Revista brasileira de iniciação científica
ISSN: 2359-232X

Processamento automático de curvas de luz para a identificação de exoplanetas por meio de uso de algoritmos de aprendizado de máquina

Automatic light curve processing for exoplanet identification using machine learning algorithms

Procesamiento automático de curvas de luz para la identificación de exoplanetas por meio de algoritmos de aprendizaje automático

Bruno Henrique Dourado Macedo[1]
Willian Zalewski[2]

Resumo: abordagens baseadas em técnicas de aprendizado de máquina têm sido propostas na literatura para auxiliar a detecção de exoplanetas por meio do processamento automatizado de curvas de luz. Apesar dos avanços, algoritmos de aprendizado de máquina considerados tradicionais ainda não foram completamente estudados para essa tarefa. Portanto, neste trabalho, define-se um *baseline* por meio de uma ampla avaliação experimental a respeito de 16 algoritmos em diferentes ajustes de parâmetros. Para atingir esse objetivo, utilizaram-se dados provenientes do telescópio Kepler, totalizando 5302 curvas de luz; cada uma com 60000 registros. Como principal resultado da avaliação experimental, o algoritmo LightGBM apresentou o melhor desempenho, com taxa de 82,92%, em termos de acurácia.
Palavras-chave: astronomia. exoplanetas. aprendizado de máquina. curvas de luz.

Abstract: approaches based on machine learning techniques have been proposed in the literature to assist in the detection of exoplanets through automated processing of light curves. Despite advancements, traditional machine learning algorithms have not yet been fully studied for this task. Therefore, in this work, we proposed the definition of a baseline through an extensive experimental evaluation involving 16 algorithms with different parameter settings. To achieve this goal, in this study, data from the Kepler telescope was used, totaling 5302 light curves, each with 60000 records. As the main result of the experimental evaluation, the LightGBM algorithm showed the best performance, with an accuracy rate of 82.92%.
Keywords: astronomy. exoplanet. machine learning. light curve.

Resumen: se han propuesto en la literatura enfoques basados en técnicas de aprendizaje automático para ayudar en la detección de exoplanetas mediante el procesamiento automatizado de curvas de luz. A pesar de los avances, los algoritmos de aprendizaje automático considerados tradicionales aún no han sido completamente estudiados para esta tarea. Por lo tanto, en este trabajo, propusimos la definición de un punto de referencia a través de una amplia evaluación experimental que involucra 16 algoritmos con diferentes ajustes de parámetros. En este estudio, se utilizaron datos provenientes del telescopio Kepler, sumando un total de 5302 curvas de luz, con 60000 registros cada. Como resultado principal de la evaluación experimental, el algoritmo LightGBM mostró el mejor rendimiento, con una precisión del 82,92%.
Palabras-clave: astronomía. exoplaneta. aprendizaje automático. curva de luz.

Submetido 17/11/2023 **Aceito 01/03/2024** **Publicado 05/04/2024**

[1] Graduando em Engenharia Física. Universidade Federal da Integração Latino-Americana – UNILA.
ORCID: https://orcid.org/0000-0002-8152-0950. E-mail: brunohdmacedo@gmail.com
[2] Doutor em Ciência da Computação. Universidade Federal da Integração Latino-Americana – UNILA.
ORCID: https://orcid.org/0000-0002-7113-5135. E-mail: willian.zalewski@unila.edu.br

Revista brasileira de iniciação científica
ISSN: 2359-232X

Considerações Iniciais

Nas últimas décadas, diversas tecnologias possibilitaram a exploração sistemática de galáxias, impulsionando novas descobertas sobre fenômenos do universo. Missões espaciais como CoRoT[3] (*COnvection ROtation et Transits planétaires*), NuSTAR[4] (*Nuclear Spectroscopic Telescope Array*), NEOWISE[5] (*Wide-field Infrared Survey Explorer*), Gaia[6], Hubble[7], Kepler[8], TESS[9] (*Transiting Exoplanet Survey Satellite*) e o mais recente Telescópio Espacial James Webb desempenharam um papel crucial em relação a esses avanços. Os dados dessas missões contribuem para o refinamento contínuo das técnicas no campo da astronomia (Souza, 2019). Dentre as principais contribuições dessas missões espaciais, destaca-se a identificação de exoplanetas, ou seja, planetas que orbitam estrelas fora do Sistema Solar. A captação de dados na forma de intensidade luminosa das estrelas tem acelerado essa tarefa. Denomina-se a intensidade luminosa de um objeto celeste registrada ao longo do tempo de curva de luz. As curvas de luz são particularmente úteis para a identificação de exoplanetas por meio do uso de método de trânsito planetário, em que planetas passam entre a estrela e o observador, alterando, temporariamente, o brilho daquela devido à sua ocultação parcial (Castrillón, 2010). Identificaram-se, por meio dessa técnica, 76% (3333) dos exoplanetas nos dados do Kepler. No entanto, o aumento exponencial no volume de dados tornou as técnicas tradicionais de análise visual impraticáveis para lidar, eficientemente, com essas informações (Babu, 2012). Como exemplo, a missão Kepler, lançada em 2009, permitiu a observação de cerca de 530.000 estrelas, produzindo 678 GB de dados até o encerramento da missão em 2018.

Diante desse cenário, estudos propõem o desenvolvimento de métodos computacionais para automatizar a identificação de exoplanetas (Blomme, 2012; Richards, 2011; Armstrong et al., 2017; Hinners et al., 2018; Malik et al., 2020; Jara-Maldonado et al., 2020), especialmente por meio da aplicação de técnicas de aprendizado de máquina (AM) (Fayyad et al., 1996; Mitchell, 1997; Rezende, 2003). Apesar dos avanços e da utilização de técnicas recentes como *deep learning,* ainda há algumas lacunas que necessitam ser adequadamente exploradas. Os

[3] Telescópio Espacial CoRoT: https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2006-063A
[4] Telescópio Espacial NuSTAR: http://www.nustar.caltech.edu/
[5] Telescópio Espacial NEOWISE: https://neowise.ipac.caltech.edu/
[6] Telescópio Espacial Gaia: https://sci.esa.int/web/gaia/
[7] Telescópio Espacial Hubble: https://science.nasa.gov/mission/hubble
[8] Telescópio Espacial Kepler: https://science.nasa.gov/mission/kepler
[9] Telescópio Espacial TESS: https://tess.mit.edu/

Revista brasileira de iniciação científica
ISSN: 2359-232X

algoritmos de AM considerados tradicionais ainda não foram completamente estudados para essa tarefa, utilizando curvas de luz. Nesse contexto, o objetivo deste trabalho consiste em definir um baseline para literatura por meio da proposição de um método de avaliação experimental considerando diferentes ajustes de parâmetros dos algoritmos de AM tradicionais.

Na busca por exoplanetas, cientistas utilizam diversas técnicas para descobrir planetas fora do nosso sistema solar, cada uma revelando diferentes características dos exoplanetas, como composição atmosférica, temperatura e massa das estrelas que eles orbitam. De acordo com os dados apresentados na tabela 1, o método de trânsito planetário é o mais produtivo em termos de detecções, e representa 76,10% das descobertas de exoplanetas.

Tabela 1 – técnicas de detecção de exoplanetas

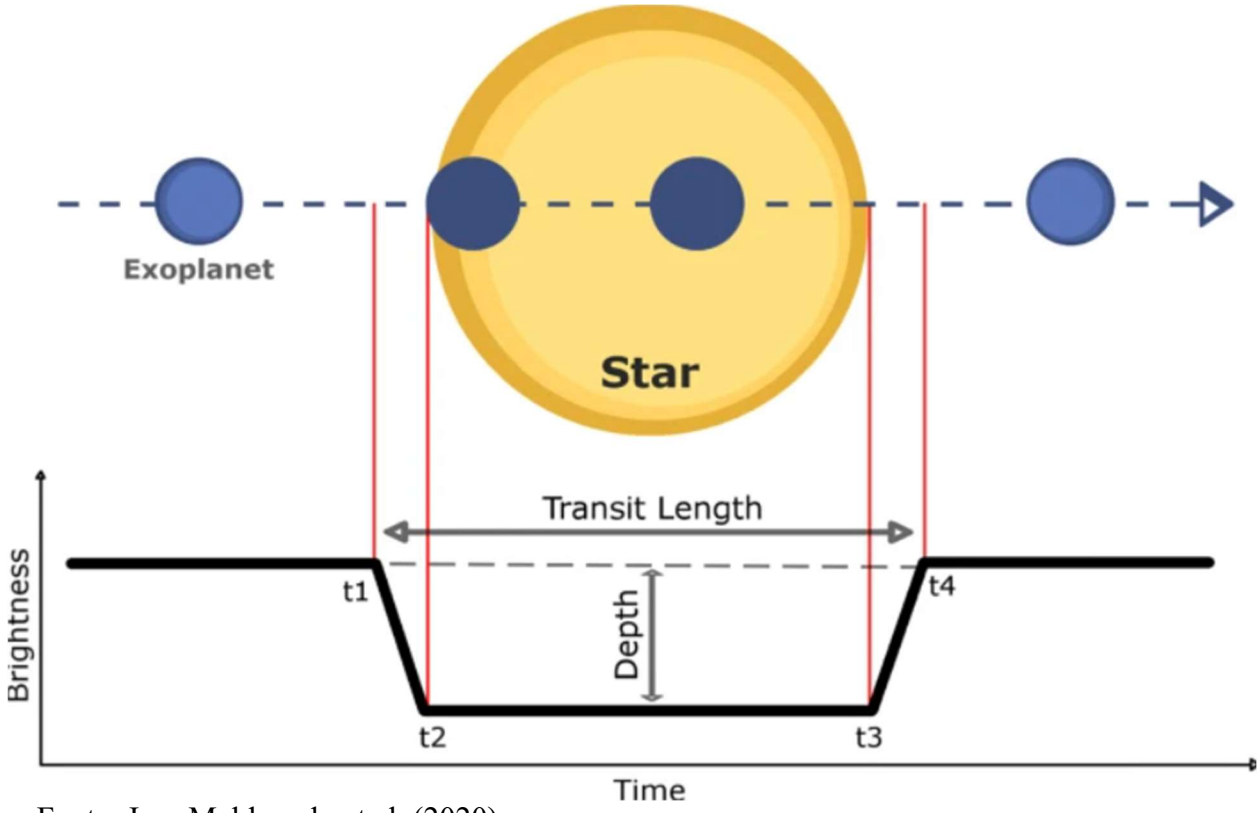
Técnica	**Detecções**
Trânsito planetário	76,10%
Velocidade radial	19,10%
Microlente gravitacional	2,40%
Imagem direta	1,20%
Outras técnicas	1,20%

Fonte: exoplanets.nasa.gov

Um trânsito planetário é semelhante a um eclipse solar, em que um exoplaneta passa entre sua estrela hospedeira e o observador, causando uma redução temporária na intensidade da luz da estrela. Observa-se esse evento por meio de curvas de luz, que revelam a intensidade da luz estelar ao longo do tempo. Durante um trânsito, a curva de luz exibe uma queda no brilho da estrela, indicando a passagem do exoplaneta. Conforme a figura 1, os parâmetros importantes derivados dessas curvas de luz incluem o início e término da entrada e saída do trânsito (t1, t2, t3, t4), a duração total do trânsito, e a profundidade do trânsito (Jara-Maldonado et al., 2020). Esse método tem sido fundamental para as descobertas significativas de exoplanetas, oferecendo *insights* sobre os sistemas planetários além do nosso.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Figura 1 – representação esquemática do trânsito planetário



Fonte: Jara-Maldonado et al. (2020).

O avanço tecnológico proporcionado por telescópios espaciais como Kepler e TESS resultou em volumes significativos de dados de curvas de luz, oferecendo oportunidades significativas para a descoberta de exoplanetas. No entanto, essa abundância de dados também apresenta desafios significativos em termos de análise devido à sua complexidade e quantidade. Métodos tradicionais de identificação tornaram-se inadequados e suscetíveis a erros humanos diante dessa imensa quantidade de informações (Armstrong et al., 2017).

Para superar esses desafios, é crucial a aplicação de técnicas automatizadas baseadas em aprendizado de máquina e inteligência artificial (Hinners et al., 2018). Estudos recentes exploram diversas abordagens inovadoras. Por exemplo, Shallue e Vanderburg (2017) introduziram o *Astronet*, uma arquitetura de aprendizado profundo; enquanto Malik et al. (2020) utilizaram séries temporais para extrair características de curvas de luz, demonstrando a eficácia desses métodos avançados.

Além disso, em pesquisas mais recentes, propuseram-se métodos inovadores para a detecção de exoplanetas em dados de trânsito. Por exemplo, Visser, Bosma e Postma (2021)

apresentaram um novo método que utiliza uma rede neural convolucional intitulada Genesis, combinando dados reais e sintéticos para aumentar a precisão na detecção de exoplanetas. Cuéllar et al. (2022) desenvolveram um método baseado em aprendizado profundo que combina dados reais e sintéticos, usando uma CNN treinada com dados combinados para aumentar a confiabilidade das descobertas. Além disso, Visser, Bosma e Postma (2022) propuseram o uso de redes neurais convolucionais esparsas, por meio do aproveitamento de informações sobre a duração do trânsito e a periodicidade das observações, a fim de selecionar, cuidadosamente, um subconjunto de imagens de curvas de luz no treinamento da rede neural.

Essas pesquisas exemplificam os avanços recentes e as múltiplas abordagens inovadoras exploradas para enfrentar os desafios complexos associados à análise de dados de trânsito e à detecção de exoplanetas, por meio da combinação de inteligência artificial, aprendizado de máquina, dados reais e sintéticos para alcançar resultados mais precisos e confiáveis na pesquisa de exoplanetas.

Neste trabalho, utilizaram-se dados coletados pelo telescópio Kepler, disponíveis no Arquivo de Exoplanetas da NASA. O Kepler observou estrelas por meio de 21 pares de módulos, totalizando 84 canais de observação. Criaram-se curvas de luz com base nesses dados, medindo a luminosidade das estrelas ao longo do tempo. Esse procedimento permitiu a visualização do comportamento das estrelas e foi crucial para a pesquisa.

Neste estudo, utilizaram-se curvas de luz de longa cadência, com taxas de captura de 29,4 minutos, obtidas com base nos dados dos KOIs do catálogo *online* da NASA. Selecionaram-se as colunas relevantes, incluindo *kepid*, *koi_disposition*, *koi_period*, *koi_time0bk*, *koi_duration* e *koi_quarters*. A biblioteca *Lightkurve* facilitou a obtenção das curvas de luz, resultando em dois fluxos de luminosidade distintos: Fotometria de Abertura Simples (SAP) e Condicionamento de Dados Pré-busca (PDCSAP).

Com os efeitos sistemáticos que afetam as curvas de luz do telescópio Kepler, como mudanças de foco, temperatura e o efeito DVA, optou-se por usar o fluxo PDCSAP, adaptado para detecção de exoplanetas, devido à sua precisão. Normalizaram-se diferenças de intensidade de luz causadas pelo movimento orbital do satélite por meio da função *stitch()*, alinhando as profundidades dos trânsitos. Após a eliminação de dados ausentes na coluna *koi_quarters*, analisaram-se 5302 objetos, dos quais 58,60% implicaram resultados falsos

positivos e 41,40% resultados confirmados, cada um com cerca de 60.000 pontos de dados devido aos 17 trimestres.

Metodologia

Pontos de dados ausentes em curvas de luz podem afetar os algoritmos de classificação projetados para análise de séries temporais. Avaliou-se o impacto desses valores ausentes nos resultados, identificando e tratando-os por meio de interpolação linear. Eliminaram-se *outliers* usando um desvio padrão de 2, permitindo descartar leituras instrumentais errôneas e medições afetadas por raios cósmicos (Shallue; Vanderburg, 2017) (Shallue; Vanderburg, 2017).

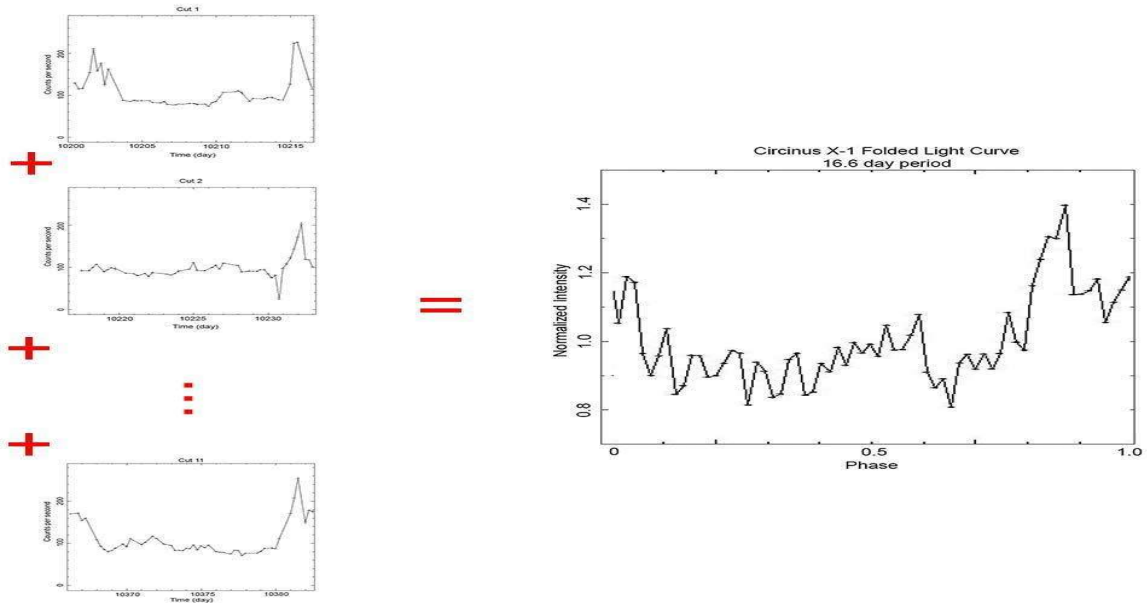
Utilizou-se a normalização por escala para padronizar cada uma das curvas de luz em nosso conjunto de dados. De acordo com esse método, ajustam-se todos os valores $t_i \in$ T considerando-se a faixa [0, 1], conforme a equação. Representam-se os menores e maiores valores de T por min(T) e max(T), respectivamente (Zalewski, 2015).

$$t_i' = \frac{(t_i - \min(T)}{\max(T) - \min(T)}$$

Para criar o conjunto de dados, utilizou-se a técnica de dobramento do período nas curvas de luz, seguindo o método descrito por Shallue e Vanderburg (2017). Esse processo envolve alinhar cada curva de luz com seu período (*koi_period*) e implementar o dobramento do período centrado no evento de trânsito (*koi_time0bk*). Empregou-se o método de *binning* para gerar um único vetor unidimensional a partir das curvas dobradas, utilizando duas representações: uma local, com 201 pontos, e uma global, com 2001 pontos. Esse procedimento resulta em uma única curva de luz amalgamada com base nas intensidades das curvas de luz divididas, conforme a figura 2.

Revista brasileira de iniciação científica
ISSN: 2359-232X

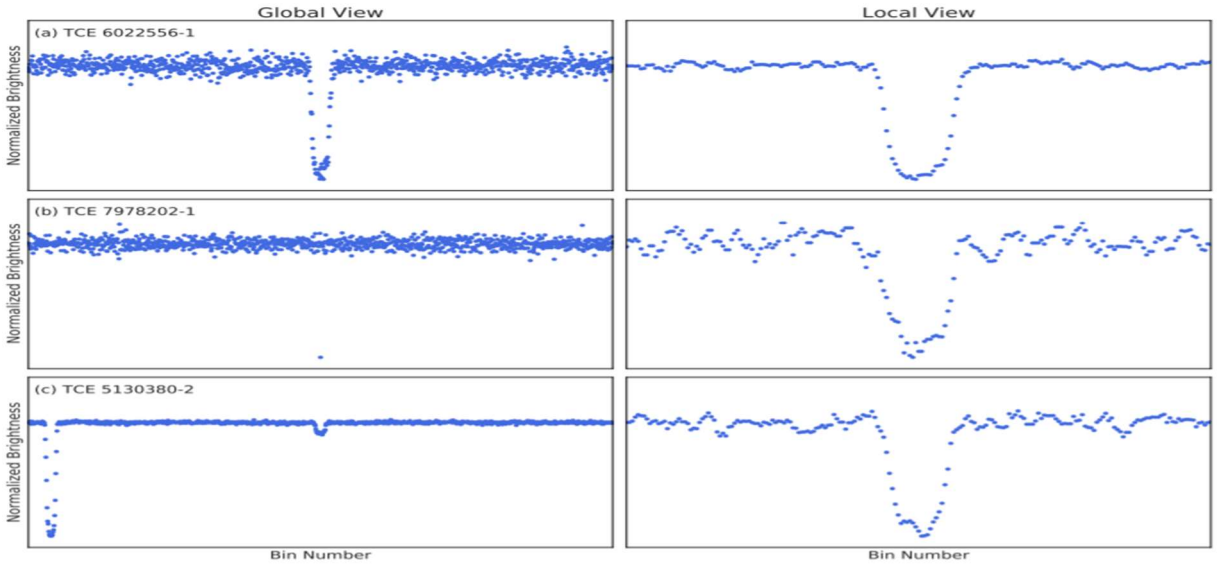
Figura 2 – combinação das curvas segmentadas formando uma única curva do mesmo tamanho que o período



Fonte: NASA's Imagine the Universe.

A representação global abrange toda a curva de luz, enquanto a representação local constitui uma janela focada em um evento de trânsito na curva em estudo, conforme a figura 3.

Figura 3 – representações local e global



Fonte: Shallue; Vanderburg (2017).

Na Figura 3, apresentam-se 3 curvas de luzes (a, b, c) com as representações global e local proposta pelo método descrito por Shallue e Vanderburg (2017).

Algoritmos de Classificação

Neste trabalho, utilizam-se 16 algoritmos tradicionais de classificação: (1) *k-Nearest Neighbor*s (*KNN*), (2) *Multi-Layer Perceptron* (*MLP*), (3) *Support Vector Machines* (*SVM*), (4) *Logistic Regression* (*LR*), (5) *Naive Bayes* (*NB*), (6) *Decision Trees* (*DT*), (7) *Gaussian Process* (*GBC*), (8) *Ridge*, (9) *Random Forest* (*RF*), (10) *Extra Trees* (*ET*), (11) *Linear Discriminant Analysis* (*LDA*), (12) *Quadratic Discriminant Analysis* (*QDA*), (13) *XGBoost* (*XGB*), (14) *LightGBM* (*LGBM*), (15) *CatBoost* (*CATB*) e (16) *AdaBoost* (*ADA*).

1. ***k-Nearest Neighbors* (*KNN*)**: o algoritmo observa os k exemplos mais próximos e faz uma votação majoritária para tarefas de classificação. Métricas de distância como a distância euclidiana ou a distância de Manhattan são frequentemente usadas para encontrar os vizinhos mais próximos (Cover; Hart, 1967).

2. ***Multi-Layer Perceptron* (*MLP*)**: essa é uma classe de rede neural artificial do tipo *feedforward* que consiste em, pelo menos, três camadas de nós: uma camada de entrada, uma camada oculta e uma camada de saída. Exceto pelos nós de entrada, cada nó constitui um neurônio que utiliza uma função de ativação não linear, o que ajuda a rede a aprender padrões complexos. MLP utiliza uma técnica de aprendizado supervisionado chamada retropropagação (*backpropagation*) para treinar a rede (Haykin, 2001).

3. ***Support Vector Machines* (*SVM*)**: dado um conjunto de dados de treinamento rotulados, o algoritmo produz um hiperplano ótimo que separa as diferentes classes. Esse hiperplano é escolhido para maximizar a margem, que é a distância entre o hiperplano e o ponto de dados mais próximo de cada classe. O kernel pode ser usado para lidar com classes que não são linearmente separáveis (Cortes; Vapnik, 1995).

4. ***Logistic Regression* (*LR*)**: apesar do nome, a Regressão Logística é um algoritmo de classificação usado para estimar a probabilidade de uma resposta binária com base em uma ou mais variáveis preditoras (independentes). Ele trabalha com uma função logística, que consiste em uma curva em forma de "S" que pode receber qualquer número real e mapeá-lo para um valor entre 0 e 1, mas nunca exatamente nos limites (Russel; Norvig, 2013).

5. ***Naive Bayes* (*NB*)**: este é um conjunto de algoritmos de aprendizado supervisionado, baseados na aplicação do teorema de *Bayes* com a suposição "ingênua" de independência condicional entre cada par de características, dada o valor da variável de classe. Embora a suposição de independência possa não ser válida em muitos casos, os classificadores *Naive Bayes*, frequentemente, apresentam um desempenho muito bom na prática (Russel; Norvig, 2013).

6. ***Decision Trees* (*DT*)**: este algoritmo cria um modelo que prevê o valor de uma variável alvo aprendendo regras de decisão simples inferidas com base nas características dos dados. Eles são simples de entender e visualizar, e podem lidar tanto com dados categóricos quanto numéricos (Breiman, 1984).

7. ***Gaussian Process* (*GBC*)**: este é um método genérico de aprendizado supervisionado, projetado para resolver problemas de regressão e classificação probabilística. Ele fornece uma abordagem probabilística, ou seja, em vez de fornecer apenas uma única melhor previsão, também fornece uma medida de incerteza, o que pode ser valioso em muitas aplicações (Rasmussen; Williams, 2006).

8. ***Ridge***: utiliza-se este método para mitigar o problema de multicolinearidade na regressão linear. Na multicolinearidade, as variáveis preditoras estão altamente correlacionadas, tornando difícil desvendar o efeito de cada preditor sobre a variável de resposta. A regressão *ridge* adiciona um grau de viés às estimativas de regressão, o que pode reduzir os erros padrão (Hoerl; Kennard, 1970).

9. ***Random Forest* (*RF*)**: este é um algoritmo de aprendizado de máquina versátil capaz de realizar tanto tarefas de regressão quanto de classificação. Também é usado para redução de dimensionalidade e tratamento de valores ausentes e valores discrepantes. É um tipo de método de aprendizado em conjunto, em que um grupo de árvores de decisão se combina para formar um único modelo (Breiman et al., 2001).

10. ***Extra Trees* (*ET*)**: *Extremely Randomized Trees* constrói várias árvores de decisão com dois tipos de aleatoriedade: em cada nó, um subconjunto aleatório de características candidatas é gerado, e um limite para cada característica é selecionado aleatoriamente para definir a condição de divisão. O melhor par de características e limite é escolhido para a divisão efetiva.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Essa aleatoriedade aumentada pode ajudar a reduzir a variância do modelo, em troca de aumentar o viés (Geurts et al., 2006).

11. ***Linear Discriminant Analysis* (*LDA*)**: este constitui uma generalização do discriminante linear de Fisher, um método utilizado em estatística, reconhecimento de padrões e aprendizado de máquina para encontrar uma combinação linear de características que caracterize ou separe duas ou mais classes de objetos ou eventos (Hastie; Tibshirani; Friedman, 2001).

12. ***Quadratic Discriminant Analysis* (*QDA*)**: está intimamente relacionado ao LDA, mas, em vez de assumir que as covariâncias das classes são idênticas, o QDA assume que cada classe possui sua própria matriz de covariância. Em outras palavras, a fronteira é quadrática (Hastie; TibshiranI; Friedman, 2001).

13. ***XGBoost* (*XGB*)**: *XGBoost significa eXtreme Gradient Boosting*. É uma biblioteca otimizada de *boosting* de gradiente distribuído, projetada para ser altamente eficiente, flexível e portátil. Ele oferece um método de *boosting* de árvore paralelo que resolve muitos problemas de ciência de dados de maneira rápida e precisa. Implementa algoritmos de aprendizado de máquina no *framework* de *Gradient Boosting* e fornece várias funcionalidades avançadas para ajuste de modelo, regularização e otimização de desempenho (Chen; Guestrin, 2016).

14. ***LightGBM* (*LGBM*)**: *LightGBM*, abreviação de *Light Gradient Boosting Machine*, é um *framework* de *gradient boosting* que utiliza algoritmos de aprendizado baseados em árvores. Ao contrário de outros algoritmos baseados em árvores, que crescem as árvores de nível em nível, o *LightGBM* cresce as árvores folha por folha, escolhendo a folha com a maior perda delta para crescer, resultando em um melhor desempenho e eficiência do modelo (Ke et al., 2017).

15. ***CatBoost* (*CATB*)**: é um algoritmo de aprendizado de máquina que utiliza o *boosting* de gradiente em árvores de decisão. Outras características importantes incluem o *boosting* ordenado, que ajuda a reduzir o *overfitting*, e as árvores *oblivious*, um tipo de árvore de decisão mais eficiente para avaliação (Prokhorenkova et al., 2017).

16. ***AdaBoost* (*ADA*)**: *Adaptive Boosting* é um algoritmo de aprendizado de máquina usado como classificador. Quando utilizado em conjunto com aprendizado de árvore de decisão, as informações coletadas em cada estágio do algoritmo *AdaBoost* sobre a "dificuldade" relativa de

Revista brasileira de iniciação científica
ISSN: 2359-232X

cada amostra de treinamento são fornecidas ao algoritmo de crescimento da árvore de forma que árvores posteriores tendem a se concentrar em exemplos mais difíceis de classificar. Esse processo é adaptativo no sentido de que os aprendizes fracos subsequentes são ajustados em favor das instâncias classificadas erroneamente pelos classificadores anteriores (Freund; Schapire, 1997).

Otimização dos Parâmetros

Neste trabalho, para identificar os melhores parâmetros para cada algoritmo de classificação, utilizou-se a técnica de Otimização Bayesiana de Hiperparâmetros (Snoek; Larochelle; Adams, 2012). Essa é uma estratégia para a seleção eficiente de hiperparâmetros em algoritmos de aprendizado de máquina. Essa estratégia visa encontrar o valor máximo de uma função desconhecida. Nesse caso, o desempenho de um algoritmo de aprendizado de máquina, com o menor número possível de etapas. Inicia-se a estratégia com uma crença prévia sobre a função, utilizando-se sequencialmente esta em uma distribuição posterior para incorporar as informações coletadas com base nos dados observados. Em cada iteração, a função é otimizada para selecionar o próximo ponto de consulta, que se espera ter a maior melhoria em relação à melhor observação atual. O procedimento é repetido até que um critério de parada seja atendido. Esse pode ser um número pré-especificado de iterações ou até que as melhorias se tornem negligenciáveis.

Tabela 2 – espaço de busca dos parâmetros dos algoritmos de classificação

Algoritmos	**Parâmetros**
Logistic Regression	'C': Real(1e-4, 1e4, prior='log-uniform'), 'fit_intercept': Categorical([True, False]), 'solver': Categorical(['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']), 'max_iter':[500], 'random_state': [1]
Support Vector Machines	C:[1,3,5,10,15], Kernel: [‘rbf’, ‘linear’], tol:[1e-3,1e-4], Random_state:[1]
Decision Tree	‘criterion’: Categorical([‘gini’, ‘entropy’]), ‘splitter’: Categorical([‘best’, ‘random’]), ‘max_depth’: Integer(3,30), ‘min_samples_split’: Integer(2,10), ‘min_samples_leaf’: Integer(1,10),

Revista brasileira de iniciação científica
ISSN: 2359-232X

	‘max_features’: Real(0.1, 1.0, prior=’uniform’), ‘random_state’:[1]
Random Forests	‘n_estimadors’: Integer(10,500), ‘criterion’: Categorical([‘gini’, ‘entropy’]), ‘max_depth’: Integer(3,30), ‘random_state’: [1]
Quadratic Discriminant Analysis	‘reg_param’: Real(0, 1, prior=’uniform’), ‘store_covariance’: Categorical([True, False]), ‘tol’: Real(1e-5, 1e-1, prior=’log-uniform’)
Linear Discriminant Analysis	‘solver’: Categorical([‘svd’, ‘lsqr’, ‘eigen’]), ‘shrinkage’: Real(0, 1, prior=’uniform’), ‘tol’: Real(1e-6, 1e-4, prior=’log-uniform’)
Extra Trees	‘n_estimators’: Integer(10, 500), ‘criterion’: Categorical([‘gini’, ‘entropy’]), ‘max_depth’: Integer(3, 30)
Naive Bayes	‘var_smoothing’: Real(1e-10, 1e-1, prior=’log-uniform’)
k-Nearest Neighbors	‘n_neighbors’: Integer(1, 50), ‘weights’: Categorical([‘uniform’, ‘distance’]), ‘algorithm’: Categorical([‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’]), ‘p’:Integer(1, 5)
Multilayer Perceptron	‘hidden_layer_sizes’: Integer(10, 100), ‘activation’: Caegorical([‘logistic’, ‘tanh’, ‘relu’]), ‘solver’: Categorical([‘sgd’, ‘adam’]), ‘max_iter’: [5000], ‘random_state’:[1]
Ridge	‘alpha’: Real(1e-4, 1e4, prior=’log-uniform’), ‘fit_intercept’: Categorical([True, False]), ‘solver’: Categorical([‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse_cg’, ‘sag’, ‘saga’]), ‘random_state’: [1]
XBoost	‘leraning_rate’: Real(0.01, 0.3, prior=’uniform’), ‘n_estimators’: Integer(50, 500), ‘max_depth’: Integer(3, 10), ‘gamma’: Real(0, 1, prior=’uniform’)
LightGBM	'learning_rate': Real(1e-3, 1, prior='log-uniform'), 'n_estimators': Integer(10, 500), 'num_leaves': Integer(2, 100), 'max_depth': Integer(3, 10)
CatBoost	‘learning_rate’: Real(1e-3, 1, prior=log-uniform’), ‘iterations’: Integer(10, 500), ‘depth’: Integer(3, 10), ‘12_leaf_reg’: Real(1, 10, prior=’uniform’), ‘border_count’: Integer(1, 255), ‘bagging_temperature’: Ral(0, 1, prior=’uniform’), ‘random_strength’: Real(1e-9, 10, prior=’log-uniform’)

Revista brasileira de iniciação científica
ISSN: 2359-232X

AdaBoost	'n_estimators': Integer(10, 500), 'learning_rate': Real(1e-3, 1, prior='log-uniform'), 'algorithm': Categorical(['SAMME', 'SAMME.R']), 'random_state': [1]
Gaussian Proccess	'optimizer': Categorical(['fmin_l_bfgs_b', None]), 'n_restarts_optimizer': Integer(0, 10), 'max_iter_predict': [500], 'random_state': [1]

Fonte: autoral 2023.

Para implementar essa estratégia neste trabalho, utilizou-se a função *BayesSearchCV* da biblioteca *scikit-optimize*, configurada com 10 iterações e acurácia como métrica de desempenho. Na tabela 2, apresenta-se o espaço de busca de parâmetros para cada algoritmo.

Avaliação Experimental

Nesta seção, apresentam-se os procedimentos e técnicas adotadas neste trabalho para a avaliação experimental.

A Validação Cruzada Aninhada (*Nested* CV) é uma técnica usada para ajuste de hiperparâmetros e avaliação do modelo que fornece uma estimativa mais precisa do desempenho do modelo (Cawley; Talbot, 2010). Ela é particularmente benéfica em cenários em que os hiperparâmetros ótimos precisam ser escolhidos antes da avaliação final do modelo. Utiliza-se, frequentemente, a Validação Cruzada padrão com *k-folds* (CV), tanto para ajuste de hiperparâmetros quanto para avaliação do modelo. Na CV padrão, dividem-se os dados em k subconjuntos (ou *folds*). O modelo é treinado em k-1 *folds* e validado no restante, de forma que cada subconjunto sirva como conjunto de validação uma vez. Se esse método for usado para ajuste de hiperparâmetros, surge um problema importante: utilizam-se os mesmos dados para ajustar os hiperparâmetros e avaliar o desempenho do modelo, o que pode levar a uma estimativa excessivamente otimista do desempenho preditivo do modelo devido a vazamento de informações (Cawley; Talbot, 2010). A CV aninhada trata esse problema em dois níveis de validação cruzada. No *loop* interno (ou CV interno), ajustam-se os hiperparâmetros. No *loop* externo (ou CV externo), avalia-se o desempenho do modelo por meio de uso dos hiperparâmetros ótimos encontrados no *loop* interno. Neste trabalho, define-se o *loop* interno como CV = 3 e o *loop* externo como CV = 5. Além disso, para evitar problemas relacionados ao desequilíbrio do conjunto de dados usado, adotou-se a técnica de Validação Cruzada

Revista brasileira de iniciação científica
ISSN: 2359-232X

Estratificada, que seleciona, proporcionalmente, o mesmo número de exemplos de cada classe em cada *fold*.

Para avaliar os modelos de classificação desenvolvidos nesta pesquisa, utilizaram-se várias métricas de desempenho, incluindo recall (também conhecido como sensibilidade), precisão, acurácia e pontuação F1. Como este estudo envolve um problema de classificação binária, calculam-se essas métricas com base nos parâmetros: Verdadeiros Positivos (TP), Verdadeiros Negativos (TN), Falsos Positivos (FP) e Falsos Negativos (FN). Definiu-se a classe positiva como objetos não exoplanetas; enquanto a classe negativa representa exoplanetas confirmados. TP refere-se ao número de exemplos corretamente classificados da classe positiva, TN representa o número de exemplos corretamente classificados da classe negativa, FP indica o número de exemplos classificados incorretamente como positivos, apesar de pertencerem à classe negativa verdadeira, e FN representa o número de exemplos classificados incorretamente como negativos, apesar de pertencerem à classe positiva verdadeira. É importante observar que o número total de exemplos é dado pela equação n = TP + TN + FP + FN. Na tabela 3, descrevem-se métricas usadas neste estudo (Faceli et al., 2011).

Tabela 3 – métricas de desempenho

Métrica	**Fórmula**
Acurácia (acc)	$acc = \frac{(TP + TN)}{n}$
Precisão (prec)	$prec = \frac{TP}{TP + FP}$
Recall (rec)	$rec = \frac{TP}{TP + FN}$
Medida F1 (f1)	$f1 = \frac{2 * prec * rec}{prec + rec}$

Fonte: autoral 2024.

Calcula-se a acurácia somando os exemplos corretamente classificados para cada classe e dividindo-os pelo número total de instâncias. A precisão refere-se à proporção de objetos corretamente rotulados como uma classe em relação à soma de todos os objetos rotulados como pertencentes a essa classe. Em outras palavras, representa a capacidade do modelo de rotular, corretamente, uma classe quando o objeto realmente pertence a essa classe. O *recall* corresponde à proporção de objetos corretamente rotulados como uma classe em relação à soma

de todos os objetos que realmente pertencem a essa classe. Em outras palavras, mede a capacidade do modelo de incluir todos os objetos que realmente pertencem a uma classe. A medida F1 é definida como a média harmônica entre precisão e recall, em que ambas as medidas têm uma contribuição relativa igual.

Organizou-se a avaliação experimental em duas partes. Para cada representação proposta por Shallue e Vanderburg (2017), nomeadamente local e global, avaliaram-se todos os algoritmos de classificação utilizados neste trabalho. Além disso, para o processo de *Nested Cross-Validation*, repetiu-se este 10 vezes, totalizando 50 execuções para cada algoritmo e cada representação. Devido ao elevado custo computacional exigido pelos experimentos, utilizou-se o serviço de computação em nuvem do Google para realizar as execuções. A máquina utilizada possuía a seguinte configuração: 16 núcleos de processamento Intel(R) Xeon(R) CPU @ 2.20GHz e 64 GB de RAM. O código-fonte do método experimental adotado neste trabalho está disponível publicamente[10].

Análise dos dados e Resultados

Estudos na literatura apresentam diferentes métodos de aprendizado de máquina para a detecção de exoplanetas por meio de curvas de luz. Essas propostas incluem desde a análise de séries temporais à aplicação de técnicas de *deep learning*. Montanger and Zalewski, et al. (2020) avaliaram algoritmos tradicionais de aprendizado de máquina como SVM, DT, RF, NB, KNN e MLP, e algoritmos de classificação baseados em séries temporais como Boss, Rocket, Weasel, Muse e Time Series Forest (Löning et al., 2019). Shallue and Vanderburg et al. (2018) avaliaram algoritmos de *deep learning* com base em um modelo CNN denominado Astronet. No estudo de Ansdell et al. (2018), desenvolveu-se uma extensão do Astronet, intitulada Exonet, que adicionou séries temporais centroides e parâmetros estelares com intenção de suprimir falsas detecções. Ansdell também propôs uma variante menor do Exonet, chamada de Exonet-XS que apresentou uma precisão de detecção de 96,6%. No entanto, apesar desses avanços, ainda há uma lacuna de conhecimento na literatura sobre o desempenho de diversos algoritmos tradicionais para esta tarefa. Nesse contexto, realizou-se uma revisão experimental abrangente envolvendo 16 algoritmos de aprendizado de máquina considerados tradicionais,

[10] Código disponível em: https://brunohdmacedo.engineer/project.html

Revista brasileira de iniciação científica
ISSN: 2359-232X

com foco na otimização do ajuste de parâmetros. Buscou-se estabelecer um referencial para pesquisa de detecção de exoplanetas, utilizando unicamente curvas de luz.

Conforme delineado no desenho experimental deste estudo, avaliou-se cada algoritmo de aprendizado dividindo o conjunto de dados em cinco *folds*. Repetiu-se esse procedimento dez vezes, resultando em 50 execuções para cada algoritmo. Nas Tabelas 3 e 4, fornece-se um resumo dos resultados experimentais para cada um dos algoritmos, considerando as representações local e global, respectivamente. Essas tabelas detalham as métricas adotadas neste estudo: acurácia, precisão, recall e F1. Expressam-se os valores em termos da média e desvio padrão das 50 execuções para cada métrica.

Tabela 4 – resultados (%) para a representação local (DP – desvio padrão)

Modelo	Acurácia no Teste		Acurácia no Treino		Precisão		Recall		F1	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
GPC	74,52	1,21	74,08	0,45	74,26	1,13	72,52	1,96	76,92	1,27
KNN	74,48	1,21	74,43	0,43	73,84	1,19	76,32	2,58	77,79	1,24
RF	74,15	1,14	73,95	0,36	73,51	1,10	75,20	2,05	77,44	1,18
SVM	74,15	1,00	73,50	0,41	73,63	1,24	74,15	2,04	77,00	1,29
ET	74,00	1,10	73,90	0,40	73,50	1,10	74,60	1,80	77,10	1,15
CATB	73,91	1,32	73,69	0,43	73,38	1,31	74,78	2,07	77,00	1,31
LGBM	73,44	1,04	73,23	0,46	72,69	1,01	75,12	2,03	76,85	1,10
XGB	73,40	1,12	73,30	0,47	72,62	1,11	75,36	2,06	76,97	1,16
ADA	71,30	1,17	71,24	0,46	70,59	1,12	73,58	2,29	75,02	1,28
MLP	71,30	1,46	71,10	0,50	70,70	1,50	76,50	32,00	75,70	1,30
QDA	70,09	1,18	70,37	0,29	69,26	1,24	74,07	3,12	74,67	1,17
LDA	69,90	1,33	69,70	0,36	70,29	1,29	64,41	2,18	71,51	1,47
LR	69,00	1,30	69,00	0,50	68,00	1,30	74,00	1,80	74,00	1,20
NB	68,49	1,11	68,49	0,29	69,70	1,05	61,19	1,67	69,48	1,27
DT	67,50	1,65	67,80	0,40	67,03	1,60	69,00	3,90	71,30	1,90
RIDGE	65,99	1,28	66,05	0,43	64,81	1,39	74,72	2,00	71,88	1,09

Fonte: autoral 2024.

Com base nos resultados apresentados, para a representação local, obteve-se o melhor desempenho em termos de acurácia pelo algoritmo *Gaussian Process Classifier* (GPC), com

Revista brasileira de iniciação científica
ISSN: 2359-232X

uma média de 74,52% ± 1,21%; enquanto a pior acurácia foi do algoritmo *Ridge*, com uma média de 65,99% ± 1,28%.

Em termos de representação global, o algoritmo LightGBM (LGBM) demonstrou o melhor desempenho em relação à acurácia, obtendo uma média de 82,92% com um desvio padrão de ± 1,03%. Por outro lado, o algoritmo *Quadratic Discriminant Analysis* (*QDA*) apresentou os resultados menos precisos, com uma acurácia média de 58,58% e um desvio padrão de ± 0,02%.

Tabela 5 – resultados (%) para a representação global (DP – desvio padrão)

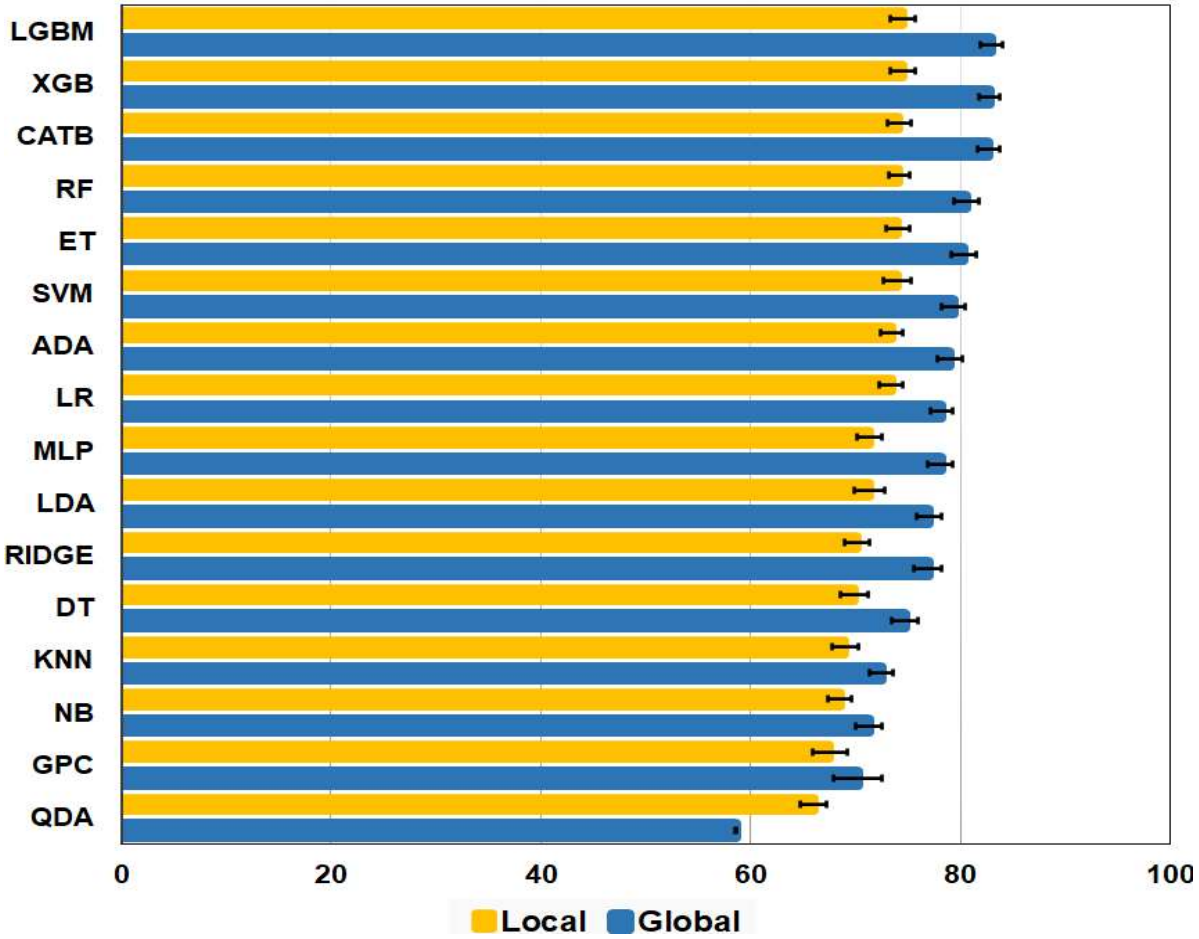
Modelo	Acurácia no Teste		Acurácia no Treino		Precisão		Recall		F1	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
LGBM	82,92	1,03	82,54	0,33	82,40	1,06	85,28	1,56	85,40	0,93
XGB	82,74	1,01	82,55	0,36	82,22	1,04	84,94	1,60	85,24	0,91
CATB	82,66	1,10	82,34	0,35	82,13	1,12	84,44	1,67	85,09	1,01
RF	80,57	1,17	80,24	0,30	80,14	1,20	84,89	1,44	83,70	0,98
ET	80,28	1,15	79,97	0,28	79,73	1,18	83,70	1,23	83,27	0,94
SVM	79,30	1,12	79,23	0,46	78,68	1,14	81,82	1,68	82,34	1,06
ADA	78,96	1,19	78,59	0,41	78,32	1,23	81,98	1,56	81,99	1,03
LR	78,17	1,08	78,27	0,35	77,55	1,09	79,65	1,70	81,00	1,02
MLP	78,08	1,22	78,57	0,54	77,80	1,17	79,00	5,37	80,72	1,44
LDA	76,99	1,18	77,10	0,33	76,35	1,16	77,29	1,91	79,85	1,16
RIDGE	76,89	1,34	76,75	0,32	76,23	1,39	80,84	1,88	80,39	1,21
DT	74,72	1,28	74,71	0,44	74,21	1,31	73,51	3,38	77,59	1,39
KNN	72,45	1,16	70,49	1,18	72,74	1,08	67,42	3,33	74,32	1,59
NB	71,27	1,27	71,59	0,37	71,72	1,30	66,56	1,48	73,11	1,22
GPC	70,24	2,33	69,37	1,68	74,11	1,52	56,15	4,72	68,84	3,37
QDA	58,58	0,02	58,60	0,00	79,29	0,01	100,00	0,00	73,88	0,02

Fonte: autoral 2024.

Na Figura 3, ilustram-se os resultados de acurácia média para cada algoritmo avaliado, considerando-se as duas representações utilizadas. Ao comparar os resultados de teste nas representações local e global, conforme a figura 3, observa-se que os valores de acurácia são maiores para a representação global. Especificamente, o melhor resultado usando a representação local, o algoritmo GPC, ocupa o décimo terceiro lugar em comparação aos

resultados obtidos com a representação global. Uma possível razão para o baixo desempenho dos algoritmos de aprendizado, usando a representação local, pode estar relacionada à etapa de pré-processamento. Isso se deve ao fato de que a média dos valores substituídos na interpolação para os dados globais foi de 1,12%, enquanto, para os dados locais, a média foi de 31,35%. O algoritmo QDA foi o único que apresentou desempenho inferior para a representação global. Além disso, o QDA foi o algoritmo que apresentou os piores resultados, em termos de acurácia, para ambas as representações analisadas.

Figura 4 – resultados de acurácia média para as representações local e global



Fonte: autoral 2024.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Para validar os resultados, utilizou-se o teste t corrigido[11] de Nadeau e Bengio (Nadeau; Bengio, 2003), definindo o nível de significância em 0,05. Elencou-se esse teste específico devido à natureza não independente das amostras produzidas no processo de validação cruzada. A hipótese nula nessa validação afirma que a acurácia dos algoritmos de classificação é equivalente. Fornece-se, na tabela 5, um subconjunto dos resultados das comparações do teste estatístico. Devido à grande quantidade de comparações realizadas, decidimos destacar aquelas relacionadas aos algoritmos com maior acurácia para cada tipo de representação. Para a representação global, o algoritmo LightGBM não apresentou diferença estatística significativa em comparação com os algoritmos XBoost e CatBoost. Para a representação local, não houve diferença estatística significativa entre o algoritmo *Gaussian Process* e outros sete algoritmos avaliados, conforme detalhado na Tabela 5.

Tabela 6 – comparações não significativas pelo teste t corrigido de Nadeau e Bengio

	Modelo	Model	t	p-valor
Global	LGBM	XGB	0,37	1,00
	LGBM	CATB	0,67	1,00
Local	GPC	KNN	0,07	1,00
	GPC	RF	0,93	1,00
	GPC	SVM	1,34	1,00
	GPC	ET	1,17	1,00
	GPC	CATB	1,09	1,00
	GPC	LGBM	2,78	0,45
	GPC	XGB	2,18	1,00

Fonte: autoral 2024.

Comparando os melhores resultados obtidos neste estudo com os resultados apresentados em Montanger and Zalewski, et al. (2020), é possível verificar que o algoritmo GPC obteve um resultado de 74,52% $\pm$ 1,21% sendo superior ao *Multilayer Perceptron* (MLP) com 72,7 $\pm$ 1,6% considerando a representação local, e o algoritmo LGBM obteve um resultado de 82,9% $\pm$ 1,92%, sendo também superior ao *Support Vector Machines* (SVM) com 80,8% $\pm$ 1,3 considerando a representação global. Em relação aos algoritmos de classificação baseados em séries temporais, o *Time Series Forest* (TSF) apresentou acurácia de

[11] Disponível em: https://scikit-learn.org/stable/auto_examples/model_selection/plot_grid_search_stats.html#comparing-two-models-frequentist-approach

78,1% ± 1,5% para a representação local e 86,1 ± 1,2% para a representação global. Comparando os resultados com os modelos de CNNs, Genesis obteve 96,4% e Exonet-XS obteve 96,0% para os dados locais e 95,3% e 94,1%, respectivamente para os dados globais.

Com base nos resultados encontrados, em relação aos algoritmos aprendizados tradicionais, resultados reportados neste estudo foram superiores em comparação com outros estudos da literatura. Em relação aos modelos de *deep learning* e baseados em séries temporais, os resultados foram inferiores em termos de acurácia. Entretanto, o objetivo deste trabalho implicou a realização de um estudo exploratório aprofundado sobre o desempenho de algoritmos tradicionais de aprendizado de máquina para a construção de um baseline que possa ser utilizado como uma referência para a literatura da área.

Em relação ao conjunto de dados utilizado neste trabalho, utilizaram-se dados da missão Kepler. Fundamentou-se essa decisão no entendimento de que a missão Kepler, uma das mais longas, forneceu uma quantidade significativa de registros disponíveis ao público por meio de várias interfaces.

Quanto às técnicas de pré-processamento utilizadas nos dados, elas se mostraram indispensáveis para corrigir falhas nos conjuntos de dados e prepará-los para uso em algoritmos de aprendizado. Fundamentou-se a fusão de todos os dados trimestrais fornecidos pela missão Kepler no princípio de que os eventos de trânsito podem ser reduzidos em relação ao uso de curvas individuais por trimestre. Assim, quando se utilizam dados que combinam todos os trimestres, o evento de interesse pode se repetir com mais frequência, facilitando a identificação de padrões pelos algoritmos de aprendizado. Na pesquisa apresentada neste trabalho, adotou-se o método de representação de dobramento de época, conforme sugerido por Shallue e Vanderburg (2017), que oferece vantagens como a redução da dimensionalidade e tem sido aplicado em outros estudos acadêmicos.

Outra contribuição significativa refere-se à introdução de um novo método para a avaliação dos resultados. A literatura existente sobre detecção de exoplanetas com base em aprendizado de máquina é composta por estudos que utilizam uma variedade de métodos de avaliação experimental. Uma prática comum é dividir o conjunto de dados em segmentos de treinamento e teste em uma única vez. No entanto, essa divisão, mesmo que aleatória, pode não representar não representar o conjunto de dados e potencialmente distorcer a interpretação dos resultados. Uma solução para mitigar esse viés inerente é a aplicação da técnica de validação

cruzada *k-fold* (James et al., 2013). O método de validação cruzada *k-fold* divide o conjunto de dados em k subconjuntos distintos e não sobrepostos. Cada um desses subconjuntos é alternadamente usado como conjunto de teste, enquanto se combinam os subconjuntos restantes para formar um conjunto de treinamento. Esse procedimento resulta no ajuste e avaliação de k modelos em k conjuntos de teste separados. Assim, calcula e relata-se o desempenho médio em relação a esses testes. No entanto, outra prática comumente adotada na literatura constitui o ajuste dos parâmetros do algoritmo de aprendizado com o objetivo de obter os melhores resultados no conjunto de dados. Nesse sentido, surge outro problema, ou seja, ao aplicar a validação cruzada *k-fold* para a otimização dos hiperparâmetros, o mesmo conjunto de dados seria utilizado tanto para ajustar os hiperparâmetros quanto para avaliar o desempenho do modelo. Isso poderia resultar em uma previsão excessivamente favorável do desempenho do modelo devido ao vazamento de informações. Assim, para minimizar esse problema, neste trabalho propôs-se o uso da técnica de validação cruzada *k-fold* aninhada. A CV aninhada trata esse problema por meio de um processo de validação cruzada em duas etapas (Cawley; Talbot, 2010). Dedica-se o nível inicial ao ajuste dos hiperparâmetros, enquanto o nível subsequente avalia o desempenho do modelo com base nos hiperparâmetros ajustados de forma ótima da primeira etapa. Não se utilizam os dados de teste do segundo nível durante a fase de ajuste dos hiperparâmetros, evitando, assim, o vazamento de informações. Isso garante uma avaliação mais imparcial da capacidade do modelo de lidar com dados previamente não vistos.

Conforme mencionado, o método de validação cruzada *k-fold* permite resultados menos enviesados do que uma divisão direta entre treinamento e teste. No entanto, há um problema adicional significativo associado à implementação do processo de validação cruzada *k-fold*: o viés também pode estar presente na divisão do conjunto de dados em *k folds*. Isso implica que cada execução do procedimento pode resultar em uma divisão diferente do conjunto de dados em *k folds*, alterando, assim, a distribuição dos escores de desempenho e levando a uma estimativa média variada do desempenho do modelo. Para mitigar esse possível problema, defende-se o uso do procedimento de validação cruzada *k-fold* repetida neste trabalho. Esse método melhora o desempenho estimado de um modelo de aprendizado de máquina, simplesmente por meio da repetição do processo de validação cruzada várias vezes e do relato do resultado médio sobre todos os *folds* de todas as repetições. Espera-se esse resultado médio para fornecer uma aproximação mais precisa do verdadeiro desempenho médio subjacente do

modelo no conjunto de dados, conforme inferido por meio do erro padrão (Kuhn; Johnson, 2013).

Considerações Finais

Nos últimos anos, propuseram-se diversas pesquisas para a automatização do processamento de grandes conjuntos de dados de curvas de luz por meio de estratégias baseadas em aprendizado de máquina. Esses estudos têm produzido resultados promissores, especialmente em relação à acurácia das previsões e à velocidade da análise dos dados. No entanto, a maioria das pesquisas existentes utilizam distintos métodos de avaliação, o que dificulta a comparação direta de seus resultados com os de outras investigações. Muitos utilizam uma única avaliação *holdout* (treino/teste), o que pode afetar a confiabilidade dos resultados devido ao possível viés na divisão do conjunto de dados. Além desse aspecto, embora tenham ocorrido avanços significativos e a adoção de métodos recentes como o *deep learning*, persistem algumas lacunas que demandam investigação mais aprofundada. Os algoritmos tradicionais de aprendizado de máquina ainda não foram exaustivamente analisados para essa aplicação, especialmente no que diz respeito ao uso de curvas de luz.

Neste estudo, com o objetivo de estabelecer uma referência (*baseline*) para a comunidade científica, realizou-se um amplo estudo experimental, examinando o desempenho de 16 algoritmos tradicionais de aprendizado de máquina. Além desse aspecto, propôs-se um método de avaliação experimental para avaliar os melhores parâmetros dos algoritmos e reduzir o viés de reamostragem dos dados.

Alcançaram-se a construção e avaliação dos modelos, resultando na definição de um *baseline* que poderá ser utilizado como referência em estudos futuros e para a literatura. A otimização de parâmetros realizada para a construção dos modelos possibilitou a exploração das melhores capacidades de acordo com cada algoritmo de aprendizado estudado.

Dentre as principais contribuições deste estudo, destaca-se que a utilização do algoritmo LightGBM em conjunto com a representação global de curvas de luz pode ser uma abordagem poderosa para a detecção de exoplanetas com um resultado de acurácia de, aproximadamente, 82,92%. Considerando a representação local, o principal resultado foi do algoritmo GPC com 74,52% em termos de acurácia.

Trabalhos futuros incluem a aplicação do método proposto neste estudo para avaliar algoritmos de aprendizado baseados em técnicas de séries temporais e aprendizado profundo.

Agradecimentos

À PRPPG/UNILA e à Fundação Araucária/PR pelo seu apoio por meio da bolsa ITI. Além disso, à PRPPG/UNILA pela promoção de recursos por meio das chamadas 104/2020 e 105/2020.

Referências

ARMSTRONG, D. J.; GAMPER, J.; DAMOULAS, T. Exoplanet validation with machine learning: 50 new validated Kepler planets. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 504, n. 4, p. 5327-5344, ago. 2020. Disponível em: <https://doi.org/10.1093/mnras/staa2498>. Acesso em 28 out. de 2023.

ARMSTRONG, D. J.; POLLACCO, D.; SANTERNE, A. Transit shapes and self-organizing maps as a tool for ranking planetary candidates: application to Kepler and K2. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 465, n. 3, p. 2634-2642, nov. 2016. Disponível em: <https://doi.org/10.1093/mnras/stw2881>. Acesso em 28 out. de 2023.

BABU, G. J.; MAHABAL, A. Skysurveys, Light Curves and Statistical Challenges. **International Statistical Review**, New Jersey, v. 84, n. 3, p. 506-527, 2016. Disponível em: <https://doi.org/10.1111/insr.12118>. Acesso em 28 out. de 2023.

BLOMME, J. **Variable star data mining techniques for time-resolved databases.** Leuven, 2012**.** Tese (Doutorado em Ciências) - Katholieke Universiteit Leuven, 2012. Disponível em: <https://fys.kuleuven.be/ster/pub/thesis-jonas-blomme/phd-jonas-blomme>. Acesso em 28 out. de 2023.

BREIMAN, L. Random Forests. **Machine Learning**, Berlin, v. 45, n. 1, p. 5-32, out. 2001. Disponível em: <https://doi.org/10.1023/A:1010933404324>. Acesso em 28 out. de 2023.

CASTRILLÓN, J. P. B. **Corot light curves analysis using different comparative processes: estimating stellar rotation periods**. Dissertação (Mestrado em Física). Universidade Federal do Rio Grande do Norte, Natal, 2010. Disponível em: <https://repositorio.ufrn.br/handle/123456789/16564>. Acesso em 28 out. de 2023.

CAWLEY, G. C.; TALBOT, N. L. C. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. **Journal of Machine Learning Research**, Brookline, v. 11, n. 70, p. 2079-2107, 2010. Disponível em: <http://jmlr.org/papers/v11/cawley10a.html>. Acesso em 28 out. de 2023.

CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. *In*: 22ND ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2016, San Francisco, California, USA. **Proceedings** […] New York, NY, USA: Association for Computing Machinery, 2016. p. 785-794. Disponível em: <https://doi.org/10.1145/2939672.2939785>. Acesso em 28 out. de 2023.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, Berlin, v. 20, n. 3, p. 273-297, set. 1995. Disponível em: <https://doi.org/10.1007/BF00994018>. Acesso em 28 out. de 2023.

COUGHLIN, Jeffrey L. *et al.* Planetary candidates observed by kepler. vii. the first fully uniform catalog based on the entire 48-month data set (q1–q17 dr24). **The Astrophysical Journal Supplement Series**, Bristol, v. 224, n. 1, p. 12, maio 2016. Disponível em: <https://dx.doi.org/10.3847/0067-0049/224/1/12>.Acesso em 28 out. de 2023.

COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE Transactions on Information Theory**, New York, v. 13, n. 1, p. 21-27, 1967. Disponível em: <https://doi.org/10.1109/TIT.1967.1053964>. Acesso em 28 out. de 2023.

CUÉLLAR, S.; GRANADOS, P.; FABREGAS, E.; CURÉ, M.; VARGAS, H.; DORMIDO-CANTO, S.; FARÍAS, G. Deep Learning Exoplanets Detection by Combining Real and Synthetic Data. **Preprints.org**, Basel, 2021. *Preprint*. Disponível em: <https://doi.org/10.20944/preprints202112.0070.v1>. Acesso em 28 out. de 2023.

EXOPLANETS NASA. **Discovery Fast Facts**. 2020. Disponível em: <https://exoplanets.nasa.gov/discovery/missions/#otp_fast_facts>. Acesso em 28 out. de 2023.

FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina.** 2. ed. Rio de Janeiro: Editora LTC, 2021.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge discovery and data mining: towards a unifying framework. *In*: SECOND INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 1996, Portland, Oregon. **Proceedings** […] Portland, Oregon: AAAI Press, 1996. p. 82-88. Disponível em: <https://dl.acm.org/doi/10.5555/3001460.3001477>. Acesso em 28 out. de 2023.

FREUND, Y.; SCHAPIRE, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. **Journal of Computer and System Sciences**, Amsterdam, v. 55, n. 1, p. 119-139, 1997. Disponível em: <https://doi.org/10.1006/jcss.1997.1504>.Acesso em 28 out. de 2023.

GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. **Machine Learning**, Berlin, v. 63, n. 1, p. 3-42, abr. 2006. Disponível em: <https://doi.org/10.1007/s10994-006-6226-1>. Acesso em 28 out. de 2023.

HAYKIN, S. **Redes Neurais: Princípios e Prática.** 2. ed. Porto Alegre: Bookman, 2001.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning:** Data Mining, Inference, and Prediction, Second Edition. 2. ed. New York, NY: Springer, 2009. Disponível em: <https://doi.org/10.1007/978-0-387-84858-7>. Acesso em 28 out. de 2023.

HINNERS, T. A.; TAT, K.; THORP, R. Machine Learning Techniques for Stellar Light Curve Classification. **The Astronomical Journal**, Bristol, v. 156, n. 1, p. 7, jun. 2018. Disponível em: <https://dx.doi.org/10.3847/1538-3881/aac16d>. Acesso em 28 out. de 2023.

HOERL, A. E.; KENNARD, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. **Technometrics**, Boston, v. 12, n. 1, p. 55-67, 1970. Disponível em: <https://doi.org/10.1080/00401706.1970.10488634>. Acesso em 28 out. de 2023.

IMAGINE NASA. **Timing Analysis.** 2013. Disponível em: <https://imagine.gsfc.nasa.gov/science/toolbox/timing2.html>. Acesso em: 28 out. 2023.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning:** with Applications in R. 1. ed. New York, NY: Springer, 2013. Disponível em: <https://doi.org/10.1007/978-1-4614-7138-7>. Acesso em 28 mar. de 2023.

JARA-MALDONADO, M. *et al.* Transiting Exoplanet Discovery Using Machine Learning Techniques: A Survey. **Earth Science Informatics**, Berlin, v. 13, n. 3, p. 573-600, 2020. Disponível em: <https://doi.org/10.1007/s12145-020-00464-7>. Acesso em 28 mar. de 2023.

KE, G. *et al.* LightGBM: a highly efficient gradient boosting decision tree. *In*: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 31., 2017, Long Beach, California, USA. **Proceedings** [...] Red Hook, NY, USA: Curran Associates Inc., 2017. p. 3149-3157. Disponível em: <https://doi.org/10.5555/3294996.3295074>. Acesso em 28 mar. de 2023.

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. 1. ed. New York, NY: Springer, 2013. XIII, 600 p. ISBN: 978-1-4614-6849-3. Disponível em: <https://doi.org/10.1007/978-1-4614-6849-3>. Acesso em 28 mar. de 2023.

CARDOSO, J. V. d. M.; HEDGES, C.; GULLY-SANTIAGO, M.; SAUNDERS, N.; CODY, A. M.; BARCLAY, T.; HALL, O.; SAGEAR, S.; TURTELBOOM, E.; ZHANG, J.; TZANIDAKIS, A.; MIGHELL, K.; COUGHLIN, J.; BELL, K.; BERTA-THOMPSON, Z.; WILLIAMS, P.; DOTSON, J.; BARENTSEN, G. **Lightkurve: Kepler and TESS time series analysis in Python.** Astrophysics Source Code Library, 2018. Disponível em: <http://adsabs.harvard.edu/abs/2018ascl.soft12013L>. Acesso em: Acesso em 28 mar. de 2023.

MALIK, A.; MOSTER, B. P.; OBERMEIER, C. Exoplanet detection using machine learning. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 513, n. 4, p. 5505-5516, 2021. Disponível em: <https://doi.org/10.1093/mnras/stab3692>. Acesso em: 4 mar. 2024.

MCCAULIFF, S. D.; JENKINS, J. M.; CATANZARITE, J.; BURKE, C. J.; COUGHLIN, J. L.; TWICKEN, J. D.; TENENBAUM, P.; SEADER, S.; LI, J.; COTE, M. Automatic Classification of Kepler Planetary Transit Candidates. **The Astrophysical Journal**, Bristol, v. 806, n. 1, p. 6, 2015. Disponível em: <https://dx.doi.org/10.1088/0004-637X/806/1/6>. Acesso em: Acesso em 28 mar. de 2023.

MITCHELL, T. M. **Machine Learning**. Boston: McGraw-Hill, 1997.

MONTANGER, P. O.; ZALEWSKI, W. Classificação automática de objetos astronômicos por meio da análise de séries temporais. **Revista Brasileira de Iniciação Científica**, Itapetininga, v.6, n.4, p.42-55, 2019. Edição Especial Universidade Federal da Integração Latino-Americana (UNILA). Disponível em <https://periodicos.itp.ifsp.edu.br/index.php/IC/article/view/1538>.Acesso em: 04 mar. 2024.

MONTANGER, P. O.; ZALEWSKI, W. Programa computacional para a identificação automática de exoplanetas. **Revista Brasileira de Iniciação Científica**, Itapetininga, 2020. Disponível em: <https://periodicos.itp.ifsp.edu.br/index.php/IC/article/view/1736>. Acesso em: 08 jan. 2021.

NADEAU, C.; BENGIO, Y. Inference for the Generalization Error. **Machine Learning**, Berlin, v. 52, n. 3, p. 239-281, 2003. Disponível em: <https://doi.org/10.1023/A:1024068626366>. Acesso em: Acesso em 28 mar. de 2023.

QUINLAN, J. R. Induction of decision trees. **Machine Learning**, Berlin, v. 1, n. 1, p. 81-106, 1986. Disponível em: <https://doi.org/10.1007/BF00116251>. Acesso em: 10 jul. 2023.

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, Montréal, Canada. **Proceedings** [...] Red Hook, NY, USA: Curran Associates Inc., 2018. p. 6639-6649. Disponível em: <https://doi.org/10.5555/3327757.3327770>. Acesso em: 15 out. 2021.

RASMUSSEN, C. E.; WILLIAMS, C. K. I. **Gaussian Processes for Machine Learning.** Cambridge: The MIT Press, 2005. ISBN: 9780262256834. Disponível em: <https://doi.org/10.7551/mitpress/3206.001.0001>. Acesso em: 15 mar. 2021.

REZENDE, S. O. **Sistemas Inteligentes:** Fundamentos e Aplicações. Barueri: Editora Manole, 2003.

RICHARDS, J. W.; STARR, D. L.; BUTLER, N. R.; BLOOM, J. S.; BREWER, J. M.; CRELLIN-QUICK, A.; HIGGINS, J.; KENNEDY, R.; RISCHARD, M. On machine-learned classification of variable stars with sparse and noisy time-series data. **The Astrophysical Journal**, Bristol, v. 733, n. 1, p. 10, 2011. Disponível em: <https://dx.doi.org/10.1088/0004-637X/733/1/10>. Acesso em: 15 mar. 2021.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial:** Uma Abordagem Moderna. 3. ed. Rio de Janeiro: Pearson, 2013.

SHALLUE, C. J.; VANDERBURG, A. Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. **The Astronomical Journal**, Bristol, v. 155, 2017. Disponível em: <https://api.semanticscholar.org/CorpusID:4535051>. Acesso em: 15 mar. 2021.

SHALLUE, C. J.; VANDERBURG, A. Identification of planetary transits using deep neural networks. **Astronomy Magazine**, Waukesha, v. 23, n. 4, p. 45-58, 2018. Disponível em: <https://lweb.cfa.harvard.edu/~avanderb/kepler90i.pdf>. Acesso em: 28 mar. 2021.

SILVA, D. F.; SOUZA, V. M. A.; BATISTA, G. E. A. P. A. Time Series Classification Using Compression Distance of Recurrence Plots. *In*: IEEE 13TH INTERNATIONAL CONFERENCE ON DATA MINING, 2013, Dallas, TX, USA. **Proceedings** [...] Dallas, TX, USA: IEEE, 2013. p. 687-696. Disponível em: <https://api.semanticscholar.org/CorpusID:6008338>. Acesso em: 28 mar. 2022.

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. *In*: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 25, 2012, Lake Tahoe, NV, United States. **Proceedings** [...] Lake Tahoe, NV: [*s.n.*], 2012. v. 4. p. 2951-2959. Disponível em: <https://papers.nips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf>. Acesso em: 28 mar. 2022.

SOUZA, A. A. de; VALIO, A. Estudo da atividade estelar da Kepler-289 a partir da modelagem de trânsitos planetários. **Revista Brasileira de Ensino de Física**, São Paulo, v. 41, n. 4, 2019. Disponível em: <https://doi.org/10.1590/1806-9126-RBEF-2018-0323>. Acesso em: 28 mar. 2022.

VISSER, K.; BOSMA, B.; POSTMA, E. A one-armed CNN for exoplanet detection from light curves. **ArXiv**, New York, 2021. *Preprint*. Disponível em: <https://arxiv.org/abs/2105.06292>. Acesso em: 20 mar. de 2022.

VISSER, K.; BOSMA, B.; POSTMA, E. Size does matter: Exoplanet detection with a sparse convolutional neural network. **Astronomy and Computing**, Amsterdam, v. 41, p. 100654, 2022. ISSN 2213-1337. Disponível em: <https://doi.org/10.1016/j.ascom.2022.100654>. Acesso em: 20 mar. de 2022.

VISSER, K.; BOSMA, B.; POSTMA, E**.** Exoplanet detection with Genesis. **Journal of Astronomical Instrumentation**, Singapore, v. 11, n. 03, p. 2250011, 2022. Disponível em: <https://doi.org/10.1142/S2251171722500118>. Acesso em: 20 mar. de 2022.

ZALEWSKI, W. **Modelagem Simbólica de Padrões Morfológicos para a Classificação de Séries Temporais**. Tese (Doutorado em Ciência da Computação). Universidade Federal do Paraná, Curitiba, 2015. Disponível em: <http://hdl.handle.net/1884/41324>. Acesso em: 28 mar. 2020.