Deep learning for pulsar detection: Investigating hyperparameter effects on TensorFlow classification accuracy

Khushi Upadhyay[1], **Josephine Wong**[2]
[1] Blue Lakes International School, Bugesera, Eastern Province, Rwanda
[2] Department of Physics, Stanford University, Stanford, California

SUMMARY

Pulsars play a critical role in astrophysics, serving as natural laboratories for studying extreme states of matter, testing general relativity, and detecting gravitational waves. With the growing volume of pulsar survey data, Convolutional Neural Networks (CNNs) are a promising approach for automating the classification of pulsar candidates. However, CNN performance is influenced by hyperparameters, such as the number of epochs, which refers to one complete iteration through the training data, and batch size. Our investigation aimed to evaluate the influence of these hyperparameters on classification accuracy using prepfold plots. Our hypotheses are as follows: first, that training and testing accuracy would increase with more epochs, and second, that training and testing accuracy would increase with smaller batch sizes. To test our hypotheses, we randomly assigned 140 samples from the Pulsar Science Collaboratory (PSC) database to training (100), validation (20), and testing (20) datasets, and then ran pulsar classification trials using different epoch values and batch sizes. The findings of this study partially support the first hypothesis; classification training and testing accuracy improved as epochs increased, but only until a threshold number of 20 epochs, after which both training and testing accuracy declined. We also found that training and testing accuracy improved with decreasing batch size, thus supporting the second hypothesis. Additionally, the model consistently achieved accuracies exceeding 90% with relatively few epochs. These results confirm the significant role of hyperparameters in determining model accuracy and offer compelling evidence for the viability of automated pulsar candidate classification for real-world applications. Hence, our work provides a basis for advancing high-accuracy pulsar classification models, with the potential of expediting the pulsar discovery process and accelerating astrophysical research.

INTRODUCTION

A pulsar is an ultra-dense, revolving neutron star that emanates a powerful magnetic field from its magnetic poles (1). It is formed when a star at least eight times the mass of the sun collapses and explodes in a supernova (2). Pulsars were first discovered by Jocelyn Bell Burnell and Anthony Hewish in 1967 (3). Today, 3000 pulsars are known, yet studies estimate the number of pulsars in our galaxy as ~$10^{4\text{-}5}$, and extrapolation to the universe suggests over a trillion pulsars in total (4, 5). Evidently, our current database of known pulsars is relatively small, and many aspects of pulsars, such as neutron star interiors, still remain a mystery (6). Therefore, broadening our known pulsar database is essential to improve our understanding of pulsar population characteristics, neutron star physics, and their broader role in astrophysical phenomena (7).

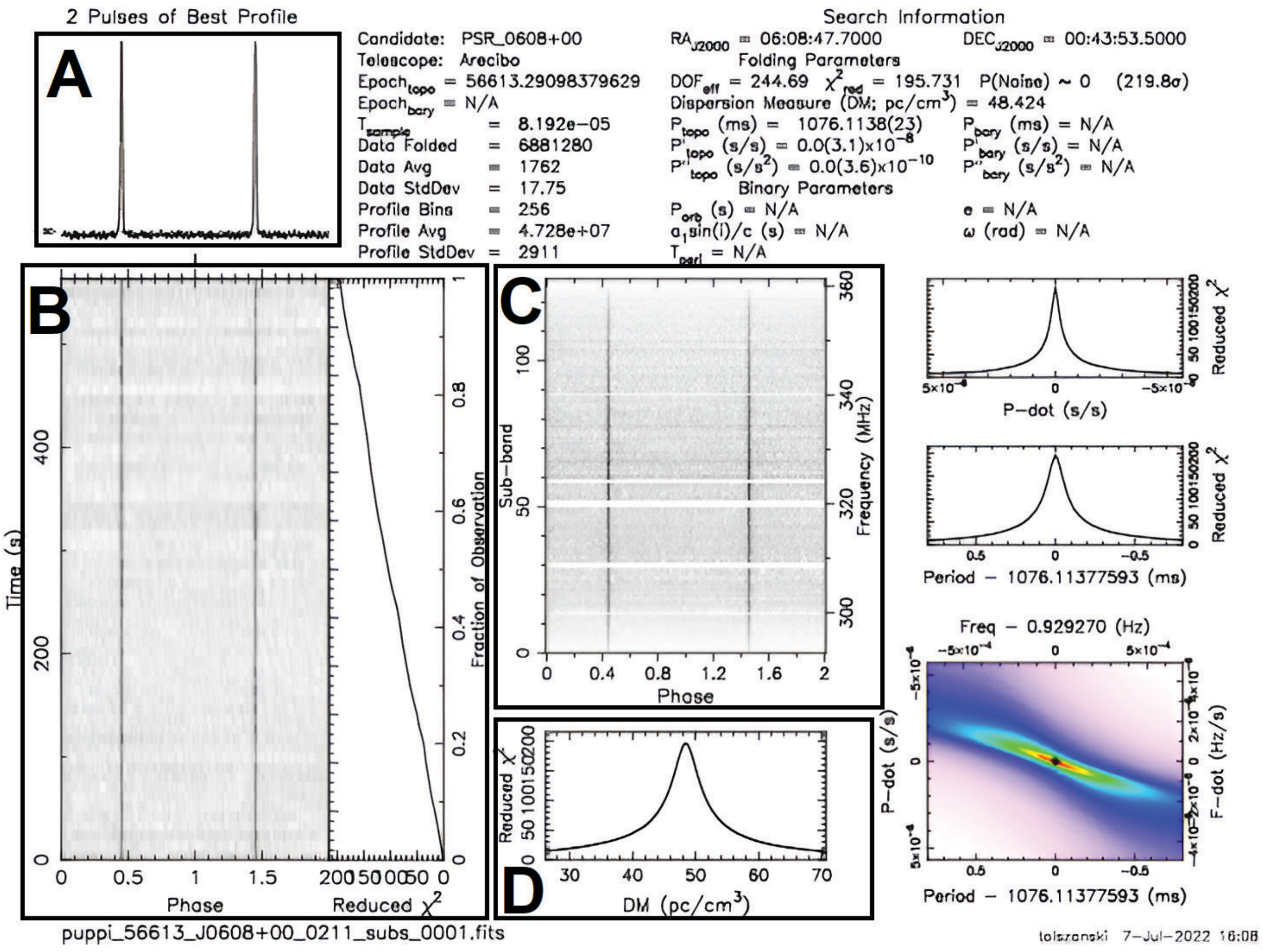
Pulsars are used in a range of astronomical research, including the study of cosmic rays and stellar evolution, and the detection of gravitational waves and dark matter (8, 9). Pulsars' exceptionally precise "blinking" can be used by scientists to detect nearby space events and for accurate calculation of cosmic distances by monitoring changes in their regularity (9). Recently, cosmic background radiation has been detected using pulsar timing arrays by the North American Nanohertz Observatory for Gravitational Waves (NANOGrav) (10). A further example is NASA's Neutron Star Interior Composition Explorer (NICER), which uses X-ray observations of 6 precise millisecond pulsars (MSPs) to explore the nature of matter at these extreme conditions (11).

Because of the misalignment between the magnetic and rotational axes of a pulsar, as the pulsar spins, radiation beams sweep through space like a lighthouse that can be observed by sensitive telescopes when they cross Earth (12). Pulsars can therefore be detected on Earth as a result of their lighthouse effect (12). Radio telescopes are primarily used for pulsar surveys and detection because most pulsars emit in the radio region of the electromagnetic spectrum, making them more reliable than X-ray or gamma-ray telescopes (13). Raw radio data can then be compiled into prepfold plots (PFPs) by the open-source software package PRESTO (14–17). By aligning and stacking numerous pulse periods to improve the periodic signal relative to noise, PRESTO facilitates visualization of pulsar periodicity through these PFP charts (14–17). This procedure increases the accuracy of pulsar detection by making it easier to identify and validate pulsar signals (14–17).

PFPs contain four major diagnostic subplots which represent patterns and features that enable pulsar signals to be distinguished from noise and radio frequency interference (RFI) (**Figure 1**) (18). PFPs have been utilized in the PSC database, which is a citizen science project for students and teachers that gives access to radio astronomy data from the Arecibo Observatory and Green Bank Telescope after requisite training (19). These data are then analyzed to search for new pulsars.

However, the classification of pulsars remains a largely manual, human-centered process that is time-intensive and laborious, which impedes the discovery of new pulsars (20). Furthermore, there are tremendous volumes of radio data already collected by telescopes, such as the 3000 terabytes

https://doi.org/10.59720/25-132



```
Candidate:  PSR_0608+00              RA_J2000 = 06:08:47.7000          DEC_J2000 = 00:43:53.5000
Telescope:  Arecibo                            Folding Parameters
Epoch_topo  = 56613.29098379629      DOF_eff = 244.69  χ²_red = 195.731  P(Noise) ~ 0  (219.8σ)
Epoch_bary  = N/A                    Dispersion Measure (DM; pc/cm³) = 48.424
T_sample         =  8.192e−05        P_topo (ms) =  1076.1138(23)      P_bary (ms) = N/A
Data Folded      =  6881280          P'_topo (s/s) = 0.0(3.1)x10⁻⁸      P'_bary (s/s) = N/A
Data Avg         =  1762             P''_topo (s/s²) = 0.0(3.6)x10⁻¹⁰   P''_bary (s/s²) = N/A
Data StdDev      =  17.75                      Binary Parameters
Profile Bins     =  256              P_orb (s) = N/A                    e = N/A
Profile Avg      =  4.728e+07        a₁sin(i)/c (s) = N/A               ω (rad) = N/A
Profile StdDev   =  2911             T_peri = N/A
```

Figure 1: Pulsar sample from the Pulsar Science Collaboratory (PSC) dataset. Shown is a Prepfold Plot (PFP) of a pulsar signal comprising the major diagnostic plots: (A) Integrated Pulse Profile, (B) Time Domain plot, (C) Sub-band plot, and (D) Dispersion measure (DM) plot. Strong peaks in the integrated pulse profile, a single set of dark parallel streaks across the time domain plot, a broadband signal in the sub-band plot, and a DM greater than zero are all indicative of a pulsar. Graphic compiled by software package PRESTO and acquired from the PSC database (19).

of radio data solely from the Arecibo telescope, with volumes of data predicted to increase exponentially in the coming years (21). For instance, Yue and Li estimate that pulsar search data volume is expected to increase a 100-fold over a decade at the Five-hundred-meter Aperture Spherical radio Telescope (FAST) alone (22). More automated classification techniques are crucial to keep pace with these expansions.

An attractive field for the development of automated classification algorithms is machine learning. Studies such as Zhao *et al.*, use Multimodal Large Language Models (MLLMs) like StarWhisper-Pulsar to classify pulsar candidates with labeled data in visual, textual, and numerical modalities (23). Convolutional neural networks (CNNs) are an especially promising option for pulsar candidate classification, as they offer a robust method for distinguishing pulsars from non-pulsar signals. Filters (kernels) applied to small regions of the input PFP graphic from convolutional layers can scan the image and then construct feature maps that capture local patterns and features such as edges or peaks in pulsar plots (24). A noteworthy study by Zeng *et al.* used a CNN to deduce patterns characteristic of pulsars from the diagnostic subplots with a classification accuracy of approximately 98.9%, missing only 4 real pulsars out of 326 candidates (25).

In machine learning, a model's learning process, and hence its learning outcomes, are governed by a configuration setting known as a hyperparameter. Unlike parameters, which are the model's internal variables learned and updated from the data during training, hyperparameter values are set before the training begins (26). Hyperparameters influence both the training and testing accuracy of machine learning models by affecting how they learn from the input data (26). Training accuracy gives us insights into how well the model can recognize and learn patterns from the training dataset (27). Testing accuracy, by contrast, gives a measure of how often a model correctly predicts the outcome on the testing dataset and evaluates the model's generalizing abilities and performance in categorizing unseen data (27). Both metrics are essential for a comprehensive evaluation of a machine learning model (27).

Optimal model hyperparameters help efficient convergence, in addition to reducing overfitting, where the model memorizes training data but fails on new data, and underfitting, where the model is too simple to capture meaningful patterns (28). Proper hyperparameter tuning ensures the model achieves both accurate learning and good generalization to unseen data. In a recent 2024 study, Wojciuk *et al.* showed a 6% improvement in CNN classification accuracy by optimizing hyperparameters (29). Therefore, understanding how individual hyperparameters affect model performance and accuracy is crucial to selecting the optimal hyperparameters.

An important hyperparameter is the number of epochs. Each full iteration of the training dataset is called an epoch, and the number of epochs greatly influences both training and testing accuracy (30). Multiple studies, such as Ajayi and Ashi (2023), observe that training and testing accuracies generally improve with increasing epoch number because the model has increased opportunities to update and optimize parameters and weights while minimizing the loss function, which quantitatively measures the difference between the model's predictions and the actual target values (31). Another critical hyperparameter to consider is batch size. The batch size indicates how many samples are analyzed in a single iteration prior to model parameters being updated, and it significantly influences the model's learning process and generalizing capabilities (30). Larger batch sizes generally lead to sharper minima in the loss landscape, meaning poorer recognition of patterns during training, resulting in lower test accuracy (32).

Our study aimed to investigate the influence of epoch and batch size on CNN's learning efficacy and generalizing capabilities for pulsar classification by measuring model training and testing accuracies. We hypothesized that training and testing accuracies would both increase with increasing epoch number and decrease with increasing batch size. To assess training and testing accuracies, we split our dataset of 140 PFP samples from the PSC database, assigning 100 samples to training, 20 to validation, and 20 to testing. All three datasets were then used as inputs to a TensorFlow CNN model. We then carried out multiple trials using varying epochs and batch sizes while recording corresponding training and testing accuracies. We found that training and testing accuracies indeed improved with increasing epochs; however, only until a critical epoch threshold of 20 epochs, after which accuracy declined. Notably, the model consistently achieved accuracies higher than 90% with a relatively small number of epochs. Furthermore, we found that training and testing accuracies decreased with larger batch sizes. These results further extend our knowledge of how hyperparameters, like epochs and batch size, affect overall model accuracy and offer powerful evidence of models being able to reach high classification accuracies with optimal hyperparameter selection. Taken together, these data demonstrate the viability of automated pulsar candidate classification for real-world applications, which could streamline the pulsar discovery process and accelerate astrophysical research.

RESULTS

Our goal was to measure the model's training and testing accuracy while systematically varying epochs and batch size, in order to examine how these hyperparameters affect the CNN's learning effectiveness and, thereby, its classification performance in pulsar identification. The study used a dataset of 140 samples, split into training, validation, and testing sets.

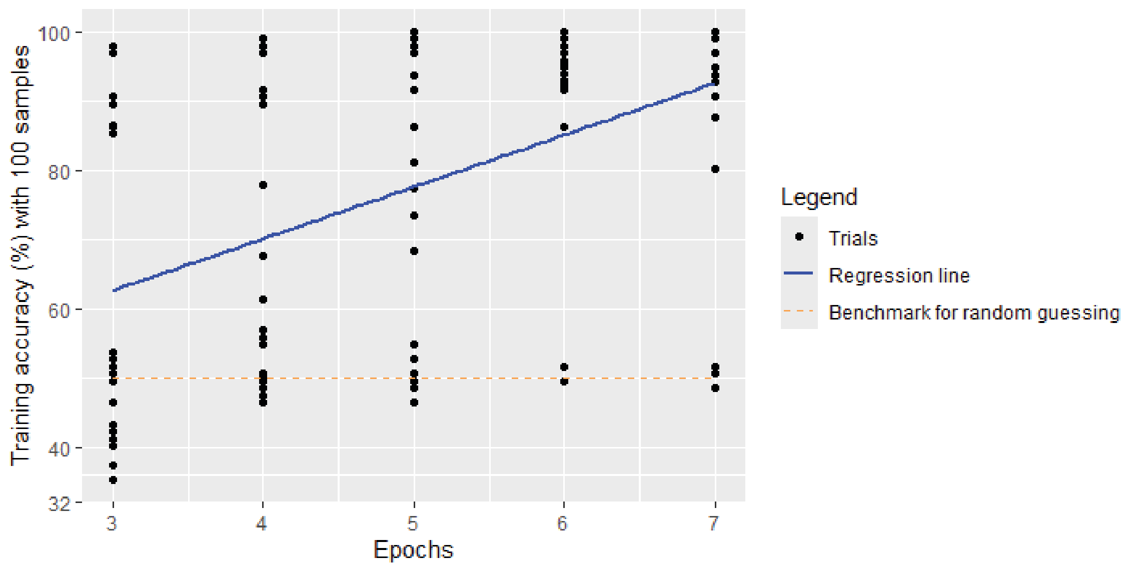


Figure 2: Positive correlation between model training accuracy and epochs. Using 100 training samples over a progressively increasing number of epochs (30 trials per number of epochs), we measured the model's training accuracy. Our model is a convolutional neural network built with TensorFlow Keras, comprising two convolutional layers that extract features from the PFP, each followed by max pooling to reduce dimensionality of the feature maps. These are then flattened before passing through two dense layers for classification. The model was trained using the Adam optimizer with binary cross-entropy loss. Three to seven training epochs were used. Each black dot represents a trial, with some hidden as their accuracy scores coincide with those of other trials. The dotted orange line represents random guessing, and the blue line represents the least-squares line of best fit. Linear regression showed a significant relationship (*t*-test, $p < 0.001$) with an R^2 value of 0.37.

For each epoch and batch size setting, we conducted 30 trials, progressively increasing these values and recording the corresponding training and testing accuracies.

Investigating the Number of Epochs Against Model Training and Testing Accuracy

Model accuracy is substantially influenced by the number of epochs trained for. Studies such as those by Ajayi and Ashi demonstrate that model accuracy rises with increasing epochs (31). A linear regression analysis of our experimental results showed a significant positive correlation between model training accuracy and number of epochs, with an R^2 value of 0.37, indicating 37% of the variance in training accuracy was explained by the number of epochs (*t*-test, $R^2 = 0.37$, $p < 0.001$) (**Figure 2**). The model's testing accuracy also improved with increasing epochs (*t*-test, $R^2 = 0.28$, $p < 0.001$) (**Figure 3**). Interestingly, at initial epochs, the CNN persistently reached low accuracies, irrespective of the initial randomly generated neural network weights (**Figures 2 and 3**). However, after roughly 6 epochs, the training and testing accuracies were observed to consistently achieve high (>90% values), regardless of the initialization of the weights (**Figures 2 and 3**).

We also ran trials with 10, 20, 50, and 100 training epochs to evaluate if there was a limit to model improvement. Training accuracies varied across different epoch values, with mean (SD) values of 85.3% (SD = 23.5), 90.0% (SD = 21.1), 81.2% (SD = 24.3), and 71.9% (SD = 24.3) at 10, 20, 50, and 100 epochs respectively. Testing accuracies followed a similar trend, with mean (SD) values of 82.0% (SD = 29.7), 89.3% (SD = 23.3), 81.3% (SD = 24.9), and 66.7% (SD = 29.3) across the same epochs. Both accuracies peaked around 20 epochs before declining; notably, training accuracy decreased from 90.0% (margin of error of 15.1%) to 71.9% (margin of error of 17.4%), and testing accuracy declined from 89.3% (margin of error of 16.7%) to 66.7% (margin of error of 21.0%) between 20 and 100 epochs. All errors are reported with a 95% confidence interval.

Investigating Batch Size Against Model Training and Testing Accuracy

Significant relationships have been identified between batch size and model performance. In a study by Thakur, smaller batch sizes improved generalization and testing accuracies, whereby a batch size of 16 performed the best, and a batch size of 256 performed the worst (33). Similarly, measuring the training and testing accuracy at increasing batch sizes allows us to determine if our model may become more accurate in pulsar signal classification.

We identified a negative correlation between batch size and both training and testing model accuracy for pulsar classification (**Figures 4 and 5**). As the batch size increased, training accuracy decreased (*t*-test, $R^2 = 0.29$, $p < 0.001$) (**Figure 4)**. Testing accuracy also decreased as batch size increased (t-test, $R^2 = 0.22$, $p < 0.001$) (**Figure 5**). Interestingly, by batch sizes of 75 and 100, the model almost consistently achieved a testing accuracy of 50% (**Figure 6**). A 96.7% (29 times out of 30) recurrence with a 2.05% margin of error (95% confidence interval) was observed for the batch size of 75. A recurrence of 96.7% (29 times out of 30) with a 2.39% margin of error (95% confidence interval) was seen for the batch size of 100, suggesting that the model failed to capture meaningful patterns from the data, and was therefore performing no better than random guessing.

DISCUSSION

We identified a positive correlation between the number of epochs and model training and testing accuracy, which held until a critical threshold of approximately 20 epochs. Beyond this point, both training and testing accuracies declined.

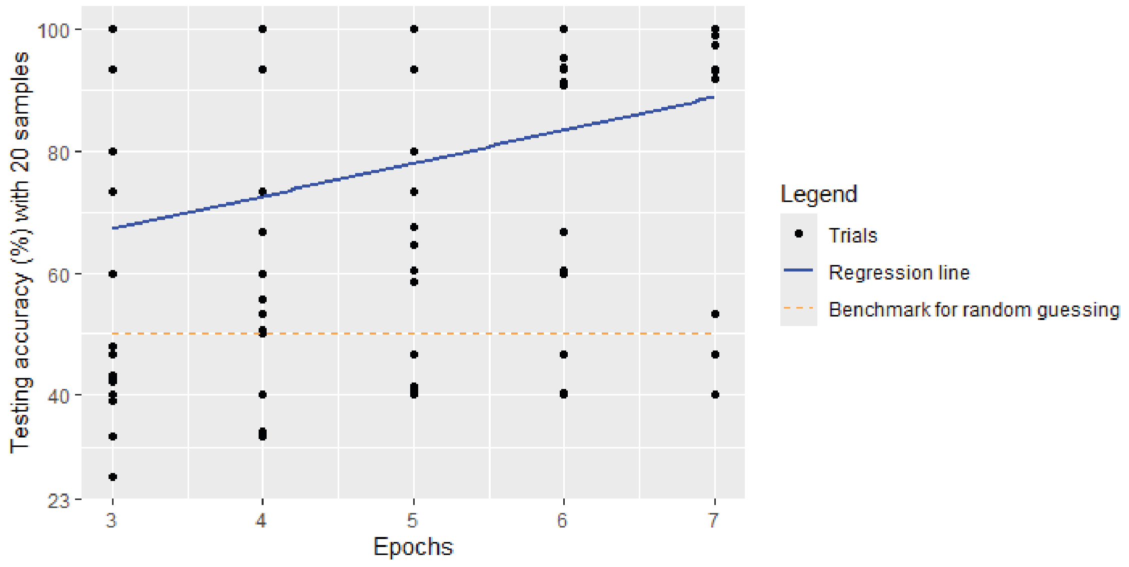


Figure 3: Positive correlation between model testing accuracy and epochs. Evaluated on 20 test samples over a progressively increasing number of epochs (30 trials per number of epochs), we measured the model's testing accuracy. Three to seven training epochs were used. Each black dot represents a trial, with some hidden as their accuracy scores coincide with those of other trials. The dotted orange line represents random guessing, and the blue line represents the least-squares line of best fit. Linear regression showed a significant relationship (*t*-test, $p < 0.001$) with an R^2 value of 0.28.

https://doi.org/10.59720/25-132

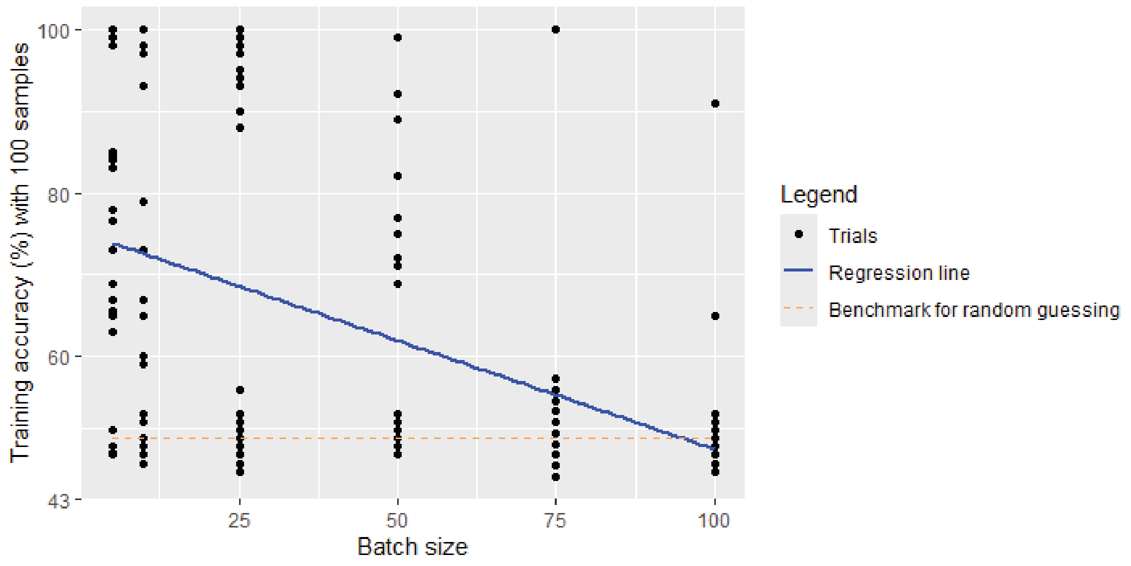


Figure 4: Negative correlation between model training accuracy and batch size. Using 100 training samples with 5 epochs over a progressively increasing batch size (30 trials per batch size), we measured the model's training accuracy. We performed trials at batch sizes of 5, 10, 25, 50, 75, and 100. Each black dot represents a trial, with some hidden as their accuracy scores coincide with those of other trials. The dotted orange line represents random guessing, and the blue line represents the least-squares line of best fit. Linear regression showed a significant relationship (*t*-test, $p < 0.001$) with an R^2 value of 0.29.

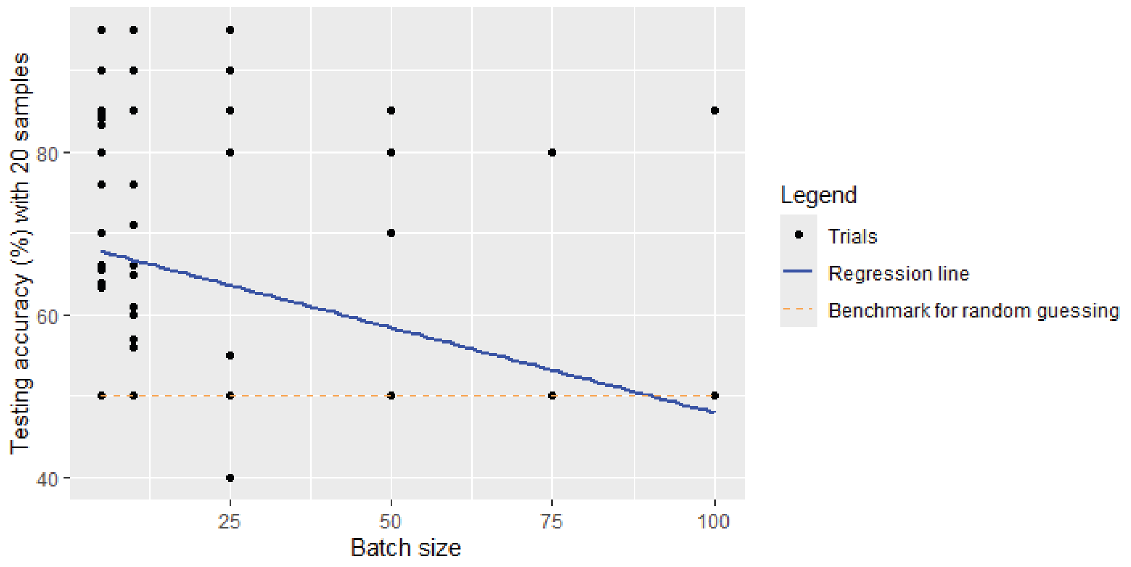


Figure 5: Negative correlation between model testing accuracy and batch size. Evaluated on 20 test samples with 5 epochs over a progressively increasing batch size (30 trials per batch size), we measured the model's testing accuracy. We performed trials at batch sizes of 5, 10, 25, 50, 75, and 100. Each black dot represents a trial, with some hidden as their accuracy scores coincide with those of other trials. The dotted orange line represents random guessing, and the blue line represents the least-squares line of best fit. Linear regression showed a significant relationship (*t*-test, $p < 0.001$) with an R^2 value of 0.22.

This result, however, should be interpreted with caution, as the observed decline is based on empirical data, and further investigation with intermediate epoch values is necessary to precisely locate the peak. We also observed a significant negative correlation between batch size and training and testing accuracies. Intriguingly, the model's accuracy consistently hovered around 50% for batch sizes of 75 and 100.

The most striking observation to emerge from the data comparison during the investigation of epoch number was that training and testing accuracies decreased after a certain number, implying a critical threshold value of epochs. We expect this is due to the occurrence of overfitting, whereby after an excessive number of epochs, models may start to memorize the training sets rather than learning generalizable patterns (34). This results in poorer performance as hidden

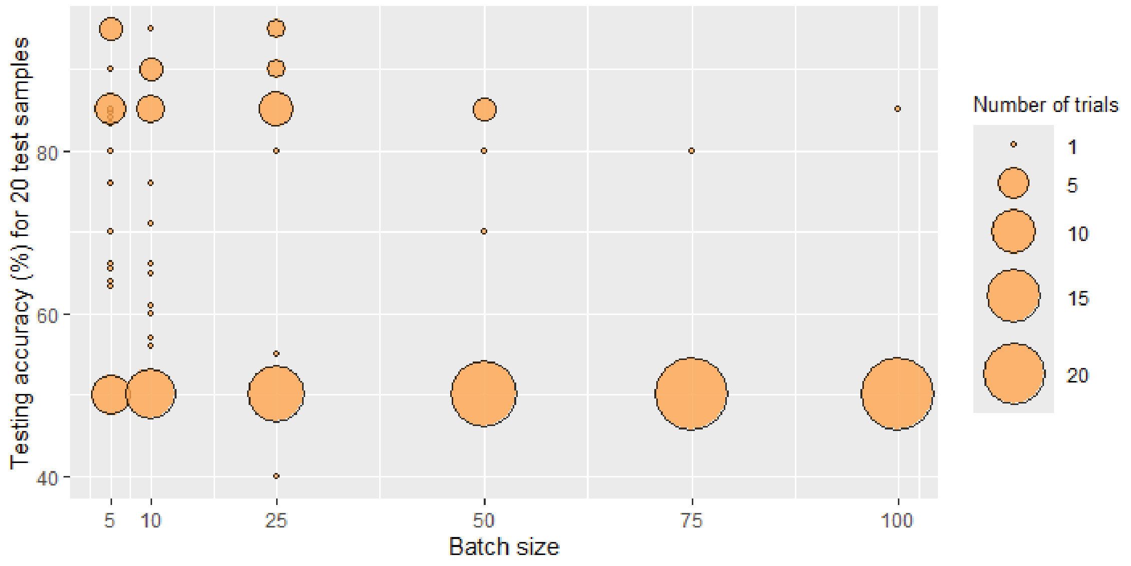


Figure 6: Consistent hovering of testing accuracy at 50%, specifically at batch sizes of 75 and 100. Bubble size represents the number of trials that achieved the corresponding testing accuracy at each batch size. Evaluated on 20 test samples with 5 epochs over a progressively increasing batch size (30 trials per batch size), we measured the model's testing accuracy. We performed trials at batch sizes of 5, 10, 25, 50, 75, and 100. For batch size 75, 29 out of 30 trials (96.7%) showed a testing accuracy of 50%, with a margin of error of 2.05% (95% confidence interval). Likewise, at batch size 100, 29 out of 30 trials (96.7%) consistently yielded a testing accuracy of 50%, with a margin of error of 2.39% (95% confidence interval).

layers in the model increasingly focus on noisy or irrelevant patterns rather than meaningful features. This also applies to training accuracy, where the model becomes excessively specialized to individual training images, thereby hindering its ability to generalize to other training images. This phenomenon is seen in other studies, such as Moshe et al., where test accuracy in PreResNet experiments initially rose with epochs, but plateaued after 50 iterations (35). Moshe et al. observed that the model reached peak validation accuracy at 80 epochs, beyond which explanation-quality metrics, like sensitivity, deteriorated despite stable loss values (35).

We should, however, raise a note of caution with regard to our findings at higher epoch values. Given that our findings are based on a limited number of trials, due to their especially lengthy run times, the results from our analysis had relatively high margins of error and should thus be treated with caution. In addition to the limited trials, we expect these high margin errors likely arise as a result of the inherent randomness involved in any machine learning model (36). Therefore, further data collection would be necessary to confirm how excessive epochs affect model accuracy in the pulsar candidate classification problem.

We also demonstrated that even with relatively few epochs, training and testing accuracies remained considerably high, and almost consistently reached accuracies exceeding 90% by 6 and 7 epochs. Achieving these results with few epochs demonstrates the model's computational efficiency, as it reduces training time as well as computational cost, thus highlighting the viability of CNNs for automated pulsar classification in real-world applications. However, it is necessary to recognize that the exact optimal number of epochs may vary with dataset size, and larger datasets may require correspondingly higher epochs to learn all the patterns.

The correlation between batch size and training and testing accuracy is also noteworthy. Importantly, we believe the negative correlation arises because smaller batch sizes can introduce beneficial noise that aids in avoiding local minima, while larger batch sizes may lead to less optimal convergence (37). This is because larger batch sizes typically result in convergence to sharp minima, which have poor generalization. Conversely, smaller batch sizes utilize stochastic noise that helps the model escape these sharp regions and find robust flat minima, resulting in better overall accuracy (30). This has been observed in previous studies, such as Keskar *et al.*, where ResNet, VGG, and LSTM architectures were trained on CIFAR-10, CIFAR-100, and Penn Treebank datasets (32). Comparing batch sizes ranging from 256 to 8192, they showed that large-batch training was prone to converging to a sharp minima, hence leading to worse generalization and poorer model performance (32).

Additionally, a notable occurrence was observed for batch sizes of 75 and 100: the model's testing accuracy plateaued around 50%, implying the model was performing no better than a random guesser in classification. This may arise as a result of an inadequacy in training, that is to say, the model not learning meaningful patterns from the data (38). This is likely due to the abovementioned reasons, whereby with large batch sizes, there are infrequent updates and therefore convergence to sharp minima.

Our work has several limitations. First, the current study was limited by the amount of data available to us. There is a marked deficiency of open-access pulsar data available, and while many deep learning models are trained and evaluated on hundreds of thousands of samples, we were restricted to the 298 PFP samples we found on the PSC database. Consequently, this limits our capacity to identify trends with vast datasets. Second, the pulsar PFPs in our dataset exhibited extremely strong signals. While this resulted in the clear detection of features homogeneous to pulsar signals, these findings may not be transferable to datasets with weak or atypical pulsar plots. Moreover, signals from unique pulsar types, such as binary pulsars, magnetars, and MSPs, with more eccentric features represented in their PFPs, were not constituents of the dataset. For this reason, our investigation with respect to pulsar classification accuracy may not capture the full range of patterns and variability; thus, our findings are not representative of broader datasets that may contain these.

Our investigations so far have only been on a small scale, given the restricted dataset size. However, for subsequent studies, we aim to expand our dataset and broaden our data pool to incorporate pulsar plots of weaker pulsar signals. This will allow us to more extensively evaluate the classification capabilities against hyperparameters, such as epochs and batch size. We also seek to diversify our dataset to include a variety of unique pulsar types to better verify that our findings are comprehensive and generalizable to an extensive dataset. This will allow us to ensure the practical application of this research.

Furthermore, since the radio data collected by telescopes is rarely evenly distributed, we plan to analyze how training the model with disproportionate distributions of pulsar and non-pulsar classes influences performance. Prior work by Lee et al. investigating pulsar prediction from the high time-resolution (HTRU2) dataset, has included dataset imbalance as a major consideration in their study, with a dataset containing only 1639 pulsar samples out of 17898 total samples and leading to a 1:10 ratio between pulsars and non-pulsars (39). By investigating optimal hyperparameters for imbalanced datasets, we can develop more robust algorithms that effectively tackle the inherent class imbalance in pulsar data, thus enabling their useful implementation in real-world pulsar classification tasks where imbalanced datasets are prevalent (40).

Additionally, while the correlations between hyperparameters and model accuracy were statistically significant ($p < 0.001$), the relatively low R^2 values (0.22 to 0.37) show that these hyperparameters explain only a modest proportion of the variance. This is often seen in complex machine learning problems involving biological and astronomical data, as many factors such as model architecture and optimization settings influence accuracy (41). Therefore, while the hyperparameters epochs and batch size have a clear effect on classification accuracy, further experimental investigations of additional model functions are necessary to fully understand and improve classification accuracy.

Moreover, we are interested in conducting an in-depth exploration of how model training and testing accuracies vary within the range of values between 20 to 50 epochs to identify the epoch value at which model accuracy ceases to improve and begins to decline. This would provide us with an approximately optimal number of epochs for classifying pulsars, given the current dataset. These insights can aid us in making more informed hyperparameter choices for other pulsar classification models and datasets, with the future potential for scalability to larger datasets. We also aim to conduct further investigations involving multi-class classification of pulsars, noise, and RFI rather than binary classification to evaluate which type would result in higher model classification accuracy with the respective hyperparameters. We conjecture that binary classification would perform best due to the increased complexities presented by multi-class classification with learning multiple decision boundaries (42). This is in line with previous results from studies, such as Murthy et al., who reported achieving up to 98.9% accuracy for binary classification in contrast to 94.6% for multiclass classification (43).

In summary, the CNN image classification model using PFP inputs improved its training and testing accuracy with more training epochs and a decreasing batch size. The correlation between epochs and training and testing accuracy is interesting; we found that these accuracies decline after a certain threshold number of epochs. Further analysis also revealed that testing accuracy plateaued at 50% for larger batch sizes, implying random guessing. These results offer compelling evidence for the substantial role played by hyperparameters, such as epochs and batch size, on overall model accuracy. This stresses the importance of informed and optimal hyperparameter choices to ensure the highest pulsar classification accuracy. Furthermore, we found that the simple TensorFlow CNN model used was able to achieve high training and testing accuracies at relatively small epochs, consistently reaching accuracies greater than 90% by 7 epochs. The CNN's ability to decipher complex patterns in PFPs through convolutional layers, as well as the data pipeline feature utilized to avoid computational memory issues, ensures exceptional scalability with larger datasets. Taken together, these findings implicate a significant practical viability for real-world applications of pulsar candidate classification using CNNs.

Our study hence provides a blueprint for ideal hyperparameter selection and constitutes important initial findings that will help models efficiently achieve high classification accuracies, automating the pulsar candidate classification process and expediting the discovery of new pulsars to augment our existing database. These discoveries of new pulsars have great potential to extend astrophysical knowledge and research in areas such as cosmology, gravitational waves, ultra-dense states of matter, extreme particle acceleration, and stellar evolution.

MATERIALS AND METHODS

Data Sources and Acquisition

The PFPs used as inputs for the model were obtained from the PSC database (19). Prior to gaining access to the database, we were required to complete the PSC Online Training Workshop and its consecutive evaluation to acquire certification (44). The data used in our study were collected through the Arecibo 327 MHz (AO327) drift pulsar survey. Radio data was collected in this survey at a frequency of 327 MHz, featuring a frequency resolution of 24 kHz and a time resolution of 82 μs (45). Raw radio data from the AO327 survey were compiled into PFPs by the open-source PRESTO software package (14). Pulsar signal PFPs show strong peaks in the integrated pulse profile, broadband signal in the sub-band plot, parallel dark streaks through the time domain plot, and a DM greater than zero (**Figure 1**) (18). Conversely, RFI is characterized by a diminutively low, or customarily zero DM, indicating its origin from terrestrial sources, and signals often appear as horizontal lines or narrow spikes at specific frequencies without the dispersion curve (18). Due to the random nature of noise, there are no clear or discernible patterns observed in noise PFPs (18).

We utilized the Python libraries 'Selenium', 'pickle', 'PIL', 'csv', 'os', 'PyDrive', and 'NumPy' to pull the radio data in JSON format directly from the PSC site database (46–52). This was followed by recompiling the complete PFP graphic and saving it into the respective folders based on plot type corresponding to either pulsar, RFI, or noise. In the total of 298 samples, 70 were pulsar instances, 91 were RFI instances, and 137 were noise instances. The resulting plots were then further split into 2 sets, that is, "Pulsars" (n = 70) and "Not Pulsars" (n = 228). To account for the imbalance in the dataset, 35 RFI and 35 noise plots were ultimately used as the non-pulsar plots, totaling 70, which is the size of the pulsar class. We refined the dataset to create evenly distributed classifications, promoting equal representation and improving the reliability of the analysis of our model.

Data Preprocessing

All image preprocessing and investigations were conducted in Google Colab using Python (version 3.10). In contrast to preloading the entirety of the dataset into memory, the Keras utility "image_dataset_from_directory" generator was used to build a data pipeline, whereby the directory specified was located in a Google Drive (53–55). This method presented a neater and more practical solution, with it being an optimal way to integrate the data into Google Colab, as well as contributing to future scalability capacities (55). This utility also built the classes and labels comprehensible by the model, with 1 corresponding to "Pulsar" and 0 corresponding to "Not Pulsar" (55). In addition to this, it implemented preprocessing, including resizing all input images to 256 by 256 pixels, with a batch size configured according to the specific investigation (55).

Following this, a numpy iterator was implemented to access the generator from the data pipeline as numpy arrays and to obtain the data required in consecutive batches using the .next() method. Lastly, the values in the resulting numpy array of shape (256, 256, 3) were normalized to between 0 and 1, which helps reduce computational complexity, making it less computationally intensive and leading to faster training times and reduced processing power requirements (56).

Experimental Setup

The total dataset was effectively split with 100 samples assigned to training, 20 assigned to validation, and 20 assigned to testing. To maintain equitable representation, each division contained equal aggregates of pulsar and non-pulsar plots.

Our study used a basic TensorFlow Keras framework consisting of two convolutional layers, two max pooling layers, one flatten layer, and two dense layers. To compile the model for training, the Adam optimizer was used, with loss evaluated using BinaryCrossentropy() and the tracked metric being accuracy (57). For testing, the y true value (true label) was compared to the the yhat value (predicted answer), and evaluated utilizing the precision, recall, and accuracy metrics. The complete code can be found at our GitHub repository (58).

We utilized R version 4.5.0 in the RStudio platform to examine the trial results (59, 60). We integrated ".csv" result files and computed statistics using the R packages "readr", "dplyr", and "tidyverse" (61–63). Additionally, we created the graphs of the findings using the R package "ggplot2" (64). A *t*-test and its *p*-values were utilized alongside linear regression analysis to measure the statistical significance of our findings. Our GitHub repository contains the statistical analysis code, located in the file "R Statistical Analysis and Graphing Script" (58).

Investigating the Number of Epochs Against Model Training and Testing Accuracy

The experiment aimed to investigate the influence of number of epochs on both training and testing accuracy. We thus varied the number of training epochs while keeping all other model hyperparameters the same. The batch size was also kept at a constant of 5. This choice was guided by the small scale of our dataset, as it allowed 24 batches per epoch, therefore allowing a sufficient number of updates per epoch and hence more opportunities to observe accuracy trends across epochs.

Our experimental setup bears a close resemblance to the one used by Nakra *et al.* (65). Thirty trials were conducted per epoch to uphold the principles of the Central Limit Theorem and establish a normal distribution, and corresponding training and testing accuracy values were recorded to a CSV file. Three epochs were initially used, and the same method was applied for additional epochs from four through seven. To establish control in our experiment, the Colab notebook kernel was killed after each trial, and the entire code was reinitialized. We then also experimented with 10, 25, 50, and 100 epochs, but because these trials had much longer run times, we only collected 10 trials for each epoch value.

Investigating Batch Size Against Model Training and Testing Accuracy

All other model hyperparameters were kept the same, with 5 epochs used. This was chosen to strike an optimal balance between computational time and the need for sufficient epochs to allow the observation of significant trends, with this number deemed appropriate owing to the dataset's small scale. Initially, a batch size of 5 was used, and we ran 30 trials and recorded training and test accuracies to a CSV file. This procedure was repeated for a batch size of 10, 25, 50, 75, and 100. To maintain control in our experiment, the Colab notebook kernel was killed after each trial, and the entire code was reinitialized.

Received: March 20, 2025
Accepted: August 19, 2025
Published: January 31, 2026

REFERENCES

1. Chiu, Hong-Yee. "A Review of Theories of Pulsars." *Publications of the Astronomical Society of the Pacific*, vol. 82, no. 486, Oct. 1970, pp. 487–493. https://doi.

org/10.1086/128932.

2. Woosley, Stan E. and Alexander Heger. "The Deaths of Very Massive Stars." *Very Massive Stars in the Local Universe*, edited by Jorick S. Vink, vol. 412, Astrophysics and Space Science Library, Springer, 2015, pp. 199–225.
3. Hewish, Antony, et al. "Observation of a Rapidly Pulsating Radio Source." *Pulsating Stars: A NATURE Reprint*, edited by Smith F. G., Hewish A., Springer US, Boston, MA, 1968, pp. 5–9.
4. Pardo-Araujo, Celsa, et al. "Radio pulsar population synthesis with consistent flux measurements using simulation-based inference." *Astronomy & Astrophysics*, vol. 696, 11 Apr. 2025, pp. A114. https://doi.org/10.1051/0004-6361/202453314.
5. Rajwade, Kaustubh, et al. "The Galactic halo pulsar population." *Monthly Notices of the Royal Astronomical Society*, vol. 479, no. 3, 21 Sept. 2018, pp. 3094–3100. https://doi.org/10.1093/mnras/sty1695.
6. Lattimer, James M. and Madappa Prakash. "Neutron Star Observations: Prognosis for Equation of State Constraints." *Physics Reports*, vol. 442, 1 Apr. 2007, pp. 109–165. https://doi.org/10.1016/j.physrep.2007.02.003.
7. Golomb, Jacob, et al. "Interplay of astrophysics and nuclear physics in determining the properties of neutron stars." *Physical Review D*, vol. 111, no. 2, 14 Jan. 2025, https://doi.org/10.1103/PhysRevD.111.023029.
8. Smith, David A., et al. "The Third Fermi Large Area Telescope Catalog of Gamma-Ray Pulsars." *The Astrophysical Journal*, vol. 958, no. 2 Nov. 2023, p. 191. https://doi.org/10.3847/1538-4357/acee67.
9. Manchester, Richard N. "Millisecond Pulsars, Their Evolution and Applications." *Journal of Astrophysics and Astronomy*, vol. 38, no. 42, 7 Sep. 2017. https://doi.org/10.1007/s12036-017-9469-2.
10. Agazie, Gabriella, et al. "The NANOGrav 15 yr Data Set: Evidence for a Gravitational-wave Background." *The Astrophysical Journal Letters*, vol. 951, no. 1, 29 June 2023, pp. L8. https://doi.org/10.3847/2041-8213/acdac6.
11. Guillot, Sebastie, et al. "NICER X-Ray Observations of Seven Nearby Rotation-powered Millisecond Pulsars." *The Astrophysical Journal Letters*, vol. 887, no. 1, 1 Jan. 2020, pp. 237.06. https://doi.org/10.3847/2041-8213/ab511b.
12. Gold, Thomas. "Rotating Neutron Stars as the Origin of the Pulsating Radio Sources." *Nature*, vol. 218, no. 5143, 25 May 1968, pp. 731–732. https://doi.org/10.1038/218731a0.
13. Stappers, Benjamin W., et al. "The Prospects of Pulsar Timing with New-Generation Radio Telescopes and the Square Kilometre Array." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2120, 16 Apr. 2018, https://doi.org/10.1098/rsta.2017.0293.
14. Ransom, Scott M. "New Search Techniques for Binary Pulsars." PhD dissertation, Harvard University, 2001. ProQuest Dissertations and Theses.
15. Ransom, Scott M., et al. "Fourier Techniques for Very Long Astrophysical Time-Series Analysis." *The Astronomical Journal*, vol. 124, 1 Sep. 2002, pp. 1788–1809. https://doi.org/10.1086/342285.
16. Andersen, Bridget C. and Scott M. Ransom. "A Fourier Domain 'Jerk' Search for Binary Pulsars." *The Astrophysical Journal*, vol. 863, 1 Aug. 2018, p. L13. https://doi.org/10.3847/2041-8213/aad59f.
17. Ransom, Scott M., et al. "A New Search Technique for Short Orbital Period Binary Pulsars." *The Astrophysical Journal*, vol. 589, 1 June 2003, pp. 911–920. https://doi.org/10.1086/374806.
18. "Searching for and Identifying Pulsars." *Pulsar Search Collaboratory*. https://pulsars.nanograv.org/app/site/media/PSC_search_guide.pdf. Accessed 6 Apr. 2025.
19. "PSC - Home." *Pulsar Science Collaboratory*. https://pulsars.nanograv.org/. Accessed 7 Mar. 2025.
20. Morello, Vincent, et al. "SPINN: A Straightforward Machine Learning Solution to the Pulsar Candidate Selection Problem." *Monthly Notices of the Royal Astronomical Society*, vol. 443, no. 2, 11 Sept. 2014, pp. 1651–1662. https://doi.org/10.1093/mnras/stu1188.
21. "Priceless Astronomy Data Saved After Collapse of Arecibo Telescope." *UT Austin News - The University of Texas at Austin*. https://news.utexas.edu/2021/05/10/priceless-astronomy-data-saved-after-collapse-of-arecibo-telescope/. Accessed 7 Mar. 2025.
22. Yue, Youling and Di Li. "Big Data Challenges of FAST." *arXiv*, vol. 11473, 13 Nov. 2019, pp. 6–9. https://doi.org/10.1007/978-3-030-28061-1_2.
23. "Pulsar Candidate Classification with Multimodal Large Language Models." *OpenReview*. https://openreview.net/forum?id=8SKgWpZiDL. Accessed 22 Nov. 2025.
24. LeCun, Yann, et al. "Gradient-based learning applied to document recognition" *Proceedings of the IEEE*, vol. 86, no. 11 Nov. 1998, pp. 2278–2324. https://doi.org/10.1109/5.726791.
25. Zeng, Qingguo, et al. "Concat Convolutional Neural Network for pulsar candidate selection." *Monthly Notices of the Royal Astronomical Society*, vol. 494, no. 3, 21 May 2020, pp. 3110–3119. https://doi.org/10.1093/mnras/staa916.
26. Arnold, Christian, et al. 'The role of hyperparameters in machine learning models and how to tune them.' *Political Science Research and Methods*, vol. 12, no. 4 Oct. 2024, pp. 841–848. https://doi.org/10.1017/psrm.2023.61.
27. Sivakumar, Muthuramalingam, et al. "Trade-off between Training and Testing Ratio in Machine Learning for Medical Image Processing." *PeerJ Computer Science*, vol. 10, 6 Sept. 2024, https://doi.org/10.7717/peerj-cs.2245.
28. Jönsson, Daniel, et al. "Visual Analysis of the Impact of Neural Network Hyper-Parameters." *International Workshop on Machine Learning in Visualisation for Big Data*, 6 July 2020. https://doi.org/10.2312/mlvis.20201101.
29. Wojciuk, Mikolaj, et al. "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization." *Heliyon*, vol. 10, no. 5, 23 Feb. 2024, pp. e26586. https://doi.org/10.1016/j.heliyon.2024.e26586.
30. LeCun, Yann, et al. "Deep learning." *Nature*, vol. 521, 27 May 2015, pp. 436–444. https://doi.org/10.1038/nature14539.
31. Ajayi, Oluibukun G. and John Ashi. "Effect of varying training epochs of a Faster Region-Based Convolutional

Neural Network on the Accuracy of an Automatic Weed Classification Scheme." Smart Agricultural Technology, vol. 3, 1 Feb. 2023, pp. 100128. https://doi.org/10.1016/j.atech.2022.100128.

32. Keskar, Nitish S., et al. "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima." *arXiv*, 9 Feb. 2017. https://doi.org/10.48550/arXiv.1609.04836.
33. Kandel, Ibrahem and Castelli Mauro. "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset." ICT Express, vol. 6, no. 4, 1 Dec. 2020, pp. 312–315. https://doi.org/10.1016/j.icte.2020.04.010.
34. Xu, Chuhan, et al. "Empirical Study of Overfitting in Deep Learning for Predicting Breast Cancer Metastasis." Cancers, vol. 15, no. 7, 25 Mar. 2023, https://doi.org/10.3390/cancers15071969.
35. Moshe, Ofir, et al. "Improving Interpretability via Regularization of Neural Activation Sensitivity." *arXiv*, 16 Nov. 2022. https://doi.org/10.48550/arXiv.2211.08686.
36. Raste, Soham, et al. "Quantifying Inherent Randomness in Machine Learning Algorithms." *SSRN Electronic Journal*, 27 June 2022. https://doi.org/10.2139/ssrn.4146989.
37. He, Fengxiang, et al. "Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence." *Advances in Neural Information Processing Systems*, vol. 32, 8 Dec. 2019, pp. 1143–1152. https://dl.acm.org/doi/10.5555/3454287.3454390.
38. Kim, Bommae and Timo von Oertzen. "Classifiers as a Model-Free Group Comparison Test." *Behavior Research Methods*, vol. 50, no. 1, 1 Feb. 2018, pp. 416–426. https://doi.org/10.3758/s13428-017-0880-z.
39. Lee, Ernesto, et al. "Predicting Pulsars from Imbalanced Dataset with Hybrid Resampling Approach." *Advances in Astronomy*, vol. 2021, 3 Dec. 2021, pp. 1–13. https://doi.org/10.1155/2021/4916494.
40. Holewik, Jakub, et al. "Imbalanced Ensemble Learning for Enhanced Pulsar Identification." *Advances in Swarm Intelligence*, vol. 12145, 2020, pp. 515–524. https://doi.org/10.1007/978-3-030-53956-6_47.
41. Martinez, Kaitlyn M., et al. "Evaluating the Factors Influencing Accuracy, Interpretability, and Reproducibility in the Use of Machine Learning Classifiers in Biology to Enable Standardization." *Scientific Reports*, vol. 15, no. 1234, 13 May 2025, https://doi.org/10.1038/s41598-025-00245-6.
42. Moral, Pablo D., et al. "Why Is Multiclass Classification Hard?" *IEEE Access*, vol. 10, 25 July 2022, pp. 80448–80462. https://doi.org/10.1109/ACCESS.2022.3192514.
43. Jha, Anupama, et al. "Comparison of Binary Class and Multi-Class Classifier Using Different Data Mining Classification Techniques." *SSRN*, 14 Oct. 2019. https://doi.org/10.2139/ssrn.3464211.
44. "PSC - Courses: PSC Online Training Workshop." *Pulsar Science Collaboratory*. https://pulsars.nanograv.org/courses/psc-online-training. Accessed 7 Mar. 2025.
45. Deneva, Julia S., et al. "The AO327 Drift Survey Catalog and Data Release of Pulsar Detections." *The Astrophysical Journal Supplement Series*, vol. 271, no. 23, 26 Feb. 2024, pp. 1–10. https://doi.org/10.3847/1538-4365/ad19da.
46. SeleniumHQ. *selenium*. Version 4.8.3, SeleniumHQ, 2023, https://pypi.org/project/selenium/.
47. Python Software Foundation. *pickle*. Version included with Python 3, 2023, https://docs.python.org/3/library/pickle.html.
48. Python Pillow Team. *Pillow*. Version 9.4.0, Python Pillow Team, 2023, https://pillow.readthedocs.io/en/stable/.
49. Python Software Foundation. *csv*. Included with Python 3, 2025, https://docs.python.org/3/library/csv.html.
50. Python Software Foundation. *os*. Included with Python 3, 2025, https://docs.python.org/3/library/os.html.
51. PyDrive developers. *PyDrive*. Version 1.3.1, 2023, https://pypi.org/project/PyDrive/.
52. NumPy Developers. *NumPy*. Version 1.25.0, 2023, https://numpy.org/doc/stable/
53. Abadi, Martín, et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." *arXiv*, 2015, https://doi.org/10.48550/arXiv.1603.04467.
54. Keras Developers. *Keras*. 2015, https://keras.io/.
55. "Keras documentation: Image data loading." *Keras*. https://keras.io/api/data_loading/image/. Accessed 7 Mar. 2025.
56. Singh, Dalwinder and Birmohan Singh. "Investigating the impact of data normalization on classification performance." *Applied Soft Computing*, vol. 97, 1 Dec. 2020, pp. 105524, https://doi.org/10.1016/j.asoc.2019.105524.
57. "Keras documentation: BinaryCrossentropy loss." *Keras*. https://keras.io/api/losses/probabilistic_losses/#binarycrossentropy-class. Accessed 22 Nov. 2025.
58. Upadhyay, Khushi. "JEI-Epochs-Batch-size-Investigation-Code." *GitHub*. https://github.com/khushiudhy/JEI-Epochs-Batch-size-Investigation-Code./tree/main. Accessed 24 Sept. 2025.
59. R Core Team. *R*. Version 4.5.0, R Foundation for Statistical Computing, 2023, https://www.r-project.org/.
60. Posit team. *RStudio: Integrated Development Environment for R*. Version 2024.12.1+563, Posit Software, PBC, 2025, http://www.posit.co/.
61. Wickham, Hadley, et al. *readr: Read Rectangular Text Data*. Version 2.1.5, RStudio, 2025, https://cran.r-project.org/package=readr.
62. Wickham, Hadley, et al. *dplyr: A Grammar of Data Manipulation*. Version 1.1.2, RStudio, 2024, https://cran.r-project.org/package=dplyr.
63. Wickham, Hadley, et al. "Welcome to the tidyverse." *Journal of Open Source Software*, vol. 4, no. 43. Version 2.0.0, RStudio, 2024, https://cran.r-project.org/package=tidyverse.
64. Wickham, Hadley. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. Version 3.4.2, RStudio, 2024, https://cran.r-project.org/package=ggplot2.
65. Nakra, Rohan, et al. "Evaluating TensorFlow image classification in classifying proton collision images for particle colliders." *Journal of Emerging Investigators*, 20 Aug. 2023. https://doi.org/10.59720/23-013.

https://doi.org/10.59720/25-132