPS). Este sistema, desenvolvido pelos norte-americanos, tornou-se completamente operacional em 1995, com um custo estimado de 10 bilhões de dólares. Consiste de uma constelação de no mínimo 24 satélites, cada um circulando a Terra duas vezes por dia, em uma configuração em que ao menos 4 satélites são visíveis de qualquer ponto da Terra. O receptor do sinal utiliza a informação enviada pelo satélite para calcular sua distância a cada um destes, utilizando a informação entre o instante de recebimento e o instante de transmissão.

Em 1991, a Organização da Aviação Civil Internacional (OACI) utilizou pela primeira vez o termo sistema de navegação por satélite (GNSS - Global Navigation Satellite System), para denominar todo e qualquer sistema semelhante ao GPS, que atualmente é usado para denominar o sistema americano, também conhecido como Navstar GPS. Outro sistema que se encontra completamente operacional é o russo GLONASS. O sistema chinês COMPASS e o europeu GALILEO se encontram em fase de implementação. OS sistemas de GNSS estão sujeitos a perturbações e interferências em seus sinais, sendo que a mais relevante é a cintilação ionosférica, que também afeta sinais de telecomunicações emitidos por satélites.

A cintilação ionosférica caracteriza-se por uma rápida variação de amplitude e fase dos sinais de radiofrequência emitidos por satélites quando atravessam irregularidades na ionosfera. A cintilação é usualmente medida em amplitude pelo índice $S_4$, definido como o desvio padrão normalizado da intensidade do sinal num intervalo de um minuto, com taxa de amostragem de 50 Hz.

A cintilação é causada por irregularidades ionosféricas eventualmente presentes nas chamadas bolhas ionosféricas, regiões da ionosfera com baixa densidade de elétrons. Essas bolhas migram na ionosfera, geralmente no sentido do Sul e do Leste magnético, podendo se expandir ou contrair. Surgem na região do equador magnético e sua ocorrência, no setor brasileiro, começa a se intensificar de outubro a novembro, apresentando picos ao longo do verão, reduzindo-se a partir de março de cada ano. Sua formação começa entre 19-20 LT (22-23 UT), encerrando-se geralmente entre 01-02 LT (04-05 UT). Além da atividade mais intensa durante o verão é possível observar uma grande variabilidade diária no decorrer do ano, o que torna difícil sua predição, (TAKAHASHI et al., 2016).

A ionosfera apresenta uma dinâmica complexa, influenciada pelo campo magnético da Terra, o qual é influenciado principalmente pelo Sol, que é a principal fonte de radiação ionizante e que rege o clima espacial e os campos magnético e elétrico no espaço. Um fenômeno de particular interesse é a Anomalia da Ionização Equatorial (EIA) que consiste na formação de uma região de alta densidade de elétrons entre 15 e 20 graus de latitude magnética. A EIA ocorre durante o dia e volta a se intensificar após o entardecer e, acoplada ao mecanismo de instabilidade do plasma, gera as bolhas ionosféricas.

Atualmente, não existe um modelo matemático baseado em primeiros princípios, que expresse a física do surgimento e a evolução das bolhas de forma completa, devido a restrições de resolução espacial e temporal. Existem alguns modelos matemáticos (que podem ou não ser baseados em dados históricos), (BÉNIGUEL; HAMEL, 2011; RETTERER, 2010; WERNIK et al., 2007), que fazem a predição do fenômeno de cintilação, porém apresentam erros relativamente grandes. Todavia, os impactos decorrentes desse fenômeno, como falhas de sistemas de navegação, para uma sociedade que apresenta cada vez mais demanda por este tipo de informação, seja em sistemas de produção como na agricultura (STAFFORD, 2000), ou na aviação, podem ser críticas, implicando, por exemplo, em perdas de vidas humanas, ou na redução na produção de alimentos. No caso da aviação, há uma tendência mundial de se utilizar unicamente navegação e procedimentos de pouso/decolagem baseados em GNSS's.

Assim, faz-se desejável uma abordagem que permita a predição da cintilação ionosférica na amplitude do sinal GNSS, representada pelo índice $S_4$. Na ausência de um modelo matemático que possa simular um fenômeno tão complexo, um enfoque possível é o uso de um modelo orientado a dados baseado em aprendizado de máquina. Apesar da evolução do aprendizado de máquina e da capacidade de processamento observados desde a década de 90, poucos trabalhos aplicam-se ao Clima Espacial devido a limitações diversas (CAMPOREALE, 2019). Observa-se que ainda menos trabalhos abordam a predição da cintilação, e raríssimos abordam essa predição em baixas latitudes magnéticas, como no caso do presente trabalho.

O primeiro trabalho de predição de cintilação em baixas latitudes magnéticas utilizando aprendizado de máquina surgiu em Rezende (2009), Rezende et al. (2010), seguido de outros dois trabalhos, Lima et al. (2014) e Lima et al. (2015). Destes dois últimos, o primeiro tinha como objetivo correlacionar a cintilação de São José dos Campos com aquela de São Luiz, enquanto o segundo prever a cintilação em São

Luiz, utilizando dados anteriores na mesma localidade. Enquanto que São José dos Campos está próximo da crista sul da EIA, São Luiz está próximo do Equador magnético. Há ainda um trabalho McGranaghan et al. (2018) que se aplica à região do polo norte, que foi o primeiro sobre predição de cintilação em altas latitudes, mas baseado no modelo de persistência, fazendo a hipótese de que as irregularidades causadoras de cintilação possam durar algumas horas.

Uma busca na base de dados *ScienceDirect* pelo termo *Ionospheric Scintillation and S4 and (predict or forecast)* retornou 68 resultados, enquanto na *ACM Digital Library*, a mesma pesquisa retornou 30 resultados. Os trabalhos encontrados se concentram principalmente em três linhas: a predição de cintilação por meio de modelagem numérica; estudo de variáveis associados ao fenômeno de cintilação utilizando alguma técnica, por exemplo, analise multiespectral, de forma a obter dados/informações que possam eventualmente serem utilizadas em modelos de predição; ou trabalhos sobre a predição da cintilação ionosférica. Destes, os mais recentes são Meziane et al. (2021), Atabati et al. (2021) e Zhao et al. (2021), detalhados a seguir.

O trabalho de Meziane et al. (2021) realiza a predição do índice de cintilação em amplitude $S_4$ e também do índice de cintilação em fase $\sigma_\Phi$ por meio de uma modelagem bayesiana. Para a variável a ser predita, considerada variando no intervalo [0,0 - 0,5], foi discretizada em 20 intervalos constantes de forma a atribuir uma probabilidade a cada um, dada pelo seu número de eventos dividido pelo número total de eventos. Os atributos de predição empregados foram a velocidade do plasma solar, a pressão do plasma solar, a componente $y$ e $z$ do campo magnético interplanetário, medidas da variação do campo magnético na zona auroral por meio do índice *SuperMAG Auroral Electrojet (SME)*, e o ângulo de elevação do Sol. O modelo foi inicialmente desenvolvido com dados de cintilação de uma estação GNSS localizada na região polar, em Eureca (Canadá), coordenadas magnéticas (86,8N,-14,6E), e posteriormente estendido com dados de outra estação na mesma região, em Pond-Inlet (Canadá), coordenadas magnéticas (80,0N, 2,6E). Os dados coletados para o trabalho são do período 2008-2018.

O trabalho de Atabati et al. (2021) realiza a predição do índice $S_4$ por meio de uma rede neural, cujos pesos são inicializados por um algoritmo genético, sendo o treinamento da rede neural feito pelo método de Levenberg-Marquardt. As variáveis utilizadas foram: altura do pico da camada F2, velocidade de deriva da camada F2, número de manchas solares, índice geomagnético $Kp$ e fluxo solar $F10.7$. Os dados

de cintilação utilizados na modelagem foram coletados em Guam, com coordenadas geográficas (13,6N, 144,9E) no período 2015-2020.

O trabalho Zhao et al. (2021) é o mas similar a esta tese, pois foi desenvolvido com dados do território brasileiro para um período semelhante. É um modelo orientado a dados gerado pelo algoritmo de *Gradient Boosting*. Emprega dados de $S_4$ para o período 2012-2020 e também a altura virtual da camada ionosférica $h'F$ medida por radares específicos chamados de ionossondas. As predições são locais a cada estação GNSS e dadas por um único valor categórico para cada noite, considerando duas classes, ocorrência ou ausência de cintilação conforme o limiar adotado de $S_4 = 0,5$. Dados de treinamento são derivados de médias de $S_4$ feitas a cada 5 minutos, mas apenas o valor máximo dessas médias é considerado para cada noite.

Nesta tese, propõem-se novas abordagens de predição de ocorrência de cintilação por modelos orientados a dados com aprendizado de máquina para uma única localidade de baixa latitude magnética. Espera-se assim, futuramente, escolher a melhor dessas abordagens, a qual poderá ter seu desempenho de predição otimizado o suficiente para que seja utilizada operacionalmente no programa de Clima Espacial do INPE (EMBRACE), para predições com antecedência horária ou maiores.

Este trabalho utilizou dados históricos para a cidade de São José dos Campos (SP) relativos à cintilação ionosférica, além de outros dados de atividade solar, geomagnéticos e ionosféricos para o período 2011-2018, o qual abrange quase todo o último Ciclo Solar. Os algoritmos testados para predição incluem dois algoritmos do tipo *Gradient Boosting Tree*, *Extreme Gradient Boosting* (XGBoost) e *Categorical Boosting* (CatBoost), e uma rede neural convolucional (Resnet-Wavenet), todos disponíveis no ambiente de programação Python. O desempenho de predição foi avaliado por métricas padrão de predição/classificação, sendo obtidos resultados promissores em todas as abordagens. Esse desempenho de predição foi limitado pelos dados disponíveis para esse período devido às interrupções de monitoramento causadas por problemas técnicos, à não cobertura completa de todo o período correspondente ao Ciclo Solar e ao desbalanceamento entre o número de amostras, uma vez que há um número relativamente pequeno de instantes de tempo com cintilação comparativamente àqueles sem cintilação.

As contribuições desta tese são constituídas pelas abordagens de predição propostas e implementadas, baseadas em aprendizado de máquina. Essas abordagens são definidas pela análise e seleção de atributos de informação (dados de entrada), pelo esquema de particionamento/seleção dos dados, pelo algoritmo de predição utili-

zado/proposto e pela predição de cintilação objetivada. Cada abordagem proposta foi testada em experimentos específicos relativos à abordagem selecionada e ao período de dados utilizado. Os algoritmos de predição utilizados foram algoritmos de *ensemble* baseados em árvores e uma rede neural específica proposta no contexto deste trabalho na Seção 4.5.6. Então, considera-se que cada abordagem aqui testada contribui, em termos de aprendizado de máquina, para as predições pretendidas.

Em cada abordagem, o algoritmo de aprendizado de máquina foi otimizado num processo iterativo pelo ajuste dos hiperparâmetros. Adicionalmente, a exploração dos diversos esquemas de particionamento/validação possibilitou a geração de modelos com melhor desempenho de predição, considerando o conjunto de dados disponíveis. Assim, o objetivo foi a obtenção de abordagens mais robustas para a predição de cintilação proposta. O desempenho de predição obtido nos vários experimentos aqui apresentados reflete as dificuldades e o caráter atual da pesquisa.

Esta tese é composta pelos demais seguintes capítulos:

- **Capítulo 2: Ionosfera**, apresenta a estrutura da ionosfera em termos de camadas, assim como uma ideia geral dos mecanismos de dínamo presentes nestas camadas, que geram a anomalia de ionização equatorial. Também são definidas as variáveis utilizadas neste trabalho, e o fenômeno de cintilação ionosférica;

- **Capítulo 3: Aprendizado de Máquina**, são tratados os principais fundamentos sobre mineração de dados e o processo de descoberta de dados em base de dados (KDD). Trata-se também de algoritmos de aprendizado de máquina em geral;

- **Capítulo 4: Algoritmos Utilizados**, descrevendo os algoritmos utilizados, que são do tipo *Gradient Boosting Tree* (XGBoost e CatBoost), e a rede neural convolucional (Resnet-Wavenet);

- **Capítulo 5: Descrição dos Experimentos Realizados**, apresenta a metodologia de pré-processamento, particionamento e validação dos dados, a qual foi utilizada para criar os conjuntos de dados de treinamento, validação e teste. São descritos os 3 experimentos realizados (e suas variações), bem como as correspondentes metodologias de teste;

- **Capítulo 6: Resultados dos Experimentos Realizados** apresenta os

resultados dos testes de predição e sua avaliação pelas métricas de desempenho escolhidas;

- **Capítulo 7: Conclusões** e trabalhos futuros previstos;
- **Apêndice A: Resultados Extras dos Experimentos**.

## 2 IONOSFERA TERRESTRE

A ionosfera é uma região ionizada da alta atmosfera, estendendo-se de 60 até 1000 km de altitude, assim, englobando partes da mesosfera, termosfera, e exosfera. Esta camada constitui-se de: gás neutro, íons e elétrons livres, estes dois últimos criados primariamente por processo de fotoionização.

A fotoionização ionosférica consiste de um processo físico-químico, onde algumas espécies químicas presentes na atmosfera ganham ou perdem elétrons decorrentes da absorção de radiação solar predominantemente nas faixas ultravioleta, extremo ultravioleta e raios-X (RISHBETH; GARRIOTT, 1969; NEGRETI, 2012).

A ionização pode também ocorrer devido a colisões com partículas altamente energéticas provindas do meio solar ou então de origem galáctica, o que é mais facilmente observado em altas latitudes em fenômenos como auroral boreal.

A composição da ionosfera, assim como a densidade dos gases, varia em função da altitude. A densidade de elétrons livres também varia, pois conforme a radiação penetra na atmosfera mais densa, a produção de elétrons aumenta até atingir um valor de pico em uma dada altitude. Abaixo desta, mesmo havendo um aumento na densidade da atmosfera neutra, a produção de elétrons decresce, pois a maior parte da radiação ionizante foi absorvida ao longo do percurso, e a taxa de recombinação predomina sobre a taxa de produção de elétrons.

Devido às diferenças marcantes em termos de processos físicos e químicos que governam seu comportamento, a ionosfera pode ser dividida em camadas, onde cada uma apresenta um processo predominante. Finalmente, devido à drástica variação de radiação absorvida devido à transição entre noite e dia, há camadas que aparecem num dado período (dia ou noite).

A ionosfera pode ser dividida nas seguintes camadas:

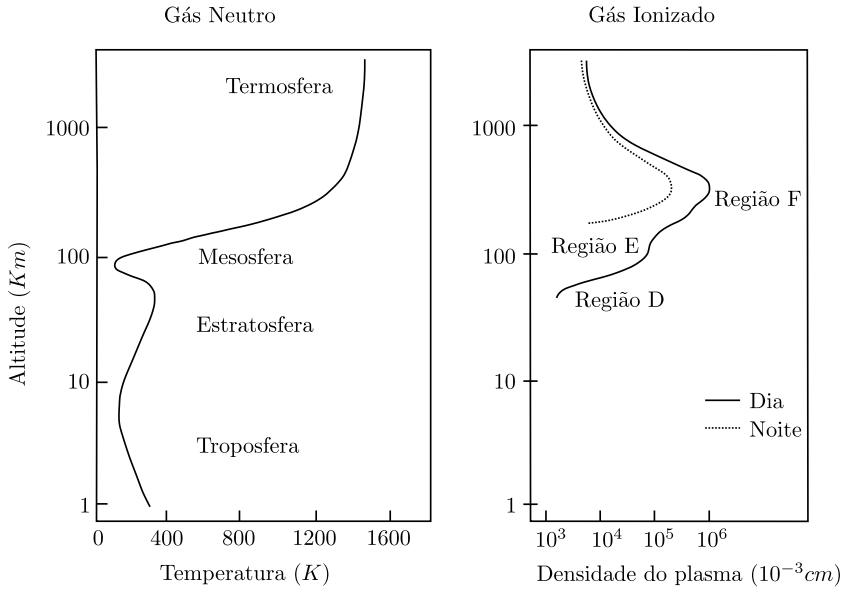
- **Camada D**: camada mais interna, situada entre 60 e 90 km acima da superfície da Terra. Apresenta moléculas ionizadas de $NO$, $N_2$ e $O_2$, assim como a maior taxa de recombinação dentre as camadas da ionosfera. A taxa de absorção para ondas de rádio de baixas e médias frequências é relativamente alta, principalmente, devido à absorção de energia pelos elétrons livres, a qual aumenta suas chances de colisão. Este efeito desaparece durante a noite, devido à menor ionização. Em altas latitudes pode haver alta densidade de elétrons livres e íons em decorrência de erupções solares

com grandes quantidades de matéria, com predominância de prótons, com uma duração de 24 a 48 horas.

- **Camada E**: camada intermediária, estando situada entre 90 e 150 km acima da superfície da Terra. A ionização decorre principalmente do espalhamento de radiação de raio-X leve e ultravioleta distante (UV) provindos do Sol nas moléculas de oxigênio. A estrutura vertical é determinada em sua maior parte pelo balanço entre efeitos de ionização e de recombinação. Esta camada é importante por apresentar fluxo de correntes elétricas, as quais interagem com o campo magnético (KIRCHHOFF, 1991). À noite, dada a ausência de sua fonte primária de fotoionização, esta camada quase desaparece.

- **Camada F**: estende-se de 150 a mais de 500 km acima da superfície da Terra e apresenta a maior concentração de elétrons. Assim, sinais de radiofrequência que chegam a esta camada podem escapar para o espaço. Há predominância de ionização de átomos de oxigênio pela radiação solar no extremo ultravioleta do espectro. Esta camada é subdivida em duas regiões, F2 que está presente durante o dia e à noite, e F1 que aparece somente durante o dia.

Figura 2.1 - Perfil de densidade ionosférica.

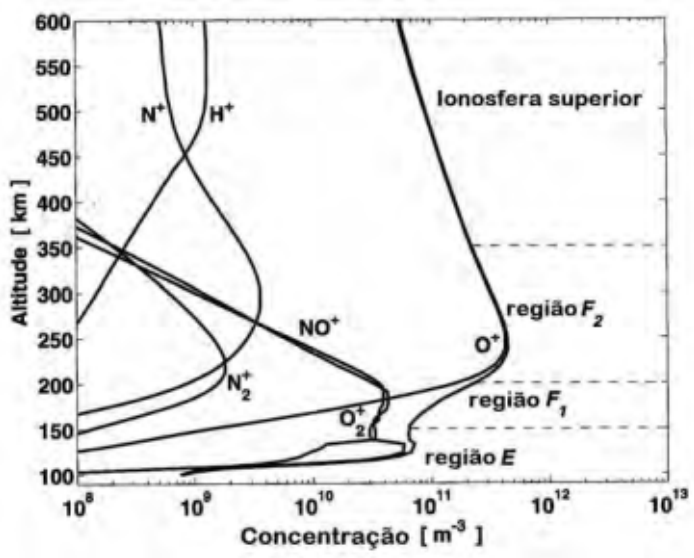



A divisão da atmosfera em função da temperatura, introduzindo a Troposfera, Estratosfera, Mesosfera e a Termosfera será discutido na Seção 2.1.

Fonte: Adaptado de Kelley (2003).

A subcamada F2 engloba toda a região superior da ionosfera e contém o pico da densidade de elétrons. Este pico máximo no perfil vertical de ionização decorre do balanço entre os processos de transporte de plasma e os processos físico-químicos. Acima deste pico, a ionosfera se encontra em equilíbrio difusivo, ou seja, o plasma se distribui em função da própria escala de altitude, sendo que o campo magnético contribui para a distribuição da ionização. A Figura 2.1 apresenta as camadas E, F1 e F2 assim como sua altitude relativamente a outras camadas da atmosfera.

Figura 2.2 - Perfis verticais padronizados dos principais constituintes ionosféricos existentes acima de 90 km durante o dia (*sunlit*): $O_2^+$, $N_2^+$, $O^+$, $H^+$ e $N^+$.




Fonte: Adaptado de Kamide e Chian (2007).

## 2.1 Camadas da atmosfera

Além da divisão baseada nas suas características físicas e químicas combinadas, a atmosfera pode ser dividida em função de um único atributo, no caso a temperatura, resultando nas seguintes camadas:

- **Troposfera**: camada mais próxima da superfície da Terra, estendendo-se até aproximadamente 15 km de altitude no equador e 8 km nos polos. Contém aproximadamente 75% da massa atmosférica e 99% do seu vapor de água. A temperatura decresce com a altitude. A energia térmica é absorvida diretamente da radiação solar nos comprimentos de onda visível e infravermelho ($\lambda > 3000$ Å), e se propaga primariamente por processos de

convecção e radiação;

- **Estratosfera**: situada aproximadamente entre 15 km e 50 km de altitude, caracterizando-se pelo movimento horizontal do ar. A temperatura aumenta com a altitude devido à absorção direta da radiação solar ultravioleta pelo ozônio ($O_3$) e pela água $H_2O$, gerando um pico de temperatura próximo de 50 km de altitude. Esta camada é relativamente estável, pois não apresenta convecção significativa;

- **Mesosfera**: encontra-se aproximadamente entre 50 km e 90 km de altitude, sendo a radiação o principal mecanismo de propagação de energia térmica. A temperatura decai com a altitude, devido à diminuição da absorção de radiação solar no ar rarefeito. Além disso, mesmo a radiação absorvida pelas moléculas de $CO_2$ causadoras do efeito estufa acaba sendo emitida de volta ao espaço devido à rarefação;

- **Termosfera**: inicia-se a aproximadamente a 90 km de altitude e não apresenta um limite superior bem definido, estimado entre 500 e 600 km de altitude, característica de órbitas de vários satélites, da estação espacial internacional e dos antigos ônibus espaciais. Em latitudes elevadas, esta camada apresenta fenômenos tais como a aurora boreal.

  Devido à sua extrema rarefação, sua temperatura não pode ser medida por instrumentos convencionais, sendo calculada por uma variável termodinâmica, a média da energia cinética das moléculas, derivada de medições feitas por satélites. Sua temperatura aumenta com a altitude atingindo valores superiores a 2000 graus Celsius devido à grande taxa de absorção de radiação solar pelo nitrogênio e o oxigênio no extremo-ultravioleta e no ultravioleta.

  A transmissão de energia térmica por convecção não é significativa, predominando a condução, ineficiente dada a rarefação. Assim como nas demais camadas da atmosfera, efeitos de gravidade e diferenças de pressão levam à formação de correntes análogas a marés oceânicas. A ionosfera está majoritariamente situada na termosfera, à qual sua dinâmica está diretamente acoplada.

  Similarmente às demais camadas da atmosfera, sua extensão vertical varia conforme a latitude.

Além disso, a atmosfera também pode ser dividida segundo a dinâmica física de

seus constituintes nas regiões descritas abaixo. Entre essas regiões há ainda regiões de transição, Turbopausa (entre a Homosfera e a Heterosfera) e Exobase (entre a Heterosfera e a Exosfera).

- **Homosfera**: estende-se até aproximadamente 100 km de altitude, predominando a difusão turbulenta. A proporção de gases constituintes é aproximadamente constante em função da altitude;
- **Heterosfera**: estende-se aproximadamente de 100 km até 500 km de altitude, predominando a difusão molecular, o que leva cada gás constituinte a se distribuir segundo sua massa. Consequentemente, a mistura de gases deixa de ser homogênea;
- **Exosfera**: inicia-se aproximadamente em 500 km de altitude sendo que as colisões entre as moléculas e/ou átomos são raras. As partículas neutras se deslocam em órbitas balísticas sob a ação da gravidade.

Finalmente, a atmosfera pode ainda ser dividida segundo a presença de elementos ionizados, tendo-se obviamente a ionosfera e, acima desta, a magnetosfera, que é uma região delimitada pelo campo geomagnético.

A magnetosfera representa um envoltório da Terra, na qual a eletrodinâmica dos processos do plasma é regida principalmente pelo acoplamento com o campo magnético. Por ser uma região mais externa, esta camada tem interação direta com os ventos solares, resultando numa distribuição espacial irregular, pois a parte voltada diretamente ao Sol sofre um processo de compressão, enquanto que a parte oposta sofre alargamento.

## 2.2 Dinâmica da atmosfera

A atmosfera é dinâmica, isto é, seus constituintes estão em constante movimento. Uma das principais dinâmicas são os ventos, ou transporte de massas de ar, decorrente de variações na pressão causadas por diferenças de temperatura.

Considere por exemplo duas regiões de volume fixo com o mesmo gás e temperaturas $T_a$ e $T_b$, tal que $T_a > T_b$. Conforme a Equação de Clapeyron, tem-se $P_a > P_b$. Se estas regiões forem conectadas, devido à diferença de pressão, o gás na região A tenderá a expandir em direção à região B até que se alcance um equilíbrio.

Analogamente, na atmosfera, ocorrerá um fluxo ou corrente de ar de A para B.

Entretanto, deve-se considerar que num volume tridimensional, a ação da gravidade causa um aumento de pressão na base da região B e uma diminuição de pressão na base de A, levando à formação de um fluxo oposto de B para A. Essas estruturas resultantes são denominada célula de circulação, regendo os ventos, que são basicamente movimentos horizontais do ar decorrentes de gradientes de pressão horizontal.

Os fluxos verticais completam a circulação horizontal, garantindo a continuidade de massa. Sem os ventos, verticalmente, a atmosfera se encontra em equilíbrio hidrostático.

Os ventos atmosféricos em altas altitudes têm origem semelhante, e obedecem às mesmas condições de continuidade, sendo os principais descritos abaixo:

- **Ventos termosféricos**: aparecem entre aproximadamente entre 100 km e 200 km de altitude (camada E da ionosfera e base da camada F, base da Termosfera), sendo decorrentes dos gradientes de pressão horizontais causados pelo aquecimento e ionização da termosfera no período diurno. Circulam na ionosfera e interagem com as partículas ionizadas, levando ao arrasto iônico, no qual partículas ionizadas são deslocadas ao colidir com partículas neutras transportadas pelo vento.

  O arrasto é proporcional à diferença entre a velocidade do vento e das partículas carregadas e um de seus principais efeitos é transportar as partículas ionizadas para regiões de maior altitude à noite;

- **Ondas planetárias**: são perturbações com períodos da ordem de dias abrangendo grandes extensões do planeta, sendo importantes abaixo da mesosfera, que atua como um limite superior;

- **Marés atmosféricas**: são oscilações com períodos e sub-períodos do dia solar e lunar, isto é, $T = 24h$, $T = 12h$, $T = 8h$, ..., $T = 24h/N$, pois decorrem de forças periódicas induzidas pelo Sol, com mais intensidade, e pela Lua. Estão associadas à radiação solar absorvida, manifestando-se como oscilações de pressão locais;

- **Ondas de gravidade**: são oscilações de períodos da ordem de minutos a horas, que decorrem da existência de gradientes de pressão e da gravidade. Propagam-se em direção à ionosfera, amplificando-se com a altura devido à conservação de energia.

## 2.3 Acoplamentos no sistema magnetosfera-ionosfera

Num material condutor, apenas os portadores de carga negativa podem se movimentar e o campo eletrostático deve ser nulo, exceto ser houver forçantes externas. Algo semelhante ocorre no comportamento do plasma sob ação de campos elétricos.

O plasma, que pode ser considerado uma nuvem de elétrons, possui também portadores de cargas positivas e elementos neutros que podem se mover. O plasma em equilíbrio deve apresentar campo elétrico médio nulo, podendo seu campo elétrico ser aproximado por $\mathbf{E_p} = 0$, para o referencial das partículas, o que consiste na chamada aproximação magneto-hidrodinâmica (MHD) (ROEDERER, 1979).

A teoria da relatividade restrita na aproximação de baixas velocidades descreve como os campos elétrico e magnético se transformam pela mudança de referencial, num processo análogo à composição de velocidades da física newtoniana.

No referencial das partículas, a velocidade do plasma é nula, pois a partícula se move juntamente com o referencial.

Considere um referencial fixo em relação à Terra e os seguintes campos vetoriais definidos neste referencial: $\mathbf{v_p}$ um vetor de velocidade associada ao plasma, $\mathbf{B}$ um vetor de campo magnético e $\mathbf{E}$ um vetor de campo elétrico. São então válidas as seguintes equações:

$$\mathbf{E_p} = \mathbf{E} + \mathbf{v_p} \times \mathbf{B}. \tag{2.1}$$

Assumindo-se a aproximação magneto-hidrodinâmica, que implica em $\mathbf{E_p} = 0$:

$$\mathbf{E} = -\mathbf{v_p} \times \mathbf{B}. \tag{2.2}$$

A segunda Equação (2.2) descreve o campo elétrico $\mathbf{E}$ perpendicular a $\mathbf{v_p}$ e $\mathbf{B}$ gerado por partículas em movimento nesse campo magnético. Define-se a velocidade de deriva do plasma $\mathbf{v_d}$ como sendo a componente de velocidade perpendicular ao campo magnético, definida então por:

$$\mathbf{v_d} = \frac{\mathbf{E} \times \mathbf{B}}{|\mathbf{B}|^2}. \tag{2.3}$$

As Equações (2.2) e (2.3) expressam que, no plasma, o campo elétrico e a velocidade estão conectados e um pode gerar o outro. Em Aeronomia, estas equações são a pedra angular para a teoria do dínamo ionosférico.

Variações no campo geomagnético já eram conhecidas há mais de 300 anos, sendo a primeira hipótese a presença de correntes elétricas fluindo na alta atmosfera. Uma vez que a existência de tais correntes pressupõe a necessidade de portadores de carga levou à concepção do processo de ionização e à definição da ionosfera.

Somente por volta de 1940 com a introdução da teoria do dínamo por Chapman e Bartels, foi elucidado o processo de formação das correntes na ionosfera, permitindo pela primeira vez estimar suas intensidades. Assume-se neste modelo que ventos de marés produzem o movimento necessário das partículas carregadas.

Em configurações fracamente excitadas, os ventos podem ser divididos em duas componentes, com origem em variações solares e em variações lunares. Como resultado do movimento de maré e da viscosidade do plasma, os elétrons e os íons são arrastados através das linhas de campo magnético gerando campos elétricos, fenômeno conhecido como efeito dínamo.

Os campos elétricos do dínamo da região E têm sua origem nos ventos de marés associados à radiação solar ultravioleta absorvida na camada de ozônio e no vapor de água na atmosfera, e aos efeitos gravitacionais gerados pela Lua (marés). Por outro lado, os campos elétricos do dínamo da região F tem sua origem nos ventos térmicos de marés que decorrem da absorção de radiação ultravioleta no extremo do espectro pela termosfera, (ABDU, 2005).

### 2.3.1 Dínamo da região E

O dínamo da região E age principalmente no lado da Terra iluminado pelo Sol. O mecanismo de geração de correntes pode ser descrito de maneira simplista como se segue.

Seja $\mathbf{v}_m$ um campo vetorial de velocidade representativo do vento de maré, e $\mathbf{B}$ o campo geomagnético. Então, o vento de maré desloca as partículas do plasma fazendo os elétrons se movimentarem com velocidade menor que os íons, o que causa uma separação entre íons e elétrons ocasionando um campo elétrico, $\mathbf{E_m}$, expresso por:

$$\mathbf{E_m} = \mathbf{v}_m \times \mathbf{B}, \tag{2.4}$$

e sua corrente elétrica associada

$$\mathbf{J_m} = \sigma \mathbf{E_m} = \sigma \mathbf{v}_m \times \mathbf{B}, \tag{2.5}$$

onde $\sigma$ é a condutividade.

O processo de separação de cargas sob influência do campo magnético leva a uma não homogeneidade na distribuição de cargas polarizando assim a ionosfera.

O campo elétrico resultante da polarização pode ser modelado em termos de seu potencial elétrico $\phi$ sendo expresso então por $\nabla\phi$ e constituindo um termo extra na expressão do campo elétrico total:

$$\mathbf{E}_t = \mathbf{v}_m \times \mathbf{B} - \nabla\phi. \tag{2.6}$$

O sistema de correntes resultante na ionosfera é então dado por

$$\mathbf{J_t} = \tilde{\sigma}[\mathbf{v}_m \times \mathbf{B} - \nabla\phi], \tag{2.7}$$

onde $\tilde{\sigma}$ é o tensor de condutividade.

### 2.3.2 Dínamo da região F

Na região F, o vento meridional (para sul) provoca um movimento das partículas carregadas ao longo das linhas de campo magnético. Existe também um movimento secundário gerado pelo vento zonal (para leste) com direção perpendicular tanto ao campo magnético quanto à direção do próprio vento (BATISTA, 1986). A velocidade resultante das partículas carregadas $\mathbf{v}$ é expressa por:

$$\mathbf{v} = \left[\frac{\nu\omega}{\nu^2 + \omega^2}\right] \frac{\mathbf{v_n} \times \mathbf{B}}{|\mathbf{B}|}, \tag{2.8}$$

onde $\mathbf{v_n}$ é a velocidade do vento neutro, $\mathbf{B}$ é o campo magnético, $\nu$ é a frequência de colisão entre partículas carregadas e partículas neutras e $\omega = qB/m$ é a giro-frequência das partículas, sendo $q$ a carga e $m$ a massa.

A dependência em relação a carga elétrica provoca movimentos opostos dos portadores de cargas positivas e negativas, causando sua separação e a consequente formação de um campo elétrico de polarização.

Durante o dia, a região E pode ser modelada como um condutor que acoplado à região F leva à dispersão das cargas e à restauração do equilíbrio. Entretanto, durante a noite, isto não ocorre, pois a região E deixa de um condutor ideal, sendo possível a formação do campo elétrico de polarização.

Durante o dia, o campo elétrico zonal, associados aos dínamos da camada E e da camada F, provocam uma deriva vertical do plasma para cima, dada por $\mathbf{E}\times\mathbf{B}/|B|^2$.

As correntes do dínamo da região F são menos intensas que aquelas da região E, porém são de grande importância logo após o por do Sol, quando surgem campos elétricos de polarização na região F provocando uma intensificação da deriva vertical do plasma para cima, a qual é denominada pico de pré-reversão.

## 2.4 Anomalia de Ionização Equatorial

As irregularidades de interesse para este trabalho são as bolhas de plasmas, as quais resultam da anomalia de ionização equatorial (EIA) que ocorre aproximadamente entre 15 e 20 graus de latitude magnética em ambos hemisférios na camada F2.

A EIA consiste na formação de uma região de alta densidade eletrônica, sendo caracterizada como anomalia pois a densidade de plasma deveria ser maior em regiões equatoriais, e não em latitudes magnéticas maiores.

É causada pela deriva vertical do plasma da camada F na região equatorial: os ventos termosféricos na camada F fazem surgir uma corrente elétrica apontando para leste, assim como um campo elétrico na mesma direção, enquanto o campo magnético aponta para o norte (o dipolo magnético que modela o campo geomagnético se encontra invertido, isto é, o norte magnético se situa no sul geográfico e vice-versa).

A força resultante $\mathbf{E} \times \mathbf{B}$ terá então sentido ascendente, deslocando o plasma para regiões de altitudes mais elevadas. Entretanto, ao ascender, o plasma sofre o efeito gravitacional e o efeito da diferença de pressão, os quais causam um movimento descendente que segue predominantemente as linhas de campo magnético. Assim, há o aumento na densidade de plasma em regiões mais afastadas do equador que caracteriza a EIA.

## 2.5 Bolhas de plasma ionosféricas

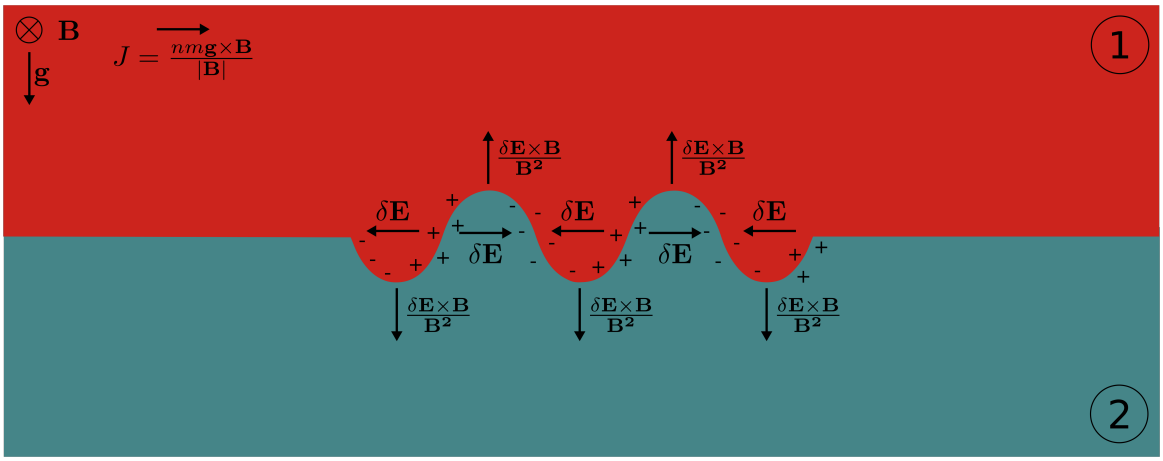
As bolhas ionosféricas ou bolhas de plasma podem ser definidas como regiões de baixa densidade de plasma ionosférico quando comparadas com a sua vizinhança. Sucintamente, surgem na região do equador magnético no contexto do pico de pré-reversão, ou seja, da intensificação da deriva vertical do plasma.

Acompanhando a dinâmica da EIA, essas bolhas sobem altitudes mais elevadas (centenas de quilômetros), e depois migram ao longo das linhas de campo magnético

por milhares de quilômetros na direções norte-sul (no hemisfério sul), alcançado cerca de 20 graus de latitude magnética.

O mecanismo de geração das bolhas é modelado pela Instabilidade de Rayleigh-Taylor, ilustrada na Figura 2.3 para o caso do plasma ionosférico.

Figura 2.3 - Ilustração da Instabilidade de Rayleigh-Taylor.




O fluído mais (menos) denso é representado pela cor vermelha (azul). As pequenas pertubações que aparecem na interface entre os dois fluidos somadas à corrente $J$ presente no meio levam ao acúmulo de cargas, causando assim um campo elétrico de polarização $\delta\mathbf{E}$. Este combinado com o campo magnético, gera uma força de deriva $\delta\mathbf{E} \times \mathbf{B}/B^2$ que amplifica a pertubação.

Fonte: Adaptado de Batista (1986).

Considere dois fluidos 1 e 2, com densidades de plasma $N_1$ e $N_2$, respectivamente, tal que $N_1 > N_2$, e ambos submetidos à gravidade $\mathbf{g}$ e ao campo magnético terrestre $\mathbf{B}$. Considere entretanto que o primeiro (mais denso) esteja acima do segundo e que estejam separados por superfície plana horizontal.

A ação dos campos $\mathbf{g}$ e $\mathbf{B}$ gera uma força perpendicular a ambos os fluidos dada por $m_i\mathbf{g} \times \mathbf{B}/q_i\mathbf{B^2}$, onde $m_i$ e $q_i$ representam a massa e a carga de uma partícula. Essa força causa o movimento das partículas carregadas, gerando a corrente $\mathbf{J}$.

Considere agora que haja uma perturbação na superfície de separação. Tal perturbação provoca um acúmulo de cargas e, consequentemente, um campo elétrico de polarização local. Este campo, na presença de campo magnético faz surgir uma nova força de deriva deslocando porções fluido 1 para baixo e do fluido 2 para cima, amplificando a perturbação.

Acoplados à deriva vertical do plasma devido ao campo elétrico de polarização na região F, essas pequenas perturbações se amplificam e ascendem, formando as bolhas de plasma.

A Instabilidade de Rayleigh-Taylor irá continuamente gerar estruturas, cada vez de menor escala. Algumas se desenvolvem em estruturas maiores que são as bolhas, enquanto outras são estruturas menores contidas nas bolhas ou nos seus contornos. A cintilação ionosférica caracteriza-se por uma variação rápida de amplitude e fase em sinais de radiofrequência quando estes atravessam irregularidades no plasma ionosférico, tais como bolhas de plasma, o que é de particular interesse para este trabalho.

## 2.6 Variáveis e índices para o estudo da cintilação ionosférica

Existem várias grandezas físicas que podem ser necessárias para um estudo completo da dinâmica ionosférica como, por exemplo, medidas do fluxo de radiação solar, do campo magnético da Terra, ou da composição da atmosfera. Nesta seção, apresentam-se os índices e variáveis utilizadas diretamente/indiretamente neste trabalho.

- **Índice AE** ("Auroral Electrojet"): foi proposto por Davis e Sugiura em 1966 para medir a atividade magnética da zona auroral, a qual é produzida pelo aumento nas correntes elétricas ionosféricas que ali fluem. Mede a taxa de variação temporal da componente horizontal do campo magnético medido.

- **Índice Dst**: mede a atividade geomagnética, sendo geralmente utilizado para quantificar a intensidade de uma tempestade magnética. Apresenta resolução temporal de uma hora, e sua unidade de medida é o nanotesla (nT), sendo dada pelo valor médio da componente horizontal do campo magnético em cada hora.

  Valores inferiores a -30 nT são considerados como sendo de atividade magnética perturbada. Na classificação de tempestades geomagnéticas, adotam-se os seguintes intervalos desse índice (GONZALEZ et al., 1994): entre -30 nT e -50 nT tempestade fraca, entre -50 nT -100 nT tempestade moderada, entre -100 nT e -250 nT tempestade muito intensa, e abaixo de -250 nT supertempestade.

- **Índice Sym-H/Sym-D**: correspondem às componentes H e D do campo

geomagnético. Expressam a perturbações magnética em médias latitudes, sendo calculados tomando por referência a variação diária do campo magnético em dia de atividade magnética calma.

- **Índice Kp** avalia as perturbações nas componentes horizontais do campo geomagnético global. É a combinação de medidas realizadas continuamente por magnetômetros distribuídos ao redor do globo, para intervalos de 3 horas. Cada estação é calibrada segundo sua localização e fornece uma quantidade denominada de índice K correspondente à atividade geomagnética medida no local.

  O índice K é uma medida quase-logarítmica com resolução temporal de três horas tomando por referência a atividade magnética em dias calmos. Cada estação mede o desvio máximo da componente horizontal do campo magnético.

  O índice Kp é gerado por uma combinação do conjunto de índices K e seu valor varia de 0 a 9 em intervalos discretos. 0 indica baixa atividade, enquanto 9 indica tempestades extremas. Valores de 0 a 4 ficam associados a períodos calmos, enquanto valores acima de 5 indicam tempestades magnéticas.

- **Índice ap** é semelhante ao índice Kp, correspondendo a uma transformação desse índice para uma escala linear. Varia entre 0 e 400 e sua relação com o Kp é dada na Tabela 2.1.

Tabela 2.1 - Tabela de conversão entre os valores dos índices Kp e ap.

| Kp | 0 | 0+ | 1- | 1 | 1+ | 2- | 2 | 2+ | 3- | 3 | 3+ | 4- | 4 | 4+ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ap | 0 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 12 | 15 | 18 | 22 | 27 | 32 |
| Kp | 5- | 5 | 5+ | 6- | 6 | 6+ | 7- | 7 | 7+ | 8- | 8 | 8+ | 9- | 9 |
| ap | 39 | 48 | 56 | 67 | 80 | 94 | 111 | 132 | 154 | 179 | 207 | 236 | 300 | 400 |

Fonte: Adaptado de National Oceanic and Atmospheric Administration (NOAA) (1999).

- **Índice F10.7** é uma medida do fluxo solar na frequência de 2800 MHz, que corresponde ao comprimento de onda 10,7 cm. É um excelente indicador da atividade solar, uma vez que as emissões de rádio nessa frequência se originam na alta cromosfera e nas regiões mais baixas da corona solar. Apresenta boa correlação com o número de manchas solares, assim como com a irradiância de radiação ultravioleta e de radiação solar visível.

  Os dados históricos para este índice são bastante extensos, com sua coleta

iniciada em 1947. É dada em unidades de fluxo solar (s.f.u), variando de valores inferiores a 50 s.f.u a valores superiores a 300 s.f.u ao longo dos ciclos solares. Uma vez que a maioria da radiação ultravioleta que produz ionização na atmosfera terrestre tem origem na cromosfera solar, o índice F10.7 é um importante fator para avaliar a dinâmica da ionosfera.

- **Índice h'F** corresponde à medida da altura virtual da base da camada F, sendo medida por uma espécie de radar chamado de ionossonda, o qual estima o perfil vertical de densidade elétrons da ionosfera em função da frequência de sondagem (BERTONI, 1998). A resolução do h'F era anteriormente de 15 min, passando a 10 min após março de 2009.

  Neste trabalho, o índice foi utilizado com resolução de 10 min, o que exigiu o uso de uma técnica de interpolação para reamostrar a série temporal anterior com resolução de 15 min.

- **Velocidade de deriva vertical do plasma**, que é dada pela taxa de variação temporal de h'F, conforme a expressão

$$vh'f = \frac{h'F_i - h'F_{i-1}}{\Delta t}. \tag{2.9}$$

- **Índice** $S_4$ é utilizado para avaliar a cintilação ionosférica. Corresponde ao desvio padrão da intensidade do sinal de GPS normalizado pela média a cada minuto, coletado com 50 amostras por segundo:

$$S_4^2 = \frac{\langle I^2 \rangle - \langle I \rangle^2}{\langle I \rangle^2}. \tag{2.10}$$

- **Total Eletron Content (TEC)**, ou conteúdo total de elétrons em português, serve para avaliar a densidade de elétrons no plasma ionosférico.

  É calculada pelo número total de elétrons integrado ao longo do *link* entre um transmissor de satélite GNSS, e um receptor GNSS na superfície, considerando uma seção unitária (HOFMANN-WELLENHOF et al., 2013).

  Pode-se deduzir que cada valor de TEC depende do *link* considerado que atravessa a ionosfera, o qual varia de instante a instante, varia de conforme o satélite para a mesma estação, e varia com a localização da estação.

  A expressão para cálculo do TEC é:

$$\text{TEC} = \int n_e(s) ds, \tag{2.11}$$

onde $ds$ especifica o elemento de integração na trajetória.

Geralmente, é reportada em unidades de TEC (TECU), definido por 1 $TECU = 10^{16}$ elétrons/m$^2$.

É importante para determinar a atrasos de fase e de grupo em sinais de radiofrequência.

A grandeza **VTEC**, ou conteúdo total de elétrons vertical é projeção do TEC na linha normal à superfície da Terra.

As bolhas de plasma apresentam uma diferença em média da ordem de 30-50 TECU em relação à sua vizinhança. (TAKAHASHI et al., 2016).

- **Número de Wolf**, também conhecido como número internacional de manchas solares, quantifica o número de manchas solares presentes na superfície do Sol num dado intervalo de tempo.

- **IMF Bz**, **IMF By**, $V_{sw}$ e $P_{sw}$ são variáveis que correspondem, respectivamente, às componentes $z$ e $y$ do campo magnético interplanetário e à velocidade e à pressão do vento solar.

## 2.7 Ciclo solar

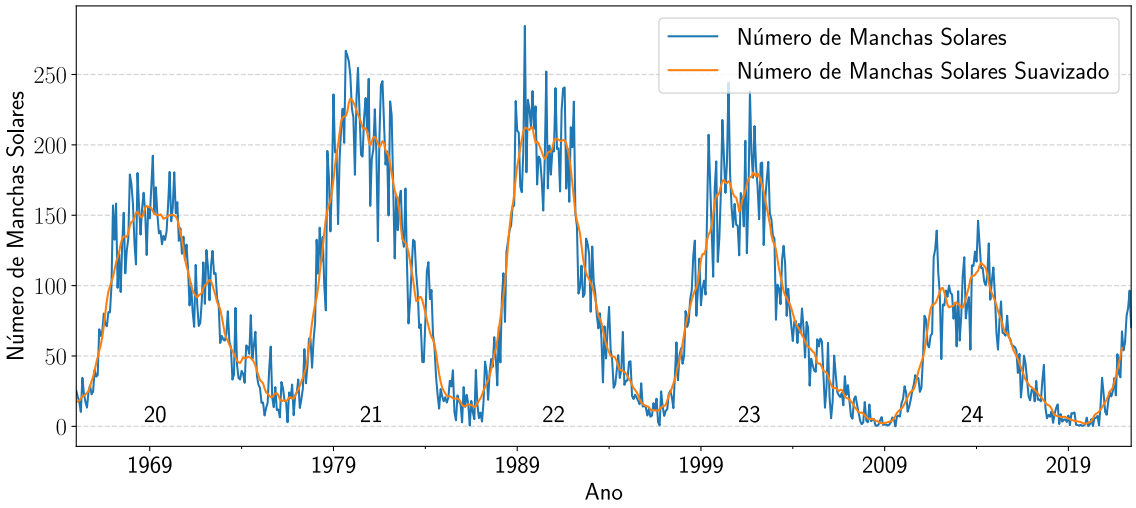
A principal fonte de energia para a fotoionização ionosférica é a radiação solar. Sua intensidade está diretamente conectada à quantidade de radiação emitida pelo Sol, a qual depende de sua dinâmica.

No século 17, foi possível observar utilizando telescópios regiões escuras na superfície do Sol, denominadas em inglês *sunspots*, e em português, manchas solares.

Em 1776, Christian Horrebow fez a primeira menção à eventual periodicidade no comportamento destas manchas, porém somente em 1844 foi confirmada a existência de ciclos solares com o trabalho de Heinrich Schwabe. Suas observações a respeito do número de manchas solares diárias ao longo de um período de 18 anos indicaram a presença de um ciclo de atividade solar com período aproximado de 10 anos.

A Figura 2.4 apresenta a curva suavizada do número de manchas solares desde 1900, a qual evidencia haver periodicidade (atualmente estimada em 11 anos).

Figura 2.4 - Curva suavizada do número diário de manchas solares em função dos anos.




Fonte: Produção do autor.

Johann Rudolf Wolf continuou os trabalhos de Schwabe, sendo que observou ser mais conveniente identificar grupos de manchas solares do que manchas individuais, introduzindo assim o número de Wolf $R$, dado por

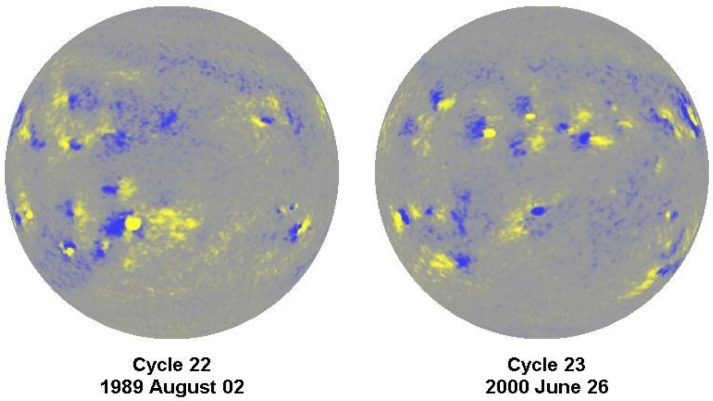
$$R = k(10g + n), \tag{2.12}$$

onde $k$ é um fator de correção para um dado observador (definido para cada observador, de forma a compatibilizar observações de diferentes observadores), $g$ é o número de grupos de manchas solares identificados e $n$ o número de manchas solares individuais. Note nesta expressão o peso maior do número de grupos.

A existência de um ciclo no comportamento das manchas solares é um forte indicativo da existência de um ciclo na dinâmica solar.

A física básica a respeito deste ciclo foi elucidada em 1908, quando George Ellery Hale e seus colaboradores demonstraram que as manchas são regiões fortemente magnetizadas. Em 1919, observaram que o campo magnético polar atingia um mínimo no período com maior número de manchas solares, denominado de máximo do ciclo solar, sendo que ocorre o oposto no mínimo do ciclo (campo com valor máximo e menor número de manchas solares). Observaram também que ocorre uma inversão da polaridade do campo magnético de um ciclo para o seguinte, como observado na Figura 2.6, sendo as polaridades opostas em hemisférios Figura 2.5.

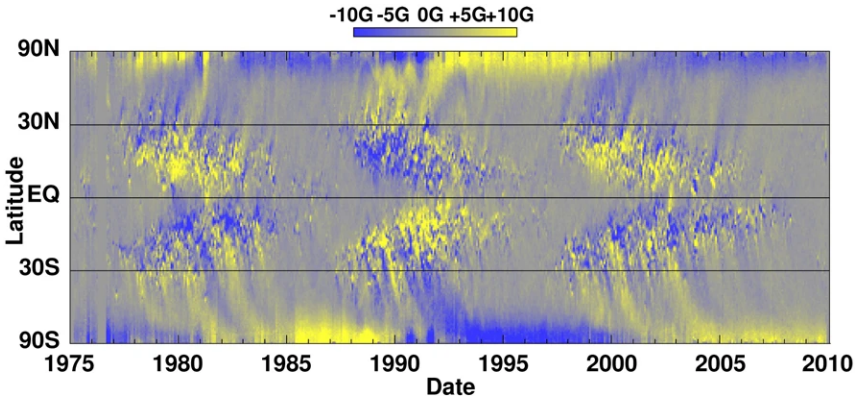
Figura 2.5 - Magnetogramas do Sol. À esquerda, em 02/08/1989 (ciclo solar 22) e à direita, em 20/06/2000 (ciclo solar 23).




A cor azul denota a polaridade negativa, enquanto a amarela, a positiva. Note que as manchas de maior destaque em um hemisfério apresentam uma correspondência no hemisfério oposto, mas com polaridade magnética invertida. Comparando o ciclo 22 (esquerda) com o ciclo 23 (direita) é possível observar a inversão de polaridade magnética (do amarelo para azul, no hemisfério sul).

Fonte: Hathaway (2015).

Figura 2.6 - Variação da projeção longitudinal da componente radial média da densidade de fluxo magnético do Sol (em Gauss) em função da latitude e do ano. Nota-se nas regiões polares que ocorre uma inversão do polaridade do campo magnético.




Fonte: Hathaway (2015).

Concluiu-se assim que o ciclo solar é melhor representado pelo campo magnético do Sol. Entretanto, observações sistemáticas com resolução diária começaram em 1970, havendo assim poucos ciclos solares caracterizados em função do campo magnético em comparação com a caracterização pelo número de manchas solares, sendo esta última grandeza usada preferencialmente.

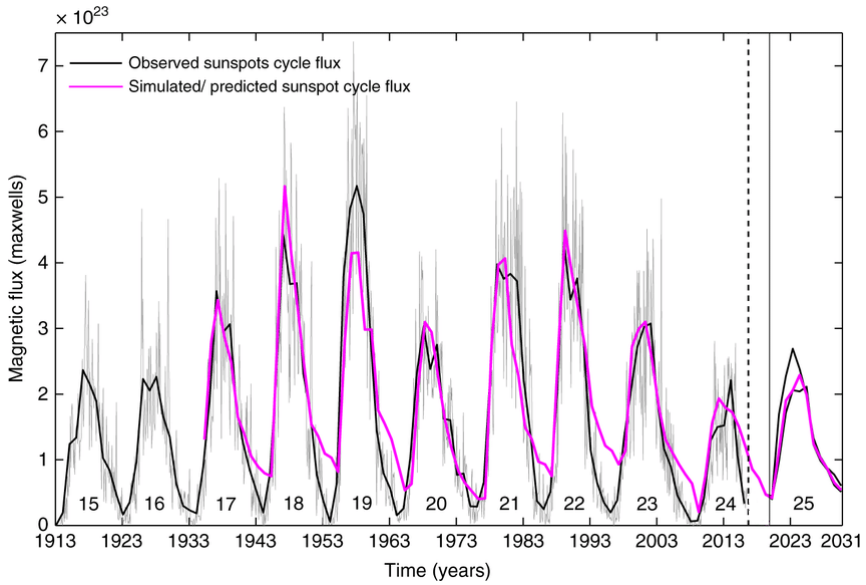
Em 1946, iniciaram-se medidas sistemáticas do índice F10.7, com resolução diária. A comparação deste índice com o número de manchas solares permitiu que Holland e Vaughn definissem em 1984 uma que relacionasse ambas essas grandezas, sendo que a correlação entre essas variáveis chegou a 99,5%, (HATHAWAY, 2015). Assim, fica evidente que variações no ciclo solar provocam variações na quantidade de radiação solar emitida e que, no máximo do ciclo, ocorrerá um máximo na emissão e vice-versa. Deste modo, é importante considerar a fase do ciclo solar quando for realizada alguma modelagem da ionosfera.

O ciclo solar 23 abrangeu o período de maio de 1996 até janeiro de 2008, o ciclo solar 24, o período de janeiro de 2008 até dezembro de 2019. O ciclo solar no momento da escrita desta tese é o ciclo solar 25.

Os dados coletados para este trabalho são do ciclo solar 24, o qual apresentou um pico duplo de máximo solar (em 2011 e em 2014), ambos parte da fase de máximo do ciclo.

O ciclo 24 foi relativamente fraco quando comparado com seu o seu antecessor, como pode ser visto na Figura 2.7, relativa à predição do fluxo magnético para o ciclo solar 25. Nesta figura é também possível observar que, conforme resultados de simulações/predições computacionais, o ciclo 25 terá uma amplitude ligeiramente maior que o ciclo 24 (BHOMWIK; NANDY, 2018).

Figura 2.7 - Estimativa da densidade de fluxo magnético para o ciclo solar 25.




Fonte: Adaptado de Bhomwik e Nandy (2018).

É importante destacar que existem diferentes abordagens que fazem a predição do ciclo 25, e que eventuais diferenças entre estas são normais, uma vez que a física do ciclo não é completamente conhecida.

# 3 APRENDIZADO DE MÁQUINA

A Mineração de Dados é uma área da Ciência da Computação que permite inferir conhecimento a partir de uma massa de dados específica de um fenômeno ou evento qualquer. Em particular, a mineração de dados contempla a inferência de modelos orientados a dados, ou seja, modelos derivados unicamente a partir de conjuntos conhecidos de dados de entrada e de saída específicos de um dado fenômeno ou evento.

Nessa área, destacam-se os chamados algoritmos de aprendizado de máquina (*machine learning*), que basicamente constituem uma forma de estatística aplicada que explora a capacidade de processamento de computadores para estimar funções, em geral, complexas, a partir de um conjunto de dados.

Nas seções seguintes, definem-se Mineração de Dados e Aprendizado de Máquina.

## 3.1 Mineração de dados

A mineração de dados é frequentemente confundida com o processo mais amplo de descoberta de conhecimento em base de dados (KDD - *Knowledge Discovery in Databases*), definido como a extração de padrões e desenvolvimento de representações associados ao conhecimento de um processo ou fenômeno a partir de um conjunto de dados associados.

A mineração de dados é uma etapa do processo de KDD, sendo definida como a extração de padrões em um conjunto de dados.

Conhecimento em um processo KDD pode ser entendido como o conjunto de objetos úteis ao especialista, gerados a partir dos dados brutos de interesse. Os objetos incluem, por exemplo, tabelas contendo correlações, métricas de desempenho de classificação ou regressão, regras de associação, indicadores categóricos decorrentes de agrupamento, entre outros. As diferenças entre o processo de KDD e a etapa de mineração de dados ficam mais claras sumarizando as etapas contidas no primeiro:

a) **Sanitização dos dados**: remoção de dados com ruído, inconsistentes e incompletos;

b) **Integração de dados**: combinação de múltiplas fontes de dados, por meio de operações como união e intersecção de tabelas;

c) **Seleção de dados**: extração de dados considerados relevantes ao processo

do banco de dados;

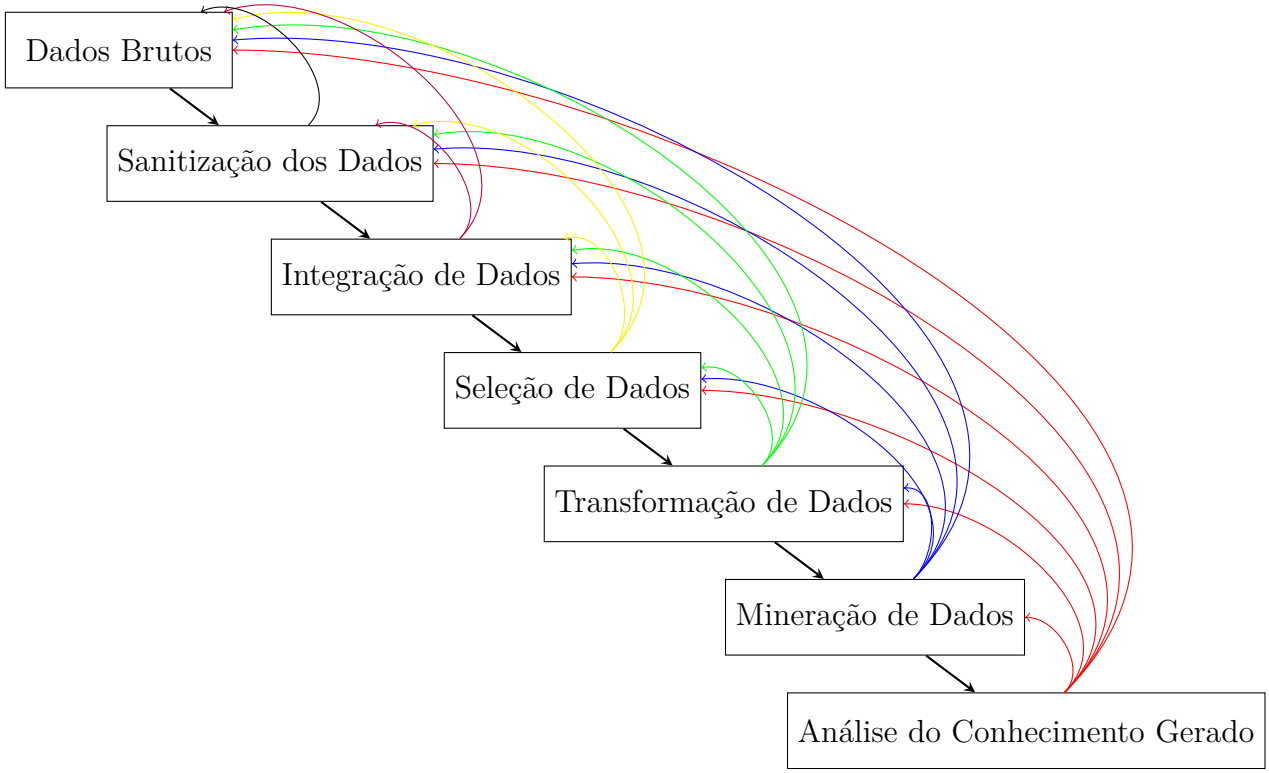
d) **Transformação de dados**: transformação e consolidação dos dados numa forma mais apropriada para mineração, sendo por exemplo discretizados, normalizados, agrupados;

e) **Mineração de dados**: é a etapa do KDD em que ocorre a aplicação de algoritmos estatísticos, de reconhecimento de padrões ou de inteligência computacional, os quais são usados conjuntamente com métricas para a avaliação dos modelos resultantes;

f) **Análise do conhecimento gerado**: análise feita utilizando-se técnicas de visualização de dados e similares.

O processo de KDD é interativo, como pode ser visto na Figura 3.1, no sentido em que etapas podem ser revistas e reexecutadas em função dos resultados obtidos.

Figura 3.1 - Passos de um processo de KDD.




Fonte: Produção do autor.

## 3.2 Aprendizado de máquina

O aprendizado de máquina faz uso de algoritmos que são capazes de "aprender" a partir dos dados, denominados assim **algoritmos de aprendizado de máquina**. Note que esta definição é incompleta, uma vez que não foi definido o que significa aprender a partir dos dados.

Considere uma tupla $(P, D, M)$ onde $P$ corresponde ao problema, $D$ ao conjunto de dados (domínio) e $M$ a métrica, diz-se que um algoritmo aprende com relação ao problema $P$, ao conjunto de dados $D$ e à métrica $M$, se seu desempenho no tratamento do problema $P$, avaliado segundo a métrica $M$ melhora conforme procedem as iterações varrendo os dados no domínio $D$.

Basicamente, o aprendizado de máquina pode ser visto como o processo de estimar (ajustar) uma função segundo algum objetivo (métrica) a um conjunto de dados. Define-se função como se segue. Sejam $X$ e $Y$ dois conjuntos não vazios, que podem ou não ser iguais, e uma regra, ou conjunto de regras, $f$ que atribui a cada amostra $x$ de $X$ uma única amostra $y$ em $Y$. Portanto, uma **função** consiste no objeto matemático formado pela tupla $(X, Y, f)$.

Os termos modelo, estimador, preditor, regressor, classificador são todos sinônimos do termo função, pois são entendidos como uma realização do algoritmo de aprendizado para um conjunto de dados, ou seja, um conjunto de regras inferidas dos dados pelo algoritmo.

Uma amostra corresponde a um conjunto de características, atributos ou variáveis que foram quantificadas de algum objeto ou evento que se deseja estudar. Quando uma amostra é da forma $(\mathbf{x}, y)$, diz-se que a amostra apresenta um rótulo ou resposta indicado por $y$, enquanto amostras da forma $(\mathbf{x})$ são ditas não rotuladas.

O termo $\mathbf{x}$ também pode ser denominado de atributo de informação, variável preditora, etc., enquanto que o termo $y$ é denominado atributo de decisão. Os dados podem ser numéricos, quantitativos ou qualitativos. Estes últimos são dados categóricos, que correspondem à atribuição de rótulos/classes que os qualificam.

Uma definição formal de um problema de aprendizado de máquina foge do escopo deste trabalho, mas podem-se distinguir dois tipos de problema, em função de como uma amostra $(\mathbf{x}, y)$ ou $(\mathbf{x})$ do domínio $D$, é mapeada para uma saída $f(\mathbf{x})$:

- Problemas de classificação: o objetivo é determinar uma função $f$ que ma-

peia cada amostra $\mathbf{x}$ a uma classe $j$ do conjunto de $n$ classes $\{0, .., n-1\}$;

- Problemas de regressão: o objetivo é determinar uma função $f$ que mapeia cada amostra $\mathbf{x}$ para um valor numérico.

Os algoritmos de aprendizado de máquina podem ainda ser divididos entre supervisionados e não-supervisionados. No caso dos supervisionados, um modelo é ajustado na fase de treinamento a partir de amostras conhecidas, que incluem atributos preditores e as correspondentes respostas. Estes algoritmos são utilizados para tarefas como classificação, predição e regressão. Por outro lado, os algoritmos não-supervisionados buscam inferir conhecimento ao agrupar os dados em conjuntos distintos ou então ao reduzir a dimensionalidade dos dados, ou ainda ao extrair padrões, de forma a permitir uma melhor análise do fenômeno ou evento de estudo.

O grande desafio das técnicas de aprendizado de máquina é apresentar bom desempenho em amostras de dados diferentes das usadas no subconjunto de treinamento. Assim, define-se generalização como a capacidade de se obter bom desempenho de classificação ou regressão em amostras de dados não observadas previamente. Para a geração do modelo, o domínio $D$ do problema é usualmente particionado em dois subconjuntos, um de treinamento e um de teste (essa discussão será estendida mais adiante em configurações com mais subconjuntos). O subconjunto de treinamento é utilizado na fase de ajuste/treinamento, na qual uma métrica específica fornece um erro de treinamento para avaliar a capacidade do modelo se ajustar às amostras deste subconjunto.

Neste ponto, a aprendizado de máquina se distingue das abordagens usuais de ajuste de funções, pois define o erro de generalização, ou erro de teste, que avalia a capacidade do modelo se ajustar a novas amostras, isto é, não conhecidas na fase de treinamento.

O erro de generalização avalia o desempenho do modelo no subconjunto de teste. Os erros de generalização e treinamento podem ser comparados entre si assumindo-se que as amostras do subconjuntos de treinamento e de teste são independentes entre si, e que ambos subconjuntos são identicamente distribuídos e apresentam a mesma distribuição de probabilidade. Tais hipóteses permitem inferir que o valor estimado de ambos os erros deve ser igual.

Em um problema real de aprendizado de máquina, as hipóteses acima não são completamente verdadeiras, pois na prática o número de amostras é finito e o processo

de amostragem não gera uma representação completa do fenômeno ou processo. Considerando-se um problema real é muito difícil obter representações de todas as configurações possíveis de um dado fenômeno ou processo. Assim, é natural que as distribuições dos dados sejam diferentes nos dois subconjuntos.

De maneira geral, como o modelo é treinado com o subconjunto de treinamento, espera-se que o modelo capture mais padrões deste subconjunto, fazendo que o erro esperado com o subconjunto de teste seja maior ou igual ao de treinamento.

Finalmente, determina-se o desempenho de um dado algoritmo de aprendizado de máquina pode ser avaliado pela sua capacidade de reduzir o erro de treinamento mantendo o erro de teste com valor próximo. Geralmente, essa capacidade depende do ajuste dos hiper-parâmetros do algoritmo considerado, para o qual não existe um procedimento assegurado. Depende também do particionamento dos dados entre os subconjuntos de treinamento e teste, bem como do esquema de validação proposto, discutido adiante.

Considerando o desempenho do algoritmo podem surgir dois casos extremos indesejáveis: o sobreajuste (*overfitting*) e subajuste (*underfitting*). O primeiro ocorre quando a diferença entre os erros de treinamento e teste divergem, decorrente do modelo se ajustar excessivamente às amostras do subconjunto de treinamento. O segundo ocorre quando o modelo não é capaz de reduzir suficientemente o erro de treinamento, ou seja, o treinamento não é bem sucedido.

Embora isso dependa do algoritmo de aprendizado de máquina considerado, é possível balancear o modelo entre sobreajuste e subajuste para os dados considerados por meio dos hiper-parâmetros e/ou dos esquemas de particionamento/validação adotados. Pode-se dizer que, considerando-se um dado algoritmo, um modelo é melhor se consegue se ajustar a um número grande de funções, mas podem se ajustar de maneira excessiva às amostras do subconjunto de treinamento e perder a capacidade de generalização, não obtendo bom desempenho com as amostras do subconjunto de teste. Ao contrário, um modelo restrito a se ajustar a um número pequeno de funções, pode nem sequer completar a fase de treinamento. Assim, dependendo do caso, deve-se escolher outro algoritmo de aprendizado de máquina.

## 3.3 Métricas e funções de erro

Métricas específicas foram criadas ou adaptadas para avaliar o desempenho de modelos de aprendizado de máquina, além de funções de erro ou perda que avaliam o

processo de treinamento desses modelos. Há casos onde a função de erro e a métrica são a mesma quantidade, tornando-se assim sinônimos, como usualmente aparece na literatura. Em mineração de dados em geral, aplicam-se métricas e funções de erro adequadas a problemas de classificação, de regressão, ou outros, como por exemplo, para agrupamento de dados.

Em aprendizado de máquina, utilizam-se métricas quantitativas (não diferenciáveis) quando se utilizam classes, enquanto que métricas numéricas (diferenciáveis) são empregadas quando se utilizam valores numéricos. O primeiro caso ocorre em problemas de predição tratados como problemas de classificação, enquanto que o segundo, quando são tratados como problemas de regressão.

### 3.3.1 Problemas de classificação

Em problemas de classificação, os quais incluem também os problemas de predição, a grande maioria das métricas de desempenho são baseadas na chamada matriz de confusão, a qual sintetiza o número de acertos e erros na classificação.

Admita um problema de classificação com $n$ categorias (classes), então a matriz de confusão será uma tabela (matriz) onde cada linha representa amostras em uma classe predita, enquanto cada coluna representa amostras em uma classe verdadeira, ou inverso. Assim, dado a $i$-ésima linha e a $j$-ésima coluna com $x$ amostras, diz-se que $x$ amostras da $j$-ésima classe foram preditas como da $i$-ésima classe. A Tabela 3.1 ilustra esta configuração, e por exemplo, pode-se dizer que $h$ amostras da Classe 1 foram previstas como sendo da Classe 2, com a segunda coluna pela terceira linha.

Tabela 3.1 - Exemplo de matriz de confusão, onde as colunas estão associadas as amostras verdadeiras e as linhas as amostras preditas.

| | | Verdadeiro | | | |
|---|---|---|---|---|---|
| | | Classe 0 | Classe 1 | Classe 2 | Total |
| Predito | Classe 0 | $a$ | $b$ | $c$ | $a+b+c$ |
| | Classe 1 | $d$ | $e$ | $f$ | $d+e+f$ |
| | Classe 2 | $g$ | $h$ | $i$ | $g+h+i$ |
| | Total | $a+d+g$ | $b+e+h$ | $c+f+i$ | N |

Fonte: Produção do autor.

Tabela 3.2 - Exemplo de matriz de confusão, onde as colunas estão associadas às amostras preditas e as linhas às amostras verdadeiras.

| | | Predito | | | |
|---|---|---|---|---|---|
| | | Classe 0 | Classe 1 | Classe 2 | Total |
| Verdadeiro | Classe 0 | $a$ | $d$ | $g$ | $a+d+g$ |
| | Classe 1 | $b$ | $e$ | $h$ | $b+e+h$ |
| | Classe 2 | $c$ | $f$ | $i$ | $c+f+i$ |
| | Total | $a+b+c$ | $d+e+f$ | $g+h+i$ | N |

Fonte: Produção do autor.

Por outro lado, na representação inversa (transposta) diz-se que $x$ amostras da $i$-ésima classe foram preditas como da $j$-ésima classe. A Tabela 3.2 apresenta esta configuração inversa, e pode-se dizer, por exemplo, que $h$ amostras da Classe 1 foram previstas como sendo da Classe 2.

Uma vez que ambas as representações são equivalentes, no sentido de fornecer a mesma informação, a escolha por uma ou outra depende da conveniência do usuário. Neste trabalho, adota-se a matriz de confusão nesta segunda configuração.

Considerando-se a matriz de confusão, pode-se definir:

a) **Verdadeiro Positivo (VP):** número de amostras que pertencem à classe $i$ e foram corretamente classificados/preditos como $i$, sendo expressos para a Classe 0 por $VP_0 = a$, enquanto que generalizando para uma classe $i$ qualquer, tem-se:

$$VP_i = A_{ii} \tag{3.1}$$

b) **Falso Positivo (FP):** número de amostras que não pertencem à classe $i$, mas que foram incorretamente classificados/preditos como $i$, sendo expressos para a Classe 0 por $FP_0 = b + c$, enquanto que generalizando para uma classe $i$ qualquer, tem-se:

$$FP_i = \sum_{\substack{j=0,\\ j\neq i}}^{n-1} A_{ji} \tag{3.2}$$

c) **Verdadeiro Negativo (VN):** número de amostras que não pertencem à classe $i$, mas foram corretamente classificados/preditos como não sendo $i$, expressos para a Classe 0 por $VN_0 = e + h + i + f$, enquanto que

generalizando para uma classe $i$ qualquer, tem-se:

$$VN_i = \sum_{\substack{j=l=0,\\ j\neq i,\\ l\neq i}}^{n-1} A_{jl} \tag{3.3}$$

d) **Falso Negativo (FN):** número de amostras que pertencem à classe $i$, mas foram incorretamente classificados/preditos como não sendo $i$, expressos para a Classe 0 por $FN_0 = d + g$, enquanto que generalizando para uma classe $i$ qualquer, tem-se:

$$FN_i = \sum_{\substack{j=0,\\ j\neq i}}^{n-1} A_{ij} \tag{3.4}$$

Nas expressões acima (3.1)-(3.4), $A$ é a matriz de confusão, $n$ é o número de classes denotadas por $i, j, l \in \{0, ..., n-1\}$, e os exemplos seguem a convenção da Tabela 3.2.

Utilizando as definições acima, é possível estabelecer muitas métricas de desempenho de classificação/regressão, como por exemplo, acurácia, precisão, especificidade, $F_1$, entre outras. A seguir, citam-se as métricas direta ou indiretamente utilizadas, assim como uma possível justificativa para a sua adoção.

Todas as métricas abaixo variam no intervalo $[0, 1]$, tendo como melhor valor 1 e como pior, zero.

- **Acurácia** (acc): é uma das métricas mais tradicionais para o tratamento de problemas de classificação, sendo definida como a razão entre número total de amostras preditas/classificadas corretamente e o número total de amostras.

  Em geral, um valor de acurácia de 0,5 indica que um modelo é tão bom quanto um sorteio de uma moeda não viciada, isto é, um caso em que a probabilidade de sair cara ou coroa são iguais.

  Seja $A$ uma matriz de confusão associada a um problema com $n$ classes, então a acurácia é dada por:

$$acc = \frac{\sum_{i=0}^{n-1} A_{ii}}{\sum_{\substack{i=0,\\ j=0}}^{n-1} A_{ij}} \tag{3.5}$$

- **Precisão**: para uma dada classe quantifica a razão entre o número de amostras corretamente preditos/classificados como sendo dessa classe e o total das amostras preditos/classificados (correta ou incorretamente) como sendo dela.

  Uma variação dessa métrica é a precisão balanceada, dada por uma média aritmética sobre a precisão de cada classe, geralmente empregada em problemas onde o número de amostras por classe não é balanceado.

  Seja $A$ a matriz de confusão já apresentada anteriormente, a precisão por classe para a classe $i$ dada por:

$$precision_i = \frac{VP_i}{VP_i + FP_i} \tag{3.6}$$

- **Sensibilidade ou revocação** (*recall*) para uma dada classe é definida como a razão entre o número de amostras corretamente preditos/classificados como sendo dessa classe e o número total de amostras que realmente pertencem a ela.

  A sensibilidade quantifica com qual frequência uma amostra que realmente pertence à classe $i$ é classificada como pertencendo a essa classe.

  Seja $A$ a matriz de confusão já apresentada anteriormente, a sensibilidade para a classe $i$ é dada por:

$$recall_i = \frac{VP_i}{VP_i + FN_i} \tag{3.7}$$

- $F_1$, para uma dada classe $i$, é definida como a média harmônica entre a precisão e a sensibilidade para aquela classe, sendo portanto:

$$F_{1i} = \frac{2}{(recall_i)^{-1} + (precision_i)^{-1}} = 2 \times \frac{(precision_i \times recall_i)}{(precision_i + recall_i)}, \tag{3.8}$$

  ou, em termos de $VP_i$, $FP_i$ e $FN_i$:

$$F_{1i} = \frac{VP_i}{VP_i + \frac{1}{2}(FP_i + FN_i)}. \tag{3.9}$$

Existem conjuntos de dados onde o número de amostras de cada classe difere por ordens de grandeza. Tais conjuntos são ditos não balanceados, sendo não adequado o uso da acurácia para avaliar os modelos gerados a partir desses conjuntos.

Como exemplo, considere um problema com 3 classes, e que o número de amostras da Classe 0 seja $x$ e das Classes 1 e 2 combinadas seja também $x$. Admita ainda que o classificador acerte todas as amostras da Classe 0, porém erre todas as demais duas classes. Então, a acurácia será 1/2 ou 0,5. Se o número de amostras da Classe 0 for o dobro ($2x$), então a acurácia será 2/3 ou 0,6666. Assim, um aumento do número de amostras da Classe 0, que é a classe predominante, leva a um aumento da acurácia, enquanto que as Classes 1 e 2 continuam sendo erroneamente classificadas. Quanto maior for esse desbalanceamento, maior a tendência do modelo classificar mais amostras como sendo da Classe 0, sendo que a acurácia não exprime o erro do modelo em classificar as demais classes.

Assim é necessário utilizar outras métricas como a precisão, a sensibilidade e o $F_1$, que permitam avaliar melhor os modelos mesmo na presença de classes que dominam a distribuição de dados.

Note que para a Classe 0, do exemplo acima, a sensibilidade seria 1, pois todas as amostras da Classe 0 são corretamente classificadas. Se considerarmos a classe 0 como sendo a classe alvo (definindo VP, VN, FP e FN em relação a ela), podemos considerar dois casos extremos possíveis: o primeiro caso corresponde a todas as amostras das Classes 1 e 2 serem classificadas como sendo da Classe 0 e assim, o número de FPs da Classe 0 seria máximo, a precisão fica menor que 1 para esta classe, mas a sensibilidade continua sendo igual 1; o segundo caso corresponde a nenhuma amostra das Classes 1 e 2 ser classificada como sendo da Classe 0, e o número de FPs da Classe 0 seria zero. Assim, a precisão em relação à Classe 0 seria 1.

Considere agora um problema com apenas duas Classes (0 e 1), e com número igual de amostras. Se um modelo classificar todas as amostras como pertencendo a Classe 0, então sua sensibilidade seria 1, pois toda amostra dessa classe é corretamente classificada, porém sua precisão será $0,5$, pois a outra metade das amostras, da classe 1, será erroneamente classificada, ou seja, o total de falsos positivos da classe 0 corresponderia à metade do número de amostras.

Assim, existem casos que uma combinação de ambas a métricas, precisão e sensibilidade, dada pelo $F_1$, é mais adequada para avaliar o desempenho de classificação. Além disso, é possível avaliar o modelo de modo mais abrangente considerando-se a média aritmética dos $F_1$ de todas as classes.

Considere um problema de classificação com $n$ classes e amostras da forma $(\mathbf{x}, y)$

definindo um domínio $D$. Um modelo pode mapear um vetor de atributos $\mathbf{x}$ diretamente para um determinado $y$, por exemplo, por meio de uma regra de associação

$$\mathbf{x} > (1.0, 0.0) \Rightarrow y \in \text{Classe}\, i.$$

Alternativamente, um modelo pode fazer esse mapeamento indiretamente por meio de uma probabilidade. Como por exemplo, dado um valor de $\mathbf{x}$, então há 95% de probabilidade que ele pertença a classe $i$ e, portanto, 5% de que ele pertença a qualquer uma das outras classes. Essa diferença na forma de mapear uma amostra no valor a ser predito leva a diferentes modelos, que por sua vez requerem formas específicas de treinamento. Modelos que apresentam o resultado na forma de probabilidade, ou a calculam internamente, são treinados de forma a aproximar a distribuição de $p(y|\mathbf{x})$ para os dados do subconjunto de treinamento. Assim, devem ser consideradas duas distribuições de probabilidade $p(y|\mathbf{x})$ para os dados e $q(y|\mathbf{x})$ para o modelo.

O objetivo do treinamento, neste caso, é aproximar as duas distribuições, ou seja, minimizar a diferença entre ambas. Essa diferença pode ser avaliado por uma distância estatística denominada de divergência de Kullback-Leibler (ou Entropia Relativa), $D_{KL}(p||q)$, a qual é definida para o caso de distribuições discretas por

$$D_{KL}(p||q) = \sum_{\mathbf{x} \in X} p(\mathbf{x}) \log \left( \frac{p(\mathbf{x})}{q(\mathbf{x})} \right). \tag{3.10}$$

onde $X$ denota o conjunto de todas as amostras $\mathbf{x}$. O problema de minimização da divergência de Kullback-Leibler para a classificação considerada acima pode ser descrito como

$$\min_q D_{KL}(p||q) = \min_q \sum_{y \in Y} p(y|\mathbf{x}) \log \left( \frac{p(y|\mathbf{x})}{q(y|\mathbf{x})} \right), \tag{3.11}$$

onde $Y$ denota as $n$ classes do problema. Expandindo o logaritmo na expressão (3.11), têm-se

$$\min_q D_{KL}(p||q) = \min_q \left[ \sum_{y \in Y} p(y|\mathbf{x}) \log p(y|\mathbf{x}) - \sum_{y \in Y} p(y|\mathbf{x}) \log q(y|\mathbf{x}) \right]. \tag{3.12}$$

Note, porém, que o primeiro termo é constante, pois é uma soma das probabilidades condicionais das amostras que compõem o subconjunto de treinamento, sendo que estas probabilidades são conhecidas e não se alteram em função da distribuição $q$. Portanto, este primeiro termo pode ser descartado do processo de minimização.

Assim, a expressão (3.12) pode ser simplificada para:

$$\min_q D_{KL}(p||q) = \min_q (-1) \sum_{y \in Y} p(y|\mathbf{x}) \log q(y|\mathbf{x}). \tag{3.13}$$

Em função da somatória acima pode-se definir a função custo a ser otimizada (minimizada) em problemas de classificação modelados por probabilidade, como sendo o Erro de Entropia Cruzada (EEC), dado por

$$EEC = -\frac{1}{N} \sum_{j=1}^{N} \left[ \sum_{y_j \in Y} p(y_j|\mathbf{x}_j) \log q(y_j|\mathbf{x}_j) \right], \tag{3.14}$$

e para o caso de $n = 2$ (somente duas classes), define-se a Entropia Cruzada Binária (ECB), dada por

$$ECB = -\frac{1}{N} \sum_{j=1}^{N} \left[ p(y_j|\mathbf{x}_j) \log q(y_j|\mathbf{x}_j) + (1 - p(y_j|\mathbf{x}_j)) \log(1 - q(y_j|\mathbf{x}_j)) \right]. \tag{3.15}$$

### 3.3.2 Problemas de regressão

Problemas de regressão, os quais podem compreender também problemas de predição, têm métricas de desempenho baseadas nos erros numéricos, como descrito a seguir. O **erro quadrático médio** é definido por

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{3.16}$$

onde $n$ é o número de amostras a serem preditas (avaliadas), $y_i$ é a variável alvo observada/real para a amostra $i$ e $\hat{y}_i$, seu valor predito.

Além de ser uma métrica, o MSE é uma função de custo, pois é utilizada para otimizar o modelo e avaliar o seu desempenho, uma vez que é diferenciável. Assim, o MSE pode ser aplicado em problemas de otimização que utilizam gradientes como as redes neurais.

Entretanto, como o MSE é definido por uma operação de média, existem distribuições de dados a serem modelados para os quais o MSE não consegue avaliar/otimizar de maneira satisfatória o modelo. Isso ocorre, por exemplo, com distribuições de dados altamente localizadas, numa analogia com o problema de classificação com classes não balanceadas. Assim, uma métrica mais conveniente para esses casos seria

o Erro Absoluto Máximo (em inglês, MAE) definido por:

$$MAE = \max|y_i - \hat{y}_i|, \tag{3.17}$$

onde $i$ varia entre 1 e $n$, o número de amostras a serem preditas, e $y_i$ e $\hat{y}_i$ sendo respectivamente, o valor real/observado e o valor predito para a amostra $i$.

## 3.4 Particionamento dos dados e avaliação do modelo

De maneira geral, como já mencionado, o treinamento do modelo requer particionar o subconjunto de amostras em dois subconjuntos, um de treinamento e um de teste. Este particionamento pode ser feito de várias maneiras, por exemplo, sorteando sem reposição $p$ amostras para o subconjunto de treinamento e $k$ amostras para o subconjunto de teste.

Em certos casos, o conjunto de amostras pode apresentar um ordenamento, ou seja, uma sequência natural. Isso ocorre, por exemplo, quando os dados são informações coletadas ao longo do tempo, em que os primeiros dados coletados vão anteceder os últimos dados coletados, estabelecendo uma ordem cronológica. Nesta situação é comum dividir os dados de forma que aqueles que antecedem um dado instante pertencerão ao subconjunto de treinamento, e aqueles posteriores a esse instante, ao subconjunto de teste.

A partição do conjunto de amostras com o intuito de avaliar a capacidade de generalização, adicionado à sua avaliação constituem um processo que pode ser chamado de **Validação**. O conceito central é o particionamento do conjunto de dados em dois novos subconjuntos mutuamente exclusivos, e a utilização de um deles para a geração do modelo (subconjunto de treinamento) e o outro para avaliar o modelo (subconjunto de teste).

As diversas formas de validação podem ser agrupadas em dois grandes grupos:

- **Exaustivas**, treina-se e avalia-se o modelo a partir de todas as possíveis maneiras de dividir o conjunto de dados em subconjuntos de treinamento e teste;
- **Não-exaustivas**, não se utilizam todas as maneiras supracitadas para dividir o conjunto de dados.

As técnicas exaustivas apresentam um custo computacional alto endo impraticá-

veis exceto por alguns casos simples. Logo, a maioria dos problemas adota alguma abordagem não-exaustiva, sendo as duas mais tradicionais:

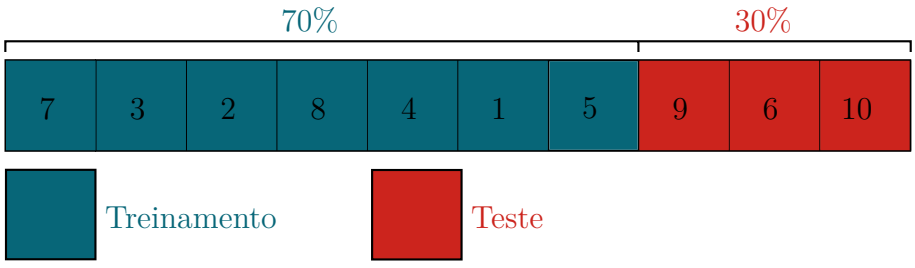
- **Holdout**, neste método as amostras são aleatoriamente distribuídas em dois subconjuntos, tal que cada amostra pertence somente a um subconjunto. Estes são designados por subconjunto de treinamento e subconjunto de teste.

  O tamanho de cada um destes subconjuntos é arbitrário, podendo assim, ser separados em quantidades iguais ou não. Usualmente, adota-se uma proporção de 70% de amostras para treinamento e 30% para teste.

  É bastante empregado em problemas com um grande número de amostras onde a distribuição de probabilidade associada aos subconjuntos é semelhante (é possível garantir isso, por exemplo, fazendo a distribuição das amostras mantendo a proporcionalidade entre o número de amostras de uma classe $n$ e o número total de amostras nos dois subconjuntos). Sem tais análises, podem ocorrer flutuações da métrica considerada na avaliação do modelo causando que o desempenho deste seja avaliado erroneamente, para melhor ou para pior.

  O *Holdout* é aplicado também em casos em que o custo computacional de treinamento do modelo é muito alto (por exemplo, semanas ou meses), inviabilizando abordagens mais sofisticadas.

Figura 3.2 - Exemplo de Holdout no qual 10 amostras foram divididas em dois subconjuntos, de treinamento e de teste, com respectivamente 70% e 30% do total de amostras.




Fonte: Produção do autor.

- **Validação cruzada com k subconjuntos**, este método consiste em dividir as amostras aleatoriamente em $k$ subconjuntos mutuamente exclusivos

de mesmo tamanho. Também é possível adotar estratégias mais sofisticadas para dividir as amostras, por exemplo, garantindo que a distribuição de probabilidade associada aos subconjuntos sejam semelhante.

Considerando os $k$ subconjuntos, $k-1$ subconjuntos são usados como subconjunto de treinamento e o subconjunto restante, como subconjunto de teste. Mantendo-se inalterados os subconjuntos, repete-se $k$ vezes o treinamento e avaliação de performance do modelo, sendo que para a $i$-ésima vez se adota um subconjunto $i$ como teste e os demais como treinamento. O resultado final é dado pela média dos $k$ resultados parciais, ou seja, pela média das $k$ métricas de desempenho obtidas.

A vantagem desta abordagem é que todas as amostras são usadas para treinamento e teste, e cada amostra é usada somente uma vez para teste, permitindo gerar uma medida mais confiável do desempenho do modelo.

Figura 3.3 - Exemplo de Validação Cruzada com 10 amostras divididas em 5 subconjuntos de duas amostras cada, em que a cada etapa, um subconjunto diferente é utilizado para teste, e os demais, para treinamento.


Fonte: Produção do autor.

As técnicas de validação têm por objetivo permitir uma avaliação mais robusta da capacidade de generalização de um modelo, isto é, de seu desempenho num conjunto não observado, ou seja, não utilizado no treinamento.

Em geral, os modelos apresentam dois conjuntos de parâmetros, aqueles que são aprendidos/otimizados diretamente dos dados automaticamente pelo algoritmo de aprendizado de máquina utilizado, e outros, os denominados hiper-parâmetros, que requerem ser ajustados por experimentação (tentativa-e-erro) ou utilizando um algoritmo adicional para esse fim. Definem-se assim os hiper-parâmetros de um algoritmo como o conjunto de variáveis que controlam e definem o processo de treinamento/aprendizado, determinando o modelo resultante.

Os hiper-parâmetros, em geral, são ajustados utilizando-se o conjunto de treinamento, mas isso não garante um bom desempenho do modelo gerado com o conjunto de testes. Assim, esse modelo pode não ter boa capacidade de generalização. Esse problema é contornado pela extensão da distribuição de amostras, anteriormente feita para dois subconjuntos, um de treinamento e outro de teste, para três subconjuntos: treinamento, validação e teste. O novo subconjunto de validação é então utilizado para ajustar os hiper-parâmetros.

Essa divisão do conjunto de amostras em três subconjuntos é uma extensão imediata do *holdout*. Extensões semelhantes existem para a validação cruzada com $k$ subconjuntos. Um exemplo seria a validação aninhada (*nested*), na qual os dados são divididos em $k$ subconjuntos num nível mais externo, separando-se $k-1$ para treinamento (denominado de conjunto de pseudo-treinamento) e um para teste. A seguir, num nível mais interno, considera-se cada conjunto de pseudo-treinamento e divide-se o mesmo num segundo nível de validação cruzada em $l$ subconjuntos, em que $l-1$ subconjuntos serão utilizados para treinamento e um para validação.

Essa abordagem de validação aninhada ou multi-nível pode ser empregada para ajuste de hiper-parâmetros, ou então, como no caso deste trabalho, para a seleção de atributos preditores, conforme discutido adiante neste capítulo.

As técnicas de validação, assim como os modelos, fazem algumas suposições a respeito das amostras de dados, uma delas é que os dados são identicamente e independentemente distribuídos, o que é adequado para uma vasta coleção de problemas. Entretanto, não são adequados no caso de séries temporais, compostas por amostras coletadas regularmente ao longo do tempo. Tais amostras podem apresentar alguma correlação entre si, isto é, dados coletados em um instante $t$ tendem a ser relacionados com os dados coletados em instantes anteriores $t-1$, $t-2$, etc. Similarmente, dados coletados posteriormente são relacionados com os dados coletados em $t$. Assim, o particionamento convencional viola essas relações entre amostras de séries temporais, podendo inviabilizar até mesmo o treinamento, devido ao chamado

vazamento de dados.

O vazamento de dados ocorre quando o modelo é treinado com alguma informação que seja compartilhada com os dados de teste. Suponha, por exemplo, que um modelo seja treinado para predizer um instante $t + 1$ dado um instante $t$ (observe a construção de um par $(t, t + 1)$) e que, devido ao particionamento, o instante $t + 1$ pertença ao subconjunto de treinamento, enquanto $t$ pertence ao subconjunto de teste. Então, devido às características do problema de predição, alguns casos podem acontecer: $t$ também irá pertencer ao subconjunto de treinamento, pois é necessário $t$ para predizer $t + 1$; ou $t + 1$ pertencerá também ao subconjunto de teste pois $t$ irá predizer $t + 1$, sendo que em ambos os casos um dos subconjuntos ficará truncado.

Na modelagem, este tipo de vazamento pode ser solucionado observando que os dados se dão em pares pela introdução de uma lacuna, se o par $(l, l + 1)$ pertencer ao subconjunto de treinamento, o de teste não poderá ter $(l - 1, l)$ e nem $(l + 1, l + 2)$. A lacuna deve ser definida de forma que não haja mais dependência entre os dados pertencentes a subconjuntos distintos, acrescida de uma margem de segurança devido às dependências indiretas não completamente conhecidas.

A escolha do tamanho da lacuna não é necessariamente difícil: se um fenômeno apresenta uma escala temporal de um dia, uma lacuna de alguns dias, ou até de uma semana seria conveniente, assumindo-se que tal lacuna garanta a independência entre fenômenos sucessivos.

Um outro problema que surge no tratamento de dados com dependência temporal é que não faz sentido utilizar dados do futuro para predizer o passado. Assim, o ordenamento temporal deve ser preservado, de forma que apenas dados do passado sejam usados para prever o futuro. Note entretanto, que certos esquemas, como a Validação Cruzada com Lacuna abaixo definido, permitem utilizar dados futuros no treinamento, mas ainda respeitando o ordenamento temporal.

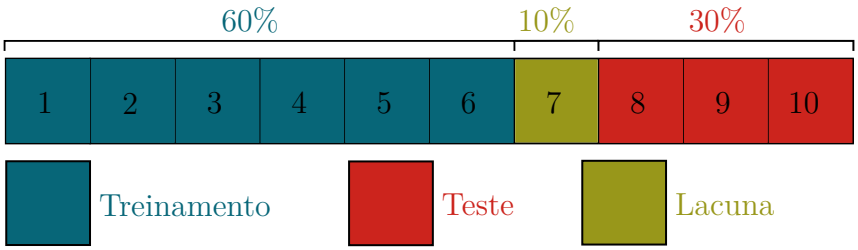
Levando em consideração a dependência temporal entre amostras sucessivas no particionamento e na avaliação do modelo, diversos esquemas foram propostos:

- ***Gap Time Series Holdout*** (GTSH): semelhante ao *holdout* tradicional, porém preservando o ordenamento temporal dos dados, e adicionando lacunas, ilustrado na Figura 3.4.

- ***Time Series Cross Validation Gap K-Fold*** (TSCV-GKF): análogo à validação cruzada com $k$ subconjuntos tradicional, porém preservando

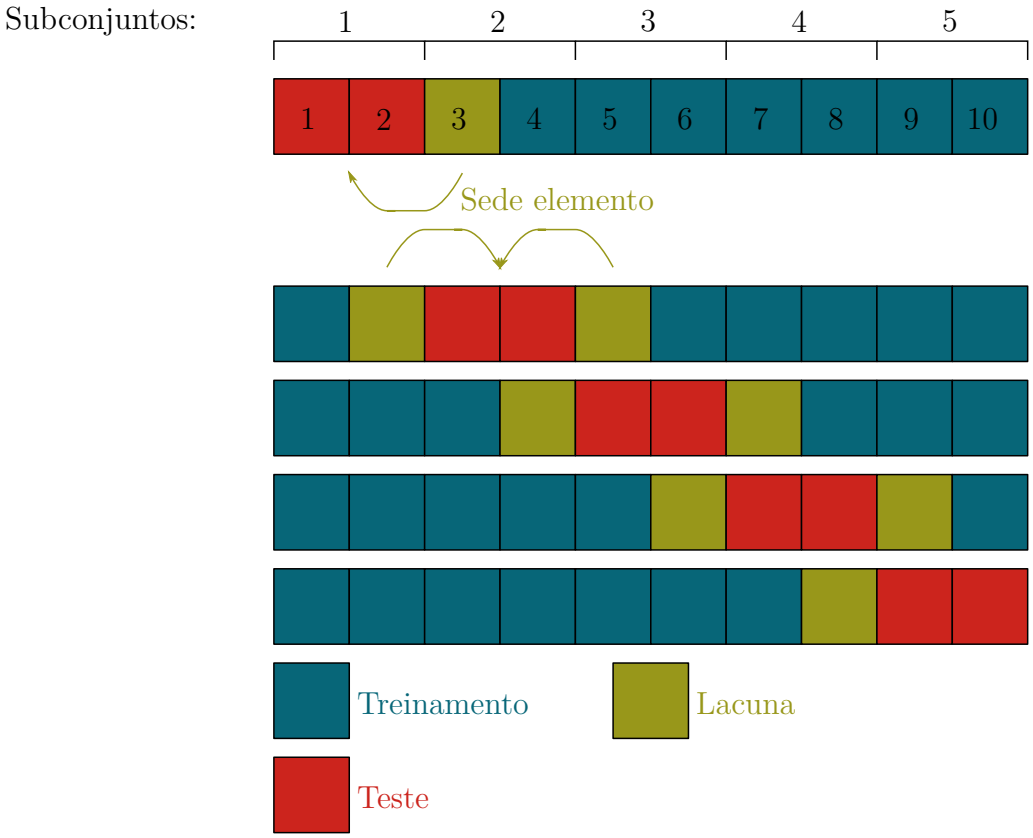
o ordenamento temporal dos dados, e adicionando lacunas compostas por amostras removidas dos subconjuntos adjacentes ao subconjunto de teste, ilustrado na Figura 3.5.

Figura 3.4 - Exemplo de particionamento GTSH, o qual preserva o ordenamento temporal e utiliza lacuna entre os subconjuntos de treinamento e teste.




Fonte: Produção do autor.

Figura 3.5 - Exemplo de particionamento TSCV-GKF, o qual preserva o ordenamento temporal e, a cada iteração, adiciona lacunas antes e depois do conjunto de teste, onde aplicável.

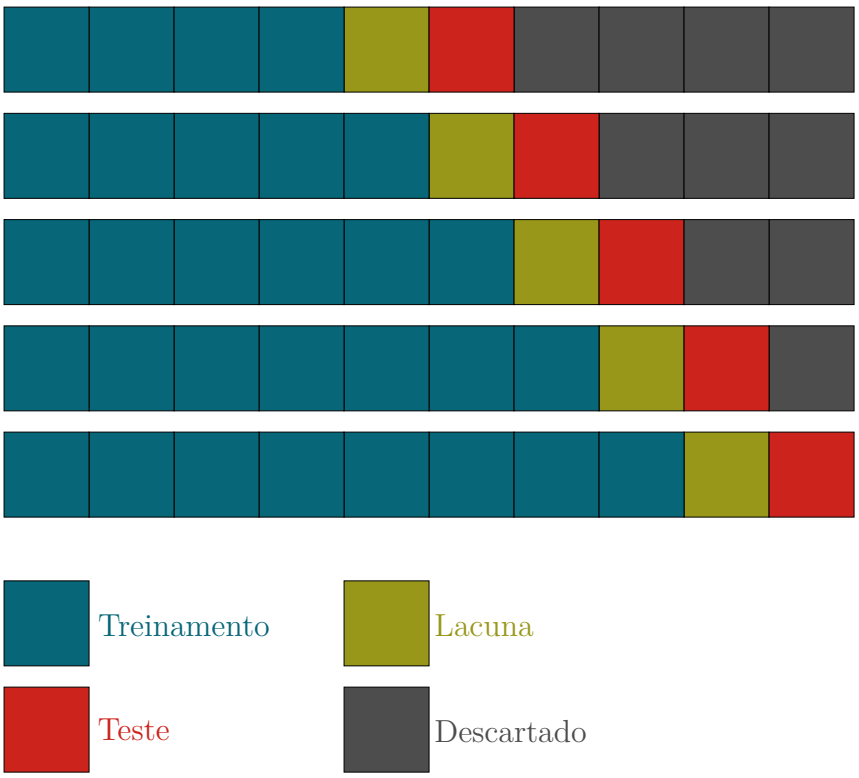



Fonte: Produção do autor.

- ***Time Series Cross Validation Gap Walk Forward*** (TSCV-GWF): apresenta semelhança com o validação cruzada temporal com lacuna, dadas as iterações com diferentes subconjuntos de treinamento e teste. Nesta abordagem, o subconjunto de treinamento sempre antecede o subconjunto de teste, sendo que a cada iteração, o tamanho do subconjunto de treinamento aumenta, como ilustrado na Figura 3.6.

  Existem algumas variações que definem os tamanhos mínimo e máximo dos subconjuntos de treinamento e teste, ou o tamanho da lacuna, ou ainda o número de iterações.

Figura 3.6 - Exemplo de particionamento TSCV-GWF, o qual preserva o ordenamento temporal e que, ao longo das iterações, utiliza um conjunto de treinamento com número crescente de amostras, separado por lacuna do subconjunto de teste, descartando amostras que sucedem temporalmente aquelas deste último subconjunto.




Fonte: Produção do autor.

## 3.5 Engenharia e seleção de atributos

A geração de um modelo baseado em dados apresenta as etapas típicas do processo KDD, incluindo a etapa de transformação dos dados, relativa à engenharia e à seleção

de atributos apresentados nesta seção.

A engenharia de atributos é a utilização de conhecimento prévio para extrair e representar os dados em novas variáveis que sejam mais adequadas para o algoritmo de aprendizado de máquina. Por exemplo, as séries temporais são uma sequência de amostras ordenadas temporalmente e que apresentam correlação entre elas. Assim, não se deve utilizar neste caso uma abordagem que assuma independência entre as amostras, uma vez que se perderia a informação referente a essas correlações. Entretanto, esse problema pode ser contornado utilizando-se uma representação que capture a correlação entre as amostras, por meio de uma nova variável.

Atualmente, existem algoritmos como as redes neurais profundas, discutidas no próximo capítulo, capazes de extrair sua própria representação dos dados, isto é executando a engenharia de atributos de maneira automática. Entretanto, tais abordagens requerem um extenso conjunto de amostras e apresentam uma interpretabilidade reduzida em comparação a uma abordagem manual. Assim, é necessário ponderar sobre as vantagens e desvantagens ao considerar abordagens automáticas ou manuais para a engenharia de atributos.

Atualmente, as redes neurais constituem o padrão para processamento de imagens e linguagem natural, que envolvem grandes conjuntos de dados e recursos computacionais por parte de grandes empresas. Uma abordagem intermediária é usar extratores automáticos de atributos, baseados em técnicas bem estabelecidas e conhecidas, tais como a Transformada de Fourier, cujos coeficientes seriam os novos atributos, ou o cálculo da entropia, entre outras.

O número possível de transformações que podem ser aplicadas em um conjunto de dados é ilimitado, de modo que o número de novos atributos derivados dos atributos originais pode ser imensamente maior. Assim, a seleção de atributos torna-se necessária, permitindo selecionar um subconjunto reduzido de atributos que melhor contribui para a modelagem.

Há várias maneiras de fazer a seleção, conforme o tipo dos atributos, mas há dois enfoques principais, descritos a seguir:

- **Seleção de atributos supervisionada**: um subconjunto de atributos é avaliado no treinamento gerando um modelo, sendo o desempenho deste modelo interpretado como sendo do próprio subconjunto. Diferentes subconjuntos devem ser testados, demandando muitos testes, eventualmente

extensos. Entretanto, uma vez que a seleção é baseada no desempenho real do modelo, os resultados são geralmente bons para uma variedade de problemas;

- **Seleção de atributos não-supervisionada**: realizam-se testes estatísticos com/entre os atributos de forma a determinar aqueles que estão correlacionadas ou então que não apresentem muita informação. Como exemplos, tem-se a variância de cada atributo ou a então a covariância entre pares de atributos. Assim, pode-se descartar atributos com variância menor que um limiar definido, ou então, descartar um dos atributos do par que apresenta covariância alta, uma vez que ambos contém a mesma informação, são linearmente dependentes.

A seleção de atributos é uma etapa desejável, uma vez que:

- Acelera o processo de treinamento do algoritmo de aprendizado de máquina escolhido;
- Reduz a complexidade do modelo e torna sua interpretação mais fácil;
- Melhora o desempenho do modelo para um subconjunto ótimo ou subótimo de atributos;
- Reduz a possibilidade de sobreajuste (*overfitting*).

### 3.5.1 Engenharia e seleção de atributos para séries temporais

Seja $D = \{\zeta_i\}_{i=1}^{N}$ um conjunto de séries temporais, composto por de $N$ séries temporais, a engenharia de atributos visa mapear cada série $\zeta_i$ em um espaço bem definido de características com dimensão $M$ por um vetor de atributos $\zeta_i = (x_{i,1}, x_{i,2}, ..., x_{i,M})$.

A forma mais simples seria escolher $M$ pontos sucessivos de cada série temporal para compor o vetor de atributos, sendo possível ainda tomar $M$ como sendo o comprimento total da série temporal. Entretanto, segundo a teoria de aprendizado de máquina e reconhecimento de padrões (BISHOP, 2006) isto pode levar à perda de informação.

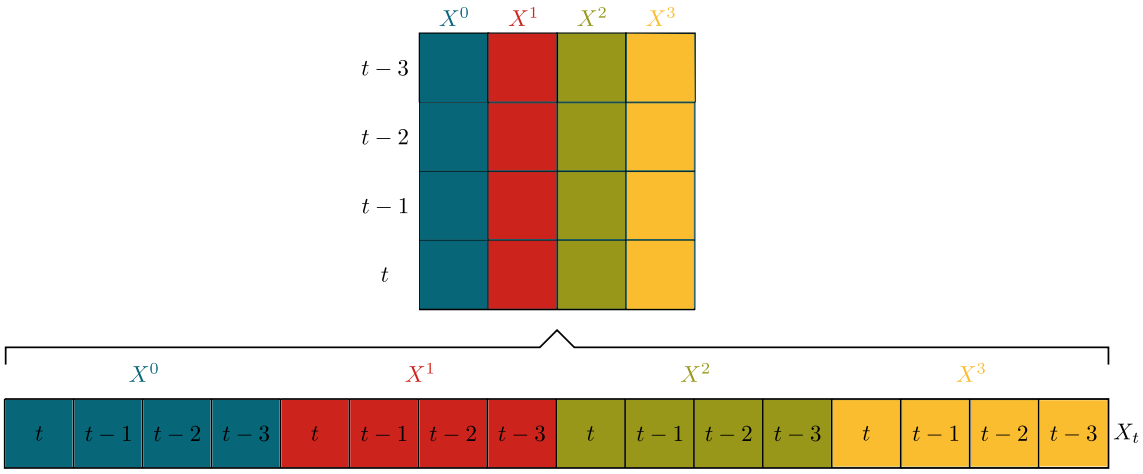
Um mapeamento mais eficiente das séries temporais em um espaço de atributos pode ser feito pela caracterização destas com respeito à distribuição dos pontos, às propriedades de correlação, à estacionaridade, à entropia, e às análises não lineares

(FULCHER, 2017). Assim, o vetor de atributos seria construído pela aplicação de uma função de caracterização $f_j$ em uma série temporal $\zeta_i$, que resulta em $\zeta = (f_1(\zeta_i), f_2(\zeta_i), ..., f_M(\zeta_i))$.

O vetor de atributos pode ser ainda estendido adicionando-se atributos extraídos de outras séries temporais ou atributos isolados. A Figura 3.7 ilustra um vetor de atributos definido com as 4 amostras mais recentes de 4 séries temporais para um instante $t$.

A engenharia de atributos para séries temporais consome muito tempo, pois demanda avaliar muitas heurísticas ou algoritmos de seleção de atributos, seja de forma automática ou manual. Essa questão levou ao desenvolvimento de diferentes bibliotecas para a extração automática de atributos nos últimos anos, tais como as bibliotecas TSFRESH,(CHRIST et al., 2016; CHRIST et al., 2018) e TSFEL, (BARANDAS et al., 2020). Ambas foram utilizadas neste trabalho, mas apenas para a geração de atributos. Uma seleção de atributos supervisionada foi feita posteriormente com auxílio do algoritmo de aprendizado de máquina empregado.

Figura 3.7 - Geração do vetor de atributos composto pelas 4 amostras mais recentes de 4 séries temporais em relação a um instante $t$.




Fonte: Produção do autor.

Existem também algoritmos de aprendizado de máquina para séries temporais que utilizam apenas uma única função de caracterização, como o algoritmo de Weasel-Muse (SCHÄFER; LESER, 2017) que emprega a transformada de Fourier. A diferença é que estes algoritmos realizam transformações adicionais sobre o resultado da função de caracterização.

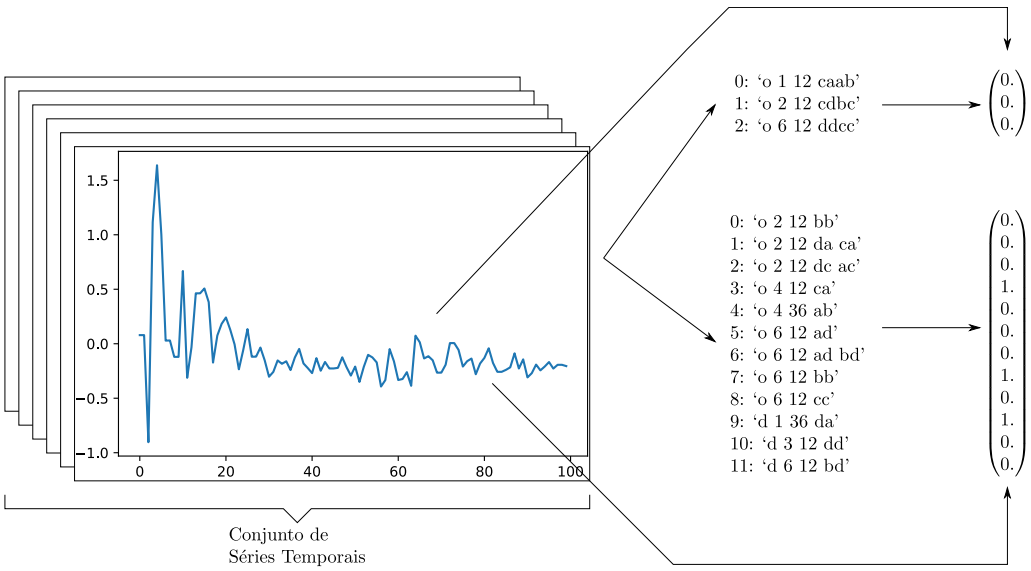
No algoritmo Weasel-Muse, os coeficientes são discretizados por alguma função em símbolos do alfabeto, que são então combinados em palavras. Assim, cada série temporal é composta por um conjunto de palavras, sendo que o conjunto de todas as palavras distintas formará um dicionário.

Diferentes dicionários podem ser construídos, por exemplo, definindo-se palavras com tamanhos diferentes. Para uma dada série temporal, verifica-se o número de vezes que cada palavra do dicionário aparece na série, sendo que estes números compõem então o vetor de atributos.

O algoritmo de Weasel-Muse realiza ainda uma seleção de atributos para descartar palavras que não sejam relevantes.

A Figura 3.8 ilustra alguns passos do algoritmo: a partir do processamento das séries temporais gera-se um dicionário contendo todas as palavras possíveis, sendo apresentados dois dicionários, o primeiro com palavras com 4 símbolos e um total de 4 palavras, e o segundo com palavras de 2 símbolos e um total de 12 palavras. Assim, cada vetor de atributos tem o tamanho do número de palavras do dicionário correspondente.

Figura 3.8 - Ilustração de dois dicionários possíveis gerados pelo algoritmo de Weasel-Muse para o mesmo conjunto de dados, com palavras de tamanho diferentes e tamanhos diferentes, tendo cada vetor de atributos a dimensão dada pelo número de palavras de cada dicionário.




Fonte: Produção do autor.

A seleção e engenharia de atributos é aplicada no próximo capítulo para algoritmos de aprendizado de máquina baseados em árvores, os quais não são adequados para utilizar séries temporais e demandam uma representação tal como a ilustrada na Figura 3.7. Uma segunda classe de algoritmos a serem apresentados no próximo capítulo são as redes neurais, as quais tem variantes específicas para tratar séries temporais.

# 4 ALGORITMOS UTILIZADOS

## 4.1 Árvores de Classificação e de Regressão

Árvores de Classificação e de Regressão (do inglês CART - *Classification and Regression Trees*) são algoritmos de aprendizado de máquina que particionam o espaço de atributos, definido como o espaço/domínio multidimensional formado pelos atributos de entrada que compõem cada amostra da base de dados.

No caso de uma árvore de classificação, os atributos de entrada podem ser numéricos ou categóricos, mas com o objetivo de atribuir uma classe a cada amostra a ser classificada. No caso de uma árvore de regressão, os atributos de entrada são geralmente numéricos, com o objetivo de atribuir um valor numérico a cada amostra a ser predita ou extrapolada.

O processo de construção de uma árvore varia conforme o problema a ser tratado é de regressão ou classificação, conforme discutido adiante.

Uma árvore pode ser interpretada como um grafo acíclico não-direcionado, composto por nós de decisão (incluindo o nó raiz, inicial), nós terminais (folhas) e os ramos que os interligam. No caso de uma árvore binária, cada nó de decisão contém uma regra na forma maior ou igual que, gerada de um atributo específico e um valor para este (o valor é escolhido em das amostras) que subdivide o nó (subconjunto de amostras associado) em dois nós derivados (cada qual com seu subconjunto de amostras disjuntos), recursivamente.

A árvore assemelha-se a um fluxograma em que as amostras, seja no fase de treinamento ou de teste, sofrem dividas segundo algum atributo num nó de decisão. Esse processo inicia-se no nó raiz, sendo efetuado sucessivamente nos nós de decisão percorrendo um conjunto de ramos da árvore até se atingir um nó terminal (folha), o qual define a classe ou valor numérico a ser atribuído àquela amostra.

Como em todo algoritmo de aprendizado de máquina, há uma fase de treinamento que utiliza amostras com atributo de saída conhecido (classe ou valor numérico). Nessa fase, a árvore vai sendo construída gradativamente a partir do nó raiz e seus nós derivados em termos da sequência/ordem de particionamento dos atributos que a caracteriza. Em cada nó de decisão, a escolha do atributo correspondente é feita considerando-o como se fosse um nó terminal que vai determinar a classe/valor numérico das amostras.

Em geral, o critério para a escolha do atributo é a minimização de alguma quantidade, por exemplo, o atributo que resulta no menor erro de classificação, ou então no menor erro numérico, para o conjunto das amostras daquele nó. Esse processo continua até que seja atingido um critério de parada, como por exemplo, um limite mínimo de número de amostras nos nós de decisão, que então passam a ser considerados nós terminais, os quais obviamente não tem nós derivados.

O número final de partições define a profundidade da árvore. Em algumas abordagens, para evitar *overfitting*, esse número geralmente é diminuído, num processo denominado poda (*pruning*).

Existem diferentes algoritmos baseados em árvores para classificação e regressão, tais como CART (aqui abrangido), C4.5, C4, etc. A maioria desse algoritmos utiliza recursão binária, em que o espaço de atributos vai sendo sucessivamente particionado em duas regiões para cada atributo considerado. Embora o problema resultante seja NP-completo, a recursão binária é mais tratável e possibilita uma interpretação mais intuitiva.

Os critérios específicos de escolha do atributo para cada nó de decisão são discutidos adiante, para as árvores de regressão e de decisão.

### 4.1.1 Árvores de regressão

No caso de árvores de regressão, a recursão binária requer a definição de um limiar $y$ para o atributo numérico $x$ para cada nó de decisão, de forma a particionar o espaço de atributos com base nesse atributo. Cada amostra será "direcionada" para o ramo correspondente ao nó derivado da esquerda ou da direita segundo o atributo da amostra seja respectivamente menor/igual ou maior que esse limiar. Assim, as sucessivas partições dividem o espaço de atributos pelo conjunto de limiares. Para cada nó de decisão, considera-se o mesmo como nó terminal e escolhe-se o atributo e o limiar que minimizem o erro de regressão para as amostras que "chegam"ao nó. Geralmente utiliza-se o erro quadrático médio (MSE). No caso de árvores de regressão, um critério de parada que pode ser adotado é atingir um valor mínimo de MSE na construção da árvore.

### 4.1.2 Árvores de decisão

Árvores de decisão são algoritmos de aprendizado de máquina supervisionados que executam classificação. Requerem um critério de avaliação da ordem/sequência dos atributos para cada nó da árvore, além de um critério de parada, sendo que existem

várias opções para ambos os critérios. No caso do primeiro critério, existem métricas específicas relativas a uma ordem que maximize a separabilidade entre classes. O segundo critério pode ser dada, como mencionado, por um número mínimo de amostras após uma partição, ou por um número limite de partições/nós, ou ainda pelo erro médio de classificação das classes (em analogia ao MSE utilizado nas árvores de regressão).

Uma árvore de decisão ótima para um determinado conjunto de amostras classifica corretamente o maior número possível de amostras, levando-se em conta restrições como critérios de parada. Uma árvore de decisão ideal classificaria perfeitamente todas as amostras do conjunto considerado e seria definida como apresentando pureza de 100%, no sentido de que cada nó terminal da árvore apresenta somente amostras de uma mesma classe. Na prática, todas as árvores apresentam um grau de impureza.

Em árvores de decisão, a escolha do atributo para cada nó de decisão, incluindo o nó raiz, considera a proporção de amostras de cada classe $k$ classificadas em cada nó. Num determinado nó $m$, deve-se considerar somente as $N_m$ amostras a serem classificadas naquele nó, uma vez que somente o nó raiz classifica o total de $N$ amostras. Denota-se por $\hat{p}_{mk}$ a proporção de amostras da classe $k$ no nó $m$. Como já mencionado, para cada nó de decisão $m$, a escolha do atributo utilizado para o particionamento é feita considerando-se o nó como sendo terminal. Assim, atribui-se a todas as amostras desse nó a classe $k_m$ que corresponde ao $\hat{p}_{mk}$ máximo, de forma a reduzir a impureza do nó, uma vez que o maior número possível de amostras seria corretamente classificada naquele nó de decisão caso fosse considerado como nó terminal.

Considerando-se um nó $m$ com atributo ótimo $k_m$, as 3 métricas de impureza comumente utilizadas, todas com valor ótimo zero, são definidas a seguir. A escolha da medida de impureza depende do algoritmo, sendo que árvores CART utilizam o índice Gini.

- **Erro de Classificação:**

$$(1 - \hat{p}_{mk_m}), \tag{4.1}$$

- **Indice Gini:**

$$\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}). \tag{4.2}$$

- **Entropia Cruzada:**

$$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}. \tag{4.3}$$

## 4.2 Abordagens de *ensemble*

A maioria dos algoritmos de aprendizagem de máquina, usualmente, ajusta um único modelo para resolver um problema. Em contraste, existe uma classe de métodos que adotam a abordagem de gerar um conjunto de modelos e combiná-los de forma a obter a solução do problema. Essa classe é denominada de métodos de *ensemble* ou métodos baseados em comitê.

O objetivo das abordagens de *ensemble* é combinar classificadores que apresentam erros de classificação da ordem de 50%, denominados de fracos em classificadores que apresentam um erro de classificação bem próximo de 0%, denominados de fortes.

Um conjunto contém um número de modelos de base, os quais são gerados a partir dos dados de treinamento por um algoritmo de aprendizagem de base. A maioria das abordagens de conjunto utilizam um único algoritmo de aprendizagem de base, e neste caso são denominadas de conjuntos homogêneos. Todavia, em princípio, não existe uma restrição para a utilização de diferentes algoritmos, neste caso, tem-se conjuntos heterogêneos.

A capacidade de generalização de um conjunto é maior do que os modelos de base. Em geral, utilizam-se algoritmos de base denominados de preditores fracos, cuja predição é um pouco melhor do que um sorteio aleatório. Assim, a abordagem de *ensemble* combina preditores fracos, obtendo um preditor forte, ou seja, gerando um modelo capaz de predições bem acuradas.

Os métodos de *ensemble* se tornaram um importante paradigma a partir da década de 90, com a publicação de dois trabalhos pioneiros (HANSEN; SALAMON, 1990) e (SCHAPIRE, 1990). O primeiro mostrou empiricamente que a predição realizada pela combinação de um conjunto de classificadores é em geral mais acurada do que a feita por um único bom classificador. O segundo, de caráter teórico, mostrou que preditores fracos podem ser combinados de forma a gerar preditores fortes.

Em geral, estes conjuntos são construídos em dois passos, a geração dos modelos de base e sua combinação. Nas próximas duas subseções serão discutidas as duas abordagens gerais de *ensemble*: o *bagging* e o *boosting*. Na primeira, os preditores base são gerados em paralelo, independentemente, enquanto que na segunda, são gerados sequencialmente (BREIMAN, 1996).

### 4.2.1 Bagging

Nesta abordagem de *ensemble*, busca-se explorar a independência entre os preditores de base, uma vez que o erro pode ser reduzido pela combinação de preditores de base independentes. Todavia, isso é praticamente impossível, uma vez que estes são gerados com base no mesmo conjunto de treinamento. Assim, uma maneira de reduzir a dependência dos preditores é introduzir aleatoriedade no processo de treinamento, inclusive na própria geração do conjunto de treinamento.

A abordagem de *bagging* introduz aleatoriedade pela geração de novas amostras no subconjunto de treinamento. Seus dois elementos principais são a amostragem com reposição e a agregação, que é a combinação dos resultados dos preditores de base.

Considere um conjunto de treinamento $A$ contendo $m$ amostras, um novo conjunto $A_1$ de $m$ amostras é gerado amostrando-se com reposição $A$. Neste caso, note que alguns exemplos do conjunto $A$ podem aparecer repetidas vezes, enquanto que algumas amostras podem não aparecer. Aplicando este processo $l$ vezes são gerados $l$ conjuntos com $m$ amostras. Então, cada conjunto é utilizado no treinamento de um modelo de base, aplicando-se o algoritmo de aprendizagem considerado.

A agregação é realizada utilizando procedimentos tradicionais. No caso da regressão, toma-se como resultado a média dos resultados gerados por cada preditor de base e, no caso de classificação, toma-se a classe mais predita pelos preditores de base, analogamente a uma votação.

### 4.2.2 Boosting

O termo *boosting* refere-se a uma família de algoritmos capazes de converter um preditor fraco em um preditor forte. A principal motivação desta abordagem é explorar a dependência entre os preditores base, uma vez que o desempenho pode ser melhorado por um processo análogo à redução de resíduo. Um exemplo simples ilustra esta abordagem. Considere um preditor fraco sobre uma distribuição qualquer, e considere um problema de classificação binária, por exemplo classificar amostras de uma coleção em positivo ou negativo. O domínio $D$ é amostrado identicamente e independentemente da distribuição. Assuma então que esse domínio (conjunto de amostras) $D$ possa ser divido em 3 subconjuntos $D_1$, $D_2$ e $D_3$, cada um compondo $1/3$ de $D$. Inicialmente, tem-se apenas um classificador fraco (preditor fraco) $f_1$ que classifica corretamente todas as amostras dos conjuntos $D_1$, mas classifica erroneamente quase todas as amostras do conjunto $D_2$ e $D_3$, resultando num erro de

classificação de 2/3, no máximo.

Define-se então um novo domínio $D'$, o qual realça erros cometidos por $f_1$ ao classificar as amostras de $D_2$ e $D_3$. Esse novo domínio $D'$ é então utilizado para treinar um novo classificador $f_2$ que combinado com classificador $f_1$ permite ter mais classificações corretas para $D_2$ e $D_3$, porém introduzindo erros de classificação em $D_1$.

Seguindo esse raciocínio, define-se ainda um domínio $D''$, o qual realça erros cometidos pela combinação de $f_1$ e $f_2$ ao classificar as amostras de $D_1$, $D_2$ e $D_3$. Esse novo domínio $D''$ é então utilizado para treinar um novo classificador $f_3$ que combinado com classificadores $f_2$ e $f_1$ permite ter mais classificações corretas para $D_1$, $D_2$ e $D_3$.

Este processo continua de maneira interativa construindo um domínio que realce os erros cometidos pela combinação de todos os preditores que o antecedem, que é então utilizado no treinamento de um novo preditor e assim sucessivamente, com avaliações do erro, até que um limiar de erro mínimo seja alcançado.

Esta maneira de combinar preditores ilustra a abordagem de *boosting* que consiste em treinar um conjunto de classificadores fracos de maneira sequencial, de modo que cada novo classificador é treinado de forma a corrigir o erro cometido pela combinação de todos os classificadores que o antecedem.

O resultado da predição é dado pela combinação das predições desses classificadores fracos, em geral, na forma de uma soma. Uma maneira de realçar os erros é definir cada novo domínio tendo a variável resposta $y$ representada pelo resíduo entre o valor verdadeiro e o valor predito pela combinação de todos os classificadores treinados sobre os domínios que precedem o classificador atual.

Este trabalho emprega dois algoritmos de *boosting*, *Extreme Gradient Boosting* (XGBoost) e *Categorical Boosting* (CatBoost), ambos pertencentes a uma classe de algoritmos denominados *Gradient Boosting Tree* (GBT) cuja ideia central é combinar árvores construídas sequencialmente sobre um conjunto de dados de forma a minimizar uma função de erro, onde cada nova árvore é construída em função do gradiente da função de erro.

Os algoritmos XGBoost e CatBoost apresentam importantes refinamentos na forma de treinamento e construção das árvores. A inovação do algoritmo XGBoost consiste numa nova função de impureza com informação sobre o gradiente da função de erro na sequência de preditores bases. A inovação do algoritmo CatBoost consiste na geração de estatísticas para o tratamento das variáveis preditoras categóricas, as

quais não são utilizadas neste trabalho, e também pela introdução do conceito de *ordered boosting*.

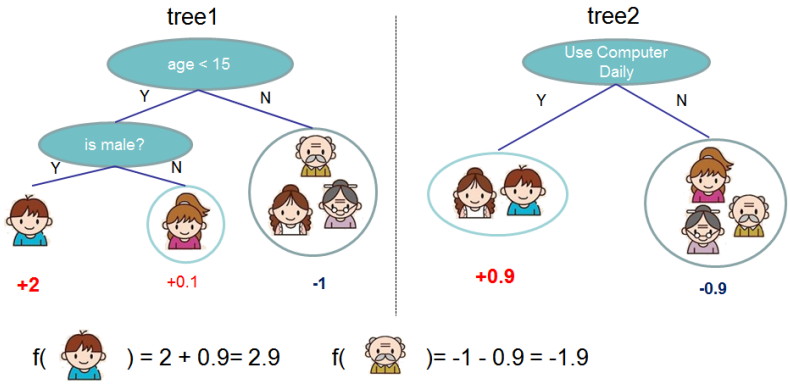
## 4.3 Extreme Gradient Boosting (XGBoost)

A explicação aqui apresentada sobre o algoritmo XGBoost é baseada em (CHEN; GUESTRIN, 2016). Considere um conjunto de dados com $N$ amostras e $p$ atributos $\mathcal{D}(\mathbf{x}_i, y_i)$ onde $\mathbf{x}_i \in \mathbb{R}^p$. Considere também $y_i \in \mathbb{R}$, um modelo baseado em um conjunto de árvores utiliza $K$ funções aditivas (árvores) para predizer/estimar, a variável resposta (saída)

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), f_k \in \mathcal{F}, \tag{4.4}$$

onde $\mathcal{F} = \{f(\mathbf{x}) = c_{q(\mathbf{x})}\}(q : \mathbb{R}^N \to M, c \in \mathbb{R}^M)$ é o espaço de árvores de regressão (CART), $q$ representa a estrutura de cada árvore que mapeia uma amostra na correspondente folha ou nó terminal, sendo $M$ é o número de folhas na árvore. Cada função $f_k$ corresponde a uma estrutura $q$ de árvore completa e independente com folhas de valor $c$. Assim, para uma dada amostra, utilizam-se as regras de decisão dadas por cada árvore $q$ para mapear a amostra em uma folha, e calcula-se o valor da predição conjunta somando os valores das correspondentes folhas (dados por $c$) das árvores $q$. A Figura 4.1 apresenta um exemplo de composição de árvores.

Figura 4.1 - Modelo de conjunto de árvores, sendo a predição final para uma amostra dada pela soma das predições de cada árvore.




Fonte: Adaptado de Chen e Guestrin (2016).

O processo de aprendizado deste algoritmo consiste em determinar as funções $f_k$, para tal, minimiza-se a seguinte função objetivo regularizada

$$\mathcal{L} = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \tag{4.5}$$

onde o termo de regularização é expresso por

$$\Omega(f_k) = \gamma M_k + \frac{1}{2}\lambda||\mathbf{c}_k||^2, \tag{4.6}$$

$l$ é uma função de erro convexa diferenciável que mede a diferença entre o valor predito $\hat{y}_i$ e o valor observado/real $y_i$. O termo de regularização penaliza a complexidade do modelo, uma vez que suaviza os coeficientes $c$ aprendidos, reduzindo a possibilidade de sobreajuste. Note que, sem esse termo de regularização, o XGBoost converte-se num algoritmo de *boosting* tradicional.

O modelo de conjunto de árvores na Equação (4.4) inclui funções não analíticas, e o problema de minimização (4.5) inclui funções com parâmetros e, portanto, não pode ser minimizado com técnicas convencionais de otimização em espaços Euclidianos. Assim, uma abordagem aditiva, análoga ao método guloso adotado na construção de árvores de regressão (CART), é utilizada.

Seja, $\hat{y}_i^{(t-1)}$ a predição da $i$-ésima amostra na iteração $t-1$, adiciona-se $f_t$, tal que

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i) \tag{4.7}$$

Objetiva-se então minimizar a seguinte função objetivo

$$\begin{aligned}\mathcal{L}^{(t)} &= \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t)}) + \Omega(f_t) \\ &= \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t).\end{aligned} \tag{4.8}$$

E, utilizando-se de uma aproximação de segunda ordem (expansão em série de Taylor) para o termo $\mathcal{L}^{(t)}$, tem-se:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{N}\left[l(y_i, \hat{y}_i^{(t-1)}) + \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}^{(t-1)}} f_t(\mathbf{x}_i) + \frac{1}{2}\frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}^{(t-1)^2}} f_t^2(\mathbf{x}_i)\right] + \Omega(f_t), \tag{4.9}$$

Essa notação pode ser simplificada definindo-se:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}^{(t-1)}} \quad \text{e} \quad h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}^{(t-1)^2}}, \tag{4.10}$$

$g_i$ e $h_i$ são respectivamente os gradientes de primeira e segunda ordem para a função de erro. Descartando-se os termos constantes e utilizando-se as simplificações

definidas em (4.10), obtém-se a seguinte função objetivo simplificada no tempo $t$

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{N} \left[ g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t). \tag{4.11}$$

Expandindo-se a Equação (4.11), e usando-se $f_t(x_i) = \sum_{m=1}^{M} c_m I(x_i \in R_m)$ e $f_t(x_i)^2 = \sum_{m=1}^{M} c_m^2 I(x_i \in R_m)$:

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{N} \left[ g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \gamma M + \frac{1}{2} \lambda \sum_{m=1}^{M} c_m^2 \\ &= \sum_{i=1}^{N} \left[ g_i \sum_{m=1}^{M} c_m I(x_i \in R_m) + \frac{1}{2} h_i \sum_{m=1}^{M} c_m^2 I(x_i \in R_m) \right] + \gamma M + \frac{1}{2} \lambda \sum_{m=1}^{M} c_m^2, \end{aligned} \tag{4.12}$$

trocando-se $i$ por $m$ e $m$ por $i$ no primeiro termo da expressão (4.12) e agrupando-se os termos, tem-se:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{m=1}^{M} \left[ c_m \sum_{i|\mathbf{x}_i \in R_m} g_i + \frac{1}{2} c_m^2 \left( \sum_{i|\mathbf{x}_i \in R_m} h_i + \lambda \right) \right] + \gamma M. \tag{4.13}$$

Tomando-se a variação da Equação (4.13) com relação a $c_m$, encontra-se o valor ótimo de $\hat{c}_m$ para a folha $m$, dado por

$$\hat{c}_m = -\frac{\sum_{i|\mathbf{x}_i \in R_m} g_i}{\lambda + \sum_{i|\mathbf{x}_i \in R_m} h_i}, \tag{4.14}$$

e o correspondente valor ótimo da função objetivo dados por

$$\tilde{\mathcal{L}}^{(t)} = -\frac{1}{2} \sum_{m=1}^{M} \frac{\sum_{i|\mathbf{x}_i \in R_m} g_i}{\lambda + \sum_{i|\mathbf{x}_i \in R_m} h_i} + \gamma M. \tag{4.15}$$

A Equação (4.15) pode ser utilizada para medir a qualidade de uma árvore, pois o valor da função objetivo é semelhante a uma medida de impureza, sendo derivada ao longo de várias iterações. Assim, à semelhança das árvores de decisão, é impossível avaliar todas as possíveis estruturas de árvores. Logo, um algoritmo guloso, que inicia com uma única folha e adiciona ramos à árvore, é utilizado.

Considere que $R_E$ e $R_D$ sejam respectivamente o nó da esquerda e da direita após a divisão. Definindo $R = R_E \cup R_D$, a redução de erro (análoga à redução de impureza),

após dividir os nós é dada por

$$\tilde{\mathcal{L}}_{split} = \frac{1}{2}\left[\frac{\sum_{i|\mathbf{x}_i \in R_E} g_i}{\lambda + \sum_{i|\mathbf{x}_i \in R_E} h_i} + \frac{\sum_{i|\mathbf{x}_i \in R_D} g_i}{\lambda + \sum_{i|\mathbf{x}_i \in R_D} h_i} - \frac{\sum_{i|\mathbf{x}_i \in R} g_i}{\lambda + \sum_{i|\mathbf{x}_i \in R} h_i}\right] - \gamma. \tag{4.16}$$

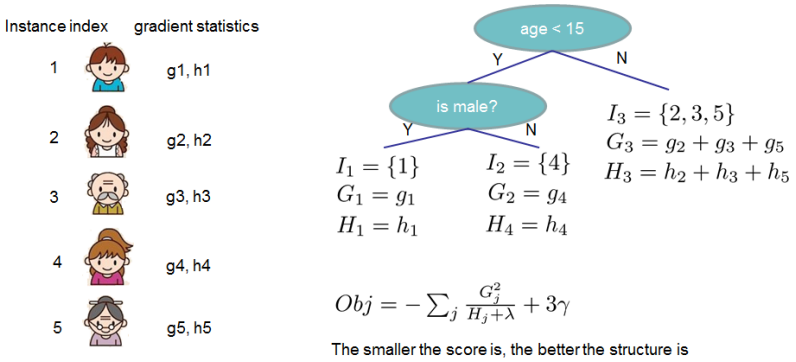
A fórmula (4.16) é usada na prática para avaliar os atributos preditores candidatos para a divisão. O algoritmo de busca guloso é apresentado abaixo em Algoritmo 1, enquanto que a Figura 4.2 ilustra o fluxograma relativo ao cálculo da divisão para um exemplo.

**Algoritmo 1** Algoritmo Guloso Exato de Busca de Divisão

valor $\leftarrow 0$
G $\leftarrow \sum_{i \in R} g_i$
H $\leftarrow \sum_{i \in R} h_i$
**para** k=1 até $N$ **faça**
    G_L $\leftarrow 0$
    H_L $\leftarrow 0$
    **para** j em R ordenado **faça**
        G_L $\leftarrow$ G_L $+ g_j$
        H_L $\leftarrow$ H_L $+ h_j$
        G_R $\leftarrow$ G $- G_L$
        H_R $\leftarrow$ H $- H_L$
        valor $\leftarrow \max(\text{valor}, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$

O algoritmo XGBOOST inclui várias outras técnicas para tratar o problema de sobreajuste e assim generalizar os resultados, algumas delas aqui empregadas, mas foge do escopo desta tese apresentá-las.

Figura 4.2 - Fluxograma relativo ao cálculo de divisão para um exemplo, bastando somar o gradiente e o gradiente de segunda ordem da função de erro em cada nó e então aplicar a fórmula (4.16) para se obter a medida de qualidade dada pela função objetivo.




Fonte: Adaptado de Chen e Guestrin (2016).

## 4.4 Categorical Boosting (CatBoost)

A cada iteração, o algoritmo CatBoost (PROKHORENKOVA et al., 2018; DOROGUSH et al., 2018) aprende uma árvore que reduz o erro cometido por todas as árvores prévias.

As árvores são construídas de maneira gulosa, i.e testando todas as possibilidades e assumindo-se que todas as árvores geradas nas iterações anteriores são fixas. Assim, ao aprender uma árvore, para cada nó, seleciona-se o atributo preditor e o limiar numérico que mais reduz a função de erro. Diferentes mecanismos de parada e de selecionar a próxima partição levam a diferentes esquemas de aprendizagem.

À semelhança de algoritmos de árvores em geral, o algoritmo CatBoost constrói os nós nível por nível até atingir uma profundidade limite, mas utiliza o mesmo atributo preditor e o mesmo limiar no particionamento de todos os nós de um mesmo nível, gerando árvores simétricas, as quais têm um custo computacional menor. Além disso, o algoritmo CatBoost discretiza os dados numéricos, reduzindo a complexidade do problema e portanto reduzindo ainda mais o custo computacional.

No algoritmo CatBoost, gera-se um conjunto de treinamento $D$ ordenado por meio de um índice fictício atribuído automaticamente (por exemplo, a primeira amostra recebe o índice 1 e assim sucessivamente). Considerando-se um conjunto de treinamento $D$ com $N$ amostras, seguem-se os seguintes passos:

a) Permutam-se aleatoriamente $s$ vezes as amostras de $D$ gerando novos conjuntos de treinamento $\{\sigma_r\}_{1,\ldots,s}$;

b) Inicializa-se com valor zero a matriz $M_{r,j}(i)$ correspondente à predição corrente (i.e. da árvore consideração nessa iteração) para a $i$-ésima amostra, com base na árvore construída com as $j$ primeiras amostras na permutação $\sigma_r$;

c) Escolhe-se aleatoriamente uma permutação do conjunto de treinamento $\sigma_r$, e para isso, geram-se estatísticas para todas as variáveis categóricas (casos existam) e constrói-se um nova árvore $T$ usando a técnica de *ordered boosting* que aproxima o gradiente de cada amostra na permutação $\sigma_r$ (para o cálculo do gradiente é necessário $M$); para a nova árvore T avaliam-se todas as amostras para todas as permutações e atualize-se $M$ com uma

estratégia de *gradient boosting*:

$$M_{r,j}(i) = M_{r,j}(i) - \alpha T(i) \tag{4.17}$$

onde $\alpha$ denota a taxa de aprendizado; este passo é repetido $I$ vezes de modo a construir $I$ árvores.

d) Calcula-se a predição final como sendo a soma das predições das I árvores, como num modelo de *boosting* tradicional.

Numa abordagem tradicional de *boosting*, o valor predito para a $k$-ésima amostra numa folha $p$ é baseado na predição de todas as amostras anteriores ou posteriores na mesma folha. Entretanto, em *ordered boosting* o valor predito para essa $k$-ésima é baseado somente na predição das amostras que a antecedem $k$ na folha $p$. O objetivo desta técnica é evitar que o gradiente tenha algum viés, devido a alguma variação na distribuição de probabilidades relativa aos subconjuntos de teste e treinamento.

## 4.5 Redes neurais artificiais

Em 30 setembro de 2012, a AlexNet, uma rede neural, desenvolvida por Alex Krizhevsky, (KRIZHEVSKY et al., 2017), competindo no *ImageNet Large Scale Visual Recognition Challenge*, marcou um momento histórico ao vencer com uma boa vantagem em relação ao segundo colocado, estabelecendo ainda um nível de acertos sem precedentes. Este evento marcou um renascimento na utilização de redes neurais, as quais eram até então abordagens "desprezadas". O conceito de redes neurais surgiu já na década de 40, em resposta à pergunta: como o cérebro funciona? Warren McCulloch, um neurofisiologista, e Walter Pitts, um matemático, tentaram responder modelando o neurônio como um simples circuito elétrico, onde dado um conjunto de entradas $\{x_1, x_1, \ldots, x_n\}$, definidas no conjunto $\{0, 1\}$, a saída $y$ é dada por

$$y = \begin{cases} 1, & \text{se } s(\mathbf{x}) \geq \theta \\ 0, & \text{se } s(\mathbf{x}) < \theta \end{cases}, \tag{4.18}$$

onde

$$s(\mathbf{x}) = \sum_{i=1}^{n} x_i. \tag{4.19}$$

Em 1949, Donnald Hebb expande as ideias de McCullcoh e Pitts, propondo que as conexões entre os neurônios sejam multiplicadas pelo uso contínuo, principalmente entre neurônios que são ativados simultaneamente.

A década de 50 tentou-se modelar um conjunto de neurônios interconectados formando uma rede por um sistema computacional, sendo que em 1954, a primeira rede Hebbiana foi implementada com sucesso no MIT. Em 1958, Frank Rosenblatt introduz a primeira rede neural artificial e também o modelo de neurônio chamado Perceptron.

As bases da técnica de treinamento por retropropagação (do inglês *backpropagation*) foram derivadas da teoria de controle desenvolvida por Kelley, em 1960 e por Bryson, em 1961. A primeira rede funcional com múltiplas camadas foi introduzida por Ivakhnenko e Lapa em 1965. Porém, em 1969, Minsky e Papert analisam uma rede constituída de único Perceptron, demonstrando que esta rede é incapaz de aprender uma simples lógica ou-exclusiva[1]. Demonstraram também que os computadores da época careciam de capacidade de processamento suficiente para tratar grandes redes neurais. Tal análise causou uma estagnação na área, com poucas pesquisas e aplicações sendo desenvolvidas. Entretanto, resultados importantes nas décadas seguintes permitiram o renascimento das redes neurais a partir de 2012.

Em 1970, Seppo pública um método geral para diferenciação automática (do inglês *Automatic Differentiation* (AD)) adequado a uma classe de redes neurais da época, sendo este método fundamental nas redes neurais da atualidade. Em 1982, Dreyfus usa retropropagação juntamente com o gradiente dos erros (este conceito será definido na próxima subseção). Em 1975, a técnica de retropropagação de Werbos permitiu treinar de maneira prática redes neurais com mais de uma camada (também denominadas multicamada). Em 1982, Werbos combinou sua técnica de retropropagação com o método de AD de Seppo, definindo o paradigma utilizado atualmente no treinamento das redes neurais.

Somente na década de 80, os avanços na tecnologia de fabricação de circuitos integrados, sua miniaturização e sua consequente explosão em termos de capacidade de processamento tornaram mais viável o uso de redes neurais. Em 1986, Rumelhart, Hinton e Williams mostram que a retropropagação permite que uma rede neural aprenda uma representação interna de palavras quando utilizada para prever a próxima palavra em uma sequência. A retropropagação já existia, mas este trabalho a popularizou desde então.

A década de 90 vai propor novas técnicas importantes para as redes neurais, mas que somente seriam exploradas nas próximas décadas graças ao aumento da capacidade

[1] A operação "ou-exclusivo" é uma função lógica que tem por argumento dois ou mais valores lógicos, e retorna uma saída 1 apenas se o número de argumentos com valor 1 for ímpar.

de processamento.

Em 2006, Hinton e outros autores propõem a máquina restrita de Boltzman de forma a modelar matematicamente as camadas de uma rede neural, com resultados excelentes. Isso permitiu modelar redes neurais com dezenas ou centenas de camadas, caracterizadas depois como sendo redes neurais profundas e sua área, como sendo *deep learning*.

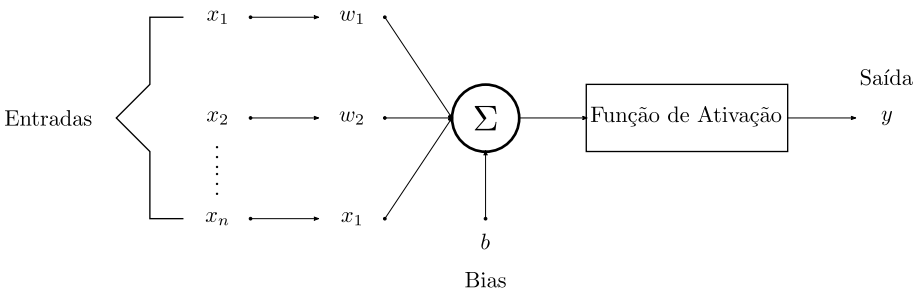
Finalmente, em 2012, tem-se a introdução da AlexNet, uma das primeiras redes convolucionais estabelecendo resultados recordes para o problema de identificar a classe de um objeto (gato, cachorro, navio). Deste ponto em diante, as redes convolucionais começam a ser utilizadas para tratar diversos problemas em visão computacional, sempre apresentando resultados excepcionais e ganhando diversas competições e prêmios na área. Acrescente-se a isso o aumento da capacidade de processamento resultante, por exemplo, da introdução de placas gráficas (GPUs) como aceleradores de processamento, tem-se então um aumento explosivo da aplicação e pesquisa em redes neurais profundas.

### 4.5.1 Redes neurais com camadas completamente conectadas

Uma rede neural artificial (ANN) é uma coleção de unidades computacionais denominada de neurônios organizados em camadas, tomando como inspiração um cérebro biológico. A Figura 4.3 apresenta uma representação gráfica da função matemática associada a um dos neurônios mais simples, o Perceptron, introduzido por Rosenblatt em 1958. Uma ANN apresenta uma camada de entrada, uma camada de saída e, eventualmente pode ter uma ou mais camadas intermediárias, chamadas de camadas ocultas, a Figura 4.4 apresenta os neurônios segundo esta organização em camadas.

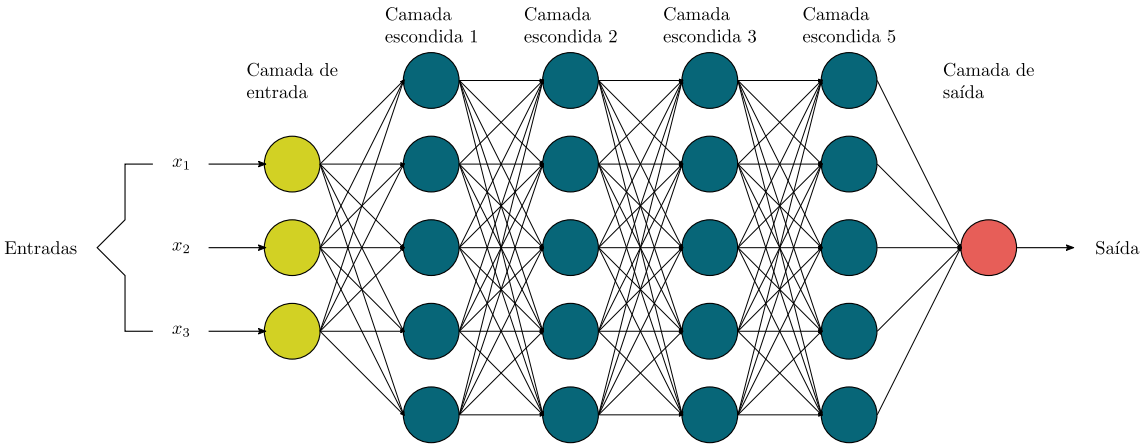
Em uma dada camada, cada neurônio recebe os sinais provenientes da camada anterior, ponderando-os por um peso e aplicando uma função não linear sobre a soma das entradas ponderadas mais um viés (*bias*), de forma a gerar um sinal de saída para a próxima camada. Esse é o neurônio apresentado na Figura 4.3, onde $\mathbf{x}$ é o sinal de entrada, $\mathbf{w}$ os pesos¸ $b$ *bias*, $y$ o sinal de saída. Os pesos e o bias serão aprendidos na fase de treinamento, sendo ambos parâmetros do neurônio. Pode-se então considerar o conjunto de parâmetros dos neurônios como sendo parâmetros da rede.

Figura 4.3 - Exemplo de neurônio.




Fonte: Produção do autor.

Figura 4.4 - Exemplo de rede neural.




Fonte: Produção do autor.

Em uma rede que apresenta propagação somente para frente *feedforward*, o sinal segue da camada de entrada para a camada de saída, atravessando cada camada somente uma única vez. Entretanto, em redes mais gerais o sinal pode passar por uma camada mais de uma vez, existindo assim algum mecanismo de recorrência. A Figura 4.4 ilustra uma rede neural *feedforward*, em que os neurônios são do tipo Perceptron. Esta rede pode ser denominada Rede com Múltipla Camadas Perceptron ou rede MLP (*Multi Layer Perceptron*).

Uma rede neural *feedforward* pode ser representada matematicamente como uma composição de várias e diferentes funções, concatenadas numa cadeia $f(\mathbf{x}) = f^{(n)}(f^{(n-1)}(...(f^{(1)}(\mathbf{x}))))$, onde o índice 1 denota a primeira camada, enquanto o índice $n$ denota a $n$-ésima camada. No caso de uma camada com neurônios tipo

Perceptron, a função $f$ será da forma

$$f(\mathbf{x}) = a\left(\sum_i x_i w_i + b\right), \tag{4.20}$$

onde $a$ é a função de ativação. Existem casos onde se utiliza a identidade como função de ativação, resultando numa camada linear. Na literatura é usual definir-se este tipo de camada como completamente conectada, em oposição a camadas que apresentam conexões esparsas, tais como a camada convolucional, discutida na próxima subseção. Adota-se aqui o termo camada linear como utilizando a identidade como função de ativação, exceto quando explicitado o contrário.

Considere agora um conjunto de $N$ amostras $(\mathbf{x}_i, y_i)$ tal que $i \in \{1, ..., N\}$ para as quais uma rede neural gera uma saída $g(\mathbf{x_i}, \mathbf{w})$, onde $\mathbf{w}$ representa os parâmetros da rede. Então, é possível definir uma função de erro $C$ que mensura a diferença entre $y_i$ (valor/classe real) e $g(\mathbf{x_i}, \mathbf{w})$ (valor/classe predito pela rede). Uma função comumente empregada é o erro quadrático médio. A soma da função de erro para todas as amostras indica o quanto a saída do modelo é diferente dos dados reais. Assim, o processo de aprendizagem pode ser baseado na minimização da função de erro sobre as amostras, resultando numa rede neural que melhor aproxima os dados de entrada.

Existem na literatura diferentes algoritmos que podem ser aplicados nessa minimização do erro, sendo os algoritmos baseados nas derivadas de primeira ordem (gradientes) majoritariamente empregados em redes neurais. Um destes algoritmos é o gradiente descendente *Steepest Gradient Descent*, cuja forma geral, quando aplicada em redes MLP, é dada por

$$w_{ij}^k = w_{ij}^k - \alpha \frac{\partial g(\mathbf{x_i}, \mathbf{w})}{\partial w_{ij}^k}, \tag{4.21}$$

onde os índices adicionais em $\mathbf{w}$, em relação aos índices da expressão para um único neurônio (4.20), se referem ao número da camada $k$ e ao número do neurônio na camada $j$.

Finalmente, no treinamento da rede, aplica-se sucessivamente a regra da cadeia para calcular as derivadas parciais, seguindo a sequência da última para a primeira camada. Este procedimento, que busca ajustar os parâmetros da rede camada-a-camada, é a retropropagação.

Na retropropagação, uma vez que cada camada recebe como informação de sua su-

cessora um gradiente, podem ocorrer dois problemas extremos: (i) pode surgir um gradiente nulo a partir da camada anterior, e assim não há atualização dos parâmetros da camada atual, e estes não contribuem mais para o processo de aprendizagem; (ii) o valor do gradiente pode tender a infinito, fazendo com que os parâmetros passem a flutuar drasticamente sem convergir (divergindo em muitos casos), e deste modo, a rede inteira não aprende.

### 4.5.2 Redes neurais com camadas convolucionais

Uma rede neural pode ser dita convolucional quando emprega ao menos uma camada convolucional, sendo designada com o termo CNN, do inglês *Convolutional Neural Network*. Uma camada convolucional é uma camada especializada em tratar dados que apresentam uma estrutura tipo grade. Existem vários exemplos de dados que apresentam tal estrutura, tais como: séries temporais, que constituem uma grade com *rank 1*, ou imagens, que constituem uma grade com *rank* 2. Pode-se aplicar ou não uma função de ativação à saída da camada convolucional. As redes CNNs foram introduzidas por LeCun et. al. em 1998 em um trabalho sobre reconhecimento de dígitos escritos a mão (LECUN et al., 1998), porém somente após o trabalho de Alex Krizhevsky, em 2012, passaram a dominar as aplicações em visão computacional.

Atualmente, a grande maioria das implementações de redes CNN emprega uma operação de correlação cruzada. Dessa forma, define-se uma camada convolucional aplicada sobre um domínio infinito unidimensional em função de um operador discreto de correlação cruzada $*$, de um conjunto $W$ de filtros (em inglês *kernel*) e de uma entrada $X$, como sendo:

$$(X * W)(i) = \sum_{-\infty}^{u=\infty} x(i+u)w(u), \tag{4.22}$$

O objeto resultante é frequentemente denominado de mapa de atributos, podendo ser interpretado como sendo as características extraídas pela correlação cruzada. Os filtros são os parâmetros a serem aprendidos e, portanto, são análogos aos pesos de uma camada linear. É possível, ainda, adicionar uma *bias* (tendência) à expressão (4.22), reforçando a semelhança com a camada linear.

Na prática, a operação é realizada sobre um domínio de entrada finito, isto é, com tamanho fixo e limitado, organizado na forma de uma grade e constituindo um tensor. Observe que os filtros também são tensores. Denotam-se aqui os tensores por letras maiúsculas, enquanto suas componentes, por letra minúsculas. Assim, $X$ indica o tensor e $x$ indica suas componentes.

Na literatura de aprendizado de máquina é normal se referir a um vetor como um tensor de *rank* 1, uma matriz como um tensor de *rank* 2, um cubóide como um tensor de *rank* 3 (um exemplo deste último seria um vídeo em preto e branco, em que a grade correspondente à imagem, com um único valor em cada ponto de grade, tem um eixo adicional, o tempo). O *shape* ou dimensão de um tensor é definido pelo vetor $(N_1, N_2, ...)$, onde $N_i$ indica o número de componentes para o i-ésimo eixo.

Uma operação de correlação cruzada aplicada num tensor corresponde ao produto escalar de um filtro sobre o tensor de entrada. Como o filtro é um tensor com tamanho de grade menor que o tensor de entrada, essa operação requer uma varredura em que o filtro é transladado sobre a grade do tensor de entrada. No caso de tensores de *rank* 1, tem-se:

$$y(i) = (X * W)(i) = \sum_{m=0}^{M-1} x(i+m)w(m), \tag{4.23}$$

onde $M$ é o número de elementos do filtro, ou tamanho do filtro, e $Y$ a saída, ou mapa de atributos.

As camadas convolucionais apresentam três propriedades importantes:

- Equivariância por translação: a translação da convolução de uma entrada deve ser igual à convolução da translação da entrada, isto é, a saída independe da ordem das operações de translação e convolução. A consequência é que as características extraídas são independentes de sua posição;

- Interações esparsas: em uma camada convolucional, os filtros têm tamanhos muitos menores que a grade formada pelos dados de entrada, ao menos em um eixo, de forma que cada saída conecta-se com um trecho dos elementos da entrada, o que reduz o número de conexões, tal que se diz que a camada convolucional apresenta interações esparsas. Em consequência dessa redução no número de conexões e parâmetros, o custo computacional diminuí;

- Parâmetros compartilhados: em uma camada linear a cada par (entrada, saída) fica associado um parâmetro, utilizado uma única vez. Em uma camada convolucional, um filtro é aplicado para todos os possíveis diferentes trechos de uma entrada, tal que, múltiplas saídas estarão associadas a um mesmo conjunto de parâmetros do filtro, ditos compartilhados. Uma consequência direta é a redução do custo computacional.

Usualmente, emprega-se mais de um filtro, assim a expressão (4.23) fica melhor

caracterizada por

$$y(i, c_o) = (X * W)(i) = \sum_{m=0}^{M-1} \sum_{c_i=0}^{C_i} x(c_i, i + m) w(c_i, c_o, m), \tag{4.24}$$

onde a entrada $X$ é uma série temporal com $C_i$ variáveis e comprimento $T$, e $W$ denota a coleção de filtros indexados por $c_o$. Cada filtro trata uma entrada com $C_i$ variáveis de comprimento $M$.

O comprimento de $Y$ é reduzido pelo comprimento do filtro, pois necessita-se que $x(c_i, i + m)$ seja uma amostra válida, o que requer que $i + m <= T - 1$. Dado que $m$ pode assumir valores de 0 até $M - 1$, tem-se que $i <= T - M$. Lembrando que a contagem inicia em zero, tem-se $T - M + 1$ valores para $i$. Assim, tem-se as seguintes dimensões para o tensor $X$, $W$ e $Y$ respectivamente $(C_i, T)$, $(C_i, C_o, M)$ e $(C_o, T - M + 1)$.

Os tensores de entrada e de saída tem *rank* 2, enquanto $W$ é um tensor de *rank* 3. No contexto dessa formulação, os segundos eixos dos tensores X e Y são denominados de comprimento (interpretados, por exemplo, como o comprimento de uma série temporal), enquanto que os demais eixos são denominados canais (por exemplo, diferentes séries temporais).

A Figura 4.5 apresenta um exemplo de convolução onde $X, W, Y$ apresentam respectivamente as dimensões $(1, 16)$, $(1, 1, 2)$ e $(1, 16 - 2 + 1 = 15)$. Note que descartando os eixos com dimensão unitária, é possível interpretar essa figura como uma operação com tensores de *rank* 1. Acima, foi apresentada uma convolução bem simples, mas nos parágrafos a seguir, a operação convolução será estendida com pela introdução dos conceitos preenchimento, passo, e dilatação.

Em alguns caso, para reduzir mais rapidamente o comprimento do tensor de entrada, aumenta-se o passo (*stride*) em que o tensor de entrada é percorrido para um valor maior que o unitário. Este procedimento pode ser interpretada como uma reamostragem com um aumento do intervalo de coleta dos dados, sendo conhecida como (*downsampling*).

Tomando-se um passo com valor $s$, pode-se definir a seguinte operação de convolução

$$y(i, c_o) = (X * W)(i) = \sum_{m=0}^{M-1} \sum_{c_i=0}^{C_i} x(c_i, i \times s + m) w(c_i, c_o, m), \tag{4.25}$$

o elemento de saída $Y$ terá as seguintes dimensões $(C_o, \lfloor (T-M)/s+1 \rfloor)$, onde a operação $\lfloor x \rfloor$ denota o maior inteiro menor ou igual a $x$.

As Figuras 4.6 e 4.7 apresentam respectivamente exemplos de convolução com passo de valores 2 e 3, note que o comprimento de elemento de saída é drasticamente reduzido para $\lfloor (16-2)/2+1 \rfloor = 8$ e $\lfloor (16-2)/3+1 \rfloor = 6$, respectivamente

Uma segunda modificação que pode ser feita na operação de convolução é a introdução de dilatação *dilation*), a qual constitui na inserção de um número $l$ de zeros entre os elementos do filtro. Por exemplo, considere um filtro $W$ com dimensões $(1,1,3)$, dado por $[1,-1,1]$, este filtro apresenta dilatação $l=1$, para $l=2$, esse filtro pode ser visto como $[1,0,-1,0,1]$ e para $l=3$ como $[1,0,0,-1,0,0,1]$. A convolução com dilatação é expressa por

$$y(i,c_o) = (X * W)(i) = \sum_{m=0}^{M-1} \sum_{c_i=0}^{C_i} x(c_i, i+l\times m) w(c_i, c_o, m). \tag{4.26}$$

Dado que $i + l \times m <= T-1$ e $m$ assume valores no conjunto $\{0,..,M-1\}$, tem-se que o valor máximo de $i$ fica limitado por $i <= T-1-l\times(M-1)$, lembrando que a contagem inicia em zero concluí-se que a saída $Y$ terá as dimensões $(C_o, T-l\times(M-1))$.

A sucessiva aplicação de camadas convolucionais leva a uma redução no comprimento do tensor, o que define um limite no número possível de camadas, pois eventualmente chega-se a uma configuração em que o comprimento será 1. Em muitos casos, o limite mínimo de camadas pode não ser suficiente para que um conjunto de dados seja modelado, sendo necessário preservar o comprimento do tensor.

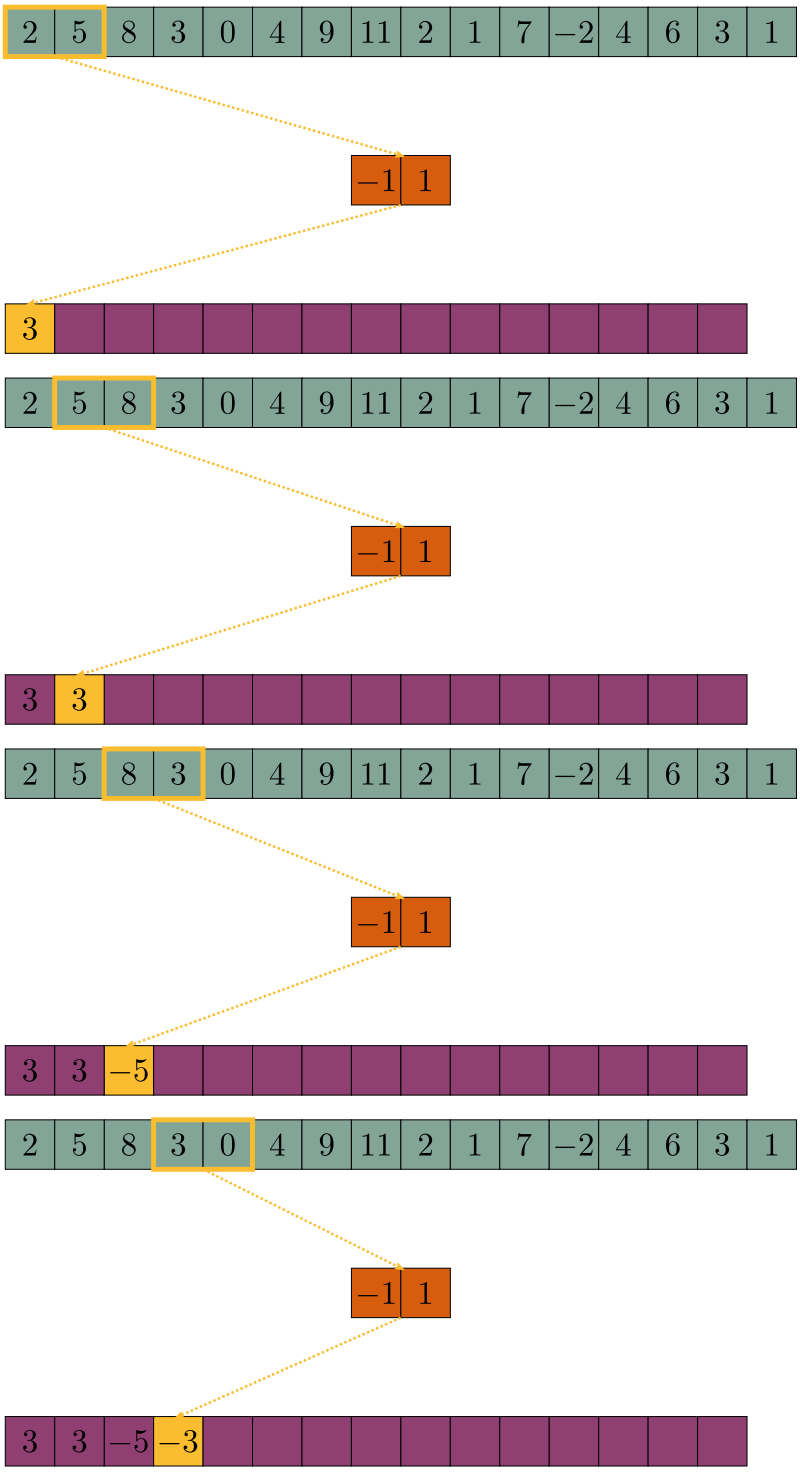
O preenchimento (do inglês *padding*) fornece um mecanismo para preservar o comprimento de um tensor, adicionando-se elementos à esquerda e à direita em $X$ tal que a dimensão fique dada por $(C_i, p_e + T + p_d)$, onde $p_e$ e $p_d$ denotam respectivamente o número de elementos de preenchimento adicionados à esquerda e à direita. A forma mais comum de preenchimento utiliza a inserção de zeros de forma a manter constante o comprimento do tensor, preservando assim seu comprimento original.

Combinando-se o conceito de passo, dilatação e preenchimento, a operação de convolução pode então ser expressa por:

$$y(i,c_o) = (X * W)(i) = \sum_{m=0}^{M} \sum_{c_i=0}^{C_i} x(c_i, i\times s + l\times m) w(c_i, c_o, m), \tag{4.27}$$

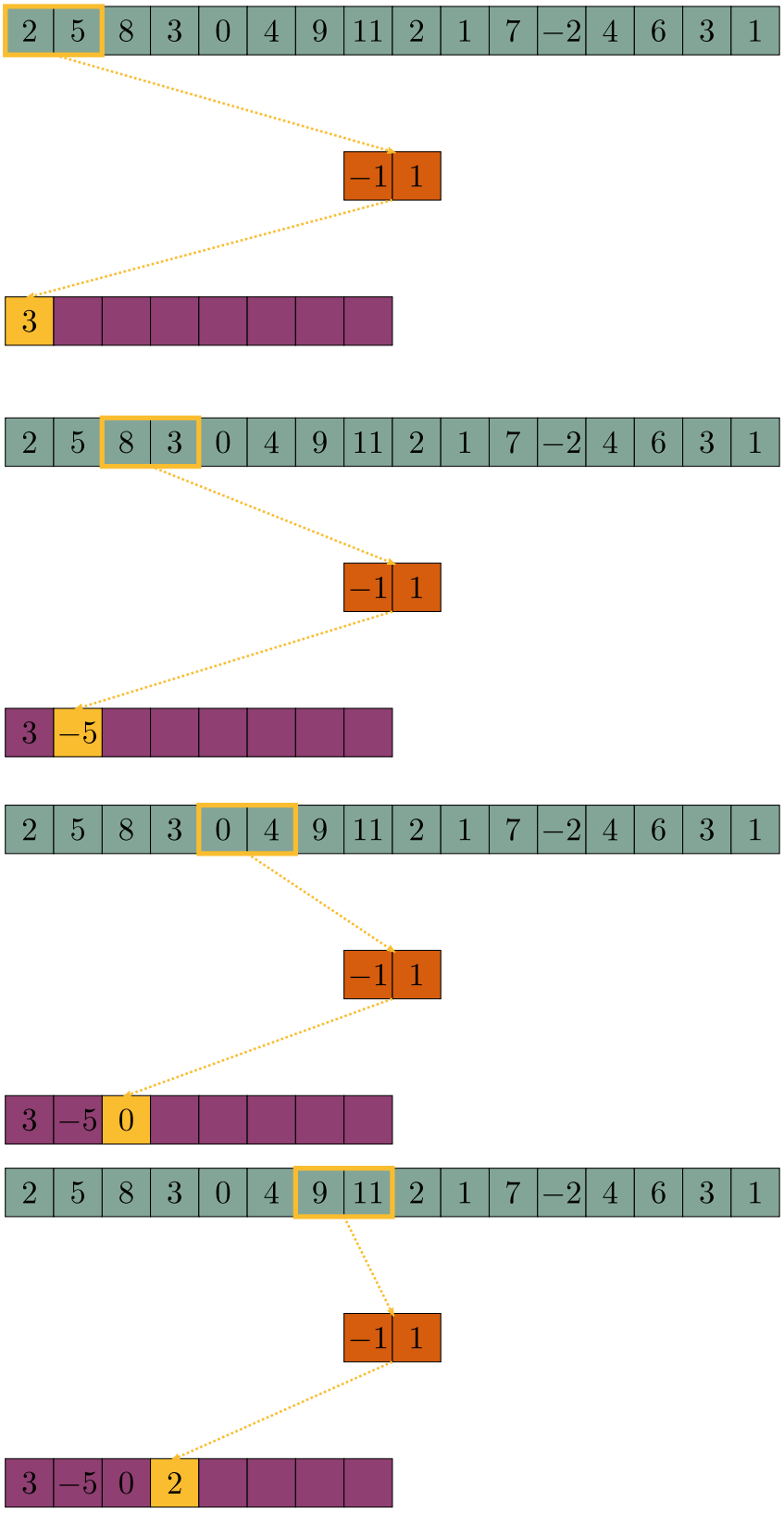
onde $X$, $W$ e $Y$ tem respectivamente as seguintes dimensões $(C_i, p_e + T + p_d)$, $(C_i, C_o, M)$ e $(C_o, \lfloor[(p_e + T + p_d) - l \times (M-1) - 1]/s + 1\rfloor)$.

Figura 4.5 - Exemplo de convolução com um filtro de comprimento 2.



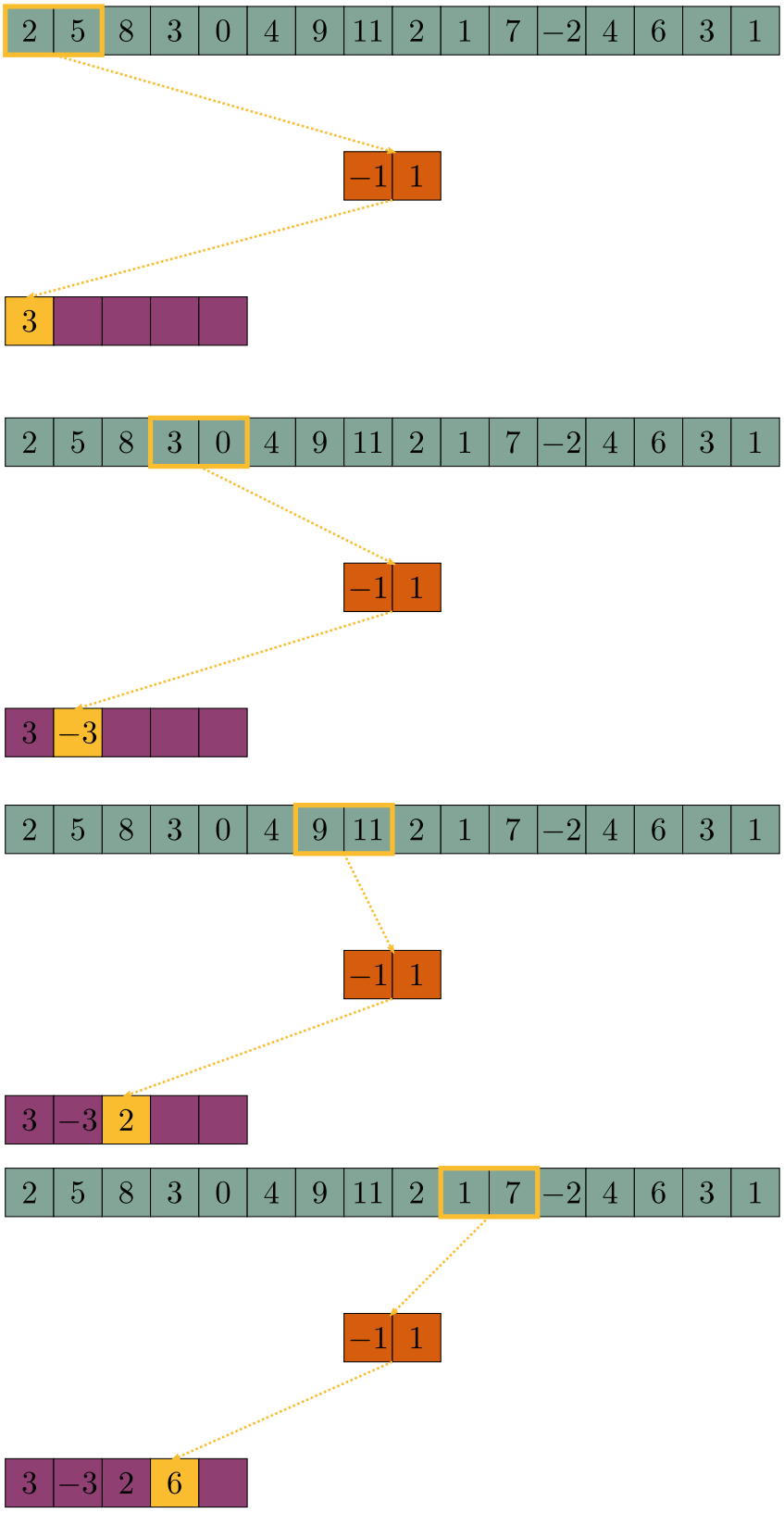

Fonte: Produção do autor.

Figura 4.6 - Exemplo de convolução com um filtro de comprimento de 2 e *stride* 2.



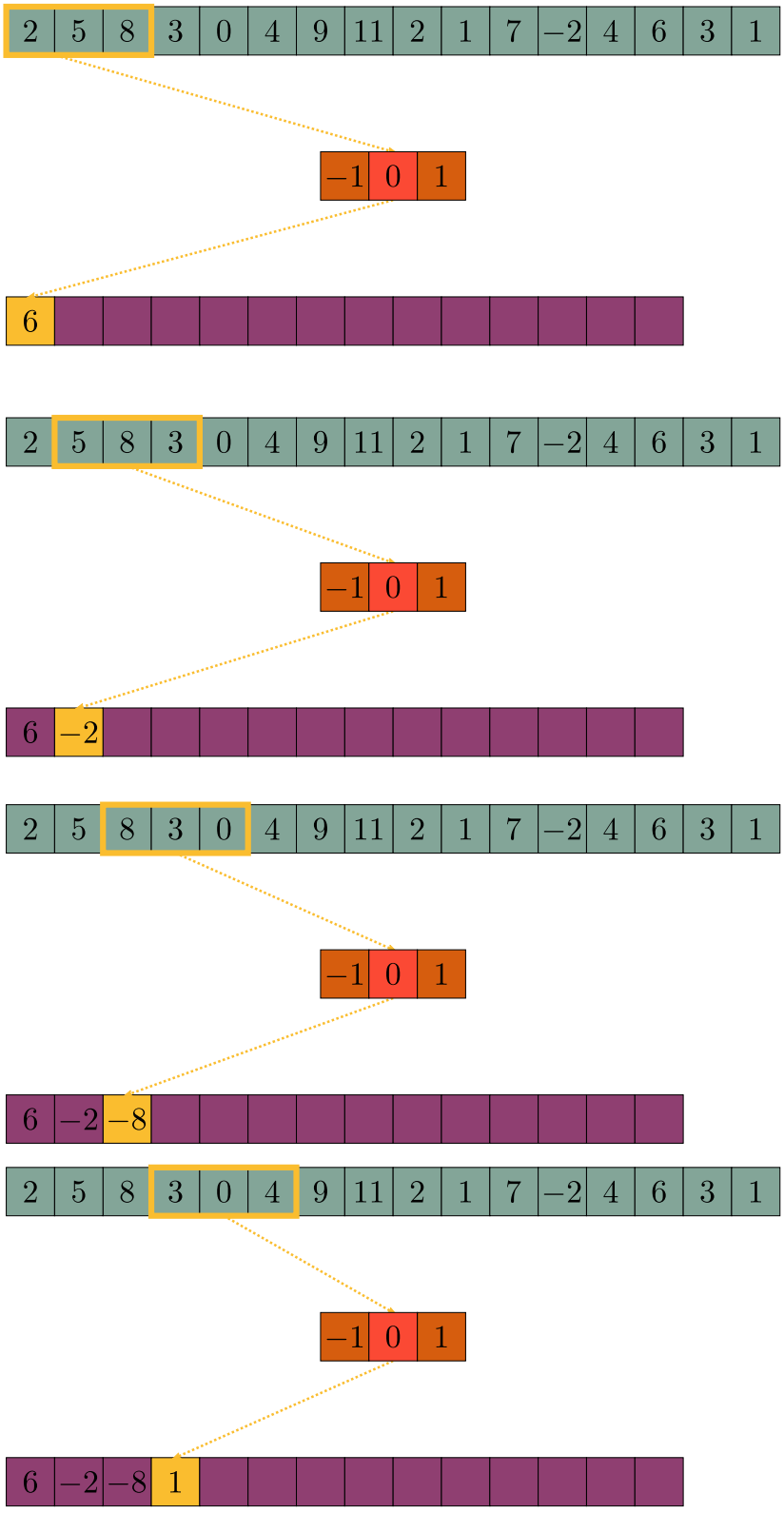

Fonte: Produção do autor.

Figura 4.7 - Exemplo de convolução com um filtro e comprimento de 2 e *stride* 3.



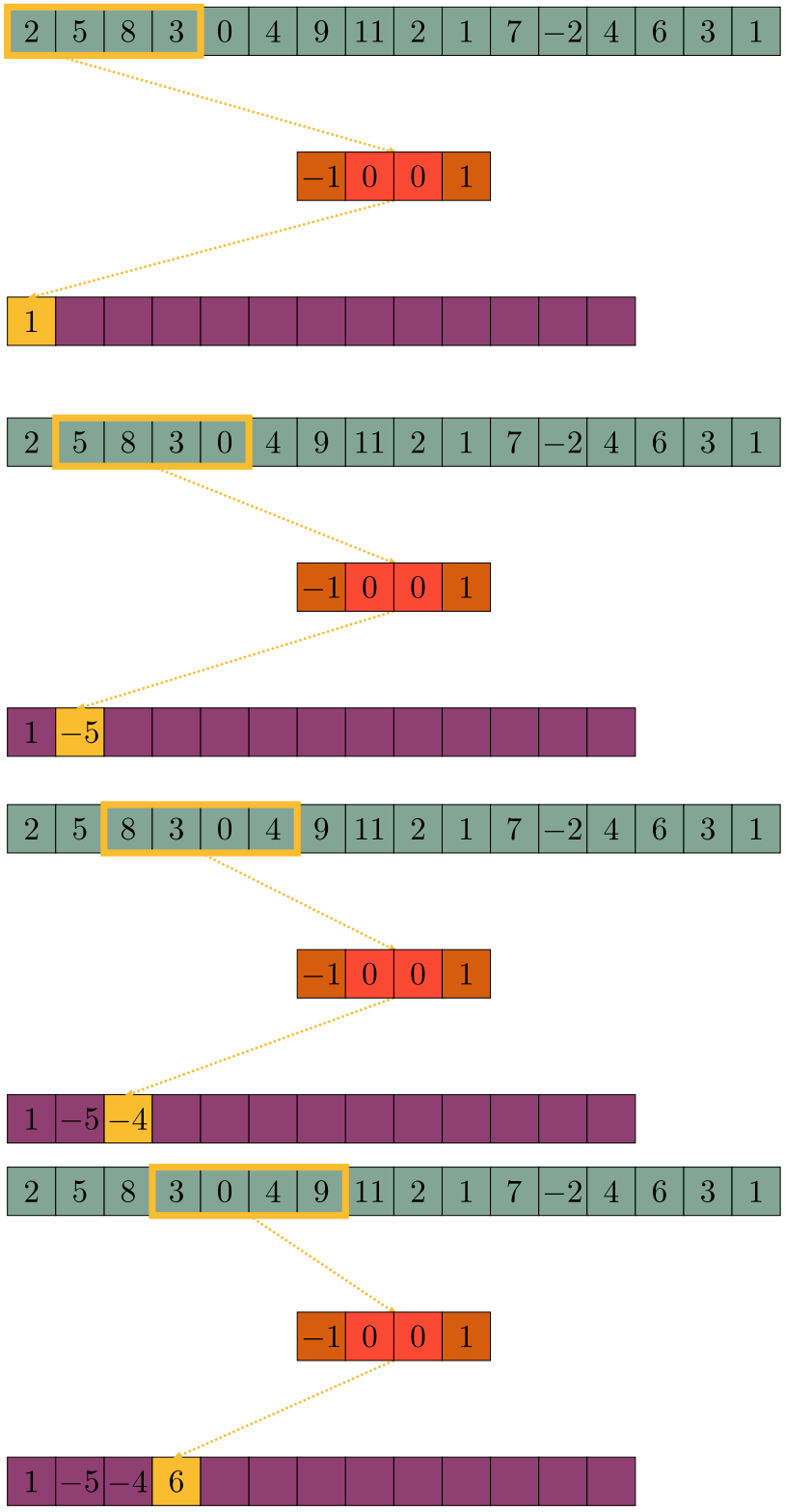

Fonte: Produção do autor.

Figura 4.8 - Exemplo de convolução com um filtro e comprimento de tamanho 2 e dilatação igual à 2.




Fonte: Produção do autor.

Figura 4.9 - Exemplo de convolução com um filtro e comprimento de tamanho 2 e dilatação igual à 3.




Fonte: Produção do autor.

### 4.5.3 Funções de ativação

As camadas lineares e convolucionais podem ser interpretadas como operações matriciais, podendo portanto realizar somente transformações afins em suas entradas, o que as impede de modelar comportamentos não-lineares. Isto é contornado pela introdução das funções de ativação, que são funções não-lineares aplicadas entre as camadas de forma a permitir que uma rede neural produza combinações não-lineares entre os elementos de entrada.

A escolha da função de ativação influi muito no tempo de treinamento, e no desempenho do modelo. As funções de ativação propostas recentemente apresentam desempenho similares.

- **Sigmoide:** A função de ativação sigmoide foi introduzida como uma aproximação da curva de resposta de um neurônio biológico, sendo largamente aplicada e utilizada em redes neurais até 2012; atualmente, é utilizada de maneira composta com outras funções de ativação, juntamente com diferentes operações lineares, permitindo construir assim elementos computacionais mais complexos que o neurônio Perceptron. Esta função também é utilizada na saída de redes neurais para problemas de classificação binária, gerando uma saída no intervalo $[0, 1]$, a qual será interpretado como a probabilidade de um elemento pertencer a uma classe ou não.

  A expressão da função sigmoide é dada por

  $$a_{sigmoide}(x) = \frac{1}{1 + e^{-x}}. \tag{4.28}$$

  Uma grande desvantagem da função sigmoide é sua suscetibilidade ao problema de o gradiente tender a zero (*vanishing gradient*), quando a derivada da função sigmoide é muito próxima do zero para todos os valores onde $|x| > 4$, como observado na Figura 4.10. Isto implica que altos valores na saída da ativação resultam em retropropagação nula do gradiente, e portanto implica na não atualização dos pesos, caracterizando um ou mais neurônios da camada ditos mortos. No caso de uma CNN, a consequência é muito pior, visto que um canal inteiro será dito morto.

- **GELU:** As primeiras redes neurais utilizavam um função de ativação binária. A função sigmoide introduziu uma aproximação suave o que permitiu o treinamento por retropropagação. Entretanto, o aumento do número de camadas da rede, o qual eventualmente causa *vanishing gradient* levou

à introdução de novas funções de ativação como a RELU, definida como $max(0, x)$, em 2010, (NAIR; HINTON, 2010). Essa função de ativação permitiu grandes avanços nas redes neurais, mas também apresenta problemas, como ocorrência de gradiente nulo para valores de entrada negativos da função. O tratamento deste problema levou ao surgimento de família RELU de funções. A função de ativação RELU ajusta muito bem os dados, mas frequentemente causa sobreajuste. Em geral, este problema é tratado pela introdução de ruído entre as camadas, porém isto aumenta a dificuldade de se definir a arquitetura da rede.

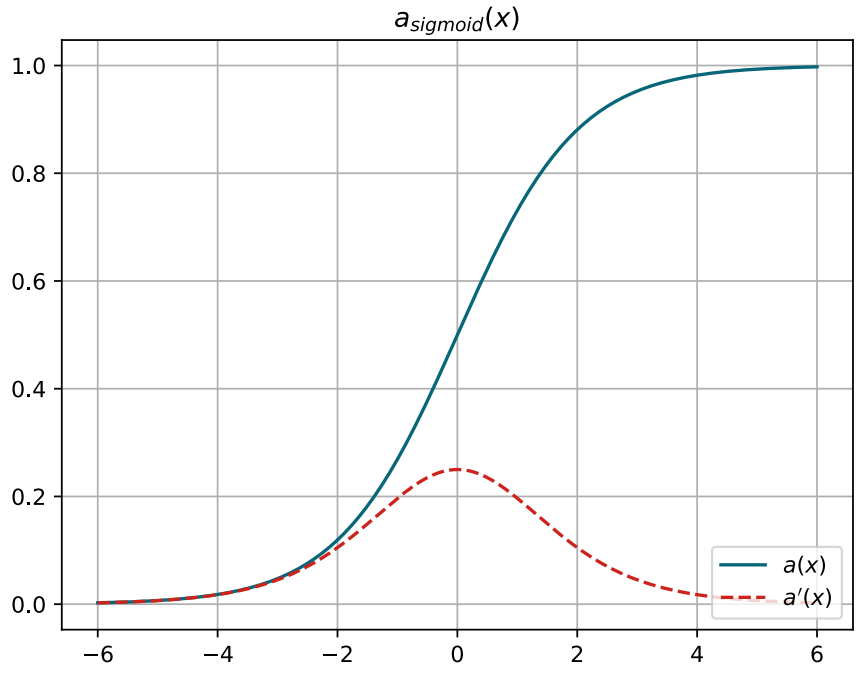
Consequentemente Hendrycks e Gimpel (2016) então introduziram a função de ativação GELU, que é membro da família RELU, aproximada por

$$a_{gelu}(x) \approx 0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)]), \tag{4.29}$$

essa função de ativação vem apresentando bons resultados em várias aplicações, eliminando a necessidade de introdução de ruído.

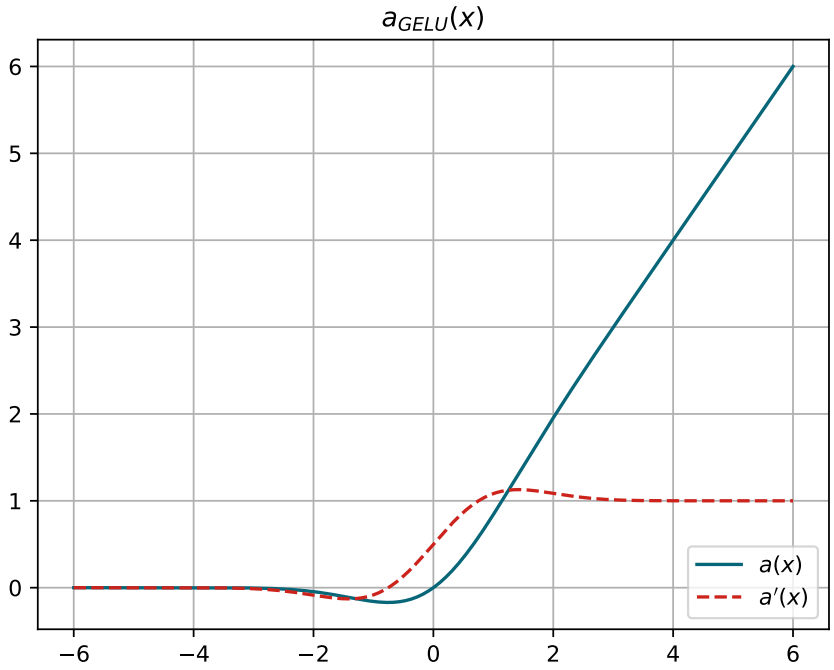
A Figura 4.11 ilustra a função de ativação GELU.

Figura 4.10 - Ilustração da função de ativação Sigmoide.




Fonte: Produção do autor.

Figura 4.11 - Ilustração da função de ativação GELU.




Fonte: Produção do autor.

### 4.5.4 Camada de normalização em *batch*

O termo *batch* é utilizado em aprendizagem de máquina, particularmente, no contexto de redes neurais, e indica o número de amostras a ser utilizado em cada iteração da fase de treinamento. O total de iterações é dado então pelo número total de amostras dividido pelo tamanho do *batch*, compondo o que se denomina uma "época do treinamento da rede.

Uma camada de normalização em *batch*, (IOFFE; SZEGEDY, 2016), aplica uma transformação nos dados de entrada e nos parâmetros dessa camada de forma que os dados de saída são mapeados para um intervalo com aproximadamente média nula e variância unitária. Em muitos casos é necessário alterar o comprimento e a localização do intervalo por uma transformação afim, que consiste numa multiplicação por um valor $a$, para escalar, e na soma de um valor $b$, para deslocar. Esses valores $a$ e $b$ são aprendidos pela rede. A camada de normalização em *batch* é implementada pelos seguintes passos:

- Cálculo do valor médio para o *batch*, pela expressão

$$\mu_{c_i} = \frac{1}{NT} \sum_{n=0}^{N-1} \sum_{l=0}^{L} x(n, c_i, l), \tag{4.30}$$

onde $\mu$ é um tensor de dimensão $(C_i)$ e $X$ um tensor de dimensão $(N, C_i, L)$, onde $N$ é o número de elementos no *batch* e $L$ o comprimento.

- Cálculo do desvio padrão para o *batch*, pela expressão

$$\sigma_{c_i} = \sqrt{\frac{1}{NT} \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} [x(n, c_i, l) - \mu_{c_i}]^2}. \tag{4.31}$$

- Normalização, pela expressão

$$\hat{x}_{n,c_i,l} = \frac{x(n, c_i, l) - \mu_{c_i}}{\sigma + \epsilon}, \tag{4.32}$$

onde o termo $\epsilon$ é um valor pequeno adicionado de forma a evitar divisão por zero.

- Transformação afim, multiplicação pelo termo de escala e adição do termo de deslocamento, dada pela expressão

$$y_{n,c_i,l} = a\hat{x}_{n,c_i,l} + b, \tag{4.33}$$

onde $Y$ é o tensor de saída como o mesmo tamanho do tensor $X$.

Os objetivos de se adicionar uma camada de normalização em *batch* é reduzir o tempo de treinamento e melhorar o desempenho de generalização. A padronização das variáveis num dado intervalo permite reduzir saturações nas funções de ativação, isto é, intervalos onde as funções de ativação apresentam valores de gradientes muito pequenos, além de causar indiretamente uma padronização no gradiente a ser retropropagado, tornando-o mais efetivo no processo de treinamento.

### 4.5.5 Conexão salto

Conforme o número de camadas de uma rede neural aumenta, o número teórico de funções que consegue aproximar aumenta. Entretanto, na prática, isso não ocorre pois esse aumento no número de camadas dificulta o treinamento, ocasionando eventualmente gradientes não-nulos, porém negligenciáveis nas camadas iniciais da rede,

impedindo-a de aprender. Assim, He et al. (2016) introduziram uma rede neural chamada ResNet, que introduz a técnica de conexão salto.

Esta técnica se baseia em adicionar a saída final de uma camada à camada de entrada da rede rede por meio de uma operação de identidade. Essa saída final da camada resulta de suas várias operações lineares e também, nas arquiteturas mais recentes, não lineares. Isso introduz um novo caminho para o fluxo de gradiente durante o treinamento, permitindo treinar redes mais profundas. Anteriormente, o treinamento dessas redes exigia o uso de certos estratagemas.

Matematicamente, a conexão de salto pode ser expressa por:

$$g(\mathbf{x})_{res} = F(\mathbf{x}) + \mathbf{x}, \tag{4.34}$$

onde o sobrescrito $res$ é um indicativo do termo resíduo, definido na conexão salto pela semelhança com o cálculo de resíduos em otimização; $F(\mathbf{x})$ representa um conjunto de operações lineares e não lineares realizadas. Uma limitação desta abordagem é que $F(\mathbf{x})$ deve apresentar a mesma dimensão que o elemento de entrada $\mathbf{x}$. Existem variações desta técnica, tal como a concatenação de $F(\mathbf{x})$ com $\mathbf{x}$, expressa por

$$g(\mathbf{x})_{res} = [F(\mathbf{x}), \mathbf{x}], \tag{4.35}$$

onde os colchetes denotam a operação de concatenação. Nesta abordagem a restrição de dimensão é mais fraca.

O conjunto das várias operações $F(\mathbf{x})$, referentes a uma dada camada, é definido como "bloco', sendo apresentado no trabalho de He et al. (2016) na forma:

$$\begin{aligned} z &= conv_1(x) \\ z &= a(x) \\ z &= conv_2(x) \\ z &= a(z + I * x)\text{x} \end{aligned}$$

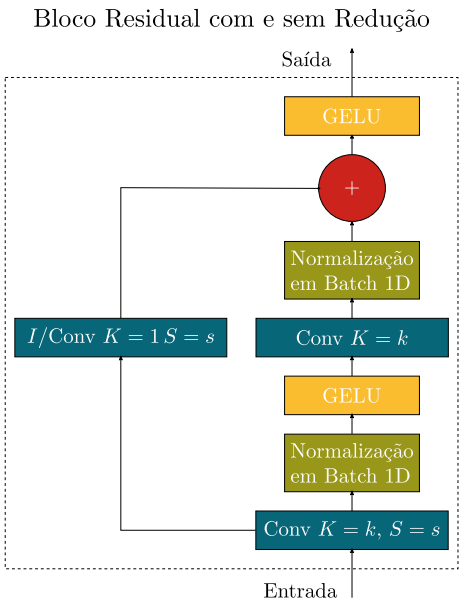
onde $conv_i$ denota a operação de convolução (seus parâmetros ficam implícitos, isto é, $conv_i$ terá os parâmetros $w_i$), $a$ é a função de ativação e $I$ é a identidade. É usual se referir a essa conexão salto como um curto-circuito.

A Figura 4.12 apresenta a arquitetura do bloco residual tal como definido para compor a ResNet, denominado de Bloco Residual com/sem Redução. $K$ denota o

comprimento do filtro e $S$ o tamanho do passo, instanciados com valores numéricos pelas correspondentes letras minúsculas. No caso, "com/sem redução" indica passo igual a 1 ou menor que 1, respectivamente, correspondendo ao elemento de saída ser igual ou menor que seu comprimento de entrada.

As principais diferenças deste bloco da ResNet em relação à versão apresentada por He et. al. são: (i) presença da camada de normalização em *batch*, (ii) troca da função de ativação RELU pela GELU e (iii) adição de uma operação de convolução no caminho de curto-circuito. No caso desta última, se as dimensões do elemento de entrada for igual à dimensão após a segunda camada de normalização em *batch*, assume-se esta operação como a identidade. Caso contrário, a operação ajusta a dimensão do elemento de entrada para que seja igual àquela.

Figura 4.12 - Arquitetura do bloco residual utilizado neste trabalho, em que $K$ representa o comprimento do filtro e $s$ o tamanho do passo.

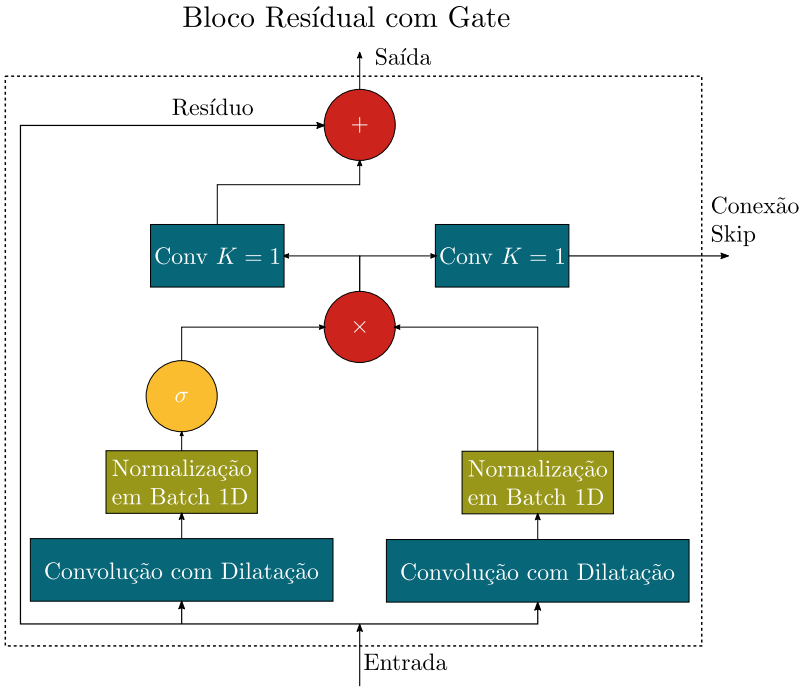



Fonte: Produção do autor.

É possível configurar mais de uma conexão salto no mesmo bloco, como exemplificado na Figura 4.13, onde há uma conexão denominada de resíduo semelhante à abordada acima, e há também uma conexão salto saindo à direita que se concatenada com ela mesma, mas é proveniente de outros blocos, e assim constituem um bloco maior. Esta variação é denominado de Bloco Residual com *Gate*, assim chamada pois a função de ativação *sigmoid*, indicada pelo círculo amarelo com a letra grega $\sigma$, funciona como uma porta que define o quanto de informação do sinal de entrada deve ser propagado para a saída. Esta variação foi introduzida por Oord et.al.

(OORD et al., 2016) para geração de sinais de áudio, particularmente voz humana, sendo adotada neste trabalho. A versão apresentada na Figura 4.13 já se refere a esta variação.

Figura 4.13 - Bloco residual com duas conexões salto.




Fonte: Produção do autor.

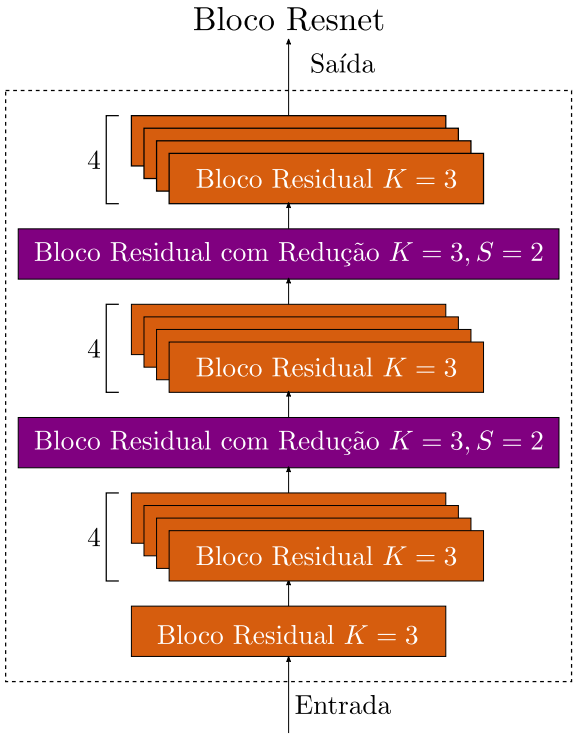
O Bloco Residual com *Gate* apresenta uma outra diferença em relação ao Bloco Residual com/sem Redução que é a utilização de convoluções com dilatação diferente de zero, no lugar das convoluções com passo diferente de 1. Embora ambas as operações sirvam para tratar dependências temporais de longo alcance, a dilatação trata melhor os detalhes do sinal de entrada, enquanto que a convolução consegue extrair mais facilmente características gerais do sinal, em termos de reamostragem.

### 4.5.6 Rede neural proposta

Neste trabalho, foi explorada a possibilidade de combinar vários blocos menores formando blocos maiores. A concatenação sequencial de vários Blocos Residuais com/sem Redução resulta no Bloco Resnet proposto, ilustrado na Figura 4.14. Este bloco constituído por: (i) um bloco residual com $K = 3$ e passo 1, utilizado para restringir número de canais usando um limite especificado; (ii) 3 grupos de blocos residuais com $K = 3$ para extrair características do sinal da camada anterior, tratando-o para a camada posterior; (iii) 2 blocos residuais com redução, neste caso

com $K = 3$ e passo $S = 2$, reduzindo o comprimento do sinal será reduzido pela metade em cada um destes blocos, de forma a codificar o sinal numa representação mais compacta, extraindo informações deste.

Figura 4.14 - Bloco residual com duas conexões salto.




Fonte: Produção do autor.

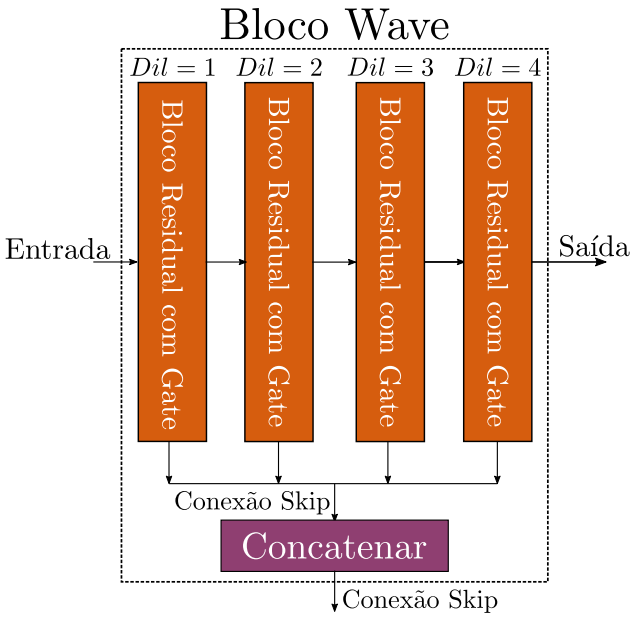
A concatenação sequencial de Blocos Residuais com *Gate* com diferentes valores de dilatação origina o chamado Bloco *Wave* Figura 4.15, o qual concatenado sequencialmente origina o Bloco *Wavenet* Figura 4.16. Ambos foram propostos por Oord et.al., mas foram utilizados de maneira diferente neste trabalho. A aplicação de diferentes valores de dilatação permite avaliar um sinal em diferentes valores de escala e, portanto, capturar padrões que ocorrem em períodos e frequências diferentes.

Neste trabalho, a rede neural proposta, ilustrado na Figura 4.17, resulta da combinação dos Blocos Wavenet e Resnet, pois ambos apresentam características específicas e desejáveis para a extração de características das séries temporais.

Além destes blocos, a rede neural proposta adiciona: (i) a operação *Flatten* para redução dimensional (por exemplo, transformando uma matriz num vetor composto por todos seus elementos; (ii) a operação de concatenação; (iii) uma camada linear

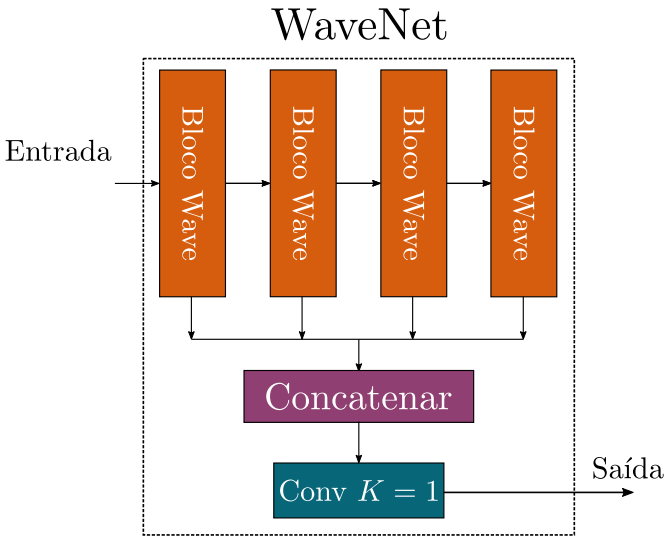
inicial para combinar as características extraídas pelos blocos Wavenet e Resnet; (iv) a função de ativação GELU; (iv) uma função linear na saída da rede que gera o sinal de entrada da função sigmoid, a qual gera um valor no intervalo $[0, 1]$, que pode ser interpretado como uma probabilidade.

Figura 4.15 - Concatenação sequencial de Blocos Residuais com *Gate* com valores de dilatação aumentando linearmente com o número de camadas (o valor da dilatação é indicado pelo termo *Dil*).



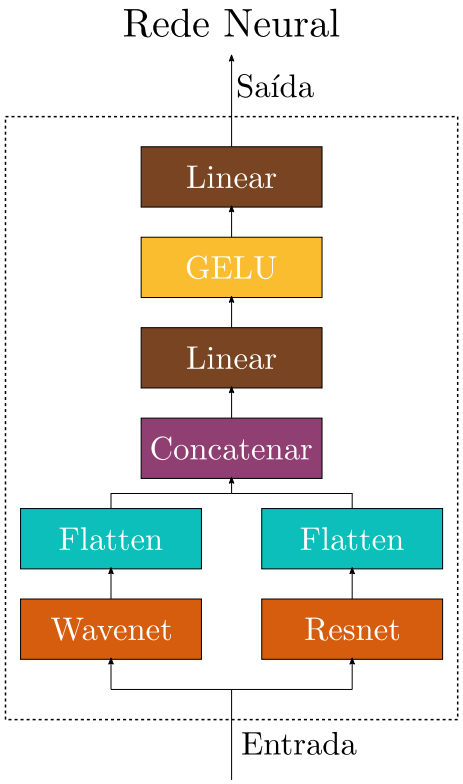

Fonte: Produção do autor.

Figura 4.16 - Múltiplos Blocos Wave agrupados de forma a gerar um Bloco Wavenet.




Fonte: Produção do autor.

Figura 4.17 - Arquitetura da rede neural utilizada neste trabalho.




Fonte: Produção do autor.

# 5 DESCRIÇÃO DOS EXPERIMENTOS REALIZADOS

No trabalho relacionado a esta tese, foram avaliadas diversas abordagens baseadas em aprendizado de máquina para a predição de cintilação ionosférica proposta. Cada abordagem é definida pelo algoritmo de aprendizado de máquina e pelo esquema de particionamento/validação dos dados. Nessa avaliação, essas diversas abordagens, aplicadas a diferentes dados ou períodos de dados, resultaram em diversos experimentos, que foram bem ou mal sucedidos. Apresentam-se aqui os três experimentos melhor sucedidos em termos de desempenho de predição. Os demais experimentos foram omitidos deste texto, por brevidade.

Os três experimentos selecionados tem suas eventuais variantes, conforme (i) o algoritmo de aprendizado de máquina utilizado, (ii) os esquemas de particionamento dos dados de entrada, (iii) os períodos abrangidos, (iv) a classe de cintilação a ser predita, e (v) sua resolução temporal. A predição é sempre local, para São José dos Campos SP, e os dados utilizados são do período 2010-2018.

- **Experimento A (GTSH/TSCV-GKF) anos 2010-2018** (Seção 5.5): predição pelos algoritmos de *ensembles* XGBoost e CatBoost, com o vetor de atributos gerado a partir de séries temporais de dias e sem/com seleção de atributos (NSA/SSA) utilizando validação com *Gap Time Series Holdout* (GTSH) ou *Time Series Cross Validation Gap K Fold* (TSCV-GKF) para predições de 30-60-90-120-150-180 minutos ao longo de uma noite, ou seja, 6 variações de modelo/preditor para cada algoritmo. Inicialmente foram realizados testes com as variáveis: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $F10.7$, $Sunspot$, $TEC$, $S_4$ e posteriormente se adicionou a variável $h'F$. Este experimento apresenta duas variantes, descritas na mesma seção:
  - **Experimento A (GTSH) anos 2012-2014**: esta variante tem metodologia idêntica ao Experimento A descrito acima, exceto pelo período abrangido menor e o esquema de particionamento/validação único.
  - **Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018**: esta variante também emprega a mesma metodologia, exceto por não utilizar seleção de atributos, e por utilizar dois níveis de validação cruzada, ambos empregando ou o esquema *Time Series Cross Validation Gap K-Fold* (TSCV-GKF), ou então o

esquema *Time Series Cross Validation Walk Forward* (TSCV-GWF).

- **Experimento B (GTSH/VCT) anos 2010-2018** (Seção 5.6): predição pelos algoritmos de *ensemble* XGBoost e CatBoost com o vetor de atributos gerado pelo algoritmo de Weasel-Muse a partir de séries temporais de atributos de predição de um mesmo dia utilizando validação com *Gap Time Series Holdout* (GTSH) ou Validação Cruzada Tradicional (VCT) para predição de um valor único para a noite do mesmo dia. Diversos modelos foram gerados, utilizando somente o TEC futuro e tamanho de palavra 8, ou então, com tamanho de palavra 4 utilizando as seguintes variáveis individualmente ou em conjunto: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $TEC$, $S_4$, $h'F$.

- **Experimento C (GTSH) anos 2010-2018** (Seção 5.7): predição por rede neural convolucional com séries temporais de 2 dias dos atributos de predição utilizando validação com *Gap Time Series Holdout* (GTSH) para antecedências de 30-60-90-120-150-180 minutos ao longo de uma noite, ou seja, gerando 6 modelos/preditores baseados na mesma rede neural, mas treinados para antecedências diferentes. Inicialmente foram realizados testes com as variáveis: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $F10.7$, $Sunspot$, $TEC$, $S_4$ e posteriormente se adicionou a variável $h'F$.

O valor do índice de cintilação $S_4$ foi discretizado e mapeado nas classes nula ($S_4 < 0,2$), fraca ($0,2 \leq S_4 < 0,4$), moderada ($0,4 \leq S_4 < 0,6$) e forte ($S_4 \geq 0,6$). Assim, as predições são categóricas, correspondentes à ocorrência (OC) ou não-ocorrência (N-OC) de cintilação. Entretanto, segundo o teste, essas duas classes podem ser definidas de três maneiras:

- OC forte (N-OC corresponde a cintilação nula, fraca ou moderada);
- OC forte-moderada (N-OC corresponde a cintilação nula ou fraca);
- OC forte-moderada-fraca (N-OC corresponde unicamente a cintilação nula).

O desempenho de um modelo de predição orientado a dados depende da qualidade e quantidade dos dados que compõem os subconjuntos de treinamento, validação e teste. É desejável que esses subconjuntos sejam balanceados, i.e. que contenham

proporções similares de amostras de cada classe. Todavia, tipicamente nos dados disponíveis, as amostras das classes moderada e forte representavam apenas uma pequena parcela do total.

Os algoritmos empregados ao longo do trabalho utilizam uma semente aleatória com valor zero, embora em alguns casos, foram utilizados 25 sementes diferentes, sendo o desempenho de predição avaliado pela média e desvio padrão das métricas adotadas.

Descrevem-se nas seções seguintes, o ambiente computacional, os esquemas de pré-processamento dos dados, de extração de atributos das séries temporais, a geração de mapas do índice $S_4$ e os três experimentos realizados.

## 5.1 Ambiente de programação e infraestrutura computacional

Nesta tese foi utilizado o ambiente de programação Python e as bibliotecas associadas (todas gratuitas), composta pelos itens a seguir, os quais suportam esse ambiente de programação. A infraestrutura computacional utilizada foi:

a) Microcomputador pessoal (PC) com 96 GB de memória principal DDR4, processador AMD Ryzen i3 com quatro núcleos físicos, 960 GB de memória secundária de estado sólido, 2 placas gráficas Nvidia RTX 2070 com 8 GB cada; esse PC pertence ao autor, mas teve sua configuração melhorada com recursos da taxa de bancada da bolsa de doutorado do CNPq;

b) Supercomputador Santos Dumont do MCTIC/LNCC [1] no escopo do projeto "Implementação de abordagens de aprendizado de máquina que demandam supercomputação para previsão de cintilação ionosférica e previsão meteorológica de eventos convectivos severos";

c) *Cluster* de alto desempenho da Ciência Espacial do INPE (CEA/INPE) [2];

d) *Storage* Asus da DIHPA/INPE com 100 TB de capacidade e que armazena dados de cintilação, bem como às demais facilidades de TI desta divisão; o que permite acessar *online* uma réplica do banco de dados de cintilação ionosférica da UNESP Presidente Prudente.

Os Experimentos propostos tem alta carga computacional requerendo portanto recursos de processamento de alto desempenho, seja por meio de nós de memória com-

[1] https://sdumont.lncc.br/machine.php?pg=machine
[2] http://mtc-m21c.sid.inpe.br/col/sid.inpe.br/mtc-m21c/2018/04.23.19.21/doc/publicacao.pdf

partilhada com processadores multi-núcleo, seja por meio de aceleradores de processamento, no caso, placas aceleradoras gráficas (*General Purpose Graphics Processing Unit - GPGPU ou simplesmente GPU*). Alguns testes podem demandar várias semanas de processamento, dependendo do ambiente computacional considerado. Como exemplo, seguem algumas valores de tempo de execução obtidos utilizando o PC mencionado no primeiro item da lista acima.

- Experimento A (GTSH/TSCV-GKF) anos 2010-2018 (Seção 5.5): aproximadamente 24 horas, considerando as 6 antecedências de predição juntas;
- Experimento A (GTSH) anos 2012-2014: aproximadamente 12 horas, considerando as 6 antecedências de predição juntas;
- Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018: aproximadamente 72 horas, considerando as 6 antecedências de predição juntas;
- Experimento B (GTSH/VCT) anos 2010-2018: de uma a duas horas dependendo do conjunto de variáveis selecionados para cada modelo;
- Experimento C (GTSH) anos 2010-2018: de 24 horas a 72 horas, para cada antecedência de predição, a depender da estação do ano (verão ou todas) e do conjunto de variáveis selecionado.

## 5.2 Esquemas de pré-processamento dos dados utilizados

Um particionamento geral de GTSH com subconjuntos de treinamento, teste e validação foi adotado para os três experimentos. Exceto quando explicitamente indicado, o subconjunto de teste é o mesmo, de forma a melhor comparar os experimentos. Nos casos particulares em que foi empregada validação cruzada, os subconjuntos de treinamento e validação foram unidos num único subconjunto, sendo posteriormente particionados segundo o esquema de validação.

Nos experimentos realizados se utilizaram algumas técnicas de pré-processamento comuns a todos e outras específicas a um experimento. As técnicas comuns segundo sua ordem de aplicação foram:

- Geração de séries temporais de mapas de $S_4$ a partir de dados brutos, a ser discutido na próxima subseção;

- Geração de séries temporais de mapas de TEC pelo método do EMBRACE/INPE (OTSUKA et al., 2002; CARMO, 2018), com resolução de 10 minutos;

- Remoção de valores negativos nos mapas de TEC, eventualmente gerados pelo modelo específico utilizado para calcular o TEC absoluto; valores negativos de TEC não tem significado físico, sendo gerados erroneamente pelo modelo e foram substituídos pelo quantil 5 % dos valores não negativos do mapa;

- Extração do valor de TEC e $S_4$ correspondentes ao ponto de grade de São José dos Campos (SJC) nos mapas de TEC e de $S_4$ de forma a gerar séries temporais desses valores;

- Re-amostragem das séries temporais de TEC, $S_4$ e demais variáveis coletadas para SJC, utilizando a operação de máximo, para resolução temporal de 30 min;

- Substituição dos valores ausentes dessas séries temporais pelo último valor válido conhecido;

- Normalização dos valores dessas séries temporais para o intervalo [0, 1].

As técnicas específicas tratam: (i) a extração de atributos das séries temporais das variáveis utilizadas no estudo; (ii) o pré-processamento dos vetores de atributos; (iii) o balanceamento do subconjunto de treinamento por meio da geração de amostras sintéticas.

## 5.3 Extração e processamento de atributos das séries temporais

a) Extração de atributos utilizando TSFRESH e TSFEL, aplicadas para os seguintes tamanhos de janela: 3 horas, 6 horas, 12 horas, 1 dia, 2 dias e 3 dias; uma janela de tamanho $a$ é definida como o intervalo $[t - a, t]$, e a aplicação de uma função característica em uma janela significa aplicá-la numa série temporal definida, considerada como uma sequência ordenada de elementos no intervalo referente à janela;

b) Cálculo das derivadas de cada série temporal, as quais serão adicionadas ao vetor de atributos $x_t$ utilizado apenas os 7 últimos elementos (incluindo $t$) para cada vetor, isto é, os elementos de $t - 6$ a $t$.

c) Cálculo das seguintes razões não lineares entre séries temporais: $v_i/(v_j + 0.001)$, $v_j/(v_i + 0.001)$, $v_i/(v_j v_l + 0.001)$; adicionam-se somente os últimos 7 elementos dessas razões ao vetor de atributos $x_t$;

d) Cálculo do sin e cos de $(2\pi t/(60 \times 36))$, com $t$ dado em minutos; adiciona-se somente os últimos 7 elementos ao vetor de atributos $x_t$;

e) Cálculo da média climatológica para cada série temporal, dada por:

$$\mu_{cl}(x_t) = \frac{1}{s} \sum_{i=0}^{s} x_{t-p \times i} \ ,$$

onde $p$ indica o período definido como 48, correspondente um dia de 24 horas amostrado em intervalos de 30 minutos, $t$ é o instante atual, $i$ é o índice sobre o qual a soma é feita e $s$ é o comprimento da janela em dias. A variável $i$ é inicializada com zero e assim, a somatória vai de $t$ até $t - s \times p$. O cálculo foi feito para os seguintes valores de janela: 30, 25, 20, 15, 10, 5, 3 e 2. A Figura 5.1 ilustra o texto acima em relação à série temporal de uma variável qualquer para o cálculo da média climatológica.

f) Cálculo do desvio climatológico, para as mesmas janelas da média climatológica, dado por:

$$d(x_t) = x_t - \mu_{cl}(x_t).$$

g) Uso do algoritmo de Weasel-Muse implementado na biblioteca PYTS para a extração de atributos, configurado com os seguintes parâmetros: 10 intervalos para discretização com estratégia uniforme, isto é, mapeando os atributos em intervalos de mesmo tamanho; palavras com 4 ou 8 caracteres (alguns testes foram feitos com 4 e outros com 8); tamanho das janelas deslizantes de 0,2, 0,4, 0,6, 0,8, 1,0; sem utilizar representação esparsa, aplicável para conjuntos de dados contendo muitos valores nulos; uso do alfabeto latino para os caracteres; os demais parâmetros configuráveis foram mantidos com valores padrão.

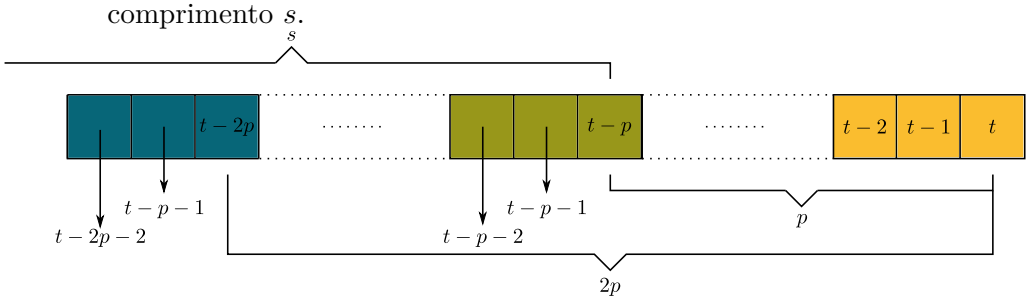
A seguir, o processamento dos vetores de atributos foi feito com as seguintes etapas:

h) Aplicação de um filtro de forma a remover variáveis cuja variância seja menor que o valor de corte $0,09$, determinado por meio de testes e pela avaliação das métricas de desempenho do modelo no subconjunto de validação;

i) Transformação dos dados de forma a se ter valor médio nulo e variância unitária.

j) Utilização do algoritmo SMOTE (*Synthetic Minority Over-sampling Technique*) disponível na biblioteca Imblearn (LEMAÎTRE et al., 2017) para a geração de amostras sintéticas a partir do subconjunto de treinamento inicial contendo o vetor de atributos, de forma a gerar um novo subconjunto de treinamento que seja balanceado em termos da distribuição de amostras entre as diversas classes de cintilação consideradas.

Figura 5.1 - Esquema ilustrando o cálculo de atributos climatológicos para uma janela de comprimento $s$.




Fonte: Produção do autor.

A aplicação das operações de média climatológica e desvio climatológico para as séries temporais gera dois grupos de novas séries temporais. Porém, o vetor de atributos para um dado instante $t$ vai conter somente informação da derivada até o instante $t - 6$, para cada um dos dois novos grupos.

## 5.4 Geração dos mapas de índice $S_4$

Neste trabalho, os mapas de cintilação utilizados correspondem a uma combinação dos dados coletados do índice $S_4$ por 3 redes de monitoramento: rede CIGALA/CALIBRA, rede do Instituto de Controle de Espaço Aéreo (ICEA) e rede *Low-Latitude Ionospheric Scintillation Network* (LISN). As redes CIGALA/CALIBRA e ICEA são mais recentes e utilizam o mesmo modelo de sensor, sendo assim seus dados compatíveis, enquanto que a rede LISN utiliza um outro tipo de sensor e outro formato de dados. Entretanto, todas as redes fornecem ao menos o índice $S_4$ medido na frequência L1 do GPS americano.

Os dados das duas primeiras redes são fornecidos no sistema de tempo próprio do GPS, enquanto a LISN utiliza o tempo UTC. Além disso, há uma diferença de tempo variável ao longo dos anos entre as duas primeiras e a LISN, que entre 2010 e 2018,

foi da ordem de dezenas de segundos. Essa diferença poderia implicar na geração de mapas diferentes, mas uma vez que a geração de mapas integra tipicamente dados relativos a 10 ou 15 minutos, tais diferenças são desprezíveis. Os trabalhos Vani (2018) e Paula et al. (2021) comparam valores do índice $S_4$ coletados por diferentes redes numa mesma localização.

Na geração dos mapas de cintilação, foram adotados aqui:

- um modelo de camada "fina" (infinitesimal) da ionosfera com altitude $hipp = 350\text{km}$;
- uma grade espacial definida na altitude *hipp* de dimensões 181 x 181 om resolução 0,5 x 0,5 graus, em particular, a célula $(0, 0)$ correspondendo à área $([239,75,\ 360,25[, [-60,25,\ -59,75[)$, com centro em $(360,0,\ -60)$. A grade abrange todo o território brasileiro e parte dos países limítrofes, correspondendo ao intervalo de coordenadas esféricas $[239,75,\ 336,25[\times[-60,25,\ 36,75[$. A curvatura da Terra foi desconsiderada;
- O ponto onde um sinal GNSS atravessa a ionosfera na altitude *hipp* é chamado de ponto de perfuração ou IPP, do inglês *ionospheric pierce point*. O valor do índice $S_4$ em cada IPP é então projetado na vertical do IPP correspondente (SPOGLI et al., 2009);

O procedimento adotado para construir o mapa do índice $S_4$ para o intervalo de tempo compreendido entre os instantes $t$ e $t - \Delta t$ é:

- para cada amostra do índice $S_4$ no intervalo de tempo considerado, calcular longitude $lon_{ipp}$ e latitude $lat_{ipp}$ do IPP correspondente e o ângulo $elv_{ipp}$ formado entre o *link* satélite-estação e a tangente à esfera no IPP (PROL; CAMARGO, 2016);
- para cada amostrada de índice $S_4$ no intervalo de tempo considerado, calcular sua projeção vertical no IPP, (SPOGLI et al., 2009; ALFONSI et al., 2011):

$$S_{4vec} = S_4 \sin(elv_{ipp}); \tag{5.1}$$

- determinar o ponto de grade mais próximo a cada IPP em função de sua longitude ($lon_{ipp}$) e latitude ($lat_{ipp}$) por meio da função "int" que converte

um número real em inteiro truncando sua parte decimal:

$$lon_{grad} = int(2(lon_{ipp} - 239,75)) \tag{5.2}$$

$$lat_{grad} = int(2(lat_{ipp} + 60,25)); \tag{5.3}$$

- considerando-se o IPP, agrupar os valores do índice $S_4$ projetados na vertical por alguma operação como máximo, média, ou desvio padrão. Considerando-se os IPPs numa mesma célula da grade, toma-se, como adotado neste trabalho, o valor máximo das correspondentes projeções, sendo este valor associado a esta célula de grade. A cada célula fica assim associado um único valor;

- uma vez projetados todos os valores de $S_4$ correspondentes aos IPPs na grade, a geração do mapa correspondente requer alguma técnica de interpolação, como por exemplo a interpolação pelo inverso da distância (PEBESMA, 2004; GRÄLER et al., 2016), adotada aqui considerando-se os quatro vizinhos mais próximos, ou então a interpolação cúbica em grade (REZENDE et al., 2007), entre outros (KIEFT et al., 2014; HAMEL et al., 2014).

A operação de agrupamento apresentada acima combina dados no espaço-tempo, isto é, combina os dados coletados em diferentes intervalos de tempo $[t, t - \Delta t[$, com os dados agrupados nas diferentes células da grade, sendo que cada célula contém dados de múltiplos pares satélites-receptores. O intervalo de tempo aqui adotado foi de 10 min, e os instantes são da forma 00:10, 00:20, 00:30, etc. cobrindo todo o período de 24 horas.

Foram gerados mapas para as datas entre 01/01/2010 e 31/12/2018. Entretanto, algumas estações GNSS das redes CIGALA/CALIBRA e ICEA só ficaram operacionais nos anos de 2010 e 2011. Consequentemente, o número total de mapas gerados para 2010 foi menor e sua qualidade possivelmente inferior dado o menor número de estações disponíveis nesse ano, o qual acarreta menor número de pontos para interpolação e assim induz um erro de interpolação maior.

## 5.5 Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - predição com *ensembles* XGBoost e/ou CatBoost

Neste experimento, os algoritmos de *ensemble* XGBoost e CatBoost utilizam vetores de atributos gerados a partir de séries temporais de vários dias dos atributos de predição. Esses vetores de atributos são gerados pela aplicação de operações mate-

máticas tais como máximo, mínimo, coeficientes da transformada de Fourier, entre outros. As predições são para 30-60-90-120-150-180 minutos ao longo de uma noite (18 h até 06 h da manhã seguinte), avançando com passos de 30 minutos num esquema de janela deslizante. Cada um dos três algoritmos gerou modelos para essas 6 antecedências de predição. Foram empregados os seguintes esquemas de validação *Gap Time Series Holdout* (GTSH) ou *Time Series Cross Validation Gap K Fold* (TSCV-GKF).

O conjunto inicial de variáveis utilizados foi: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $F10.7$, $Sunspot$, $TEC$, $S_4$. Posteriormente, foram realizados testes com a adição da variável $hF$.

O vetor de atributo de uma amostra, no instante $t$, é constituído por: (i) valores extraídos por meio das bibliotecas TSFRESH e TSFEL, (ii) derivadas das variáveis de entrada, (iii) relações não lineares entre as variáveis, (iv) média e desvio padrão climatológico e (v) variáveis originais do instante $t - 6$ até um $t$.

A predição será feita para o instante $t + n$, onde $n$ varia de 1 até 6. O algoritmo SMOTE da biblioteca Imblearn foi empregado, assim como o filtro por variância e a normalização para média nula e variância unitária. Os algoritmos de aprendizagem empregados foram o XGBoost e o CatBoost.

Foram avaliados, também, 3 períodos de predição com seus respectivos modelos: das 18h até 06h do dia seguinte, das 18h até às 24h, e das 24h até às 06h. Considera-se que uma amostra pertence a um dado período se ao menos uma predição (do total de 6 antecedências de predição) pertencer ao mesmo.

As etapas deste experimento foram:

- Geração dos atributos de informação (preditores);
- Particionamento dos dados segundo o esquema de particionamento, quando aplicada alguma forma de validação cruzada será sempre empregado com 5 subconjuntos;
- Filtragem por variância, normalização e aplicação do algoritmo SMOTE no conjunto de vetores de atributos (o SMOTE é aplicado somente ao subconjunto de treinamento, as demais duas operações são aplicadas em todos os subconjuntos);
- Treinamento do modelo com a consequente geração de um vetor que avalia

a importância de cada atributo (vetor de importância) na classificação. A métrica de desempenho sobre o subconjunto de validação controla uma parada antecipada da fase de treinamento, evitando assim um sobreajuste ao dados de treinamento;

- Avaliação do modelo no subconjunto de teste;
- Seleção dos atributos: realiza-se um laço sobre um vetor $A$ igualmente espaçado entre os valores máximo e mínimo do vetor de importância. A cada iteração um elemento do vetor $A$ é definido como um valor de corte e somente atributos com importância maior que o valor de corte serão utilizados no treinamento do modelo. Este modelo é avaliado no subconjunto de validação, e a sua métrica de desempenho é salva. Finalmente, seleciona-se o valor de corte que produziu o melhor valor de métrica $F_1$ no subconjunto de validação;
- Treinamento do modelo final utilizando o conjunto mínimo de atributos encontrado na etapa anterior;
- Avaliação do modelo final utilizando o subconjunto de teste;

Nas variantes do Experimento A, quando empregado algum esquema de validação cruzada, sempre considerando o mesmo tempo de antecedência de predição (30, 60, 90, 120, 150 ou 180 minutos), o modelo final é dado pela combinação dos 5 modelos gerados ao longo das iterações sobre o esquema de validação, de forma a reduzir a dependência do modelo do subconjunto avaliado pelo mecanismo de parada antecipada. Assim, o valor predito será a média das probabilidades de ocorrência de cintilação fornecida por cada modelo para a mesma antecedência de predição.

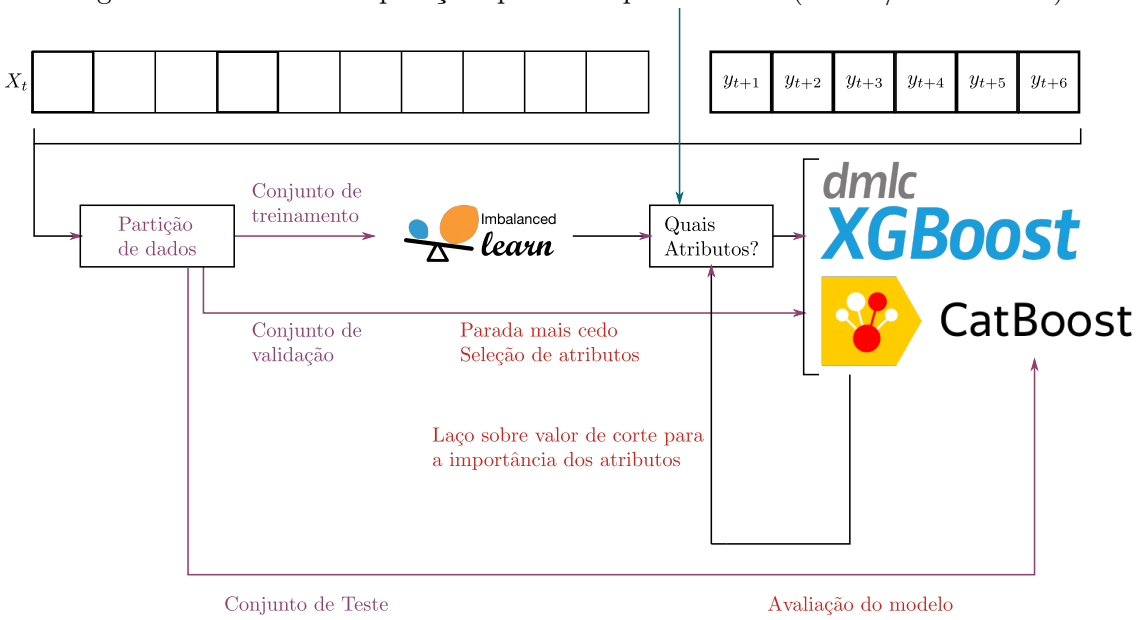
Este experimento apresenta duas principais variantes: o Experimento A (GTSH) anos 2012-2014 e o Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018. Ambos adotam as mesmas variáveis e metodologia apresentadas acima, mas há diferenças, pois o primeiro faz seleção de atributos e utiliza um períodos mais curto, enquanto que o segundo não faz tal seleção. O primeiro utiliza GTSH (*Gap Time Series Holdout*, mas o segundo emprega dois esquemas separadamente, o TSCV-GKF (*Time Series Cross Validation Gap K-Fold*) e o TSCV-GWF e (*Time Series Cross Validation Gap Walk Forward*).

Nesta segunda variante do experimento, os esquemas TSCV-GKF e TSCV-GWF foram utilizados com validação aninhada com dois níveis. No nível de particiona-

mento mais externo, os dados foram divididos em 5 subconjuntos (I-II-III-IV-V), e para cada iteração 4 destes são usados como subconjunto de pseudo-treinamento e 1 como subconjunto de teste. No nível mais interno, divide-se cada subconjunto de pseudo-treinamento também em 5 subconjuntos, e para cada iteração 4 destes são usados como subconjunto de treinamento e 1 como subconjunto de validação. Neste trabalho adotou-se uma abordagem em que tal divisão multi-nível possibilita obter 25 triplas de subconjuntos (treinamento-validação-teste), mas para cada um dos 5 níveis externos, combinam-se os 5 modelos do nível interno correspondentes, pela média das predições. Note-se que o particionamento em cada nível ocorre segundo o esquema TSCV-GKF ou então TSCV-GWF.

A Figura 5.2 apresenta o fluxo das operações do Experimento A.

Figura 5.2 - Fluxo das operações para o Experimento A (GTSH/TSCV-GKF).




Na primeira execução, todos os atributos são utilizados, nas execuções seguintes, será utilizado um subconjunto dos atributos cuja importância é maior que um valor de corte. Na execução final, utiliza-se o subconjunto de atributos que apresentou o melhor desempenho em termos da métrica $F_1$ sobre o subconjunto de validação.

Fonte: Produção do autor.

## 5.6 Experimento B (GTSH/VCT) anos 2010-2018 - predição com *ensembles* XGBoost e/ou CatBoost e codificação de atributos por Weasel-Muse

Neste experimento, a predição é feita pelos algoritmos de *ensemble* XGBoost e CatBoost, os quais utilizam séries temporais de até 24 horas dos atributos de predição

codificados pelo algoritmo de Weasel-Muse, o qual utiliza a transformada de Fourier da série temporal e representa os valores numéricos resultantes por palavras, i.e. sequências de caracteres. Cada dia é tratado independentemente, ou seja, são utilizados dados de um determinado dia para se fazer a predição de um valor único para a noite desse mesmo dia. Esse enfoque baseia-se na hipótese de se tratar cada dia independentemente dos demais, isto é, descartando a possibilidade de que a dinâmica ionosférica de um determinado dia possa influenciar nos dias consecutivos. O particionamento dos dados e avaliação do modelo foram realizadas com o *Gap Time Series Holdout* (GTSH) e com a Validação Cruzada Tradicional (VCT).

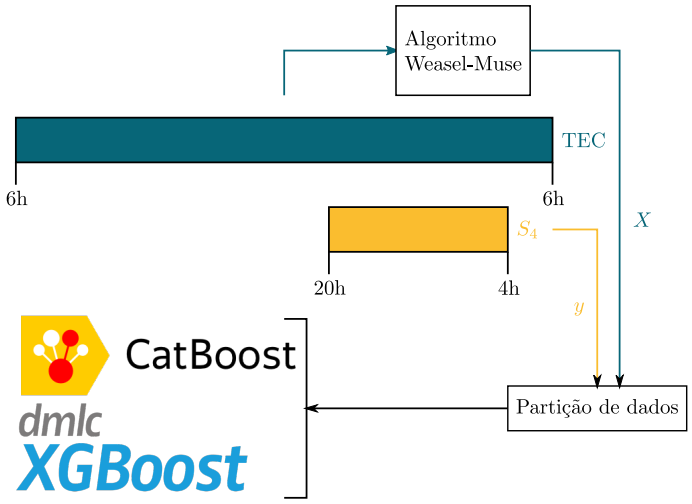
Foram realizados testes individuais com as variáveis: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $TEC$, $S_4$, $h'F$. Também, foi feito testes utilizando destas mais o TEC, em um total de 2 variáveis e utilizando todas as variáveis.

Foram avaliadas duas janelas de tempo referentes aos atributos preditores, 06-18 h (total de 12 h) e 06-06 h (total de 24 h), como ilustrado nas Figuras 5.3 e 5.4. É feita a predição de um único valor categórico de cintilação (classe OC/N-OC) para o período 20-04 h (total de 8h) de cada noite.

A configuração com janela de 24 horas foi testada utilizando-se apenas o TEC, assumindo-se o uso de valores preditos de TEC (supostamente preditos, pois estavam incluídos nos dados post-mortem), o que resulta numa série temporal suficientemente longa para que o algoritmo de Weasel-Muse gere palavras de 8 caracteres. Por outro lado, a configuração com janela de 12 horas foi testada com o TEC (sem assumir valores preditos) e também com as demais variáveis, resultando em séries temporais mais curtas, que resultaram em palavras de 4 caracteres.

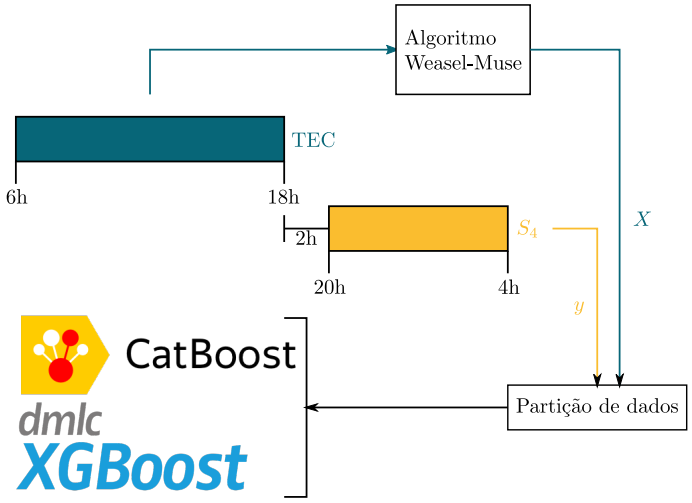
As Figuras 5.3 e 5.4 apresentam um esquema geral das etapas definidas acima, respectivamente para a configuração com janela de 24 horas e de 12 horas. A comparação de resultados entre ambas as configurações tem o intuito de verificar se a possível disponibilidade de informação futura da variável TEC (predita) poderia melhorar o resultado da predição de cintilação. Faça-se a ressalva que ainda não estão disponíveis predições de TEC com acurácia suficiente no escopo deste trabalho, sejam geradas por modelos numéricos ou por modelos orientados a dados.

Figura 5.3 - Variante do Experimento B (GTSH/VCT) anos 2010-2018 incluindo informações a respeito do TEC futuro.




Fonte: Produção do autor.

Figura 5.4 - Fluxo de operações e ilustração do tamanho das séries temporais para o Experimento B (GTSH/VCT) anos 2010-2018.




Fonte: Produção do autor.

As etapas deste experimento foram:

- Geração dos atributos preditores pelo algoritmo de Weasel-Muse, o qual inclui um esquema interno de seleção de atributos;
- Particionamento dos dados, quando aplicada alguma forma de validação cruzada será sempre empregado 5 subconjuntos;
- Treinamento do modelo. Assim como no Experimento A (GTSH/TSCV-

GKF) anos 2010-2018, a métrica de desempenho sobre o subconjunto de validação controla uma parada antecipada da fase de treinamento, evitando assim um sobreajuste ao dados de treinamento;

- Avaliação do modelo final utilizando o subconjunto de teste;

Assim como no Experimento A (GTSH/TSCV-GKF) anos 2010-2018, quando empregado algum esquema de validação cruzada, o modelo final é dado pela combinação dos 5 modelos gerados ao longo das iterações sobre o esquema de validação.

## 5.7 Experimento C (GTSH) anos 2010-2018 - predição com redes neurais convolucionais

Neste experimento, a predição é feita por uma rede neural convolucional a partir de séries temporais de 2 dias dos atributos de predição. Geram-se 6 modelos com base na mesma rede neural para efetuar predições de 30-60-90-120-150-180 minutos ao longo de uma noite. Os modelos diferem por ter sido treinados para antecedências diferentes. Foi empregado o *Gap Time Series Holdout* (GTSH) para validação do modelo.

O conjunto inicial de variáveis utilizados foi: $AE$, $IMFB_y$, $IMFB_z$, *Sym-H*, *Sym-D*, $V_{sw}$, $P_{sw}$, *ap*, $Dst$, $F10.7$, *Sunspot*, $TEC$, $S_4$. Posteriormente, foram realizados testes com a adição da variável $h'F$.

Consideram-se séries temporais com janela temporal de 2 dias consecutivos para a predição das 3 horas seguintes em intervalos de 30 minutos, que correspondem às antecedências de predição de cada uma das 6 redes, isto é, uma rede prediz com antecedência de 30 min, outra 60 min até um total de 180 minutos. Os dados passaram pelas técnicas de pré-processamento comuns aos três experimentos. Nenhuma técnica de extração de atributos é empregada, pois se entende que a rede realizará internamente sua própria extração de atributos. As etapas deste experimento foram:

- Particionamento dos dados;

- Treinamento das 6 redes neurais (uma para cada antecedência de predição), por 1000 épocas. Para cada antecedência de predição, a rede neural é avaliada continuamente no subconjunto de validação para cada época, mantendo uma cópia da rede que apresentou o melhor desempenho na métrica $F_1$, a qual será adotada ao final do treinamento;

- Avaliação final dos modelos utilizando o subconjunto de teste.

No treinamento da rede neural, os pesos são atualizados pelo algoritmo de otimização estocástica Adam, (KINGMA; BA, 2014), com os seguintes hiperparâmetros: 1000 amostras por iteração, taxa de aprendizado 0,0003, regularização com norma L2 com multiplicador 0,01, taxas de decaimento exponencial para as estimativas de momentos ambas iguais à 0,5. A função objetivo a ser otimizada foi a entropia cruzada binária. A predição final é categórica com valor OC, sendo que a saída da rede neural expressa a probabilidade de uma amostra pertencer a classe OC.

# 6 RESULTADOS DOS EXPERIMENTOS REALIZADOS

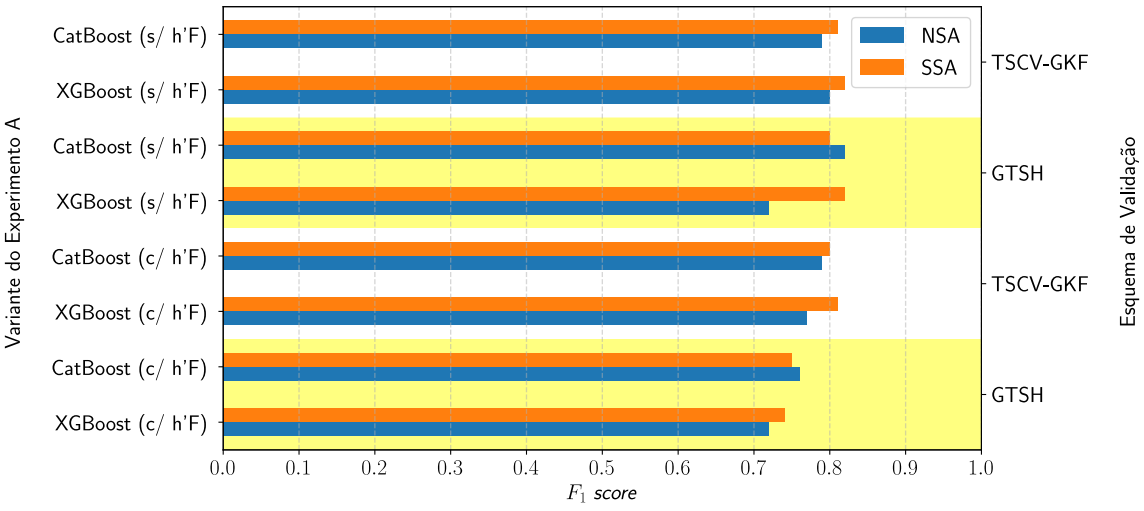
Apresentam-se aqui os resultados dos experimentos realizados (A, B e C) e suas variantes. O Apêndice A apresenta um conjunto estendido dos resultados apresentados neste capítulo.

## 6.1 Experimento A (GTSH/TSCV-GKF) anos 2010-2018

Os resultados do Experimento A (GTSH/TSCV-GKF, anos 2010-2018) para o período 18-06 h são apresentados nas Figuras 6.1 à 6.6. Analisando estes resultados, pode-se afirmar que existem casos onde a seleção de atributos melhora o desempenho de predição e outros em que não. O desempenho de predição foi avaliado pela métrica $F_1$ considerando-se os resultados para o subconjunto de teste. Observou-se que o desempenho de predição degrada com o aumento da antecedência de predição.

O subconjunto de teste é mesmo ao longo do Experimento A (GTSH/TSCV-GKF, anos 2010-2018). O subconjunto de validação muda em função do esquema de validação adotado. A validação TSCV-GKF gera um modelo mais independente do subconjunto de validação, pois envolve uma média entre os múltiplos subconjuntos de validação.

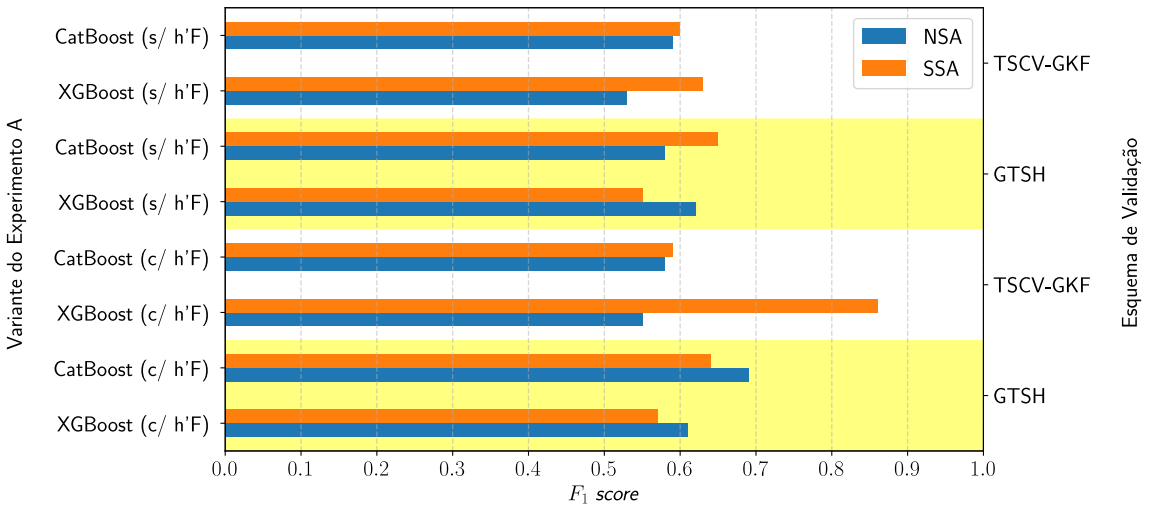
Figura 6.1 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 30 min.




Denota-se respectivamente por "c/" e "s/", a inclusão ou não da variável preditora h'F. SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

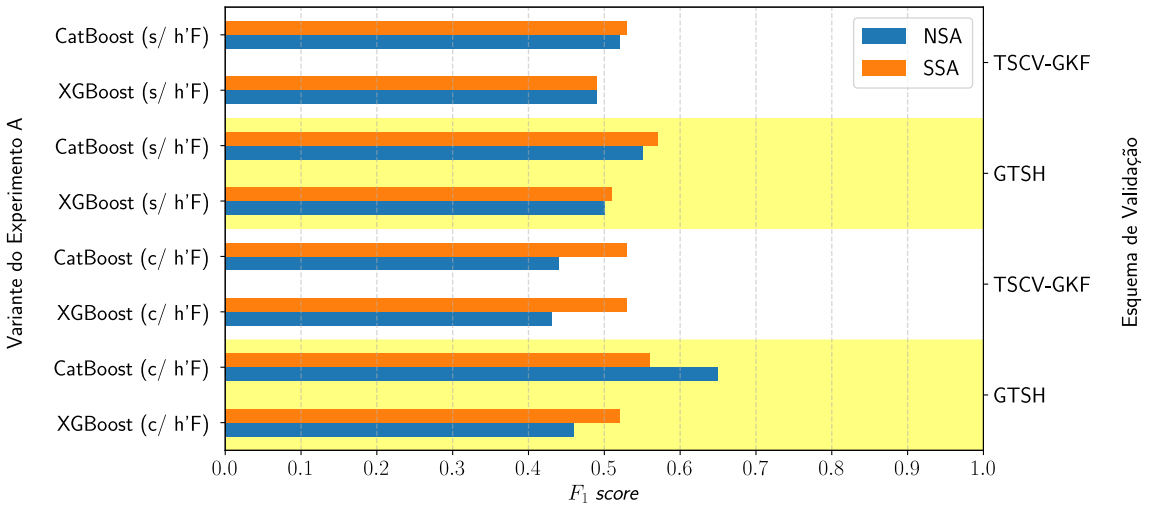
Figura 6.2 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 60 min.




Denota-se respectivamente por "c/" e "s/", a inclusão ou não da variável preditora h'F. SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

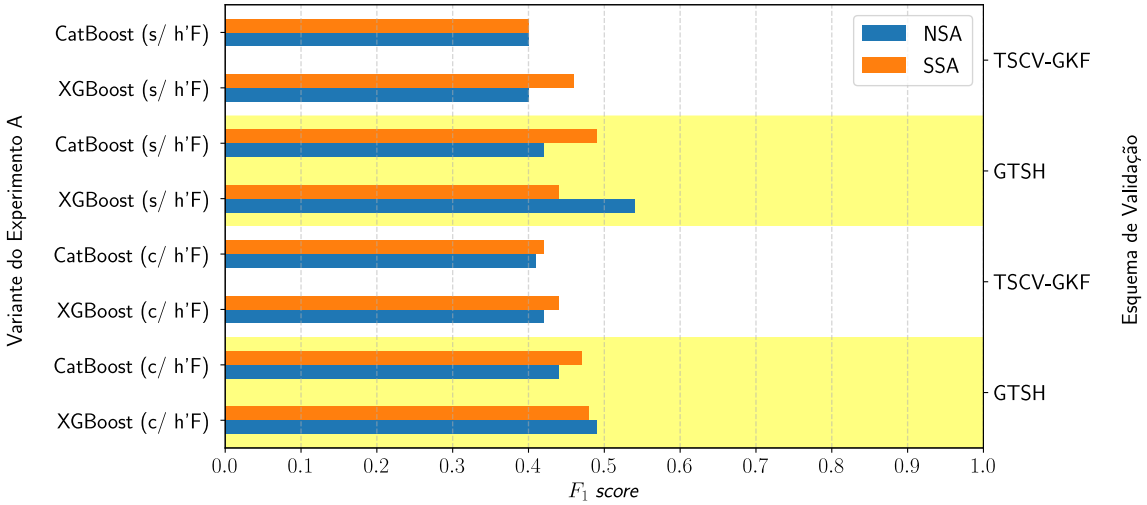
Figura 6.3 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 90 min.




Denota-se respectivamente por "c/" e "s/", a inclusão ou não da variável preditora h'F. SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

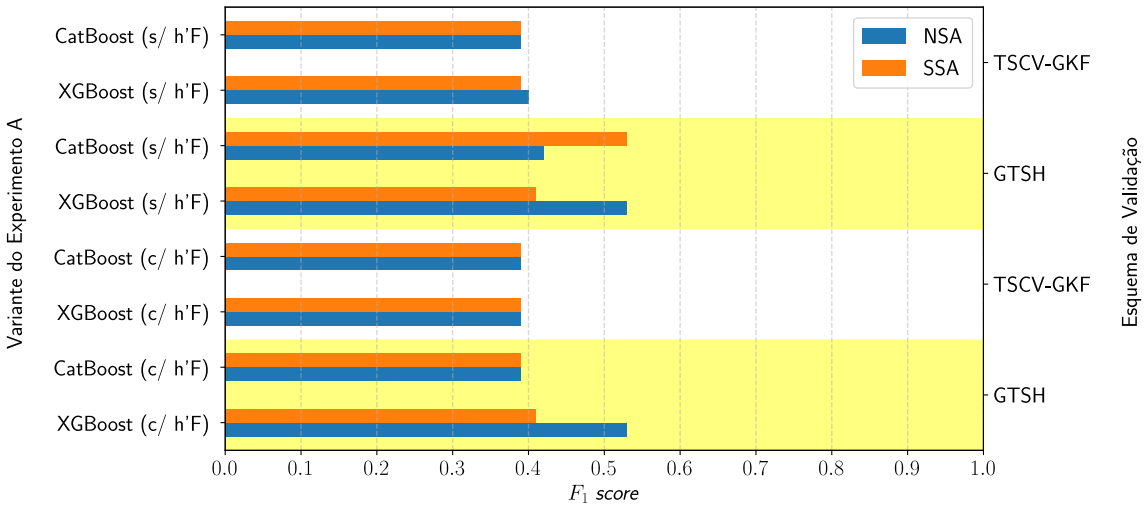
Figura 6.4 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 120 min.




Denota-se respectivamente por "c/" e "s/", a inclusão ou não da variável preditora h'F. SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

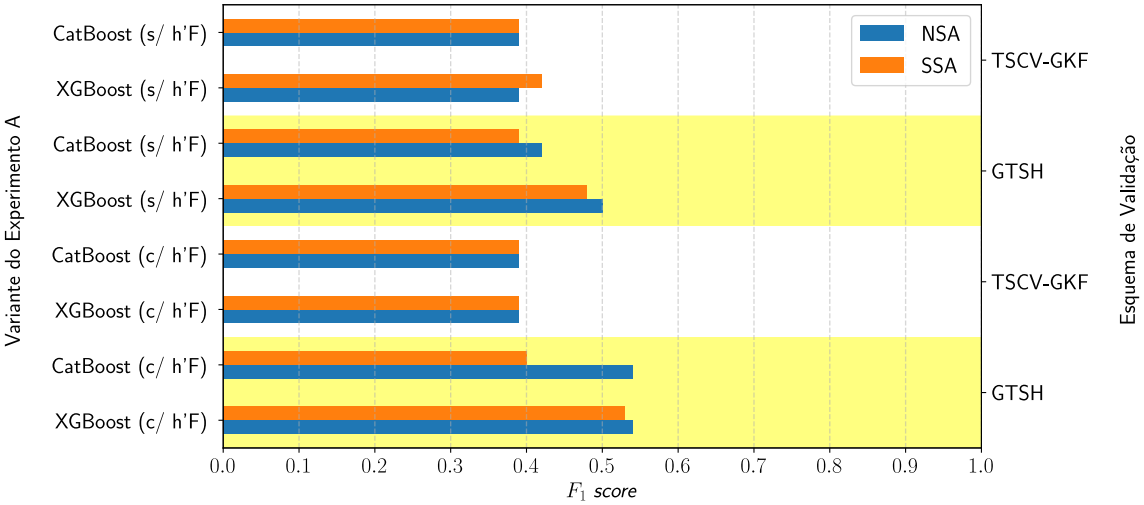
Figura 6.5 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 150 min.




Denota-se respectivamente por "c/" e "s/", a inclusão ou não da variável preditora h'F. SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

Figura 6.6 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 180 min.




Denota-se respectivamente por "c/" e "s/", a inclusão ou não da variável preditora h'F. SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

## 6.2 Experimento A (GTSH) anos 2012-2014

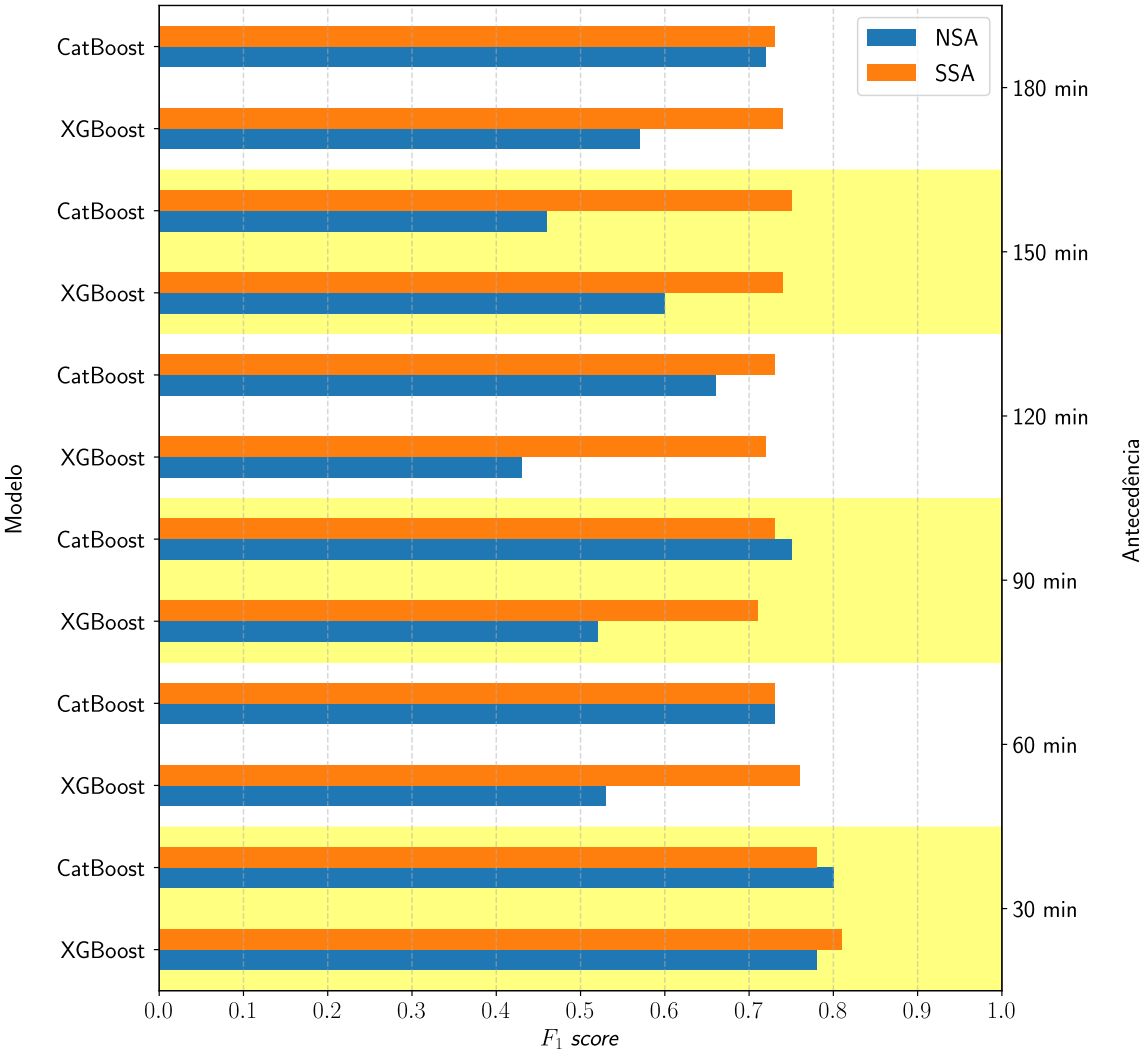
Nesta variante do experimento anterior, foi adotado como esquema de validação apenas o GTSH. Além disso, os dados referem-se a um período menor, compreendido pelos anos de 2012, 2013 e 2014, os quais foram utilizados respectivamente, para treinamento, validação e teste. Um ponto interessante é que 2014 foi ano de máximo do ciclo solar, contrastando com os testes anteriores do Experimento A, em que o treinamento e validação utilizaram dados de anos de máximo do ciclo solar, enquanto que o teste utilizou dados do mínimo do ciclo solar.

A Figura 6.7 apresenta os resultados para o Experimento A (GTSH, anos 2012-2014). Comparando estes resultados com os anteriores do Experimento A (GTSH/TSCV-GKF, anos 2010-2018), Figuras 6.1 à 6.6, pode-se observar que estes resultados apresentam valores de $F_1$ melhores, mas ainda apresentando uma degradação desses valores com o aumento da antecedência de predição, embora essa degradação seja mais suave. Além disso, o processo de seleção de atributos contribui mais para a melhora de desempenho de predição.

A redução na degradação com o aumento da antecedência de predição e a melhora resultante da seleção de atributos podem ser explicados pelas diferenças de intensi-

dade no ciclo considerado, em que aparecem períodos onde a cintilação varia mais nitidamente, correlacionando-se melhor com as variáveis preditoras, enquanto em outros períodos, essa variação é mais aleatória.

Figura 6.7 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição.




SSA e NSA denotam respectivamente com seleção de atributos e sem seleção de atributos.

Fonte: Produção do autor.

## 6.3 Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018

Esta variação do Experimento A utiliza um esquema de validação multi-nível, com 5 subconjuntos (I-II-III-IV-V) no nível mais externo. Cada nível externo dividido em 5 subconjuntos internos (1-2-3-4-5). As Tabelas 6.1 e 6.3 ilustram respectivamente

para o esquema TSCV-GKF e para o esquema TSCV-GWF o número de amostras por subconjunto para a predição com 30 minutos de antecedência.

Tabela 6.1 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Número de amostras considerando a predição 30 min à frente para TSCV-GKF.

| Subconjunto | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| I | Treinamento | 17406 | 16216 | 16450 | 17006 | 18700 |
| | Validação | 3256 | 3256 | 3255 | 3255 | 3255 |
| | Teste | 4406 | 4406 | 4406 | 4406 | 4406 |
| II | Treinamento | 15508 | 14288 | 14564 | 15158 | 17242 |
| | Validação | 2987 | 2987 | 2987 | 2986 | 2986 |
| | Teste | 4406 | 4406 | 4406 | 4406 | 4406 |
| III | Treinamento | 15510 | 14290 | 14566 | 15160 | 17244 |
| | Validação | 2987 | 2987 | 2987 | 2987 | 2986 |
| | Teste | 4405 | 4405 | 4405 | 4405 | 4405 |
| IV | Treinamento | 15510 | 14290 | 14566 | 15160 | 17244 |
| | Validação | 2987 | 2987 | 2987 | 2987 | 2986 |
| | Teste | 4405 | 4405 | 4405 | 4405 | 4406 |
| V | Treinamento | 17408 | 16218 | 16450 | 17008 | 18700 |
| | Validação | 3256 | 3256 | 3255 | 3255 | 3255 |
| | Teste | 4405 | 4405 | 4405 | 4405 | 4405 |

Fonte: Produção do autor.

Tabela 6.2 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Número de amostras considerando a predição 30 min à frente.

| Subconjunto | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| I | Treinamento | 974 | 2144 | 3322 | 4514 | 5706 |
| | Validação | 596 | 596 | 596 | 596 | 596 |
| | Teste | 3671 | 3671 | 3671 | 3671 | 3671 |
| II | Treinamento | 2176 | 4584 | 6976 | 8252 | 10186 |
| | Validação | 1207 | 1207 | 1207 | 1207 | 1207 |
| | Teste | 3671 | 3671 | 3671 | 3671 | 3671 |
| III | Treinamento | 3392 | 7008 | 9200 | 12048 | 15130 |
| | Validação | 1819 | 1819 | 1819 | 1819 | 1819 |
| | Teste | 3671 | 3671 | 3671 | 3671 | 3671 |
| IV | Treinamento | 4614 | 8286 | 12060 | 16182 | 19342 |
| | Validação | 2431 | 2431 | 2431 | 2431 | 2431 |
| | Teste | 3671 | 3671 | 3671 | 3671 | 3671 |
| V | Treinamento | 5836 | 10282 | 15172 | 19350 | 23566 |
| | Validação | 3043 | 3043 | 3043 | 3043 | 3043 |
| | Teste | 3671 | 3671 | 3671 | 3671 | 3671 |

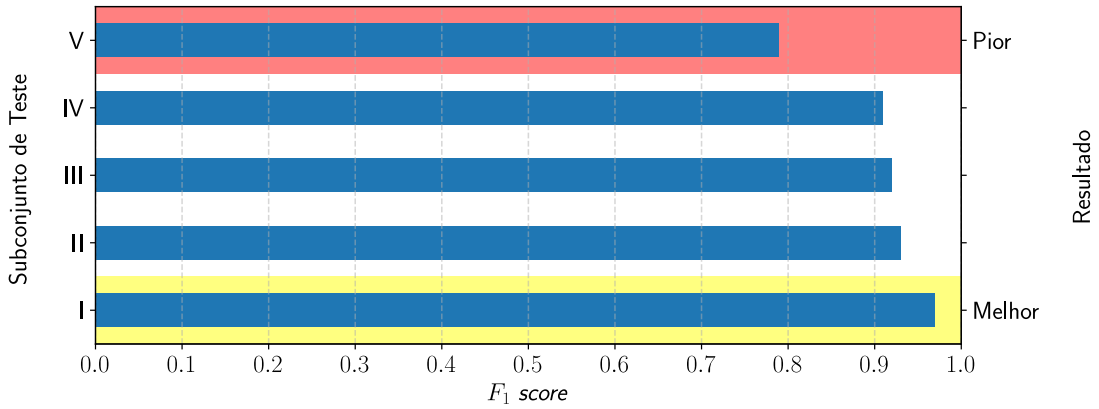
Fonte: Produção do autor.

O número de amostras para as demais antecedências são semelhantes em cada esquema. Nota-se nas Tabelas acima que os subconjunto de teste têm tamanhos fixo, enquanto que os subconjuntos de treinamento e validação variam de tamanho, em consequência do esquema de validação.

Considerando a variante com esquema TSCV-GKF, os resultados são apresentado nas Figuras 6.8 à 6.13, sendo os melhores obtidos ao longo do trabalho, embora exibam tendências comuns aos resultados dos outros experimentos, como por exemplo, o decaimento de desempenho com o aumento do tempo de antecedência da predição. Os modelos gerados para os quatro primeiros subconjuntos de teste (I, II, III, IV) obtiveram melhor desempenho de predição, enquanto que o modelo gerado para o subconjunto de teste V apresentou desempenho similar àquele das variações anteriores do Experimento A (GTSH/TSCV-GKF, anos 2010-2018), uma vez que utiliza dados de teste de um período semelhante.

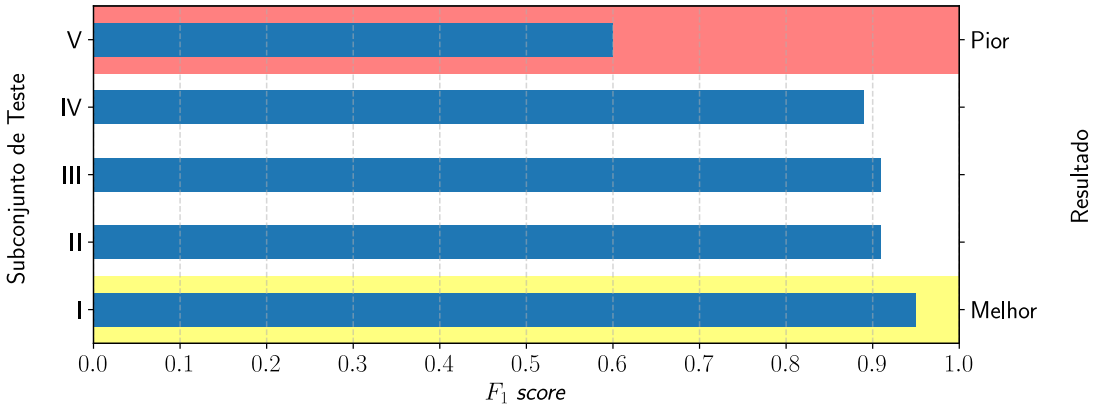
Um problema da variante TSCV-GKF é a utilização de dados futuros para treinamento, uma vez que, por exemplo, quando o subconjunto I for utilizado para teste, os subconjuntos de II-III-IV-V serão utilizados para treinamento e validação. Assim, dados futuros em relação àqueles do teste serão utilizados para gerar o modelo. Por outro lado, quando subconjunto $V$ for usado como teste, apenas dados anteriores cronologicamente serão utilizados para gerar o modelo. Embora o modelo gerado pelo subconjunto $V$ tenha obtido pior desempenho de predição, reproduz melhor condições operacionais reais, por não apresentar vazamento no particionamento dos dados.

Figura 6.8 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF com antecedência de 30 min.



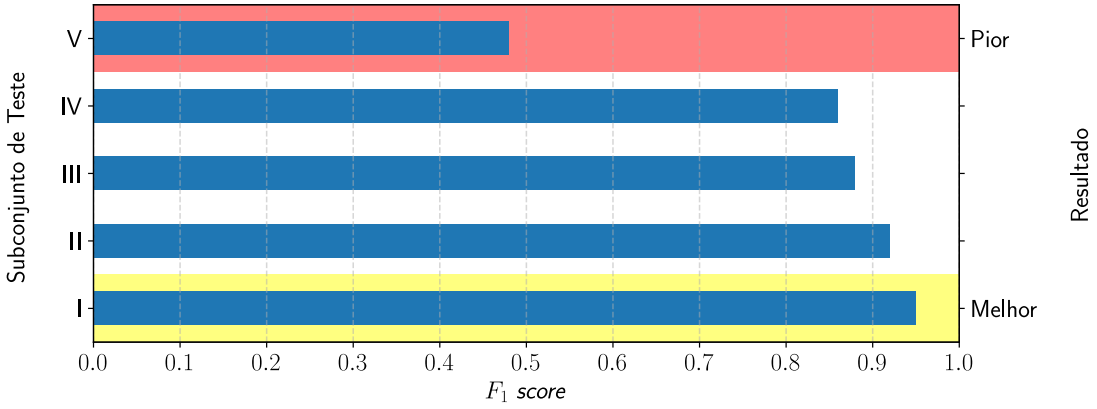

Fonte: Produção do autor.

Figura 6.9 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF com antecedência de 60 min.



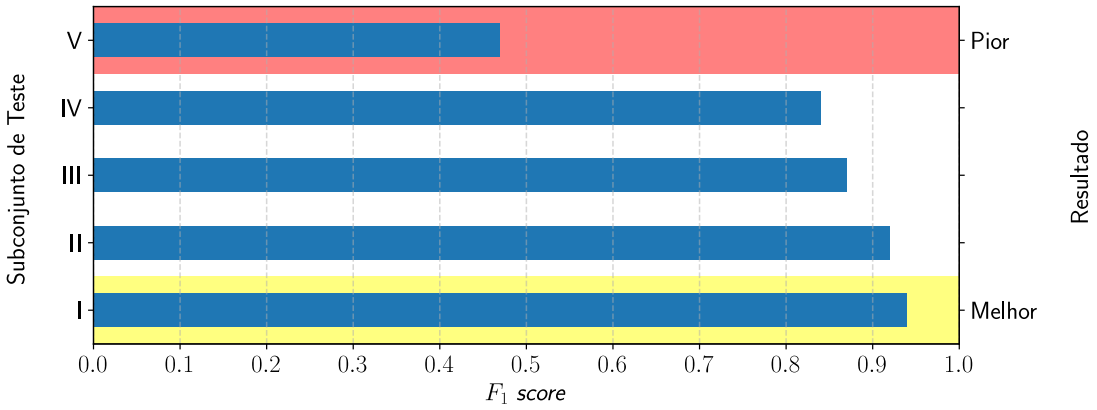

Fonte: Produção do autor.

Figura 6.10 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF com antecedência de 90 min.



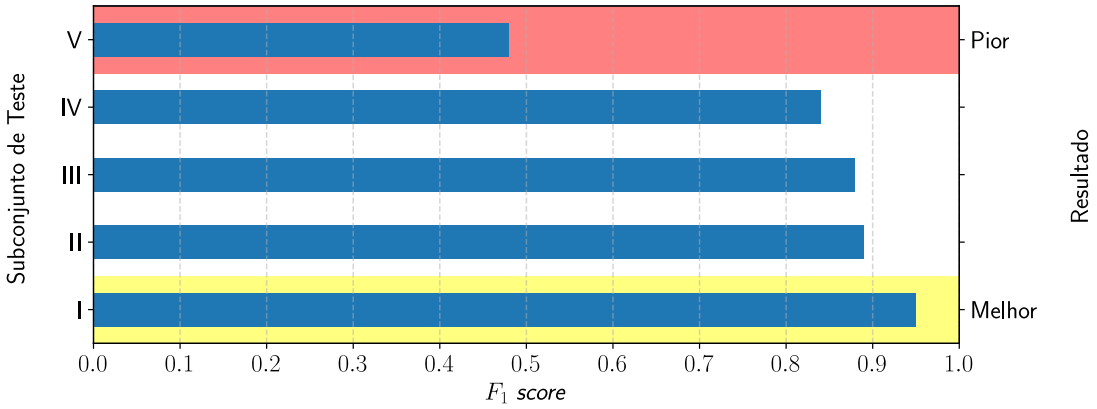

Fonte: Produção do autor.

Figura 6.11 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF com antecedência de 120 min.



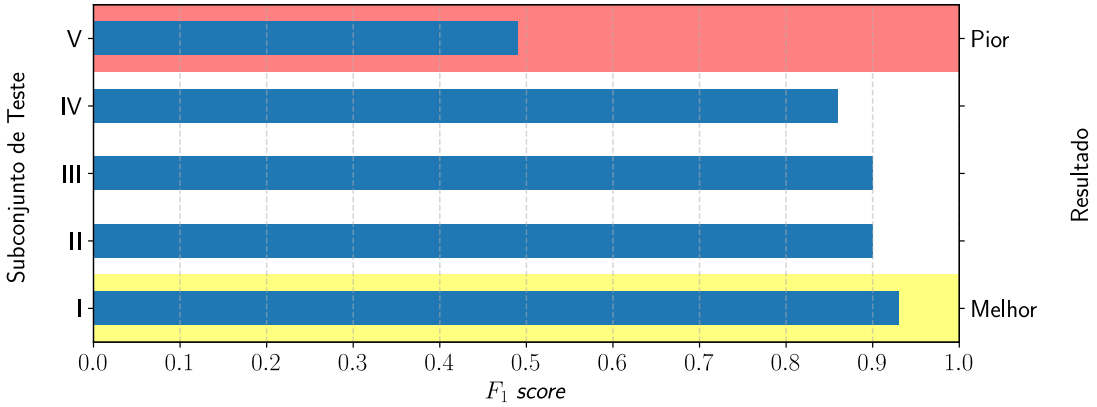

Fonte: Produção do autor.

Figura 6.12 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF com antecedência de 150 min.




Fonte: Produção do autor.

Figura 6.13 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF com antecedência de 180 min.




Fonte: Produção do autor.

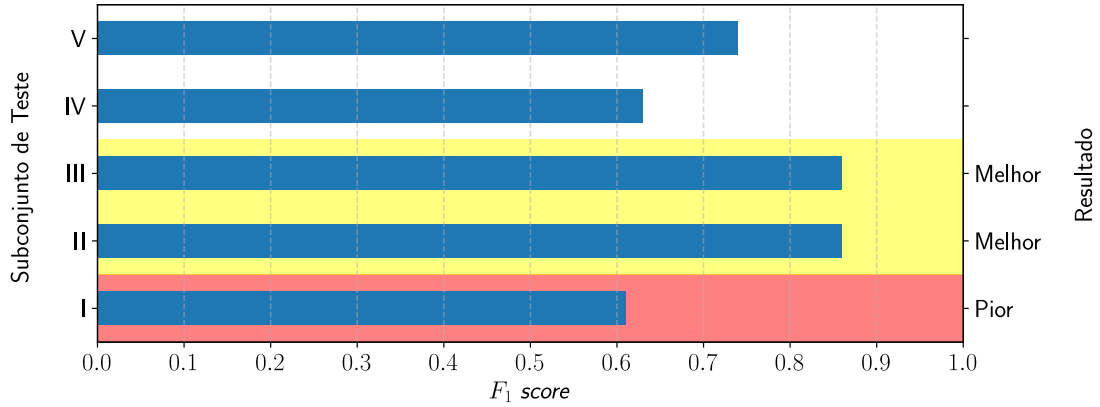
A variante com o esquema TSCV-GWF assegura que as amostras dos subconjuntos de treinamento e validação, utilizadas para gerar o modelo, antecedem cronologicamente as amostras de subconjunto de teste, reproduzindo melhor condições operacionais reais. As Figuras 6.14 à 6.19 apresentam os resultados para essa variante.

Nas figuras associadas ao esquema TSCV-GWF, observa-se que os subconjuntos II, III e IV geram modelos com melhor desempenho de predição, similarmente ao que ocorria com o esquema TSCV-GKF, mas é importante notar que os conjuntos I-II-III-IV-V gerados são diferentes para ambos esquemas, embora possam conter períodos parcialmente sobrepostos.

Os períodos em que o desempenho de predição foi melhor correspondem ao máximo

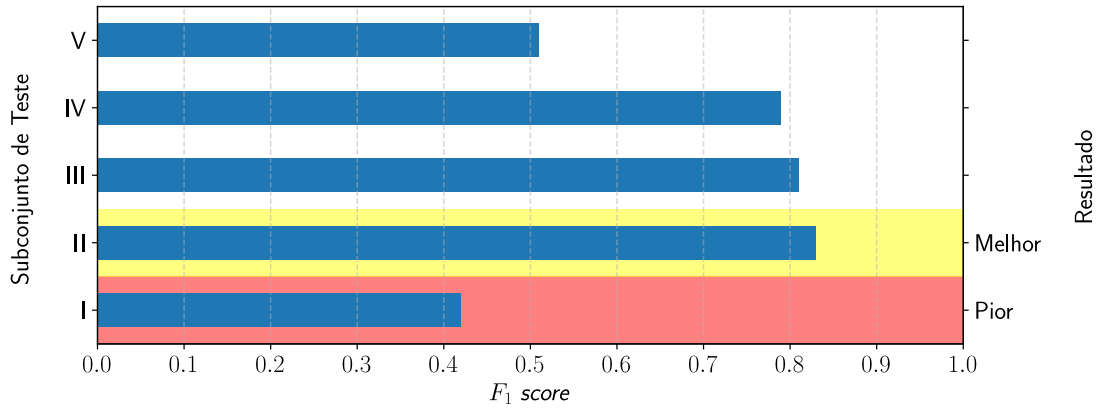
do ciclo colar, como já ocorreu com testes anteriormente mencionados neste capítulo. Embora o subconjunto I tenha gerado um modelo com péssimo desempenho, pode-se observar na Tabela 6.3 que o número de amostras dos conjuntos de treinamento e validação é muito pequeno. Uma vez que o conjunto de validação serve para interromper o treinamento num certo ponto de forma a evitar *overfitting*, os tamanhos pequenos desses conjuntos podem levar a uma interrupção muito precoce, limitando o modelo a tratar apenas um número restrito de amostras. Uma comparação dos resultados de ambos os esquemas para os subconjuntos II, III e IV indica que o esquema TSCV-GWF gera modelos com pior desempenho de predição, mas sem utilizar informação futura. Isso implica que, conforme o subconjunto, o esquema TSCV-GWF implique no uso de um número restrito de amostras, insuficientes para caracterizar todo o espectro de observações.

Figura 6.14 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF com antecedência de 30 min.



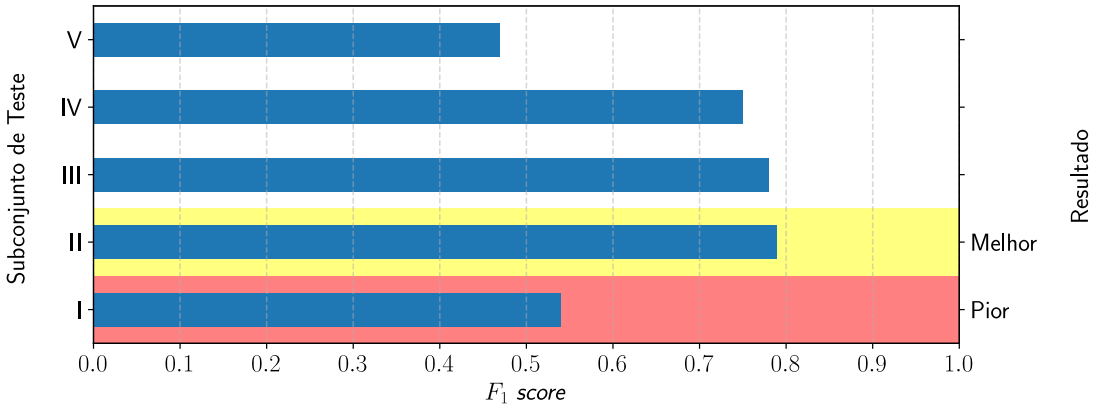

Fonte: Produção do autor.

Figura 6.15 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF com antecedência de 60 min.



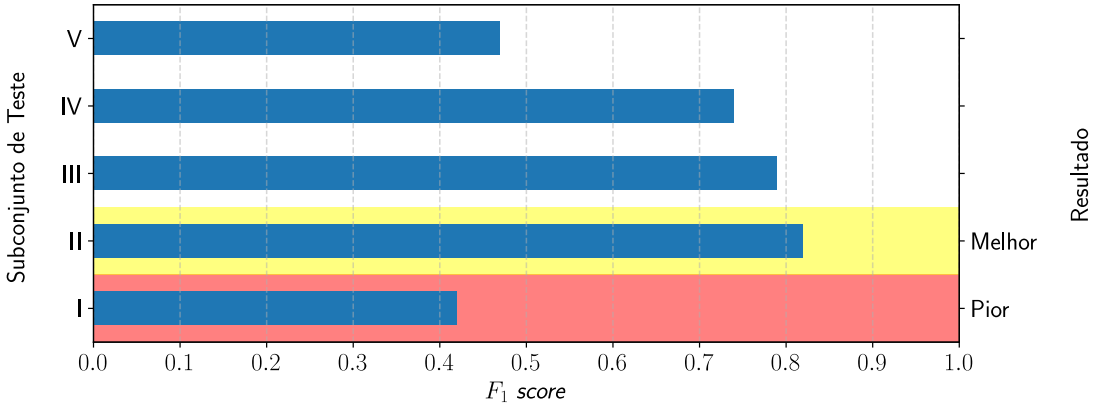

Fonte: Produção do autor.

Figura 6.16 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF com antecedência de 90 min.



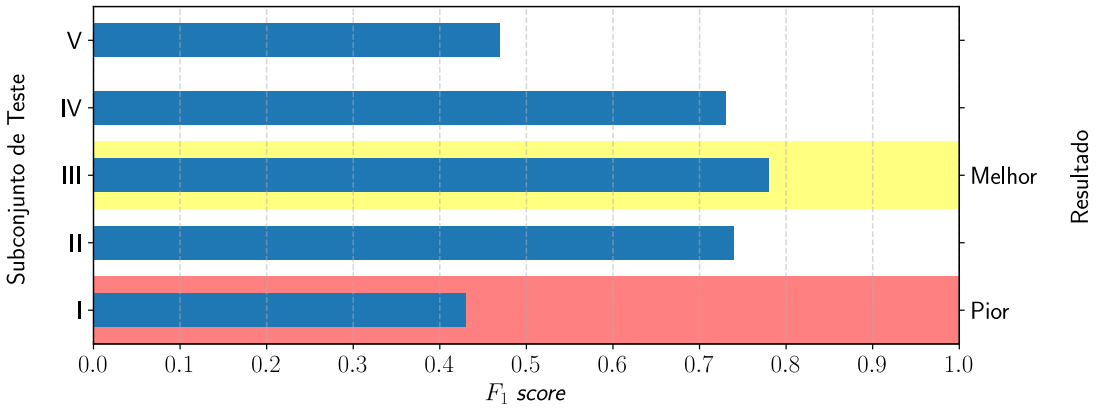

Fonte: Produção do autor.

Figura 6.17 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF com antecedência de 120 min.



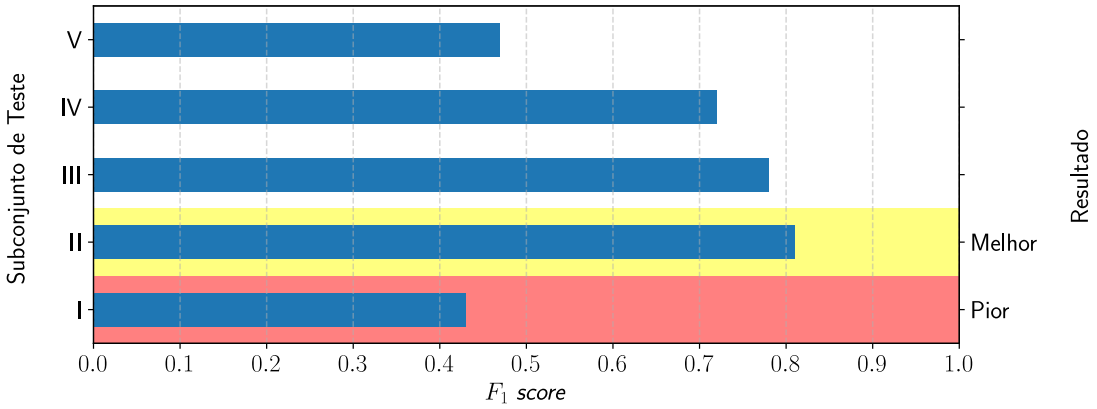

Fonte: Produção do autor.

Figura 6.18 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF com antecedência de 150 min.




Fonte: Produção do autor.

Figura 6.19 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF com antecedência de 180 min.




Fonte: Produção do autor.

## 6.4 Experimento B (GTSH/VCT) anos 2010-2018

O Experimento B (GTSH/VCT, anos 2010-2018) executa a predição com ensembles XGBoost e CatBoost, separadamente, ambos com codificação de atributos pelo algoritmo de Weasel-Muse. Os melhores resultados aparecem na Tabela 6.3, que apresenta os resultados utilizando como único atributo preditor os valores futuros de TEC (i.e. correspondentes ao instante-alvo da predição), tamanho de palavra 8, e definindo-se a classe OC como composta por amostras de cintilação [forte-moderada-fraca]. Assumiu-se que o valor de TEC estaria disponível nos horários-alvo das previsões à noite, ou seja, que esteja disponível a predição do TEC futuro, embora essa predição não tenha sido realizada, mas sim a utilização dos correspondentes valores reais de TEC. A comparação entre as variantes GTSH e VCT foi inconclusiva.

Tabela 6.3 - Experimento B (GTSH/VCT) anos 2010-2018 - Desempenho de predição com tamanho de palavra 8 e tendo como único atributo preditor o TEC futuro.

| Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|
| Holdout Cronológico | XGBoost | 0,73 | 0,64 | 0,65 |
| | CatBoost | 0,76 | 0,60 | 0,60 |
| Validação Cruzada Tradicional | XGBoost | 0,66 | 0,66 | 0,63 |
| | CatBoost | 0,54 | 0,66 | 0,55 |

Fonte: Produção do autor.

Os demais casos, relativos a palavras de tamanho 4 encontram-se no Apêndice B, sendo a Tabela A.57 para a classe OC composta por amostras de cintilação [forte-

moderada-fraca] e a Tabela A.58, por amostras de cintilação [forte-moderada]. Nesses casos, foram tentadas várias combinações de atributos preditores, incluindo ou não o TEC, mas esses valores de TEC não são valores futuros. Conclui-se que o desempenho de predição é pior com palavras de tamanho 4 do que com palavras de tamanho 8, sendo que os modelos gerados apresentaram valores de $F_1$ próximos ou (geralmente) abaixo de $0,5$. Comparando-se os valores de $F_1$ dessas tabelas para tamanho de palavra 4, pode-se inferir que prever classe OC como sendo [forte-moderado] é mais difícil do que sendo [forte-moderado-fraco].

Ainda considerando os testes com tamanho de palavra 4 apresentados na Tabela A.57, nota-se a importância da variável $h'F$, que utilizada isoladamente na predição resultou nos melhores valores de $F_1$, inclusive melhores do que utilizando todas as variáveis. Entretanto, isso não se observou nos resultados da Tabela A.58, na qual a utilização de todas as variáveis foi melhor. Novamente, a comparação entre as variantes GTSH e VCT foi inconclusiva. Quando da utilização de todos os atributos preditores ("Todos"), aparecem duas opções: "Todos (i)", em que se aplica o algoritmo de Weasel-Muse a todas as variáveis para geração do vetor de atributos; e "Todos (ii)", em que esse algoritmo é aplicado independentemente a cada variável sendo que posteriormente essas codificações para as variáveis são concatenadas.

## 6.5 Experimento C (GTSH) anos 2010-2018

O Experimento C (GTSH, anos 2010-2018) foi realizado com um conjunto inicial de variáveis: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $F_1 0.7$, $Sunspot$, $TEC$, $S_4$, denominado de conjunto "Original" de variáveis. Posteriormente, tornou-se disponível e foi adicionada a variável $h'F$. E, finalmente, também foram adicionadas outras 4 variáveis: (i) tempo (instante do dia em minutos) representado como uma onda de período 1440 ($24 \times 60$) minutos pelo correspondente sin e cos do tempo; (ii) $TEC$ e $S_4$ em São Luiz; (iii) TEC e $S_4$ dos primeiros vizinhos do ponto de grade associado à São José dos Campos totalizando 8 elementos; (iv) Laplaciano do TEC em função dos 4 primeiros vizinhos do mesmo ponto de grade.

As Figuras 6.20 à 6.25 apresentam os resultados do Experimento C (GTSH, anos 2010-2018) que utilizaram como algoritmo de aprendizado de máquina a rede neural apresentada na Figura 4.17. Nestas figuras, as barras vermelhas ilustram o desempenho dos modelos gerados com dados do verão, enquanto as azuis, com dados para todo o ano. "O" indica o conjunto de variáveis originais, "T" é tempo UTC em minutos, "viz." indica pontos de grade vizinhos, "(TEC+S4)/SLuiz" indica valores de ambas variáveis em São Luiz, "(TEC+S4)/8-viz." idem para as médias dos 8 pon-

tos de grade vizinhos a SJC, "$\nabla^2(TEC)$/4-viz." é o Laplaciano do TEC para os 4 pontos de grade vizinhos a SJC e "Todas" indica o conjunto completo de variáveis preditoras.

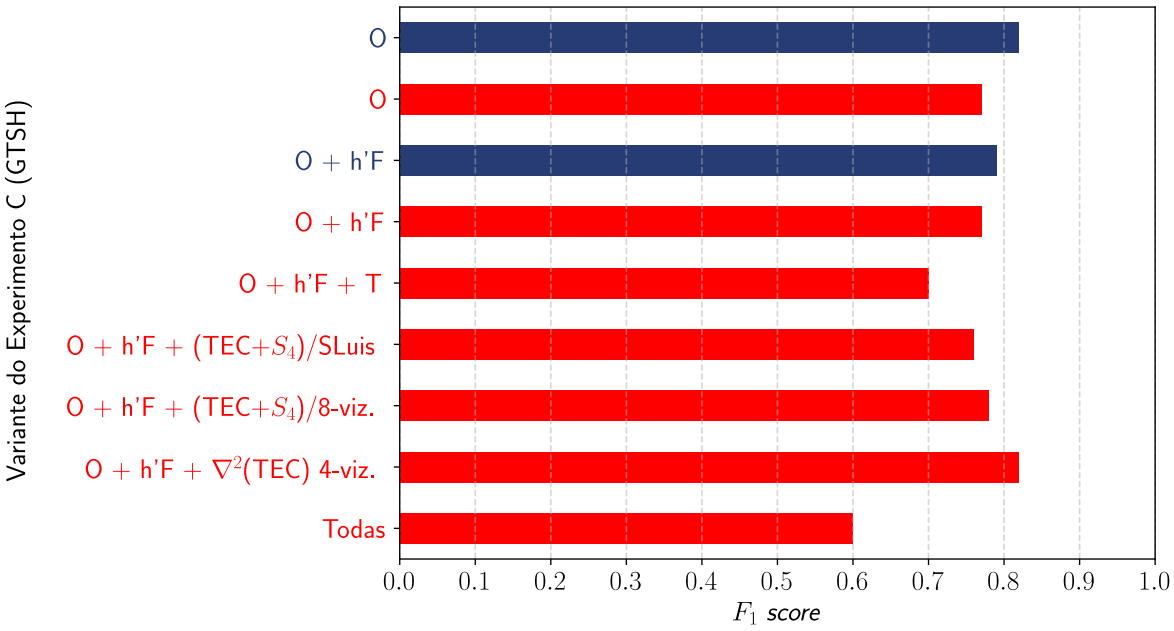
Assim, como acontecia no Experimento A (GTSH/TSCV-GKF, anos 2010-2018), quanto maior a antecedência de predição menor o valor de $F_1$. O Experimento C (GTSH, anos 2010-2018) utiliza o esquema de validação GTSH, e tem os subconjuntos de teste, treinamento e validação definidos no mesmo período do Experimento A (GTSH/TSCV-GKF, anos 2010-2018). Assim, os resultados obtidos são semelhantes para ambos, como por exemplo, com valores de $F_1$ no intervalo $[0,7 - 0,8]$ para antecedência de 30 min.

À semelhança entre os resultados do Experimento A (GTSH/TSCV-GKF, anos 2010-2018) e Experimento C (GTSH, anos 2010-2018) se reduz com o aumento da antecedência de predição, levando em consideração as variantes de cada experimento, uma vez que estas apresentaram maior variabilidade nos valores de $F_1$.

Observando as Figuras 6.20 à 6.25 é possível notar uma leve tendência para que que os modelos desenvolvidos somente para o verão sejam melhores que os desenvolvidos para qualquer estação do ano. Também é possível notar que, em geral, a adição da variável $h'F$ melhora os resultados, porém existem casos em que isso não acontece, ou seja, degrada o desempenho do modelo.

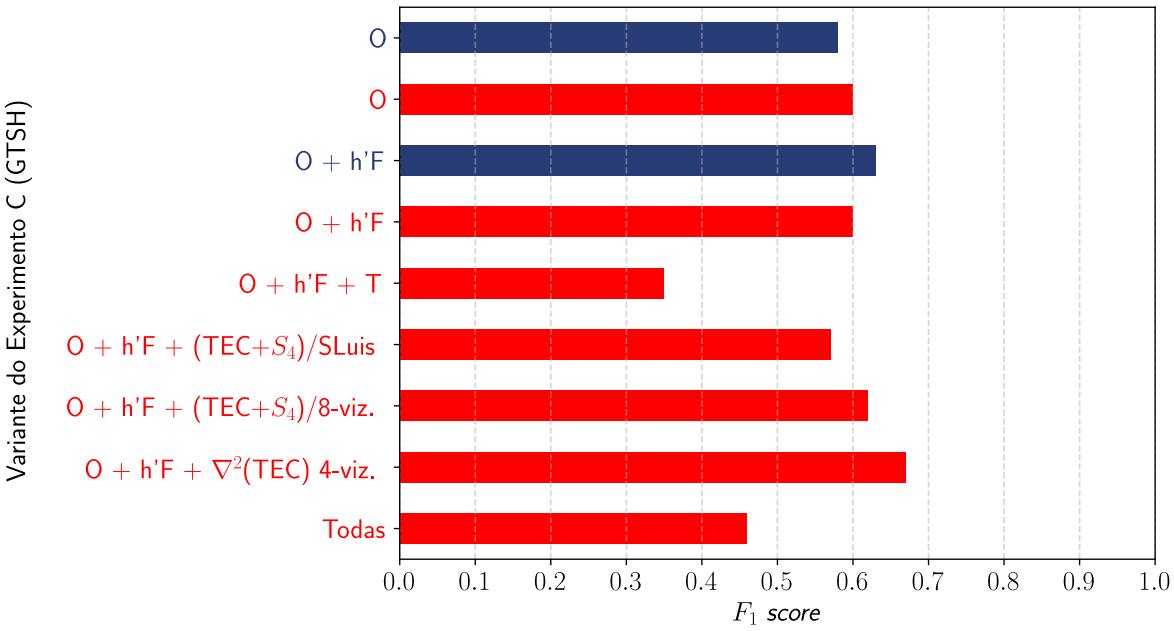
Adição de informação sobre o Tempo degrada o desempenho de predição do modelo; a adição de informação local de São Luiz pode ou não degradar o desempenho, conforme o caso; a adição do $TEC$ e $S_4$ dos primeiros 8 vizinhos em SJC degrada o desempenho para antecedências de predição até 120 minutos, mas o melhora para 150 e 180 minutos; a adição do laplaciano do TEC dos primeiros 4 vizinhos em SJC degrada o desempenho para antecedências de 60 e 90 minutos, mas melhora para as demais; finalmente a adição de todas as variáveis degrada o desempenho do modelo.

Figura 6.20 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para antecedência de 30 min.



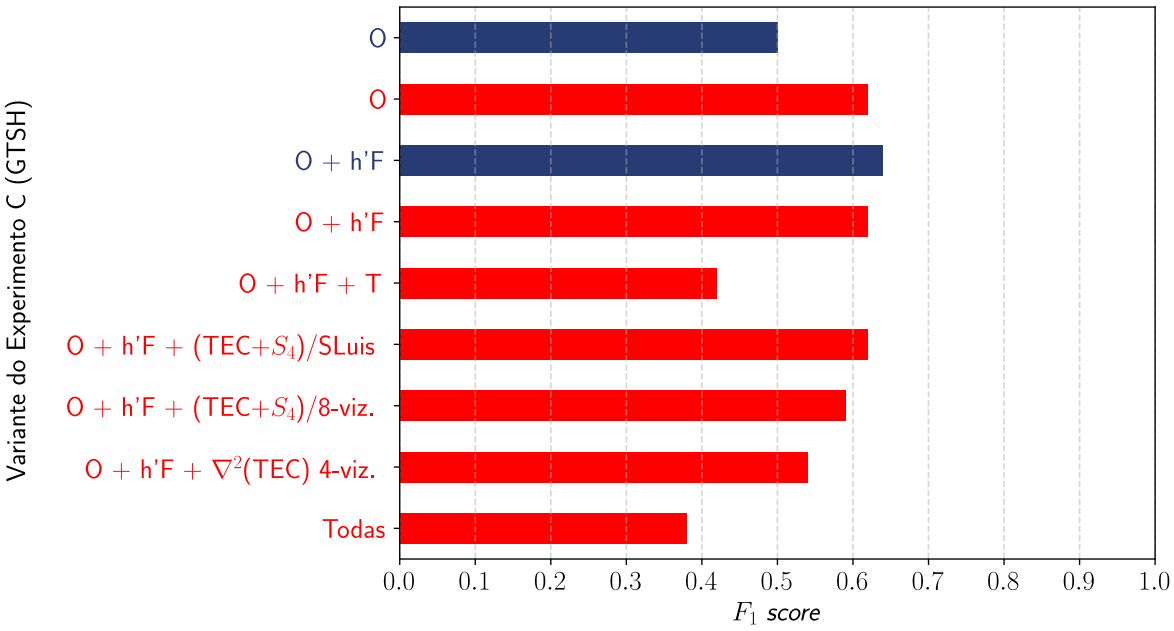

Fonte: Produção do autor.

Figura 6.21 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para antecedência de 60 min.



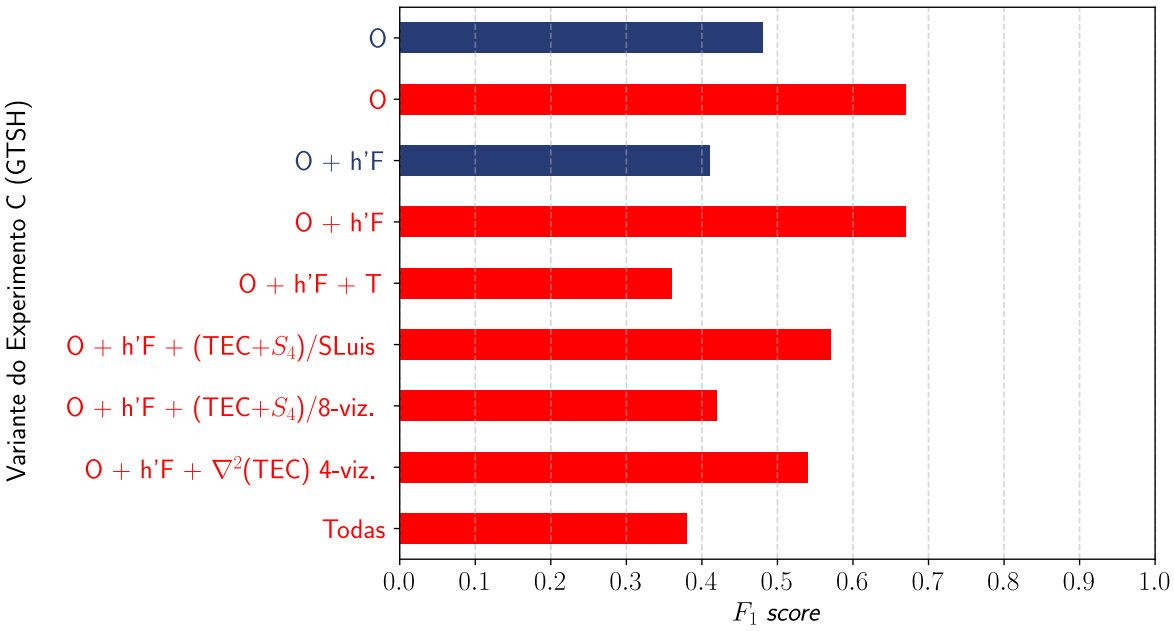

Fonte: Produção do autor.

Figura 6.22 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para antecedência de 90 min.



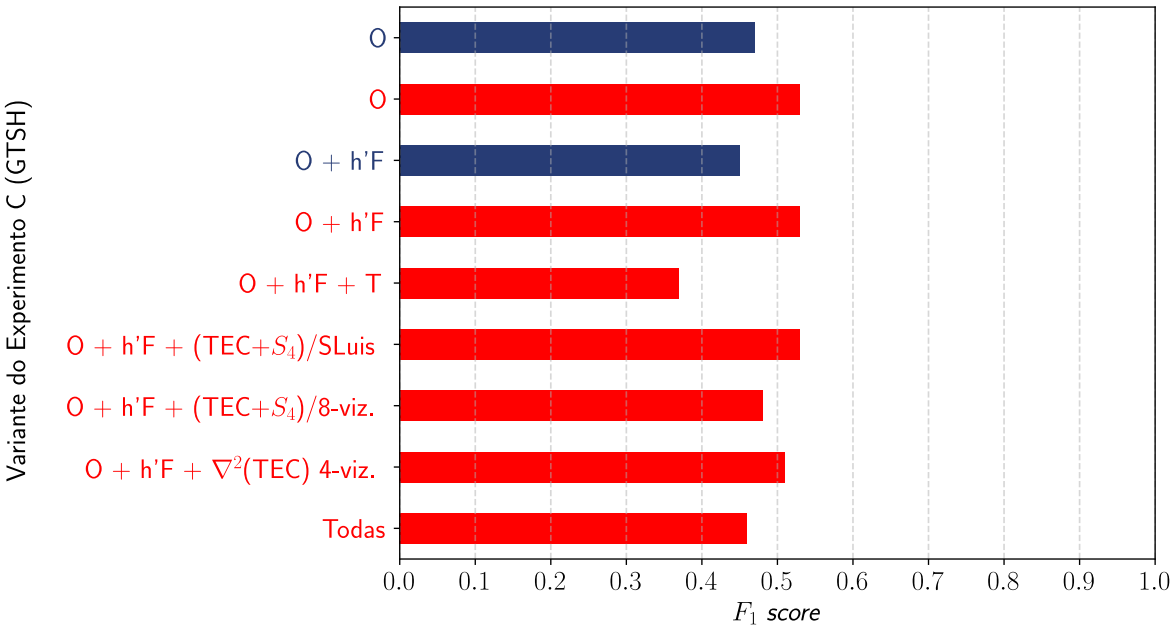

Fonte: Produção do autor.

Figura 6.23 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para antecedência de 120 min.



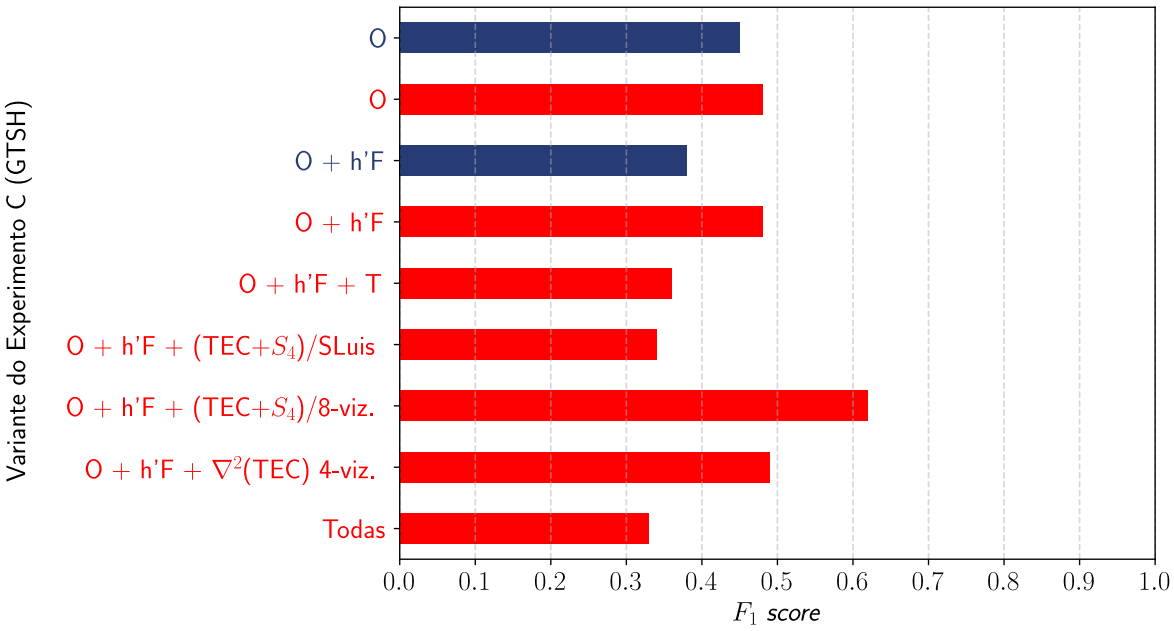

Fonte: Produção do autor.

Figura 6.24 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para antecedência de 150 min.




Fonte: Produção do autor.

Figura 6.25 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para antecedência de 180 min.




Fonte: Produção do autor.

# 7 CONCLUSÕES

O monitoramento e a predição da cintilação ionosférica é um tema de pesquisa corrente e tem importância para emissão de alerta de falha de serviço de sistemas de navegação global por satélite (GNSS), principalmente no tocante à navegação aérea e procedimentos de pouso e decolagem. Na ausência de modelos matemáticos que possam ser executados com resolução espacial e temporal suficiente para possibilitar a predição de cintilação, modelos orientados a dados têm sido propostos. Esses modelos são gerados a partir do treinamento de algoritmos de aprendizado de máquina específico com dados históricos, possibilitando fazer predições com dados novos. Uma busca recente revelou um único trabalho recente propondo a predição de cintilação em baixas latitudes magnéticas (ZHAO et al., 2021), como já mencionado. Entretanto, a predição proposta no referido trabalho fornece um único valor categórico (ocorrência ou ausência de cintilação) para cada noite, o que demonstra as dificuldades inerentes à predição de cintilação, e indica ser esse um tema de pesquisa corrente.

Esta tese apresenta algumas metodologias implementadas para a predição local de cintilação ionosférica, especificamente para a cidade de São José dos Campos (SP), de baixa latitude magnética. Utilizaram-se dados históricos de cintilação ionosférica, atividade solar, geomagnéticos e ionosféricos para os meses de verão do período 2010-2018, abrangendo quase todo o último ciclo solar. Esses dados foram particionados em conjuntos mutuamente exclusivos para treinamento, validação e teste. Um robusto e complexo conjunto de esquemas de pré-processamento, particionamento e validação foi proposto e implementado, de forma a melhor assegurar a isenção dos resultados.

Os algoritmos testados para predição são do tipo *Gradient Boosting Tree*, *Extreme Gradient Boosting* (XGBoost) e *Categorical Boosting* (CatBoost), além de uma rede neural convolucional (Resnet-Wavenet), todos disponíveis no ambiente de programação Python. As predições foram realizadas de forma categórica, ocorrência (OC) ou ausência de cintilação (N-OC), por exemplo, com base no limiar escolhido, de forma a incluir somente cintilação forte, ou forte-moderada, ou forte-moderada-fraca na classe OC. Foram testados diversos esquemas de validação, referentes ao particionamento das amostras de treinamento, validação e teste, os quais resultaram nas correspondentes variantes dos experimentos: Experimento A (variantes GTSH/TSCV-GKF anos 2010-2018, GTSH anos 2012-2014, multi-TSCV-GKF/multi-TSCV-GWF anos 2010-2018), Experimento B (variantes GTSH/VCT anos 2010-2018) e Experi-

mento C (sem variantes). Adicionalmente, foi feita a seleção de atributos com base nos algoritmos de *ensemble*, sendo avaliado o desempenho de predição com (SSA) ou sem (NSA) seleção de atributos. Entretanto, essa avaliação foi quase sempre inconclusiva. Os Experimentos A e C permitem predições a cada 30 minutos para antecedências de 30 a 180 minutos, enquanto que o Experimento B permite uma única predição para a noite inteira. No caso de antecedências de 30 a 180 minutos, as menores antecedências tiveram melhor desempenho de predição, como seria de se esperar.

O desempenho de predição foi avaliado para os algoritmos de *ensemble* e para a rede neural convolucional, obtendo-se resultados similares com os Experimentos A e C para o período 2010-2018, relativo à maioria dos testes. O Experimento B difere muito desses experimentos, mas seu desempenho de predição foi pior.

Outro ponto importante foi a seleção das amostras dos subconjuntos de treinamento, validação e teste. Exceto pela variação multi-TSCV-GKF do Experimento A, todas as variantes dos Experimentos A, B e C, ordenam temporalmente esses subconjuntos. Entretanto, para a maioria das predições, em que foram usados dados do período 2010-2018, isso implicou em treinamento (e, conforme o caso, a validação) com amostras do período de Solar Máximo e teste com amostras do período de Solar Mínimo. Isso torna a predição mais demandante, explicando desempenhos de predição mais baixos. Por outro lado, no Experimento A variante GTSH anos 2012-2014, todos esses subconjuntos continham amostras do período de Solar Máximo, obtendo-se desempenhos de predição melhores.

Objetivando a publicação de um artigo para periódico (SANTOS et al., 2022) (submetido), foi selecionado o Experimento A, variante multi-TSCV-GKF/multi-TSCV-GWF anos 2010-2018, sendo comparado o desempenho de predição de ambos os esquemas de validação, o primeiro sem ordenamento temporal e o segundo, com ordenamento temporal. O primeiro esquema obteve melhor desempenho, mas o segundo esquema reproduz melhor condições reais de predição operacional.

A avaliação por métricas padrão de predição/classificação mostrou resultados promissores para todas as implementações. O desempenho de predição foi também limitado pelos dados disponíveis para esse período devido a problemas técnicos das estações GNSS, à não disponibilidade de dados abrangendo todo o período do ciclo solar e ao desbalanceamento entre classes de intensidade de cintilação nos dados, o qual é decorrente do número relativamente pequeno de amostras com cintilação, comparativamente àquelas sem cintilação.

Assim, as contribuições desta tese são os experimentos realizados (incluindo o projeto/concepção dos mesmos), os esquemas de particionamento/validação dos dados e o ajuste dos algoritmos de aprendizado de máquina propostos para os experimentos, bem como o ajuste de seus hiper-parâmetros e, particularmente, a rede neural proposta apresentada na Seção 4.5.6.

Nesse sentido, espera-se que as metodologias aqui propostas representem um avanço nessa linha de pesquisa. Espera-se também, conforme detalhado na seção seguinte sobre trabalhos futuros, melhorar o desempenho de predição suficientemente para utilização operacional das abordagens propostas no programa EMBRACE/INPE.

## 7.1 Trabalhos futuros

Pretende-se explorar as variantes com ordenamento temporal deste trabalho para o ciclo solar atual (ciclo 25), ou seja, com amostras do ano de 2019 em diante. em que os dados do subconjunto de treinamento e de validação antecedem os dados do subconjunto de teste. Isso permitiria selecionar amostras desses subconjuntos para o mesmo período do ciclo solar (período de ciclo máximo ou mínimo).

Outros trabalhos futuros incluem o eventual refinamento dos modelos pelo ajuste dos hiper-parâmetros dos algoritmos, e a geração de novos modelos utilizando o maior volume de dados disponível a partir de 2019, decorrente da operacionalização mais completa das 4 redes de estações GNSS existentes no Brasil (LISN, CIGALA-CALIBRA/INCT, ICEA, RBMC/IBGE). Outro item seria a extensão das predições propostas para outros locais do território brasileiro com estações GNSS, incluindo a possibilidade de usar um conjunto de dados comum a todas as estações para treinamento e validação dos modelos, embora executando os testes de predição separadamente para cada estação. Adicionalmente, pode-se considerar a inclusão nos dados das coordenadas magnéticas das estações GNSS.

Finalmente, quaisquer trabalhos futuros na linha proposta demandam processamento de alto desempenho, seja processamento paralelo provido por múltiplos nós de memória compartilhada, cada um com processadores multi-núcleo, seja por um ou mais nós providos de placas aceleradoras gráficas (GPUs). A tendência atual é de um volume crescente de dados e de algoritmos e esquemas de validação cada vez mais complexos e portanto com alta carga computacional.

# REFERÊNCIAS BIBLIOGRÁFICAS


ABDU, M. Equatorial ionosphere–thermosphere system: electrodynamics and irregularities. **Advances in Space Research**, v. 35, n. 5, p. 771 – 787, 2005. ISSN 0273-1177. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0273117705004898>. 14

ALFONSI, L.; SPOGLI, L.; FRANCESCHI, G. D.; ROMANO, V.; AQUINO, M.; DODSON, A.; MITCHELL, C. N. Bipolar climatology of gps ionospheric scintillation at solar minimum. **Radio Science**, v. 46, n. 3, 2011. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010RS004571>. 94

ATABATI, A.; ALIZADEH, M.; SCHUH, H.; TSAI, L.-C. Ionospheric scintillation prediction on s4 and roti parameters using artificial neural network and genetic algorithm. **Remote Sensing**, v. 13, n. 11, 2021. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/13/11/2092>. 3

BARANDAS, M.; FOLGADO, D.; FERNANDES, L.; SANTOS, S.; ABREU, M.; BOTA, P.; LIU, H.; SCHULTZ, T.; GAMBOA, H. Tsfel: time series feature extraction library. **SoftwareX**, v. 11, p. 100456, 2020. ISSN 2352-7110. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352711020300017>. 48

BATISTA, I. S. **Dínamo da região F Equatorial: assimetrias sazonais e longitudinais no setor americano**. 169 p. INPE-3760-TDL/206. Tese (Doutorado em Ciência Espacial) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 1986. Disponível em: <http://marte3.sid.inpe.br/col/sid.inpe.br/iris@1905/2005/07.26.21.28.37/doc/publicacao.pdf>. 15, 17

BÉNIGUEL, Y.; HAMEL, P. A global ionosphere scintillation propagation model for equatorial regions. **Journal of Space Weather and Space Climate**, v. 1, n. 1, p. A04, 2011. Disponível em: <https://doi.org/10.1051/swsc/2011004>. 2

BERTONI, F. C. P. **Estudos de derivas ionosféricas por meio de ionosondas digitais**. 135 p. INPE-7169-TDI/675. Dissertação (Mestrado em Geofísica Espacial) — Instituto Nacional de Pesquisas Espaciais (INPE), São José

dos Campos, 1998. Disponível em:
<http://urlib.net/sid.inpe.br/deise/1999/09.14.13.10>. 20

BHOMWIK, P.; NANDY, D. Prediction of the strength and timing of sunspot cycle 25 reveal decadal-scale space environmental conditions. **Nature Communications**, v. 9, 12 2018. 24

BISHOP, C. M. **Pattern recognition and machine learning (information science and statistics)**. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738. 47

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, Aug 1996. ISSN 1573-0565. Disponível em:
<https://doi.org/10.1023/A:1018054314350>. 54

CAMPOREALE, E. The challenge of machine learning in space weather: nowcasting and forecasting. **Space Weather**, v. 0, n. ja, 2019. Disponível em:
<https:
//agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018SW002061>. 2

CARMO, C. de sousa do. **Estudo de diferentes técnicas para o cálculo do conteúdo eletrônico total absoluto na ionosfera equatorial e de baixas latitudes**. 149 p. (02.21.19.14-TDI). Dissertação (Mestrado em Geofísica Espacial/Ciências do Ambiente Solar-Terrestre) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2018. Disponível em:
<http://urlib.net/8JMKD3MGP3W34P/3QJNJHE>. 91

CHEN, T.; GUESTRIN, C. Xgboost: a scalable tree boosting system. In: ACM SIGKDD INTERNACIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 22., 2016. **Proceedings...** New York: ACM, 2016. p. 785–794. ISBN 978-1-4503-4232-2. Disponível em:
<http://doi.acm.org/10.1145/2939672.2939785>. 57, 60

CHRIST, M.; BRAUN, N.; NEUFFER, J.; KEMPA-LIEHR, A. W. Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package). **Neurocomputing**, v. 307, p. 72–77, 2018. ISSN 0925-2312. Disponível em: <https:
//www.sciencedirect.com/science/article/pii/S0925231218304843>. 48

CHRIST, M.; KEMPA-LIEHR, A. W.; FEINDT, M. Distributed and parallel time series feature extraction for industrial big data applications. **arXiv e-prints**, p. arXiv:1610.07717, out. 2016. 48

DOROGUSH, A. V.; ERSHOV, V.; GULIN, A. Catboost: gradient boosting with categorical features support. **ArXiv**, abs/1810.11363, 2018. 61

FULCHER, B. D. Feature-based time-series analysis. **arXiv e-prints**, p. arXiv:1709.08055, set. 2017. 48

GONZALEZ, W. D.; JOSELYN, J. A.; KAMIDE, Y.; KROEHL, H. W.; ROSTOKER, G.; TSURUTANI, B. T.; VASYLIUNAS, V. M. What is a geomagnetic storm? **Journal of Geophysical Research: Space Physics**, v. 99, n. A4, p. 5771–5792, 1994. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/93JA02867>. 18

GRÄLER, B.; PEBESMA, E.; HEUVELINK, G. Spatio-temporal interpolation using gstat. **The R Journal**, v. 8, n. 1, p. 204–218, 2016. Disponível em: <https://doi.org/10.32614/RJ-2016-014>. 95

HAMEL, P.; SAMBOU, D. C.; DARCES, M.; BENIGUEL, Y.; HéLIER, M. Kriging method to perform scintillation maps based on measurement and gism model. **Radio Science**, v. 49, n. 9, p. 746–752, 2014. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014RS005470>. 95

HANSEN, L. K.; SALAMON, P. Neural network ensembles. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 12, n. 10, p. 993–1001, out. 1990. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/34.58871>. 54

HATHAWAY, D. H. The solar cycle. **Living Reviews in Solar Physics**, v. 12, n. 1, p. 4, set. 2015. 23, 24

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016. **Proceedings...** [S.l.]: IEEE, 2016. p. 770–778. 80

HENDRYCKS, D.; GIMPEL, K. Gaussian Error Linear Units (GELUs). **arXiv e-prints**, p. arXiv:1606.08415, jun. 2016. 77

HOFMANN-WELLENHOF, B.; LICHTENEGGER, H.; COLLINS, J. **Global positioning system: theory and practice**. Vienna: Springer, 2013. ISBN 9783709133118. Disponível em: <https://books.google.com.br/books?id=bQntCAAAQBAJ>. 20

IOFFE, S.; SZEGEDY, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 32., 2015. **Proceedings...** [S.l.]: JMLR, 2016. p. 448–456. 78

KAMIDE, Y.; CHIAN, A. **Handbook of the solar-terrestrial environment**. Berlin: Springer, 2007. Disponível em: <https://cds.cern.ch/record/1338942>. 9

KELLEY, M. Ionosphere. In: HOLTON, J. R. (Ed.). **Encyclopedia of Atmospheric Sciences**. Oxford: Academic Press, 2003. p. 1022–1030. ISBN 978-0-12-227090-1. Disponível em: <https://www.sciencedirect.com/science/article/pii/B0122270908001846>. 8

KIEFT, P.; AQUINO, M.; DODSON, A. Using ordinary kriging for the creation of scintillation maps. In: NOTARPIETRO, R.; DOVIS, F.; FRANCESCHI, G. D.; AQUINO, M. (Ed.). **Mitigation of ionospheric threats to GNSS**. Rijeka: IntechOpen, 2014. cap. 6. Disponível em: <https://doi.org/10.5772/58781>. 95

KINGMA, D.; BA, J. Adam: a method for stochastic optimization. In: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS, 12., 2014. **Proceedings...** [S.l.], 2014. 102

KIRCHHOFF, V. W. J. H. **Introdução à geofísica espacial**. [S.l.: s.n.], 1991. 152 p. ISBN 9788572330015. 8

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Communications of the ACM**, v. 60, n. 6, p. 84–90, maio 2017. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3065386>. 62

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. 67

LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. **Journal of Machine Learning Research**, v. 18, n. 17, p. 1–5, 2017. Disponível em: <http://jmlr.org/papers/v18/16-365>. 93

LIMA, G. R. T. d.; STEPHANY, S.; PAULA, E. R. d.; BATISTA, I. S.; ABDU, M. A.; REZENDE, L. F. C.; AQUINO, M. G. S.; DUTRA, A. P. S. Correlation analysis between the occurrence of ionospheric scintillation at the magnetic equator and at the southern peak of the equatorial ionization anomaly. **Space Weather**, v. 12, n. 6, p. 406–416, June 2014. ISSN 1542-7390. 2

LIMA, G. R. T. d.; STEPHANY, S.; PAULA, E. R. d.; BATISTA, I. S.; ABDU, M. A. Prediction of the level of ionospheric scintillation at equatorial latitudes in brazil using a neural network. **Space Weather**, v. 13, n. 8, p. 446–457, Aug 2015. ISSN 1542-7390. 2

MCGRANAGHAN, R. M.; MANNUCCI, A. J.; WILSON, B.; MATTMANN, C. A.; CHADWICK, R. New capabilities for prediction of high-latitude ionospheric scintillation: a novel approach with machine learning. **Space Weather**, v. 16, n. 11, p. 1817–1846, 2018. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018SW002018>. 3

MEZIANE, K.; KASHCHEYEV, A.; JAYACHANDRAN, P. T.; HAMZA, A. M. A bayesian inference-based empirical model for scintillation indices for high-latitude. **Space Weather**, v. 19, n. 6, p. e2020SW002710, 2021. E2020SW002710 2020SW002710. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020SW002710>. 3

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: INTERNATIONAL CONFERENCE ON INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 27., 2010. **Proceedings...** Madison: Omnipress, 2010. p. 807–814. ISBN 9781605589077. 77

National Oceanic and Atmospheric Administration (NOAA). **Geomagnetic kp and ap Indices**. 1999. Disponível em: <url{https://www.ngdc.noaa.gov/stp/GEOMAG/kp_ap.html}>. Acesso em: 03 maio 2019. 19

NEGRETI, P. M. S. **Estudo do conteúdo eletrônico total na região brasileira em períodos magneticamente perturbados**. 323 p. Sid.inpe.br/mtc-m19/2012/05.10.21.43-TDI. Tese (Doutorado em Geofísica Espacial/Ciências do Ambiente Solar-Terrestre) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2012. Disponível em: <http://urlib.net/sid.inpe.br/mtc-m19/2012/05.10.21.43>. 7

OORD, A. V. D.; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A. W.; KAVUKCUOGLU, K. Wavenet: a generative model for raw audio. **SSW**, v. 125, p. 2, 2016. 82

OTSUKA, Y.; OGAWA, T.; SAITO, A.; TSUGAWA, T.; FUKAO, S.; MIYAZAKI, S. New technique for mapping of total electron content using gps network in Japan. **Earth and Planetary Science Letters**, v. 54, p. 63–70, 01 2002. 91

PAULA, E. R. de; MARTINON, A. R. F.; MORAES, A. O.; CARRANO, C.; NETO, A. C.; DOHERTY, P.; GROVES, K.; VALLADARES, C. E.; CROWLEY, G.; AZEEM, I.; REYNOLDS, A.; AKOS, D. M.; WALTER, T.; BEACH, T. L.; SLEWAEGEN, J.-M. Performance of 6 different global navigation satellite system receivers at low latitude under moderate and strong scintillation. **Earth and Space Science**, v. 8, n. 2, p. e2020EA001314, 2021. E2020EA001314 2020EA001314. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020EA001314>. 94

PEBESMA, E. J. Multivariable geostatistics in s: the gstat package. **Computers 'I&' Geosciences**, v. 30, n. 7, p. 683 – 691, 2004. ISSN 0098-3004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0098300404000676>. 95

PROKHORENKOVA, L.; GUSEV, G.; VOROBEV, A.; DOROGUSH, A. V.; GULIN, A. Catboost: unbiased boosting with categorical features. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 32., 2018. **Proceedings...** Red Hook: Curran Associates, 2018. (NIPS'18), p. 6639–6649. 61

PROL, F.; CAMARGO, P. Review of tomographic reconstruction methods of the ionosphere using GNSS. **Revista Brasileira de Geofísica**, v. 33, 08 2016. 94

RETTERER, J. M. Forecasting low-latitude radio scintillation with 3-d ionospheric plume models: 2. scintillation calculation. **Journal of Geophysical Research: Space Physics**, v. 115, n. A3, 2010. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JA013840>. 2

REZENDE, L. F. **Mineração de dados aplicada à análise e predição de cintilação ionosférica**. 176 p. (INPE-16080-TDI/1537). Dissertação (Mestrado em Compuatação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2009. Disponível em: <http://urlib.net/sid.inpe.br/mtc-m18@80/2009/06.22.15.52>. 2

REZENDE, L. F. C.; PAULA, E. R. de; STEPHANY, S.; KANTOR, I. J.; MUELLA, M. T. A. H.; SIQUEIRA, P. M. de; CORREA, K. S. Survey and prediction of the ionospheric scintillation using data mining techniques. **Space Weather**, v. 8, n. 6, 2010. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2009SW000532>. 2

REZENDE, L. F. C. de; PAULA, E. R. de; KANTOR, I. J.; KINTNER, P. M. Mapping and survey of plasma bubbles over brazilian territory. **Journal of Navigation**, v. 60, n. 1, p. 69–81, 2007. 95

RISHBETH, H.; GARRIOTT, O. K. **Introduction to ionospheric physics**. [S.l.: s.n.], 1969. 7

ROEDERER, J. Earth's magnetosphere - global problems in magnetospheric plasma physics. 02 1979. 13

SANTOS, P. A. dos; STEPHANY, S.; PAULA, E. R. de. A new approach for low-latitude ionospheric scintillation prediction. **Computer & Geociences**, 2022. Submetido. 122

SCHÄFER, P.; LESER, U. Multivariate time series classification with WEASEL+MUSE. **arXiv e-prints**, p. arXiv:1711.11343, nov. 2017. 48

SCHAPIRE, R. E. The strength of weak learnability. **Machine Learning**, v. 5, n. 2, p. 197–227, Jun 1990. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00116037>. 54

SPOGLI, L.; ALFONSI, L.; FRANCESCHI, G. D.; ROMANO, V.; AQUINO, M. H. O.; DODSON, A. Climatology of gps ionospheric scintillations over high and mid-latitude european regions. **Annales Geophysicae**, v. 27, n. 9, p. 3429–3437, 2009. Disponível em: <https://www.ann-geophys.net/27/3429/2009/>. 94

STAFFORD, J. V. Implementing precision agriculture in the 21st century. **Journal of Agricultural Engineering Research**, v. 76, n. 3, p. 267 – 275, 2000. ISSN 0021-8634. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0021863400905778>. 2

TAKAHASHI, H.; WRASSE, C. M.; DENARDINI, C. M.; PáDUA, M. B.; PAULA, E. R.; COSTA, S. M. A.; OTSUKA, Y.; SHIOKAWA, K.; MONICO, J. F. G.; IVO, A.; SANT'ANNA, N. Ionospheric tec weather map over South America. **Space Weather**, v. 14, n. 11, p. 937–949, 2016. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016SW001474>. 1, 21

VANI, B. C. **Investigações sobre modelagem, mitigação e predição de cintilação ionosférica na região brasileira**. 182 p. Tese (Doutorado em Ciências Cartográficas) — Universidade Estadual Paulista. Faculdade de Ciências e Tecnologia, Presidente Prudente, Presidente Prudente, 2018. Disponível em: <http://hdl.handle.net/11449/153701>. 94

WERNIK, A. W.; ALFONSI, L.; MATERASSI, M. Scintillation modeling using in situ data. **Radio Science**, v. 42, n. 1, 2007. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006RS003512>. 2

ZHAO, X.; LI, G.; XIE, H.; HU, L.; SUN, W.; YANG, S.; LI, Y.; NING, B.; TAKAHASHI, H. The prediction of day-to-day occurrence of low latitude ionospheric strong scintillation using gradient boosting algorithm. **Space Weather**, v. 19, n. 12, p. e2021SW002884, 2021. E2021SW002884 2021SW002884. Disponível em: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021SW002884>. 3, 4, 121

# APÊNDICE A RESULTADOS EXTRAS DOS EXPERIMENTOS

## A.1 Experimento A (GTSH/TSCV-GKF) anos 2010-2018

Os resultados do **Experimento A (GTSH/TSCV-GKF) anos 2010-2018** estão organizados em 4 grupos de 6 tabelas, o primeiro número corresponde às variações na configuração do experimento, enquanto o segundo está associado às diferentes antecedências de predição, de 30-60-90-120-150-180 min. As tabelas apresentam as seguintes colunas: Período, Modelo, Acurácia, Precisão, $F_1$. A coluna Período indica o período da noite relativo ao horário-alvo das predições. A coluna Modelo indica se o algoritmo empregado foi o XGBoost ou CatBoost, enquanto as demais colunas apresentam as métricas de desempenho. Cada métrica de desempenho apresenta duas subcolunas que indicam o valor resultante da métrica sem utilizar seleção de atributos (NSA) ou com seleção de atributos (SSA). Caso essa seleção resulte numa métrica melhor, o resultado correspondente é marcado com o símbolo $^*$.

As Tabelas A.1, A.3, A.5, A.7, A.9 e A.11 apresentam os resultados para o **Experimento A (GTSH/TSCV-GKF) anos 2010-2018** empregando como método de validação o GTSH e utilizando adicionalmente os dados de ionossonda, variável $h'F$, enquanto que as Tabelas A.2, A.4, A.6, A.8, A.10 e A.12 apresentam as Matrizes de Confusão para o modelo treinado considerando o período das 18-06 h.

De maneira geral, analisando estes resultados, pode-de afirmar que existem casos onde a seleção de atributos melhora o desempenho de predição e outros onde ela não melhora, sempre considerando o subconjunto de teste. Note-se que o subconjunto de teste é mantido fixo durante todo o Experimento A (GTSH/TSCV-GKF) anos 2010-2018, e o subconjunto de validação muda em função do esquema de validação adotado em cada caso. A validação TSCV-GKF deve gerar um modelo mais independente do subconjunto de validação, pois envolve uma média entre múltiplos subconjuntos de validação.

Pode-se afirmar ainda que as métricas de desempenho de predição, principalmente o $F_1$ e a precisão, degradam rapidamente com o aumento da antecedência de predição, como seria de se esperar. A cintilação ocorre com maior frequência antes das 0 h (meia noite), e é possível observar uma tendência do $F_1$ ser menor neste período em comparação ao período após 0 h, o que indica possivelmente que a predição para a primeira parte da noite possa ser mais difícil.

Em relação ao uso de seleção de atributos (NSA ou SSA), existem alguns casos

atípicos, no sentindo que o $F_1$ apresenta um bom resultado em um ou outro algoritmo, mas não em ambos, por exemplo, na Tabela A.11 é estranho observar um aumento de $F_1$ de $0,47$ para $0,85$ com seleção de atributos, no modelo XGBoost, quando comparado com o correspondente modelo CatBoost. Este tipo de comportamento pode indicar que a abordagem de seleção de atributos empregada pode não ser completamente capaz de tratar o problema, principalmente quando levando em consideração que o subconjunto de validação e o de teste contém amostras de anos diferentes do ciclo solar.

Finalmente, comparando os resultados de $F_1$ para o período das 18-24 h, das 24-06 h e das 18-06 h, é possível notar que a utilização de modelos diferentes para a predição de valores das 18-24 h e das 24-06 h é possivelmente melhor que um único modelo para o período 18-06 h. A configuração com período das 18-06 h cujos resultados são apresentados nas Tabelas A.1, A.3, A.5, A.7, A.9 e A.11, também foi avaliada considerando o uso de 25 sementes diferentes para geração de números aleatórios, mas os valores médios das métricas ficaram próximos dos valores já apresentados com semente única. Como exemplo, o desvio padrão do $F_1$ resultante da utilização de várias sementes foi de 0,04 para a predição com antecedência de 30 min (menor desvio padrão observado) e chegou até 0,09 para antecedência de 180 min (maior desvio padrão observado). Isso parece indicar que os algoritmos de *ensemble* utilizados são robustos quanto à escolha da semente para geração de números aleatórios.

As Tabelas A.13 à A.18 apresentam os resultados para o **Experimento A (GTSH/TSCV-GKF) anos 2010-2018** empregando TSCV-GKF e utilizando dados de ionossonda, diferindo assim do teste anterior unicamente pelo esquema de validação. Os resultados também são muito semelhantes apesar do esquema de validação diferente, mas o subconjunto de teste é mantido fixo para todos os casos do Experimento A (GTSH/TSCV-GKF) anos 2010-2018. Porém, com este esquema de validação, não se observaram métricas $F_1$ atípicas em relação à variante utilizada quanto ao uso ou não de seleção de atributos (SSA ou NSA).

Duas variações do Experimento A (GTSH/TSCV-GKF) anos 2010-2018 foram realizadas sem utilização de dados de ionossonda (variável $h'F$). As Tabelas A.19 à A.24 apresentam os resultados para a variante que empregou como método de validação o GTSH, enquanto as Tabelas A.25 à A.30 apresenta os resultados para a variante que empregou TSCV-GKF.

Os desempenhos de predição com e sem dados de ionossonda se mostraram relativamente equivalentes, sendo que em alguns casos o uso dessa variável melhorava o

desempenho e em outros piorava. Considerando-se seleção de atributos, o aumento do número de atributos não necessariamente melhora o desempenho de predição, além de aumentar a dimensionalidade das amostras do conjunto de treinamento. Neste experimento, na geração do vetor de atributos derivado dos atributos de predição, a adição de uma nova variável implica em novas combinações dos atributos preditores (por exemplo, com base na média, desvio padrão, etc. de cada atributo) e portanto pode melhorar ou não o desempenho de predição. A técnica de seleção de atributos empregada, por sua vez, parece não ser suficientemente capaz de selecionar um subconjunto de atributos ótimo em todas as circunstâncias, e é portanto um ponto a ser investigado no futuro.

Tabela A.1 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 30 min com dados de ionossonda e validação GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,83 | 0,81 | 0,80 | 0,76 | 0,81 | 0,77 |
| | CatBoost | 0,78 | 0,69 | 0,73 | 0,72 | 0,73 | 0,61 |
| 24-06 h | XGBoost | 0,85 | 0,84 | 0,79 | 0,77 | 0,81 | 0,79 |
| | CatBoost | 0,87 | 0,90* | 0,82 | 0,87* | 0,84 | 0,89* |
| 18-06 h | XGBoost | 0,78 | 0,79* | 0,70 | 0,73* | 0,72 | 0,74* |
| | CatBoost | 0,81 | 0,80 | 0,74 | 0,73 | 0,76 | 0,75 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.2 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Matriz de Confusão para a predição 30 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,72$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 704 | 4 |
| | OC | 230 | 167 |

| CatBoost-NSA $F_1 = 0,76$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 701 | 1 |
| | OC | 196 | 201 |

| XGBoost-SSA $F_1 = 0,74$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 698 | 10 |
| | OC | 214 | 183 |

| CatBoost-SSA $F_1 = 0,75$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 694 | 14 |
| | OC | 205 | 192 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.3 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 60 min com dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,67 | 0,72* | 0,60 | 0,65* | 0,59 | 0,64* |
| | CatBoost | 0,62 | 0,66* | 0,50 | 0,57* | 0,39 | 0,54* |
| 24-06 h | XGBoost | 0,75 | 0,78* | 0,65 | 0,69* | 0,66 | 0,70* |
| | CatBoost | 0,85 | 0,81 | 0,80 | 0,74 | 0,82 | 0,76 |
| 18-06 h | XGBoost | 0,72 | 0,71 | 0,62 | 0,59 | 0,61 | 0,57 |
| | CatBoost | 0,77 | 0,73 | 0,68 | 0,64 | 0,69 | 0,64 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.4 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Matriz de Confusão para a predição 60 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,61$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 703 | 9 |
| | OC | 291 | 102 |

| CatBoost-NSA $F_1 = 0,69$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 694 | 18 |
| | OC | 241 | 152 |

| XGBoost-SSA $F_1 = 0,57$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 701 | 11 |
| | OC | 313 | 80 |

| CatBoost-SSA $F_1 = 0,64$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 694 | 18 |
| | OC | 275 | 118 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.5 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 90 min com dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,62 | 0,70* | 0,50 | 0,67* | 0,42 | 0,68* |
| | CatBoost | 0,62 | 0,73* | 0,49 | 0,68* | 0,39 | 0,69* |
| 24-06 h | XGBoost | 0,69 | 0,69 | 0,57 | 0,57 | 0,53 | 0,53 |
| | CatBoost | 0,81 | 0,80 | 0,74 | 0,73 | 0,76 | 0,75 |
| 18-06h | XGBoost | 0,66 | 0,67* | 0,53 | 0,55* | 0,46 | 0,52* |
| | CatBoost | 0,73 | 0,69 | 0,65 | 0,58 | 0,65 | 0,56 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.6 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Matriz de Confusão para a predição 90 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,46$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 706 | 7 |
| | OC | 364 | 28 |

| CatBoost-NSA $F_1 = 0,65$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 664 | 49 |
| | OC | 249 | 143 |

| XGBoost-SSA $F_1 = 0,52$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 690 | 23 |
| | OC | 336 | 56 |

| CatBoost-SSA $F_1 = 0,64$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 684 | 29 |
| | OC | 313 | 79 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.7 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 120 min com dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,77 | 0,64 | 0,71 | 0,50 | 0,73 | 0,39 |
| | CatBoost | 0,65 | 0,67* | 0,51 | 0,55* | 0,44 | 0,52* |
| 24-06 h | XGBoost | 0,67 | 0,67 | 0,52 | 0,52 | 0,44 | 0,44 |
| | CatBoost | 0,80 | 0,73 | 0,73 | 0,63 | 0,74 | 0,62 |
| 18-06 h | XGBoost | 0,66 | 0,66 | 0,54 | 0,52 | 0,49 | 0,48 |
| | CatBoost | 0,67 | 0,66 | 0,54 | 0,53 | 0,44 | 0,47 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.8 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Matriz de Confusão para a predição 120 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,49$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 687 | 25 |
| | OC | 347 | 46 |

| CatBoost-NSA $F_1 = 0,48$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 700 | 12 |
| | OC | 354 | 39 |

| XGBoost-SSA $F_1 = 0,44$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 710 | 2 |
| | OC | 374 | 19 |

| CatBoost-SSA $F_1 = 0,47$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 693 | 19 |
| | OC | 357 | 36 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.9 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 150 min com dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,66 | 0,82* | 0,50 | 0,84* | 0,40 | 0,82* |
| | CatBoost | 0,66 | 0,66 | 0,50 | 0,50 | 0,40 | 0,40 |
| 24-06 h | XGBoost | 0,66 | 0,67* | 0,53 | 0,53 | 0,45 | 0,45 |
| | CatBoost | 0,70 | 0,66 | 0,59 | 0,52 | 0,58 | 0,43 |
| 18-06 h | XGBoost | 0,68 | 0,65 | 0,56 | 0,51 | 0,53 | 0,41 |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.10 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Matriz de Confusão para a predição 150 min à frente com dados de ionossonda e GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,53$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 685 | 26 |
| | OC | 329 | 65 |

| CatBoost-NSA $F_1 = 0,39$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 710 | 1 |
| | OC | 393 | 1 |

| XGBoost-SSA $F_1 = 0,41$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 706 | 5 |
| | OC | 386 | 8 |

| CatBoost-SSA $F_1 = 0,39$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 711 | 0 |
| | OC | 394 | 0 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.11 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 180 min com dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,67 | 0,86* | 0,52 | 0,85* | 0,47 | 0,85* |
| | CatBoost | 0,67 | 0,67 | 0,50 | 0,50 | 0,40 | 0,40 |
| 24-06 h | XGBoost | 0,63 | 0,74* | 0,50 | 0,67* | 0,40 | 0,68* |
| | CatBoost | 0,63 | 0,94* | 0,51 | 0,95* | 0,39 | 0,94* |
| 18-06 h | XGBoost | 0,68 | 0,68 | 0,58 | 0,56 | 0,54 | 0,53 |
| | CatBoost | 0,66 | 0,64 | 0,56 | 0,50 | 0,54 | 0,40 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.12 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Matriz de Confusão para a predição 180 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,54$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 679 | 33 |
| | OC | 322 | 71 |

| CatBoost-NSA $F_1 = 0,54$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 652 | 60 |
| | OC | 312 | 81 |

| XGBoost-SSA $F_1 = 0,53$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 692 | 20 |
| | OC | 331 | 62 |

| CatBoost-SSA $F_1 = 0,40$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 712 | 0 |
| | OC | 390 | 3 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.13 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 30 min com dados de ionossonda e TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,81 | 0,80 | 0,76 | 0,76 | 0,77 | 0,77 |
| | CatBoost | 0,83 | 0,81 | 0,78 | 0,77 | 0,80 | 0,78 |
| 24-06 h | XGBoost | 0,86 | 0,80 | 0,80 | 0,80 | 0,82 | 0,82 |
| | CatBoost | 0,88 | 0,86 | 0,83 | 0,81 | 0,85 | 0,83 |
| 18-06 h | XGBoost | 0,81 | 0,84* | 0,75 | 0,79* | 0,77 | 0,81* |
| | CatBoost | 0,83 | 0,83 | 0,76 | 0,78* | 0,79 | 0,80* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.14 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 60 min com dados de ionossonda e TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,62 | 0,67* | 0,50 | 0,57* | 0,39 | 0,52* |
| | CatBoost | 0,63 | 0,65* | 0,51 | 0,54* | 0,41 | 0,47* |
| 24-06 h | XGBoost | 0,75 | 0,81* | 0,65 | 0,74* | 0,65 | 0,76* |
| | CatBoost | 0,80 | 0,76 | 0,72 | 0,67 | 0,75 | 0,68 |
| 18-06 h | XGBoost | 0,70 | 0,70 | 0,58 | 0,59 | 0,55 | 0,86 |
| | CatBoost | 0,70 | 0,71* | 0,60 | 0,60 | 0,58 | 0,59 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.15 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 90 min com dados de ionossonda e TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,63 | 0,64* | 0,50 | 0,52* | 0,39 | 0,44* |
| | CatBoost | 0,63 | 0,64* | 0,50 | 0,50 | 0,39 | 0,39 |
| 24-06 h | XGBoost | 0,71 | 0,69 | 0,58 | 0,56 | 0,55 | 0,51 |
| | CatBoost | 0,74 | 0,69 | 0,63 | 0,60 | 0,64 | 0,58 |
| 18-06 h | XGBoost | 0,66 | 0,69* | 0,52 | 0,57* | 0,43 | 0,53* |
| | CatBoost | 0,68 | 0,69* | 0,52 | 0,56* | 0,44 | 0,53* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.16 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 120 min com dados de ionossonda e TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,64 | 0,65* | 0,50 | 0,50 | 0,39 | 0,39 |
| | CatBoost | 0,64 | 0,65* | 0,50 | 0,51* | 0,39 | 0,41* |
| 24-06 h | XGBoost | 0,67 | 0,66 | 0,52 | 0,51 | 0,44 | 0,41 |
| | CatBoost | 0,67 | 0,68* | 0,52 | 0,54* | 0,45 | 0,49* |
| 18-06 h | XGBoost | 0,64 | 0,66* | 0,51 | 0,52* | 0,42 | 0,44* |
| | CatBoost | 0,65 | 0,65 | 0,51 | 0,51 | 0,41 | 0,42* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.17 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 150 min com dados de ionossonda e TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,66 | 0,66 | 0,50 | 0,51* | 0,40 | 0,42* |
| | CatBoost | 0,66 | 0,66 | 0,50 | 0,50 | 0,39 | 0,39 |
| 24-06 h | XGBoost | 0,65 | 0,65 | 0,50 | 0,51* | 0,39 | 0,42* |
| | CatBoost | 0,68 | 0,66 | 0,55 | 0,53 | 0,49 | 0,45 |
| 18-06 h | XGBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.18 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 180 min com dados de ionossonda e TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,67 | 0,67 | 0,50 | 0,50 | 0,40 | 0,40 |
| | CatBoost | 0,67 | 0,67 | 0,50 | 0,50 | 0,40 | 0,40 |
| 24-06 h | XGBoost | 0,62 | 0,72* | 0,50 | 0,50 | 0,38 | 0,39* |
| | CatBoost | 0,63 | 0,63 | 0,51 | 0,51 | 0,41 | 0,40 |
| 18-06 h | XGBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.19 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 30 min sem dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,73 | 0,80* | 0,66 | 0,75* | 0,65 | 0,76* |
| | CatBoost | 0,78 | 0,83* | 0,72 | 0,80* | 0,73 | 0,81* |
| 24-06 h | XGBoost | 0,86 | 0,86 | 0,79 | 0,79 | 0,82 | 0,82 |
| | CatBoost | 0,88 | 0,88 | 0,83 | 0,83 | 0,85 | 0,85 |
| 18-06 h | XGBoost | 0,79 | 0,85* | 0,71 | 0,80* | 0,72 | 0,82* |
| | CatBoost | 0,85 | 0,83 | 0,80 | 0,79 | 0,82 | 0,80 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.20 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 60 min sem dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,68 | 0,68 | 0,61 | 0,61 | 0,61 | 0,61 |
| | CatBoost | 0,74 | 0,79* | 0,67 | 0,77* | 0,68 | 0,77* |
| 24-06 h | XGBoost | 0,73 | 0,73 | 0,62 | 0,62 | 0,61 | 0,61 |
| | CatBoost | 0,82 | 0,83* | 0,75 | 0,77* | 0,77 | 0,79* |
| 18-06 h | XGBoost | 0,73 | 0,70 | 0,63 | 0,58 | 0,62 | 0,55 |
| | CatBoost | 0,71 | 0,74* | 0,60 | 0,65* | 0,58 | 0,65* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.21 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 90 min sem dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,67 | 0,73* | 0,56 | 0,69* | 0,53 | 0,70* |
| | CatBoost | 0,64 | 0,71* | 0,52 | 0,62* | 0,42 | 0,61* |
| 24-06 h | XGBoost | 0,68 | 0,68 | 0,54 | 0,54 | 0,49 | 0,49 |
| | CatBoost | 0,80 | 0,80 | 0,73 | 0,73 | 0,75 | 0,75 |
| 18-06 h | XGBoost | 0,68 | 0,67 | 0,55 | 0,55 | 0,50 | 0,51* |
| | CatBoost | 0,70 | 0,70 | 0,58 | 0,59* | 0,55 | 0,57* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.22 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 120 min sem dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,81 | 0,79 | 0,81 | 0,79 | 0,80 | 0,78 |
| | CatBoost | 0,63 | 0,84* | 0,49 | 0,83* | 0,40 | 0,83* |
| 24-06 h | XGBoost | 0,71 | 0,71 | 0,59 | 0,59 | 0,57 | 0,57 |
| | CatBoost | 0,76 | 0,79* | 0,68 | 0,71* | 0,68 | 0,73* |
| 18-06 h | XGBoost | 0,68 | 0,66 | 0,57 | 0,52 | 0,54 | 0,44 |
| | CatBoost | 0,65 | 0,66* | 0,51 | 0,54* | 0,42 | 0,49* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.23 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 150 min sem dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,73 | 0,68 | 0,65 | 0,60 | 0,66 | 0,59 |
| | CatBoost | 0,77 | 0,67 | 0,70 | 0,50 | 0,71 | 0,40 |
| 24-06 h | XGBoost | 0,68 | 0,72* | 0,56 | 0,61* | 0,51 | 0,59* |
| | CatBoost | 0,87 | 0,94* | 0,83 | 0,94* | 0,85 | 0,93* |
| 18-06 h | XGBoost | 0,68 | 0,64 | 0,57 | 0,51 | 0,53 | 0,41 |
| | CatBoost | 0,65 | 0,64 | 0,51 | 0,55* | 0,42 | 0,53* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.24 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 180 min sem dados de ionossonda e validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,71 | 0,68 | 0,60 | 0,52 | 0,59 | 0,46 |
| | CatBoost | 0,77 | 0,66 | 0,71 | 0,50 | 0,72 | 0,40 |
| 24-06 h | XGBoost | 0,68 | 0,64 | 0,58 | 0,53 | 0,54 | 0,44 |
| | CatBoost | 0,65 | 0,79* | 0,53 | 0,73* | 0,49 | 0,74* |
| 18-06 h | XGBoost | 0,68 | 0,67 | 0,55 | 0,54 | 0,50 | 0,48 |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,42 | 0,39 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.25 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 30 min sem dados de ionossonda e empregando TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,82 | 0,82 | 0,77 | 0,77 | 0,78 | 0,78 |
| | CatBoost | 0,82 | 0,81 | 0,76 | 0,75 | 0,78 | 0,77 |
| 24-06 h | XGBoost | 0,85 | 0,86* | 0,78 | 0,81* | 0,81 | 0,83* |
| | CatBoost | 0,88 | 0,87 | 0,84 | 0,82 | 0,86 | 0,85 |
| 18-06 h | XGBoost | 0,84 | 0,85* | 0,78 | 0,80* | 0,80 | 0,82* |
| | CatBoost | 0,83 | 0,85* | 0,77 | 0,79* | 0,79 | 0,81* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.26 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 60 min sem dados de ionossonda e empregando TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,63 | 0,62 | 0,51 | 0,50 | 0,40 | 0,39 |
| | CatBoost | 0,64 | 0,65* | 0,53 | 0,54* | 0,45 | 0,48* |
| 24-06 h | XGBoost | 0,78 | 0,76 | 0,70 | 0,66 | 0,71 | 0,67 |
| | CatBoost | 0,78 | 0,77 | 0,69 | 0,68 | 0,70 | 0,69 |
| 18-06 h | XGBoost | 0,69 | 0,74* | 0,57 | 0,64* | 0,53 | 0,63* |
| | CatBoost | 0,72 | 0,72 | 0,60 | 0,61* | 0,59 | 0,60* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.27 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 90 min sem dados de ionossonda e empregando TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,64 | 0,64 | 0,50 | 0,51* | 0,39 | 0,41* |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |
| 24-06 h | XGBoost | 0,69 | 0,72* | 0,55 | 0,60* | 0,51 | 0,58* |
| | CatBoost | 0,76 | 0,75 | 0,66 | 0,64 | 0,67 | 0,64 |
| 18-06 h | XGBoost | 0,68 | 0,68 | 0,55 | 0,55 | 0,49 | 0,49 |
| | CatBoost | 0,69 | 0,69 | 0,56 | 0,56 | 0,52 | 0,53* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.28 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 120 min sem dados de ionossonda e empregando TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,65 | 0,66* | 0,50 | 0,52* | 0,39 | 0,43* |
| | CatBoost | 0,65 | 0,65 | 0,50 | 0,50 | 0,39 | 0,39 |
| 24-06 h | XGBoost | 0,66 | 0,66 | 0,50 | 0,50 | 0,40 | 0,40 |
| | CatBoost | 0,68 | 0,68 | 0,54 | 0,53 | 0,48 | 0,46 |
| 18-06 h | XGBoost | 0,65 | 0,66* | 0,50 | 0,53* | 0,40 | 0,46* |
| | CatBoost | 0,65 | 0,65 | 0,50 | 0,50 | 0,40 | 0,40 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.29 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 150 min sem dados de ionossonda e empregando TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,66 | 0,66 | 0,50 | 0,50 | 0,40 | 0,40 |
| | CatBoost | 0,66 | 0,66 | 0,50 | 0,50 | 0,39 | 0,39 |
| 24-06 h | XGBoost | 0,65 | 0,65 | 0,50 | 0,51* | 0,39 | 0,42* |
| | CatBoost | 0,65 | 0,66* | 0,51 | 0,52* | 0,41 | 0,43* |
| 18-06 h | XGBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,40 | 0,39 |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.30 - Experimento A (GTSH/TSCV-GKF) anos 2010-2018 - Desempenho de predição para 180 min sem dados de ionossonda e empregando TSCV-GKF.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,67 | 0,67 | 0,50 | 0,51* | 0,41 | 0,42* |
| | CatBoost | 0,67 | 0,67 | 0,50 | 0,50 | 0,40 | 0,40 |
| 24-06 h | XGBoost | 0,62 | 0,63* | 0,50 | 0,51* | 0,38 | 0,42* |
| | CatBoost | 0,62 | 0,62 | 0,50 | 0,50 | 0,38 | 0,38 |
| 18-06 h | XGBoost | 0,64 | 0,65* | 0,50 | 0,51* | 0,39 | 0,42* |
| | CatBoost | 0,64 | 0,64 | 0,50 | 0,50 | 0,39 | 0,39 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

## A.2 Experimento A (GTSH) anos 2012-2014

Nesta variação do Experimento A (GTSH/TSCV-GKF) anos 2010-2018, foi utilizado como esquema de validação o GTSH, também utilizado anteriormente, porém para estação seção os dados foram restritos aos anos 2012, 2013 e 2014, sendo o primeiro usado como subconjunto de treinamento, o segundo como subconjunto de validação e o terceiro como subconjunto de teste. Um ponto interessante deste intervalo é que 2014 é o ano de máximo do ciclo solar, o que contrasta com o período de teste usado para os experimentos anteriores que é o final da redução da intensidade do ciclo indo para o mínimo do ciclo solar.

As Tabelas A.31, A.33, A.35, A.37, A.39 e A.41 apresentam os resultados para o Experimento A (GTSH) anos 2012-2014. Elas tem 5 colunas principais: Período, Modelo, Acurácia, Precisão, $F_1$. A coluna Período indica se o modelo foi treinado para realizar predições: das 18h até 06h do dia seguinte, 18-06 h; iniciando às 18h até as 24h, 18-24 h; ou das 24h até 06h, 24-06 h. A coluna Modelo indica se o algoritmo de aprendizagem de máquina empregado foi o XGBoost ou CatBoost, enquanto as demais colunas apresentam as métricas de desempenho de predição. Cada métrica de desempenho apresenta duas subcolunas NSA e SSA que indicam o valor da métrica antes da operação de seleção de atributos e depois da seleção de atributos respectivamente, se houver uma melhora devido a seleção o resultado é marcado com o símbolo $^*$. Além das 6 tabelas já introduzidas existem mais 6 Tabelas A.32, A.34, A.36, A.38, A.40 e A.42 que apresentam a matriz de confusão para o modelo treinado considerando o período das 18-06 h.

As Tabelas A.31, A.33, A.35, A.37, A.39 e A.41 podem ser comparadas com as Tabelas A.1, A.3, A.5, A.7, A.9 e A.11, apresentadas no início da Seção A.1 (Seção Resultados Experimento A (GTSH/TSCV-GKF) anos 2010-2018). Primeiro, é evidente que os resultados aqui são superiores em termos de $F_1$. Segundo, ainda existe uma degradação no valor de $F_1$ com aumento da antecedência de predição, porém esse é mais suave quando comparado com resultados do Experimento A (GTSH/TSCV-GKF) anos 2010-2018. Esse fato pode provavelmente pode ser explicado considerando as diferenças de intensidade no ciclo, que levam a períodos onde a cintilação é mais preditiva, enquanto em outros ela é mais aleatória, ao menos em termos das variáveis adotadas para a modelagem empregada. Terceiro, o processo de seleção de atributos aparenta funcionar melhor, considerando o número de símbolos $^*$ presente.

A configuração com o período das 18-06 h cujos resultados são apresentados nas Tabelas A.31, A.33, A.35, A.37, A.39 e A.41, também foi avaliado considerando

múltiplas sementes aleatória, neste caso os valores médios se apresentam próximos dos valores já apresentados com uma única semente fixa e o valor do desvio padrão para o $F_1$, por exemplo, varia de 0,02 para a predição com antecedência de 30 min (menor desvio padrão observado) até 0,13 para antecedência de 180 min (maior desvio padrão observado).

Tabela A.31 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição para 30 min com validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,64 | 0,74* | 0,65 | 0,74* | 0,64 | 0,74* |
| | CatBoost | 0,74 | 0,78* | 0,73 | 0,77* | 0,72 | 0,77* |
| 24-06 h | XGBoost | 0,84 | 0,93* | 0,62 | 0,85* | 0,64 | 0,87* |
| | CatBoost | 0,91 | 0,94* | 0,87 | 0,90* | 0,87 | 0,90* |
| 18-06 h | XGBoost | 0,82 | 0,84* | 0,77 | 0,79* | 0,78 | 0,81* |
| | CatBoost | 0,84 | 0,80 | 0,78 | 0,80* | 0,80 | 0,78 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.32 - Experimento A (GTSH) anos 2012-2014 - Matriz de Confusão para a predição 30 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,78$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1536 | 166 |
| | OC | 299 | 543 |

| CatBoost-NSA $F_1 = 0,80$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1599 | 103 |
| | OC | 313 | 529 |

| XGBoost-SSA $F_1 = 0,80$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1568 | 134 |
| | OC | 283 | 559 |

| CatBoost-SSA $F_1 = 0,78$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1385 | 317 |
| | OC | 186 | 656 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.33 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição para 60 min com validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,53 | 0,68* | 0,54 | 0,68* | 0,53 | 0,68* |
| | CatBoost | 0,67 | 0,65 | 0,68 | 0,66 | 0,65 | 0,63 |
| 24-06 h | XGBoost | 0,85 | 0,92* | 0,69 | 0,81* | 0,70 | 0,84* |
| | CatBoost | 0,87 | 0,88* | 0,84 | 0,88* | 0,79 | 0,81* |
| 18-06 h | XGBoost | 0,64 | 0,80* | 0,54 | 0,73* | 0,53 | 0,76* |
| | CatBoost | 0,80 | 0,77 | 0,71 | 0,73* | 0,73 | 0,73 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.34 - Experimento A (GTSH) anos 2012-2014 - Matriz de Confusão para a predição 60 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,53$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1428 | 279 |
| | OC | 635 | 202 |

| CatBoost-NSA $F_1 = 0,76$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1666 | 41 |
| | OC | 470 | 367 |

| XGBoost-SSA $F_1 = 0,76$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1505 | 202 |
| | OC | 313 | 524 |

| CatBoost-SSA $F_1 = 0,73$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1459 | 248 |
| | OC | 331 | 506 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.35 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição para 90 min com validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,58 | 0,50 | 0,61 | 0,51 | 0,57 | 0,50 |
| | CatBoost | 0,56 | 0,69* | 0,60 | 0,70* | 0,54 | 0,69* |
| 24-06 h | XGBoost | 0,89 | 0,87 | 0,67 | 0,85* | 0,69 | 0,75* |
| | CatBoost | 0,92 | 0,93* | 0,82 | 0,80 | 0,81 | 0,82* |
| 18-06 h | XGBoost | 0,68 | 0,75* | 0,54 | 0,72* | 0,52 | 0,71* |
| | CatBoost | 0,77 | 0,74 | 0,77 | 0,77 | 0,75 | 0,73 |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.36 - Experimento A (GTSH) anos 2012-2014 - Matriz de Confusão para a predição 90 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,52$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1589 | 129 |
| | OC | 694 | 132 |

| CatBoost-NSA $F_1 = 0,75$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1317 | 401 |
| | OC | 181 | 645 |

| XGBoost-SSA $F_1 = 0,71$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1382 | 336 |
| | OC | 308 | 518 |

| CatBoost-SSA $F_1 = 0,73$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1201 | 517 |
| | OC | 138 | 668 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.37 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição para 120 min com validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,45 | 0,56* | 0,51 | 0,52* | 0,41 | 0,51* |
| | CatBoost | 0,48 | 0,54* | 0,55 | 0,59* | 0,43 | 0,52* |
| 24-06 h | XGBoost | 0,94 | 0,94 | 0,76 | 0,76 | 0,78 | 0,77 |
| | CatBoost | 0,93 | 0,93* | 0,74 | 0,75* | 0,75 | 0,75* |
| 18-06 h | XGBoost | 0,59 | 0,73* | 0,46 | 0,74* | 0,43 | 0,72* |
| | CatBoost | 0,75 | 0,75 | 0,65 | 0,75* | 0,66 | 0,73* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.38 - Experimento A (GTSH) anos 2012-2014 - Matriz de Confusão para a predição 120 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,43$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1406 | 334 |
| | OC | 717 | 87 |

| CatBoost-NSA $F_1 = 0,66$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1594 | 146 |
| | OC | 494 | 310 |

| XGBoost-SSA $F_1 = 0,72$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1231 | 509 |
| | OC | 175 | 629 |

| CatBoost-SSA $F_1 = 0,73$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1296 | 444 |
| | OC | 197 | 607 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.39 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição para 150 min com validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,44 | 0,60* | 0,41 | 0,56* | 0,40 | 0,56* |
| | CatBoost | 0,59 | 0,60* | 0,60 | 0,59 | 0,59 | 0,59 |
| 24-06 h | XGBoost | 0,92 | 0,93* | 0,57 | 0,56 | 0,57 | 0,58* |
| | CatBoost | 0,94 | 0,93 | 0,57 | 0,56 | 0,57 | 0,57 |
| 18-06 h | XGBoost | 0,72 | 0,77* | 0,60 | 0,76* | 0,60 | 0,74* |
| | CatBoost | 0,70 | 0,76* | 0,52 | 0,78* | 0,46 | 0,75* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.40 - Experimento A (GTSH) anos 2012-2014 - Matriz de Confusão para a predição 150 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,60$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1607 | 173 |
| | OC | 548 | 216 |

| CatBoost-NSA $F_1 = 0,46$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1759 | 21 |
| | OC | 728 | 36 |

| XGBoost-SSA $F_1 = 0,74$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1421 | 359 |
| | OC | 216 | 548 |

| CatBoost-SSA $F_1 = 0,75$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1281 | 499 |
| | OC | 110 | 654 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

Tabela A.41 - Experimento A (GTSH) anos 2012-2014 - Desempenho de predição para 180 min com validação por GTSH.

| Período | **Modelo** | Acurácia | | Precisão | | $F_1$ | |
|---|---|---|---|---|---|---|---|
| | | NSA | SSA | NSA | SSA | NSA | SSA |
| 18-24 h | XGBoost | 0,51 | 0,57* | 0,50 | 0,58* | 0,50 | 0,57* |
| | CatBoost | 0,42 | 0,60* | 0,46 | 0,59* | 0,41 | 0,59* |
| 24-06 h | XGBoost | 0,96 | 0,96 | 0,52 | 0,52 | 0,53 | 0,53 |
| | CatBoost | 0,92 | 0,96* | 0,52 | 0,51 | 0,52 | 0,51 |
| 18-06 h | XGBoost | 0,63 | 0,77* | 0,59 | 0,76* | 0,57 | 0,74* |
| | CatBoost | 0,80 | 0,79 | 0,70 | 0,72* | 0,72 | 0,73* |

O símbolo * indica melhoria ao utilizar seleção de atributos (SSA) em relação a não utilizar (NSA).

Fonte: Produção do autor.

Tabela A.42 - Experimento A (GTSH) anos 2012-2014 - Matriz de Confusão para a predição 180 min à frente com dados de ionossonda e validação por GTSH, considerando o período das 18-06 h.

| XGBoost-NSA $F_1 = 0,57$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1237 | 592 |
| | OC | 360 | 355 |

| CatBoost-NSA $F_1 = 0,74$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1679 | 150 |
| | OC | 366 | 349 |

| XGBoost-SSA $F_1 = 0,72$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1408 | 421 |
| | OC | 173 | 542 |

| CatBoost-SSA $F_1 = 0,73$ | | PRED | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV | N-OC | 1611 | 218 |
| | OC | 319 | 396 |

PRED/OBSV indica predito/observado e NSA/SSA indica sem/com seleção de atributos, respectivamente.

Fonte: Produção do autor.

## A.3 Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018

Os Experimentos Experimento A (GTSH/TSCV-GKF) anos 2010-2018 à Experimento C (GTSH) anos 2010-2018 herdam um subconjunto de teste fixo definido por meio de um GTSH, os resultados da Seção A.2, por outro lado são oriundos de um subintervalo dos dados (2012-2014), com um subconjunto de teste diferente, porém

fixo e único. Observando a necessidade de avaliar outros intervalos como subconjunto de teste, optou-se por adotar um esquema de validação cruzada em níveis: uma mais externa fazendo o papel do antigo GTSH, definindo um subconjunto de teste e treinamento e uma mais interna que toma o subconjunto de treinamento oriundo da parte mais externa e particiona em um novo subconjunto de treinamento e um subconjunto de validação. Essa abordagem permite avaliar diferentes intervalos de tempo (note que os modelos são diferentes, porém com a mesma metodologia de geração). Este experimento segue a metodologia do Experimento A (GTSH/TSCV-GKF) anos 2010-2018 com TSCV-GKF, exceto que: não é feito o a seleção de atributos, os resultados gerados pelo XGBoost e CatBoost são combinados em um único resultado e a presença de um subconjunto de teste que varia. Além da TSCV-GKF foi avaliado também a TSCV-GWF.

Considerando a variante usando o esquema TSCV-GKF, os resultados são apresentados na Tabela A.43, enquanto as Tabelas A.44 à A.49 apresentam as matrizes de confusão, e a Tabela 6.1 indica o número de amostras por subconjunto para a predição com 30 minutos de antecedência. Somente foi apresentado a tabela com o número de amostras para um único passo de predição, os valores são semelhantes, mas não iguais, para os demais passos (60 min até 180 min). Em algarismos romanos fica indicado o subconjunto de teste e em algarismos árabes o subconjunto de validação. Observando a Tabela 6.1 fica evidente que há 5 subconjuntos para teste e há 5 subconjuntos para validação para casa subconjunto de teste. Os subconjuntos estão ordenados no tempo, assim os dados do subconjunto I antecedem o subconjunto II.

Os resultados apresentados na Tabela A.43 são os melhores obtidos ao longo do trabalho, mas existem comportamentos comuns em relação aos outros resultados, por exemplo, o decaimento de desempenho com o aumento do tempo de antecedência da predição. Os quatro primeiros subconjuntos (I, II, III, IV) são os que apresentam melhores resultados, O subconjunto de teste V apresenta um intervalo de dados semelhante aos utilizados nas variações anteriores do Experimento A (GTSH/TSCV-GKF) anos 2010-2018 e assim apresenta resultados equivalentes.

Um problema da variação TSCV-GKF é a utilização de dados futuros para treinamento, note, por exemplo, que quando o subconjunto I for utilizado para teste, os subconjuntos de II até V serão utilizados para treinamento e validação, logo dados futuros em relação ao teste serão utilizados no desenvolvimento do modelo. Por outro lado, para o subconjunto $V$ sendo usado como teste, apenas dados do passado serão utilizados no modelo, o que permite argumentar que os bons resultados decorrem

da utilização de dados futuros.

Tabela A.43 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GKF.

| Antecedência (min) | Subconjunto de Teste | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|
| 30 | I | 0,99 | 0,97 | 0,97 |
| | II | 0,95 | 0,92 | 0,93 |
| | III | 0,93 | 0,93 | 0,92 |
| | IV | 0,95 | 0,90 | 0,91 |
| | V | 0,93 | 0,72 | 0,79 |
| 60 | I | 0,98 | 0,95 | 0,95 |
| | II | 0,94 | 0,92 | 0,91 |
| | III | 0,92 | 0,92 | 0,91 |
| | IV | 0,94 | 0,88 | 0,89 |
| | V | 0,90 | 0,57 | 0,60 |
| 90 | I | 0,98 | 0,96 | 0,95 |
| | II | 0,94 | 0,93 | 0,92 |
| | III | 0,90 | 0,90 | 0,88 |
| | IV | 0,92 | 0,86 | 0,86 |
| | V | 0,88 | 0,50 | 0,48 |
| 120 | I | 0,98 | 0,96 | 0,94 |
| | II | 0,95 | 0,94 | 0,92 |
| | III | 0,88 | 0,89 | 0,87 |
| | IV | 0,91 | 0,86 | 0,84 |
| | V | 0,88 | 0,50 | 0,47 |
| 150 | I | 0,98 | 0,96 | 0,95 |
| | II | 0,92 | 0,93 | 0,89 |
| | III | 0,90 | 0,90 | 0,88 |
| | IV | 0,91 | 0,86 | 0,84 |
| | V | 0,89 | 0,50 | 0,48 |
| 180 | I | 0,98 | 0,96 | 0,93 |
| | II | 0,93 | 0,93 | 0,90 |
| | III | 0,91 | 0,92 | 0,90 |
| | IV | 0,92 | 0,87 | 0,86 |
| | V | 0,89 | 0,51 | 0,49 |

Fonte: Produção do autor.

Tabela A.44 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 30 min à frente para 18-06 h, com TSCV-GKF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,97$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 3920 | 22 |
| | OC | 26 | 438 |

| $F_1 = 0,93$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 3336 | 111 |
| | OC | 113 | 846 |

| $F_1 = 0,92$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2823 | 164 |
| | OC | 129 | 1289 |

| $F_1 = 0,91$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3567 | 76 |
| | OC | 139 | 623 |

| $F_1 = 0,79$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3891 | 12 |
| | OC | 275 | 227 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.45 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 60 min à frente para 18-06 h, com TSCV-GKF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,95$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 3914 | 45 |
| | OC | 40 | 407 |

| $F_1 = 0,91$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 3305 | 148 |
| | OC | 115 | 838 |

| $F_1 = 0,91$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2768 | 229 |
| | OC | 124 | 1284 |

| $F_1 = 0,89$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3535 | 112 |
| | OC | 156 | 602 |

| $F_1 = 0,60$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3890 | 14 |
| | OC | 426 | 75 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.46 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 90 min à frente para 18-06 h, com TSCV-GKF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,95$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 3933 | 50 |
| | OC | 30 | 393 |

| $F_1 = 0,92$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 3289 | 177 |
| | OC | 83 | 857 |

| $F_1 = 0,88$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2676 | 341 |
| | OC | 115 | 1273 |

| $F_1 = 0,86$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3468 | 184 |
| | OC | 166 | 587 |

| $F_1 = 0,64$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3892 | 9 |
| | OC | 498 | 6 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.47 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 120 min à frente para 18-06 h, com TSCV-GKF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,94$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 3950 | 59 |
| | OC | 26 | 371 |

| $F_1 = 0,92$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 3315 | 175 |
| | OC | 67 | 849 |

| $F_1 = 0,87$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2646 | 410 |
| | OC | 107 | 1242 |

| $F_1 = 0,84$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3425 | 241 |
| | OC | 158 | 581 |

| $F_1 = 0,47$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3892 | 8 |
| | OC | 503 | 2 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.48 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 150 min à frente para 18-06 h, com TSCV-GKF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,95$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 3990 | 49 |
| | OC | 25 | 342 |

| $F_1 = 0,89$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 3252 | 277 |
| | OC | 60 | 817 |

| $F_1 = 0,88$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2764 | 349 |
| | OC | 108 | 1184 |

| $F_1 = 0,84$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3466 | 232 |
| | OC | 157 | 550 |

| $F_1 = 0,48$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3902 | 5 |
| | OC | 494 | 4 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.49 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 180 min à frente para 18-06 h, com TSCV-GKF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,93$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 4009 | 61 |
| | OC | 25 | 311 |

| $F_1 = 0,90$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 3334 | 248 |
| | OC | 56 | 768 |

| $F_1 = 0,90$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2889 | 306 |
| | OC | 82 | 1128 |

| $F_1 = 0,86$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3528 | 205 |
| | OC | 136 | 536 |

| $F_1 = 0,49$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3898 | 16 |
| | OC | 483 | 8 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Levando em consideração o problema de utilização de dados futuros, optou-se por usar a mesma metodologia porém com o esquema TSCV-GWF, o qual garante que as amostras utilizadas para validação sucedem as amostras de treinamento, e as amostras de teste sucedem as amostras de validação. A Tabela A.50 apresenta os resultados para essa variação, enquanto as Tabelas A.51 à A.56 apresentam as correspondentes matrizes de confusão considerando o período das 18-06 h. Finalmente a Tabela 6.3 apresenta o número de amostras subconjunto para a predição com 30 minutos de antecedência, valores similares são encontradas para as outras antecedências.

Observando as Tabelas associadas ao esquema TSCV-GWF fica claro que os subconjuntos II, III e IV são os que apresentam os melhores resultados, é importante destacar que os subconjuntos I, II, III, IV, V para o esquema TSCV-GKF são diferentes do esquema TSCV-GWF, porém podem ter trechos com intervalos comuns.

A região com melhores resultados intersecta com o período de máximo do ciclo colar, e já foi observado em resultados anteriores que a metodologia tende a funcionar bem para este intervalo. O subconjunto I apresentou um péssimo resultado, porém observando a Tabela 6.3 com o número de amostras é possível notar que este é muito pequeno para o treinamento e para a validação, lembrando que o subconjunto de validação é utilizada para controlar a parada do algoritmo de treinamento de forma a evitar um sobreajuste, um subconjunto muito pequeno pode levar a uma parada muito antecipada de tal modo que o modelo somente seja capaz de tratar bem um subconjunto restrito de amostras. Uma comparação dos resultados para os subconjuntos II, III e IV em relação aos dois esquemas de validação indica que o esquema TSCV-GWF apresenta um menor desempenho, porém se tem a troca do problema de informação futura, pelo problema de subconjuntos que não contém amostras suficientes para caracterizar todo o espectro de observações.

Tabela A.50 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Desempenho de predição para TSCV-GWF.

| Antecedência (min) | Subconjunto de Teste | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|
| 30 | I | 0,75 | 0,61 | 0,61 |
| | II | 0,90 | 0,84 | 0,86 |
| | III | 0,88 | 0,85 | 0,86 |
| | IV | 0,93 | 0,82 | 0,83 |
| | V | 0,92 | 0,68 | 0,74 |
| 60 | I | 0,69 | 0,50 | 0,42 |
| | II | 0,87 | 0,82 | 0,83 |
| | III | 0,83 | 0,82 | 0,81 |
| | IV | 0,92 | 0,78 | 0,79 |
| | V | 0,87 | 0,52 | 0,51 |
| 90 | I | 0,71 | 0,55 | 0,54 |
| | II | 0,86 | 0,76 | 0,79 |
| | III | 0,82 | 0,78 | 0,78 |
| | IV | 0,90 | 0,74 | 0,75 |
| | V | 0,87 | 0,50 | 0,47 |
| 120 | I | 0,71 | 0,50 | 0,42 |
| | II | 0,86 | 0,83 | 0,82 |
| | III | 0,82 | 0,80 | 0,79 |
| | IV | 0,90 | 0,75 | 0,74 |
| | V | 0,87 | 0,50 | 0,47 |
| 150 | I | 0,73 | 0,50 | 0,43 |
| | II | 0,84 | 0,72 | 0,74 |
| | III | 0,82 | 0,78 | 0,78 |
| | IV | 0,89 | 0,73 | 0,73 |
| | V | 0,87 | 0,50 | 0,47 |
| 180 | I | 0,75 | 0,50 | 0,43 |
| | II | 0,87 | 0,80 | 0,81 |
| | III | 0,83 | 0,80 | 0,78 |
| | IV | 0,90 | 0,73 | 0,72 |
| | V | 0,87 | 0,50 | 0,47 |

Fonte: Produção do autor.

Tabela A.51 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 30 min à frente para 18-06 h, com TSCV-GWF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,61$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 2490 | 37 |
| | OC | 873 | 271 |

| $F_1 = 0,86$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 2642 | 104 |
| | OC | 267 | 658 |

| $F_1 = 0,86$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2391 | 189 |
| | OC | 243 | 848 |

| $F_1 = 0,83$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3156 | 110 |
| | OC | 135 | 270 |

| $F_1 = 0,74$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3188 | 6 |
| | OC | 303 | 174 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.52 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 60 min à frente para 18-06 h, com TSCV-GWF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,42$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 2530 | 16 |
| | OC | 1108 | 17 |

| $F_1 = 0,83$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 2572 | 183 |
| | OC | 278 | 638 |

| $F_1 = 0,81$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2224 | 363 |
| | OC | 248 | 836 |

| $F_1 = 0,79$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3141 | 125 |
| | OC | 165 | 240 |

| $F_1 = 0,51$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3191 | 4 |
| | OC | 456 | 20 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.53 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 90 min à frente para 18-06 h, com TSCV-GWF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,54$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 2444 | 136 |
| | OC | 916 | 175 |

| $F_1 = 0,79$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 2622 | 145 |
| | OC | 384 | 520 |

| TSCV \| $F_1 = 0,78$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2274 | 326 |
| | OC | 332 | 739 |

| $F_1 = 0,75$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3101 | 166 |
| | OC | 190 | 214 |

| $F_1 = 0,47$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3192 | 0 |
| | OC | 479 | 0 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.54 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 120 min à frente para 18-06 h, com TSCV-GWF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,42$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 2619 | 8 |
| | OC | 1042 | 2 |

| $F_1 = 0,82$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 2474 | 315 |
| | OC | 194 | 688 |

| $F_1 = 0,79$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2819 | 405 |
| | OC | 259 | 788 |

| $F_1 = 0,74$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3064 | 211 |
| | OC | 173 | 223 |

| $F_1 = 0,47$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3191 | 0 |
| | OC | 480 | 0 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.55 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 150 min à frente para 18-06 h, com TSCV-GWF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,43$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 2670 | 20 |
| | OC | 974 | 7 |

| $F_1 = 0,74$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 2673 | 144 |
| | OC | 443 | 411 |

| $F_1 = 0,78$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2328 | 344 |
| | OC | 306 | 693 |

| $F_1 = 0,73$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3085 | 207 |
| | OC | 179 | 200 |

| $F_1 = 0,47$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3197 | 0 |
| | OC | 474 | 0 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

Tabela A.56 - Experimento A (multi-TSCV-GKF/multi-TSCV-GWF) anos 2010-2018 - Matriz de Confusão para a predição 180 min à frente para 18-06 h, com TSCV-GWF e com dados de ionossonda, subconjuntos I-II-III-IV-V.

| $F_1 = 0,43$ | | PRED(I) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(I) | N-OC | 2761 | 0 |
| | OC | 910 | 0 |

| $F_1 = 0,81$ | | PRED(II) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(II) | N-OC | 2674 | 195 |
| | OC | 274 | 528 |

| $F_1 = 0,78$ | | PRED(III) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(III) | N-OC | 2339 | 390 |
| | OC | 245 | 697 |

| $F_1 = 0,72$ | | PRED(IV) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(IV) | N-OC | 3103 | 207 |
| | OC | 170 | 191 |

| $F_1 = 0,47$ | | PRED(V) | |
|---|---|---|---|
| | | N-OC | OC |
| OBSV(V) | N-OC | 3203 | 0 |
| | OC | 468 | 0 |

PRED/OBSV indica predito/observado.

Fonte: Produção do autor.

## A.4 Experimento B (GTSH/VCT) anos 2010-2018

Tabela A.57 - Experimento B (GTSH/VCT) anos 2010-2018 - Desempenho de predição sem informação de TEC futuro e com tamanho de palavra 4, classe OC significando cintilação forte-moderado-fraco.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| TEC | GTSH | XGBoost | 0,52 | 0,52 | 0,49 |
| | | CatBoost | 0,49 | 0,40 | 0,40 |
| | VCT | XGBoost | 0,32 | 0,48 | 0,36 |
| | | CatBoost | 0,33 | 0,49 | 0,31 |
| AE | GTSH | XGBoost | 0,56 | 0,48 | 0,48 |
| | | CatBoost | 0,59 | 0,54 | 0,53 |
| | VCT | XGBoost | 0,56 | 0,49 | 0,49 |
| | | CatBoost | 0,32 | 0,48 | 0,36 |
| AP | GTSH | XGBoost | 0,69 | 0,60 | 0,60 |
| | | CatBoost | 0,62 | 0,55 | 0,55 |
| | VCT | XGBoost | 0,55 | 0,53 | 0,51 |
| | | CatBoost | 0,51 | 0,50 | 0,48 |
| IMF By | GTSH | XGBoost | 0,60 | 0,46 | 0,45 |
| | | CatBoost | 0,69 | 0,54 | 0,53 |
| | VCT | XGBoost | 0,41 | 0,56 | 0,41 |
| | | CatBoost | 0,28 | 0,49 | 0,23 |
| IMF Bz | GTSH | XGBoost | 0,64 | 0,47 | 0,45 |
| | | CatBoost | 0,61 | 0,48 | 0,50 |
| | VCT | XGBoost | 0,40 | 0,57 | 0,39 |
| | | CatBoost | 0,29 | 0,51 | 0,36 |
| Dst | GTSH | XGBoost | 0,53 | 0,49 | 0,48 |
| | | CatBoost | 0,59 | 0,44 | 0,42 |
| | VCT | XGBoost | 0,52 | 0,51 | 0,49 |
| | | CatBoost | 0,48 | 0,55 | 0,48 |
| h'F | GTSH | XGBoost | 0,81 | 0,75 | 0,76 |
| | | CatBoost | 0,76 | 0,69 | 0,69 |
| | VCT | XGBoost | 0,55 | 0,63 | 0,54 |
| | | CatBoost | 0,77 | 0,71 | 0,71 |

Continua

Tabela A.57 – Continuação.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| $P_{sw}$ | GTSH | XGBoost | 0,63 | 0,46 | 0,44 |
| | | CatBoost | 0,60 | 0,45 | 0,43 |
| | VCT | XGBoost | 0,32 | 0,50 | 0,36 |
| | | CatBoost | 0,28 | 0,49 | 0,23 |
| $V_{sw}$ | GTSH | XGBoost | 0,60 | 0,52 | 0,52 |
| | | CatBoost | 0,67 | 0,49 | 0,47 |
| | VCT | XGBoost | 0,56 | 0,49 | 0,49 |
| | | CatBoost | 0,36 | 0,49 | 0,28 |
| Sym-D | GTSH | XGBoost | 0,47 | 0,53 | 0,46 |
| | | CatBoost | 0,67 | 0,46 | 0,40 |
| | VCT | XGBoost | 0,48 | 0,55 | 0,48 |
| | | CatBoost | 0,36 | 0,48 | 0,36 |
| Sym-H | GTSH | XGBoost | 0,57 | 0,47 | 0,47 |
| | | CatBoost | 0,70 | 0,53 | 0,52 |
| | VCT | XGBoost | 0,52 | 0,43 | 0,44 |
| | | CatBoost | 0,32 | 0,51 | 0,29 |
| AE+TEC | GTSH | XGBoost | 0,36 | 0,40 | 0,35 |
| | | CatBoost | 0,49 | 0,46 | 0,45 |
| | VCT | XGBoost | 0,33 | 0,45 | 0,33 |
| | | CatBoost | 0,28 | 0,36 | 0,28 |
| Ap+TEC | GTSH | XGBoost | 0,46 | 0,53 | 0,46 |
| | | CatBoost | 0,46 | 0,46 | 0,44 |
| | VCT | XGBoost | 0,38 | 0,47 | 0,38 |
| | | CatBoost | 0,40 | 0,53 | 0,39 |
| IMF By+TEC | GTSH | XGBoost | 0,42 | 0,46 | 0,42 |
| | | CatBoost | 0,46 | 0,46 | 0,44 |
| | VCT | XGBoost | 0,32 | 0,47 | 0,31 |
| | | CatBoost | 0,26 | 0,47 | 0,21 |
| IMF Bz+TEC | GTSH | XGBoost | 0,43 | 0,46 | 0,42 |
| | | CatBoost | 0,52 | 0,48 | 0,47 |
| | VCT | XGBoost | 0,31 | 0,47 | 0,29 |
| | | CatBoost | 0,28 | 0,47 | 0,36 |

Continua

Tabela A.57 – Continuação.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| Dst+TEC | GTSH | XGBoost | 0,52 | 0,55 | 0,50 |
| | | CatBoost | 0,51 | 0,50 | 0,48 |
| | VCT | XGBoost | 0,40 | 0,53 | 0,40 |
| | | CatBoost | 0,28 | 0,45 | 0,25 |
| h'F+TEC | GTSH | XGBoost | 0,55 | 0,55 | 0,52 |
| | | CatBoost | 0,63 | 0,58 | 0,57 |
| | VCT | XGBoost | 0,42 | 0,51 | 0,42 |
| | | CatBoost | 0,40 | 0,54 | 0,40 |
| $P_{sw}$+TEC | GTSH | XGBoost | 0,45 | 0,47 | 0,44 |
| | | CatBoost | 0,50 | 0,46 | 0,47 |
| | VCT | XGBoost | 0,33 | 0,49 | 0,32 |
| | | CatBoost | 0,29 | 0,50 | 0,25 |
| $V_{sw}$+TEC | GTSH | XGBoost | 0,48 | 0,51 | 0,46 |
| | | CatBoost | 0,51 | 0,48 | 0,47 |
| | VCT | XGBoost | 0,34 | 0,50 | 0,33 |
| | | CatBoost | 0,32 | 0,48 | 0,36 |
| Sym-D+TEC | GTSH | XGBoost | 0,45 | 0,49 | 0,44 |
| | | CatBoost | 0,41 | 0,42 | 0,39 |
| | VCT | XGBoost | 0,36 | 0,49 | 0,35 |
| | | CatBoost | 0,32 | 0,48 | 0,36 |
| Sym-H+TEC | GTSH | XGBoost | 0,44 | 0,42 | 0,41 |
| | | CatBoost | 0,49 | 0,49 | 0,47 |
| | VCT | XGBoost | 0,36 | 0,50 | 0,35 |
| | | CatBoost | 0,36 | 0,51 | 0,35 |
| Todos (i) | GTSH | XGBoost | 0,64 | 0,63 | 0,60 |
| | | CatBoost | 0,60 | 0,52 | 0,52 |
| | VCT | XGBoost | 0,42 | 0,53 | 0,43 |
| | | CatBoost | 0,51 | 0,58 | 0,50 |

Continua

Tabela A.57 – Continuação.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| Todos (ii) | GTSH | XGBoost | 0,64 | 0,63 | 0,61 |
| | | CatBoost | 0,65 | 0,61 | 0,60 |
| | VCT | XGBoost | 0,42 | 0,53 | 0,43 |
| | | CatBoost | 0,54 | 0,50 | 0,49 |

Todos se apresentam com duas possibilidades: (i) aplica-se o algoritmo de Weasel-Muse a todas as variáveis para geração do vetor de atributos; e (ii) aplica-se o algoritmo de Weasel-Muse a cada variável independentemente e posteriormente concatenam-se as codificações obtidas para cada variável.

Fonte: Produção do autor.

Tabela A.58 - Experimento B (GTSH/VCT) anos 2010-2018 - Desempenho de predição sem informação de TEC futuro e com tamanho de palavra 4, classe OC significando cintilação forte-moderado.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| TEC | GTSH | XGBoost | 0,51 | 0,51 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| AE | GTSH | XGBoost | 0,45 | 0,46 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,39 |
| | VCT | XGBoost | 0,52 | 0,53 | 0,39 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| AP | GTSH | XGBoost | 0,47 | 0,47 | 0,39 |
| | | CatBoost | 0,48 | 0,49 | 0,32 |
| | VCT | XGBoost | 0,48 | 0,49 | 0,32 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| IMF By | GTSH | XGBoost | 0,50 | 0,51 | 0,38 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| IMF Bz | GTSH | XGBoost | 0,48 | 0,49 | 0,36 |
| | | CatBoost | 0,51 | 0,51 | 0,40 |

Continua

Tabela A.58 – Continuação.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Dst | GTSH | XGBoost | 0,57 | 0,58 | 0,54 |
| | | CatBoost | 0,49 | 0,50 | 0,35 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,37 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| h'F | GTSH | XGBoost | 0,41 | 0,42 | 0,34 |
| | | CatBoost | 0,48 | 0,49 | 0,32 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| $P_{sw}$ | GTSH | XGBoost | 0,52 | 0,53 | 0,41 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| $V_{sw}$ | GTSH | XGBoost | 0,48 | 0,49 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Sym-D | GTSH | XGBoost | 0,51 | 0,51 | 0,40 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,51 | 0,51 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Sym-H | GTSH | XGBoost | 0,53 | 0,54 | 0,47 |
| | | CatBoost | 0,48 | 0,49 | 0,32 |
| | VCT | XGBoost | 0,48 | 0,49 | 0,32 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| AE+TEC | GTSH | XGBoost | 0,47 | 0,47 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,35 |
| | VCT | XGBoost | 0,52 | 0,52 | 0,39 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |

Continua

Tabela A.58 – Continuação.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| Ap+TEC | GTSH | XGBoost | 0,48 | 0,48 | 0,38 |
| | | CatBoost | 0,49 | 0,50 | 0,35 |
| | VCT | XGBoost | 0,51 | 0,51 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| IMF By+TEC | GTSH | XGBoost | 0,52 | 0,53 | 0,40 |
| | | CatBoost | 0,50 | 0,51 | 0,36 |
| | VCT | XGBoost | 0,51 | 0,51 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| IMF Bz+TEC | GTSH | XGBoost | 0,48 | 0,49 | 0,36 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,48 | 0,49 | 0,32 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Dst+TEC | GTSH | XGBoost | 0,52 | 0,52 | 0,46 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| h'F+TEC | GTSH | XGBoost | 0,48 | 0,49 | 0,41 |
| | | CatBoost | 0,50 | 0,51 | 0,38 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| $P_{sw}$+TEC | GTSH | XGBoost | 0,56 | 0,56 | 0,48 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| $V_{sw}$+TEC | GTSH | XGBoost | 0,51 | 0,51 | 0,40 |
| | | CatBoost | 0,51 | 0,51 | 0,36 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Sym-D+TEC | GTSH | XGBoost | 0,57 | 0,58 | 0,50 |
| | | CatBoost | 0,49 | 0,50 | 0,35 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |

Continua

Tabela A.58 – Continuação.

| Variável | Validação | Modelo | Acurácia | Precisão | $F_1$ |
|---|---|---|---|---|---|
| Sym-H+TEC | GTSH | XGBoost | 0,52 | 0,52 | 0,44 |
| | | CatBoost | 0,48 | 0,49 | 0,42 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Todos (i) | GTSH | XGBoost | 0,47 | 0,47 | 0,37 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| Todos (ii) | GTSH | XGBoost | 0,45 | 0,46 | 0,35 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |
| | VCT | XGBoost | 0,49 | 0,50 | 0,33 |
| | | CatBoost | 0,49 | 0,50 | 0,33 |

Todos se apresentam com duas possibilidades: (i) aplica-se o algoritmo de Weasel-Muse a todas as variáveis para geração do vetor de atributos; e (ii) aplica-se o algoritmo de Weasel-Muse a cada variável independentemente e posteriormente concatenam-se as codificações obtidas para cada variável.

Fonte: Produção do autor.

## A.5 Experimento C (GTSH) anos 2010-2018

O Experimento C (GTSH) anos 2010-2018 foi realizado com um conjunto inicial de variáveis: $AE$, $IMFB_y$, $IMFB_z$, $Sym\text{-}H$, $Sym\text{-}D$, $V_{sw}$, $P_{sw}$, $ap$, $Dst$, $F10.7$, $Sunspot$, $TEC$, $S_4$, denominado de conjunto Original de variáveis. Posteriormente, tornou-se disponível e foi adicionada a variável $h'F$. E, finalmente, foram adicionadas outras 4 variáveis: (i) tempo (instante do dia em minutos) representado como uma onda de período 1440 ($24\times60$) minutos pelo correspondente sin e cos do tempo; (ii) $TEC$ e $S_4$ em São Luiz; (iii) TEC e $S_4$ dos primeiros vizinhos do ponto de grade associado à São José dos Campos totalizando 8 elementos; (iv) Laplaciano do TEC em função dos 4 primeiros vizinhos do mesmo ponto de grade.

As Tabelas A.59 à A.62 apresentam os resultados do Experimento C (GTSH) anos 2010-2018 que utilizaram como modelo as redes neurais, em particular a rede apresenta na Figura 4.17. Essas tabelas tem 5 colunas: a primeira indica as variáveis utilizadas; a segunda indica a(s) estação(ões) do ano das amostras utilizados no treinamento (verão, ou as 4 estações do ano); a terceira coluna indica a antecedência da predição; e a quarta e a quinta colunas apresentam as métricas de desempenho de predição Acurácia e $F_1$.

Tabela A.59 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para OC significando forte-moderado-fraco.

| Variáveis | Estações do Ano | Antecedência (min) | Acurácia | $F_1$ |
|---|---|---|---|---|
| Original | Todas | 30 | 0,82 | 0,82 |
| | | 60 | 0,62 | 0,58 |
| | | 90 | 0,58 | 0,50 |
| | | 120 | 0,56 | 0,48 |
| | | 150 | 0,56 | 0,47 |
| | | 180 | 0,54 | 0,45 |
| | Verão | 30 | 0,77 | 0,77 |
| | | 60 | 0,64 | 0,60 |
| | | 90 | 0,65 | 0,62 |
| | | 120 | 0,67 | 0,67 |
| | | 150 | 0,59 | 0,53 |
| | | 180 | 0,56 | 0,48 |
| Original + h'F | Todas | 30 | 0,81 | 0,79 |
| | | 60 | 0,66 | 0,63 |
| | | 90 | 0,65 | 0,64 |
| | | 120 | 0,52 | 0,41 |
| | | 150 | 0,55 | 0,45 |
| | | 180 | 0,52 | 0,38 |
| | Verão | 30 | 0,81 | 0,80 |
| | | 60 | 0,74 | 0,73 |
| | | 90 | 0,70 | 0,69 |
| | | 120 | 0,60 | 0,54 |
| | | 150 | 0,51 | 0,35 |
| | | 180 | 0,51 | 0,34 |

Fonte: Produção do autor.

Assim, como acontecia no Experimento A (GTSH/TSCV-GKF) anos 2010-2018, quanto maior a antecedência de predição menor o valor de $F_1$. O Experimento C (GTSH) anos 2010-2018 foi avaliado utilizando GTSH, e tem o subconjunto de teste, treinamento e validação definido no mesmo período do Experimento A (GTSH/TSCV-GKF) anos 2010-2018, o que justifica uma semelhança entre os resultados obtidos, note que essa comparação é melhor feita considerando o período das 18-06 h do Experimento A (GTSH/TSCV-GKF) anos 2010-2018, pois os modelos do Experimento C (GTSH) foram treinados apenas para o período das 18-06 h.

A Tabela A.59 apresenta os resultados considerando que a classe OC seja forte-moderado-fraco, enquanto a Tabela A.60 para OC sendo forte-moderado, e finalmente A.60 para OC sendo apenas forte. Comparando os resultados, em termos do $F_1$, apresentados nessas tabelas fica claro que o $F_1$ cai conforme a classe OC fica mais restrita. Comportamento semelhante foi evidenciado no Experimento B (GTSH/VCT) anos 2010-2018, para o caso

forte-moderado, que foi pior que o caso forte-moderado-fraco.

Situações com $F_1$ próximo ao valor de $0,5$ indicam que o modelo não foi capaz de aprender a diferenciar entre OC e N-OC, classificando praticamente todas as amostras como N-OC.

Observando a Tabela A.59 é possível concluir que os modelos desenvolvidos somente para o verão tendem a ser levemente melhores que os desenvolvidos para qualquer estação do ano. Também é possível notar que em geral a adição da variável $h'F$ melhora os resultados, porém existem casos em que isso não acontece, ou seja, a adição de $h'F$ degrada o desempenho do modelo.

Tabela A.60 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para OC significando forte-moderado.

| Variáveis | Estações do Ano | Passo de Tempo (min) | Acurácia | $F_1$ |
|---|---|---|---|---|
| Original | Todas | 30 | 0,90 | 0,66 |
| | | 60 | 0,88 | 0,47 |
| | | 90 | 0,87 | 0,47 |
| | | 120 | 0,87 | 0,46 |
| | | 150 | 0,88 | 0,47 |
| | | 180 | 0,88 | 0,47 |
| | Verão | 30 | 0,90 | 0,66 |
| | | 60 | 0,88 | 0,47 |
| | | 90 | 0,87 | 0,47 |
| | | 120 | 0,87 | 0,47 |
| | | 150 | 0,88 | 0,47 |
| | | 180 | 0,88 | 0,47 |
| Original+h'F | Todas | 30 | 0,90 | 0,64 |
| | | 60 | 0,88 | 0,49 |
| | | 90 | 0,87 | 0,47 |
| | | 120 | 0,87 | 0,47 |
| | | 150 | 0,88 | 0,47 |
| | | 180 | 0,88 | 0,47 |
| | Verão | 30 | 0,89 | 0,47 |
| | | 60 | 0,88 | 0,47 |
| | | 90 | 0,88 | 0,47 |
| | | 120 | 0,88 | 0,47 |
| | | 150 | 0,88 | 0,47 |
| | | 180 | 0,88 | 0,47 |

Fonte: Produção do autor.

Tabela A.61 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição para OC significando forte.

| Variáveis Adicionais | Estações do Ano | Antecedência (min) | Acurácia | $F_1$ |
|---|---|---|---|---|
| Original | Todas | 30 | 0,99 | 0,50 |
| | | 60 | 0,99 | 0,50 |
| | | 90 | 0,98 | 0,50 |
| | | 120 | 0,98 | 0,50 |
| | | 150 | 0,98 | 0,50 |
| | | 180 | 0,98 | 0,50 |
| | Verão | 30 | 0,99 | 0,50 |
| | | 60 | 0,99 | 0,50 |
| | | 90 | 0,98 | 0,50 |
| | | 120 | 0,98 | 0,49 |
| | | 150 | 0,98 | 0,49 |
| | | 180 | 0,98 | 0,49 |
| Original+h'F | Todas | 30 | 0,99 | 0,50 |
| | | 60 | 0,99 | 0,50 |
| | | 90 | 0,98 | 0,49 |
| | | 120 | 0,98 | 0,49 |
| | | 150 | 0,98 | 0,49 |
| | | 180 | 0,98 | 0,49 |
| | Verão | 30 | 0,99 | 0,50 |
| | | 60 | 0,99 | 0,50 |
| | | 90 | 0,98 | 0,50 |
| | | 120 | 0,98 | 0,50 |
| | | 150 | 0,98 | 0,49 |
| | | 180 | 0,98 | 0,49 |

Fonte: Produção do autor.

Comparando as Tabelas A.59 e A.62 é possível notar que a adição de informação sobre o Tempo, degrada o modelo; a adição de informação sobre São Luiz degrada o modelo para alguns valores de passo e melhora o modelo para outros; a adição do $TEC$ e $S_4$ dos primeiros vizinhos degrada os resultados para os quatro primeiros passos de tempo, porém melhora para os dois últimos e apresenta o melhor valor de $F_1$ para o passo de 180 min entre os modelos baseados em redes neurais; a adição do laplaciano do TEC nos primeiros vizinhos degrada os passos de 60 min e 90 min e melhora os demais; finalmente a adição de todas as variáveis degrada o desempenho do modelo.

Tabela A.62 - Experimento C (GTSH) anos 2010-2018 - Desempenho de predição em função de uso de variáveis adicionais, para classe OC significando cintilação forte-moderado-fraco.

| Variáveis | Estações do Ano | Antecedência (min) | Acurácia | $F_1$ |
|---|---|---|---|---|
| Original + h'F + Tempo UTC (min) | Verão | 30 | 0,72 | 0,70 |
| | | 60 | 0,50 | 0,35 |
| | | 90 | 0,53 | 0,42 |
| | | 120 | 0,50 | 0,36 |
| | | 150 | 0,52 | 0,37 |
| | | 180 | 0,51 | 0,36 |
| Original + h'F + TEC e $S_4$ (São Luís) | Verão | 30 | 0,77 | 0,76 |
| | | 60 | 0,62 | 0,57 |
| | | 90 | 0,65 | 0,62 |
| | | 120 | 0,61 | 0,57 |
| | | 150 | 0,59 | 0,53 |
| | | 180 | 0,50 | 0,34 |
| Original + h'F + TEC e $S_4$ nos 8 vizinhos | Verão | 30 | 0,79 | 0,78 |
| | | 60 | 0,65 | 0,62 |
| | | 90 | 0,62 | 0,59 |
| | | 120 | 0,54 | 0,42 |
| | | 150 | 0,57 | 0,48 |
| | | 180 | 0,64 | 0,62 |
| Original + h'F + Laplaciano do TEC nos 4 primeiros vizinhos | Verão | 30 | 0,82 | 0,82 |
| | | 60 | 0,69 | 0,67 |
| | | 90 | 0,59 | 0,54 |
| | | 120 | 0,60 | 0,54 |
| | | 150 | 0,58 | 0,51 |
| | | 180 | 0,56 | 0,49 |
| Todas | Verão | 30 | 0,63 | 0,60 |
| | | 60 | 0,57 | 0,46 |
| | | 90 | 0,52 | 0,38 |
| | | 120 | 0,52 | 0,38 |
| | | 150 | 0,56 | 0,46 |
| | | 180 | 0,51 | 0,33 |

Fonte: Produção do autor.