
HYPERPARAMETER-FREE NEUROCHAOS LEARNING ALGORITHM FOR CLASSIFICATION

Akhila Henry

Department of Mathematics
Amrita Vishwa Vidyapeetham
Amritapuri Campus
Kollam, 690525, Kerala, India.
akhilahenryu@am.amrita.edu

Nithin Nagaraj

Complex Systems Programme
National Institute of Advanced Studies
Indian Institute of Science Campus
Bengaluru, 560012, Karnataka, India.
nithin@nias.res.in

ABSTRACT

Neurochaos Learning (NL) is a brain-inspired classification framework that employs chaotic dynamics to extract features from input data and yields state-of-the-art performance on classification tasks. However, NL requires the tuning of multiple hyperparameters and computing of four chaotic features per input sample. In this paper, we propose *AutochaosNet* – a novel, hyperparameter-free variant of the NL algorithm that eliminates the need for both training and parameter optimization. AutochaosNet leverages a universal chaotic sequence derived from the Champernowne constant and uses the input stimulus to define firing time bounds for feature extraction. Two simplified variants—TM AutochaosNet and TM-FR AutochaosNet—are evaluated against existing NL architecture - ChaosNet. Our results demonstrate that AutochaosNet achieves competitive or superior classification performance while significantly reducing training time due to reduced computational effort. In addition to eliminating training and hyperparameter tuning, AutochaosNet exhibits excellent generalisation capabilities, making it a scalable and efficient choice for real-world classification tasks. Future work will focus on identifying universal orbits under various chaotic maps and incorporating them into the NL framework to further enhance performance.

Keywords NeurochaosLearning · Tracemean · Firing Rate · Universal Orbit · AutochaosNet

1 Introduction

The current trajectory of artificial intelligence research is geared toward enabling machines to learn autonomously. A wide variety of machine learning algorithms are being developed at an accelerated pace. Among these, brain-inspired learning algorithms have recently garnered significant attention. One such approach is the *Neurochaos Learning (NL)* algorithm [1], which draws inspiration from the dynamical behaviour of the human brain, particularly the presence of chaos in neural activity.

In 2023, researchers introduced two distinct architectures under the NL framework: ChaosNet and CFX (Chaotic Feature Extraction) + ML [2]. These architectures employ chaos through the use of a one-dimensional chaotic map known as the Skew Tent Map. For each dataset, a neural trace is computed, which terminates when it falls within a predefined noise threshold.

Given a dataset with m samples and n features, the ChaosNet and CFX+ML algorithms extract four chaotic features - Firing Time, Firing Rate, Energy, and Entropy - for each input feature, resulting in a transformed feature set of size $4n$. These features are subsequently fed into classical machine learning classifiers or used directly with cosine similarity for classification. The algorithm involves hyperparameters such as the initial condition and skew value of the chaotic map, as well as the noise level, all of which require tuning. This process increases the computational cost and complexity of feature extraction.

In this study, we introduce a novel variant of the Neurochaos Learning algorithm that completely eliminates the need for hyperparameter tuning by leveraging a recently proposed universal chaotic orbit [3], derived using the Decimal Shift Map. This approach aims to streamline the feature extraction process while maintaining the effectiveness of chaotic representations for learning tasks.

2 Neurochaos Learning using Universal Orbit

Neurochaos Learning (NL) is an interdisciplinary approach that integrates concepts from neuroscience, chaos theory, and machine learning. A recent advancement in this field involves the introduction of a novel class of chaotic orbits known as universal orbits, specifically defined with respect to the Decimal Shift Map (DSM) [4] and the Continued Fraction Map [3]. In this study, we employ the universal orbit under the DSM for chaotic feature extraction.

The orbit of the DSM is generated by successively shifting the digits of the initial point. Universal orbits under the DSM can be generated from uncountably many initial points. Among these, we focus on the orbit derived from the Champernowne constant, denoted by

$$C = 0.1234567891011 \dots 498499,$$

truncated after the digits corresponding to the number 499. Also, this constant is known to be normal [5] in base 10, ensuring that all finite digit sequences appear with equal frequency [3]. The following lemma provides the theoretical basis for the occurrence of any natural number within this constant.

Lemma 1. *Let $c = 0.1234567891011 \dots N(N+1)(N+2) \dots$ be the Champernowne constant and $c_N = 0.1234567891011 \dots (N-2)(N-1)$ be the truncated portion of c , where $N = a_1 a_2 \dots a_d$, $d \geq 2$. Then the number of digits after the decimal point in c_N is given by*

$$dN - (10 + 10^2 + \dots + 10^{d-1}) - 1. \tag{1}$$

In other words, the number N will occur in the decimal expansion of c after

$$dN - (10 + 10^2 + \dots + 10^{d-1}) - 1$$

digits from the decimal point. [3].

A detailed proof of the lemma can be found in [3].

The use of the DSM with champernowne's constant as initial point eliminates the need for tuning parameters such as the skew value and noise threshold, which are typically required in conventional NL algorithms. Instead of using a predefined noise level to terminate the neural trace, the trace is halted when its first three decimal places match those of the input stimulus (feature value). The threshold of three decimal places was chosen arbitrarily in this study and may be adjusted in future work based on empirical performance. This is equivalent to the noise threshold of 0.001.

Suppose the stimulus (feature value) is $0.b_1 b_2 b_3 b_4 b_5$. Then by the lemma 1, the number $N' = b_1 b_2 b_3$ will occur in the decimal expansion of c (Champernowne's constant) after $\alpha = 3N' - (10 + 10^2) - 1$ digits from the decimal point

(Here, we are considering a threshold of three decimal places i.e, $d = 3$). Thus α will be an upper bound for the firing time.

Specifically, lemma 1 allows us to determine firing time bound and firing rate associated with each stimulus, which serves as the key chaotic feature in the proposed variant of the Neurochaos Learning algorithm.

3 Proposed Algorithm: AutochaosNet

In this section, we will employ universal orbit generated by decimal shift map with Champernowne’s constant to develop a hyperparameter-free Neurochaos Learning Architecture i.e, AutochaosNet. There are two versions of AutochaosNet based on the features extracted:

- TM AutochaosNet (Tracemean AutochaosNet)
- TM-FR AutochaosNet (Tracemean and Firing Rate AutochaosNet)

The proposed architecture is as follows:

- (a) **Input** : Suppose the input data is of size $m \times n$, where m represents the number of samples and n denotes the number of raw input features (or attributes). Let

$$\{(x_{11}, x_{12}, \dots, x_{1n}), (x_{21}, x_{22}, \dots, x_{2n}), \dots, (x_{m1}, x_{m2}, \dots, x_{mn})\}$$

be the entire dataset. After suitable normalisation, every data instance can undergo feature transformation sequentially. Min-max normalisation is utilised here. The normalised data can be represented as :

$$\{(z_{11}, z_{12}, \dots, z_{1n}), (z_{21}, z_{22}, \dots, z_{2n}), \dots, (z_{m1}, z_{m2}, \dots, z_{mn})\}$$

where $z_{ij} = \frac{x_{ij} - \min(\{x_{ij}:1 \leq i \leq m\})}{\max(\{x_{ij}:1 \leq i \leq m\}) - \min(\{x_{ij}:1 \leq i \leq m\})}$ for $1 \leq j \leq n$.

- (b) **Feature Transformation**: Each feature attribute x_{ij} of the data instance has to be transformed into a chaos-based feature.

In both versions of AutochaosNet, the firing time bound value corresponding to each x_{ij} is calculated using the formula defined in Lemma 1 within three decimal places. Let T be the firing time bound corresponding to $x_{(ij)}$ and the Champernowne’s constant (truncated till 500),

$$c = 0.1234567891011 \dots 498499$$

be the initial point. Then the neural trace corresponding to x_{ij} is

$$\tau = \{c, f(c), f^{(2)}(c), \dots, f^{(T-1)}(c)\}$$

- (c) **Feature Extraction**: In TM AutochaosNet, corresponding to each x_{ij} , mean of the neural trace τ is calculated. Suppose if t_{ij} is the mean of the neural trace corresponding to x_{ij} . Then the extracted feature set will be :

$$\{(t_{11}, t_{12}, \dots, t_{1n}), (t_{21}, t_{22}, \dots, t_{2n}), \dots, (t_{m1}, t_{m2}, \dots, t_{mn})\}$$

In TM-FR AutochaosNet, corresponding to each x_{ij} , mean of the neural trace τ and its firing rate are calculated. Firing rate corresponding to each x_{ij} is defined as fraction of time for which the neural trace τ exceeds the value 0.5 to recognise the stimulus.

- (d) **Classification:** Datasets with the transformed features can be classified using cosine similarity. First, compute the mean representation vector of each class. Suppose the m data instances belong to k classes. Let

$$\{(x_{l_11}, x_{l_12}, \dots, x_{l_1n}), (x_{l_21}, x_{l_22}, \dots, x_{l_2n}), \dots, (x_{l_r1}, x_{l_r2}, \dots, x_{l_rn})\}$$

be the r samples in class l . After normalisation, the extracted data be:

$$\{(f_{l_11}, f_{l_12}, \dots, f_{l_1n}), (f_{l_21}, f_{l_22}, \dots, f_{l_2n}), \dots, (f_{l_r1}, f_{l_r2}, \dots, f_{l_rp})\}$$

In TM AutochaosNet, the value of p will be n itself, since there is only one feature. But in TM-FR AutochaosNet, the value of index p will be $2n$. Then the mean representation vector corresponding to the class l can be defined as:

$$M^{(l)} = \left(\frac{\sum_{i=l_1}^{l_r} f_{i1}}{r}, \frac{\sum_{i=l_1}^{l_r} f_{i2}}{r}, \dots, \frac{\sum_{i=l_1}^{l_r} f_{ip}}{r} \right)$$

In order to classify a particular data instance

$$X_i = (x_{i1}, x_{i2}, \dots, x_{in}),$$

calculate the cosine similarity of extracted feature vector of the data instance $F_i = (f_{i1}, f_{i2}, \dots, f_{ip})$ with the mean representation vectors of each class $\{M^{(1)}, M^{(2)}, \dots, M^{(k)}\}$. Cosine similarity is defined as follows:

$$\cos \theta = \frac{M^{(j)} \cdot F_i}{\|M^{(j)}\| \|F_i\|},$$

where $j = 1, 2, \dots, k$. The data instance will belong to the class with least value.

- (e) **Output :** The output refers to the class to which each test data instance is assigned. AutochaosNet gives output based on mean representation vector of each class .

The proposed algorithm is illustrated in Figure 1.

3.1 Results and Analysis

This section presents a comparative analysis of the proposed hyperparameter free algorithms : TM AutochaosNet and TM-FR AutochaosNet against chaosNet using F1 Score as evaluation metric. ChaosNet algorithm utilises four chaotic features and three hyperparameters whereas the proposed algorithms utilises maximum two chaotic features and operate without any hyperparameters. The table 1 reports the F1 Scores of AutochaosNet aand chaosNet algorithms across ten distinct datasets. The F1 Scores with the highest values are highlighted in bold. Notably, despite the absence of hyperparameters, the performance of AutochaosNet algorithm remains comparable to that of chaosNet algorithm.

3.1.1 Computational Complexity

Beyond achieving comparable F1 Scores, a key advantage of AutochaosNet Algorithm is its ability to classify datasets in a shorter time compared to chaosNet. To compare the time complexity of these algorithms, four datasets namely - Iris, Seeds, Statlog and Sonar were selected for evaluation. Each algorithm was executed over 50 iterations and the average value of elapsed time over 50 iterations was computed. The mean elapsed time over 50 iterations for TM-AutochaosNet, TM- FR AutochaosNet and ChaosNet is indicated in the table 2.

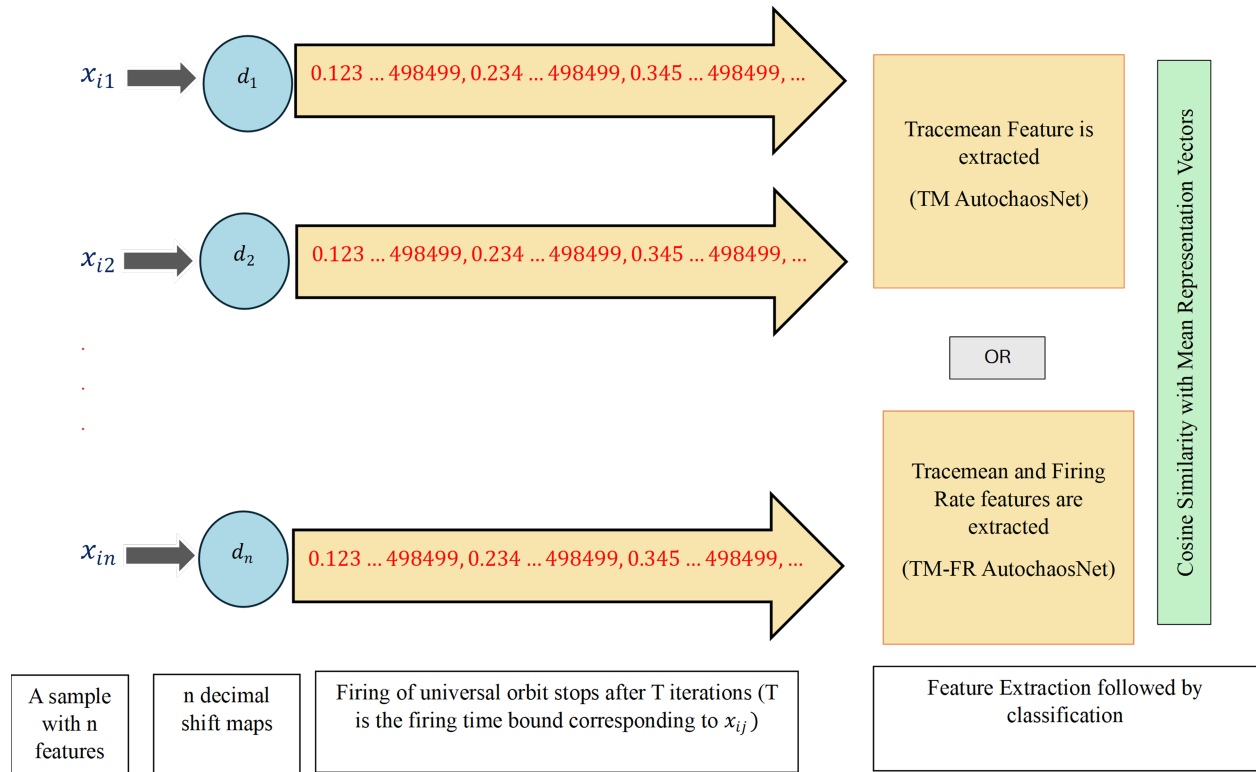


Figure 1: Architecture of AutochaosNet algorithm for feature extraction and classification.

Table 1: Comparison of F1 Scores for AutochaosNet with ChaosNet.

Dataset	TM-FR AutochaosNet	TM AutochaosNet	ChaosNet
Iris	0.868	0.838	1.000
Haberman	0.537	0.516	0.560
Seeds	0.878	0.878	0.845
Statlog	0.755	0.755	0.738
Breast Cancer	0.784	0.717	0.927
Bank	0.821	0.806	0.845
Ionosphere	0.823	0.813	0.860
Wine	0.858	0.824	0.976
Sonar	0.785	0.760	0.643
Penguin	0.907	0.885	0.964

Table 2: Comparison of Macro F1 Scores, Mean and Standard Deviation of Elapsed Time over 50 iterations for different models.

Dataset	TM-FR AutochaosNet			TM AutochaosNet			ChaosNet (Single Iteration)	
	F1 Score	Mean Time (s)	Standard Deviation	F1 Score	Mean Time (s)	Standard Deviation	F1 Score	Mean Time (s)
<i>Iris</i>	0.868	0.741	0.123	0.838	0.596	0.018	1	52.15
<i>Seeds</i>	0.878	1.577	0.044	0.878	4.99	0.475	0.845	7310.19
<i>Statlog</i>	0.755	2.384	0.241	0.755	2.531	0.279	0.738	19750
<i>Sonar</i>	0.785	14.506	0.858	0.76	14.702	0.584	0.643	10686.01

4 Conclusion

In this study, we introduced two hyperparameter-free variants of the Neurochaos Learning algorithm—TM AutochaosNet and TM-FR AutochaosNet—which leverage a universal chaotic orbit generated using the Decimal Shift Map with Champernowne’s constant (truncated till 500) as the initial point. By eliminating the need for hyperparameter tuning and relying on a simplified set of chaotic features, these models significantly reduce computational complexity while preserving classification accuracy.

Extensive evaluations on multiple benchmark datasets demonstrate that the proposed AutochaosNet algorithms deliver competitive performance compared to ChaosNet. Notably, AutochaosNet achieves this while using fewer features and no hyperparameters, offering substantial gains in efficiency. The ability of AutochaosNet to generalise across datasets without hyperparameter tuning makes it a scalable and practical solution for real-world classification problems.

In future work, we aim to identify sets of points that generate universal orbits under various chaotic maps. These truncated orbits will be used as starting triggers within the Neurochaos Learning framework, with the goal of further enhancing classification accuracy while preserving the algorithm’s simplicity and parameter-free nature.

Acknowledgements

NN would like to acknowledge the financial support of National Board of Higher Mathematics (NBHM), Dept. of Atomic Energy, Govt. of India (Grant No. 02011/8/2025/NBHM(R.P)/R&D II/1259).

Code Availability

The implementation of our proposed method is available at: <https://github.com/akhilahenry98/AutochaosNet.git>

Appendix

A Dataset Description

This section presents a summary of the ten datasets employed in this study, detailing the number of samples and the number of classes associated with each dataset.

Table 3: **Number of features and samples in each dataset used in this study.**

Dataset	Number of Features	Number of Classes	Number of Samples
<i>Iris</i>	4	3	150
<i>Haberman’s Survival</i>	3	2	306
<i>Seeds</i>	7	3	210
<i>Statlog (Heart)</i>	13	2	270
<i>Ionosphere</i>	34	2	351
<i>Bank Note Authentication</i>	4	2	1372
<i>Breast Cancer Wisconsin</i>	31	2	569
<i>Wine</i>	13	3	178
<i>Penguin</i>	4	3	342
<i>Sonar</i>	60	2	208

A.1 Iris

Iris [6] dataset contains information about three *Iris* plant varieties: “*Iris Setosa*, *Iris Versicolor*, and *Iris Virginica*”. This dataset contains 150 data objects, each having four attributes: “sepal length, width, petal length and width”. All attributes are measured in *cms*. The class distribution in Table 4 is as follows: *Setosa*, *Versicolor*, *Virginica*.

Table 4: **Iris : Number of Samples in each class.**

Class	Label	Number of Samples
Iris Setosa	0	50
Iris Versicolor	1	50
Iris Virginica	2	50

A.2 Haberman’s Survival

The dataset includes breast cancer surgical survival records from a 1958–1970 study at the University of Chicago’s Billings Hospital [7]. It contains 306 cases, each of which is characterised by three numerical characteristics: the number of positive axillary lymph nodes found, the year of surgery (represented as the year minus 1900), and the patient’s age at the time of surgery. The classification is based on whether the patient died within five years after surgery or survived for at least five years, as indicated in Table 5.

Table 5: **Haberman : Number of Samples in each class.**

Class	Label	Number of Samples
Less than 5 years	0	225
Greater than or equal to 5 years	1	81

A.3 Seeds

The *Seeds* [8] dataset offers measurements of the geometrical features of kernels from three distinct types of wheat - *Kama*, *Rosa* and *Canadian*. Seven real-valued attributes namely compactness, length, width etc. were constructed using a soft X-ray approach and the GRAINS package. The class distribution is shown in Table 6.

Table 6: **Seeds: Number of Samples in each class.**

Class	Label	Number of Samples
Kama	0	70
Rosa	1	70
Canadian	2	70

A.4 Statlog (Heart)

The *Statlog* [8] dataset provides information on the absence or presence of heart disease using 13 attributes including resting blood pressure, chest pain type, exercise induced angina and so on. The class distribution is shown in Table 7.

Table 7: **Heart: Number of Samples in each class.**

Class	Label	Number of Samples
Absence	0	150
Presence	1	120

A.5 Ionosphere

The *Ionosphere* [9] dataset is a binary classification dataset with dimensionality 34. The classes indicate the status of receiving a radar signal from the Ionosphere. The objective of this experiment is to determine the composition of the Ionosphere through the analysis of radar data. The class distribution is shown in Table 8.

Table 8: **Ionosphere: Number of Samples in each class.**

Class	Label	Number of Samples
Bad	0	126
Good	1	225

A.6 Bank Note Authentication

Bank Note Authentication [10] dataset is on differentiating Genuine and Forgery currencies. Data were retrieved from photos of authentic and counterfeit banknote-like samples. The dataset contains four attributes such as variance of Wavelet Transformed image (continuous), skewness of Wavelet Transformed image (continuous) etc. obtained from the images using wavelet transformation. The class distribution is shown in Table 9.

Table 9: **Bank: Number of Samples in each class.**

Class	Label	Number of Samples
Genuine	0	762
Forgery	1	610

A.7 Breast Cancer Wisconsin

Breast Cancer Wisconsin [11] (Diagnostic) dataset predict if the cancer is benign or malignant. The 31 features such as radius, perimeter, texture, smoothness etc. for each cell nucleus are derived from a digitised image of a fine needle aspirate (FNA) of a breast tumour. The class distribution is shown in Table 10.

Table 10: **Cancer: Number of Samples in each class.**

Class	Label	Number of Samples
Malignant	0	212
Benign	1	357

A.8 Wine

The results of a chemical analysis of wines produced from three different grape varieties in the same region of Italy are available in the *Wine dataset* [12]. Thirteen chemical constituent concentrations were measured for every wine sample in the study. With the labels “1, 2, and 3,” the wines are divided into three classes. Table 11 shows the distribution of these classes.

A.9 Palmer Penguins Dataset

The *Palmer Archipelago Penguin* data [13] is a dataset that includes size measurements for three species of penguins. Data was gathered from three islands in the Palmer Archipelago, Antarctica, comprising information on 344 penguins.

Table 11: **Wine: Number of Samples in each class.**

Class	Label	Number of Samples
1	0	59
2	1	71
3	2	48

There are three penguin species in the dataset: *Adelie*, *Gentoo*, and *Chinstrap*. The class distribution is shown in Table 12.

Table 12: **Penguin: Number of Samples in each class.**

Class	Label	Number of Samples
Adelie Penguin	0	151
Chinstrap Penguin	1	68
Gentoo Penguin	2	123

A.10 Sonar

The *Sonar* dataset [14] contains 97 patterns from rocks under similar conditions and 111 patterns from sonar signals bounced off a metal cylinder at various angles and conditions. The 60 numbers in the pattern range from 0.0 to 1.0. The label for each record indicates whether the item is a mine (represented by “M”) or a rock (represented by “R”). Table 13 displays the distribution of classes.

Table 13: **Sonar: Number of Samples in each class.**

Class	Label	Number of Samples
Mine	0	111
Rock	1	97

References

- [1] Harikrishnan N B, Aditi Kathpalia, Snehanshu Saha, and Nithin Nagaraj. Chaosnet: A chaos based artificial neural network architecture for classification. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29:113125, 11 2019.
- [2] Devashish Sethi and Nithin Nagaraj. Neurochaos feature transformation for machine learning. *Integration*, 90:157–162, 2023.
- [3] Akhila Henry, Nithin Nagaraj, and Rajan Sundaravaradhan. Universal orbits: Unveiling the connection between chaotic dynamics, normal numbers, and neurochaos learning. *Chaos Theory and Applications*, 7(1):61–69, 2025.
- [4] Steven H Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, Taylor & Francis Group, 2018.
- [5] D. Khoshnevisan. On the normality of normal numbers. Clay Mathematics Institute Annual Report, 2006.
- [6] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.
- [7] Shelby J Haberman. The analysis of residuals in cross-classified tables. *Biometrics*, pages 205–220, 1973.
- [8] Dheeru Dua, Casey Graff, et al. Uci machine learning repository. 2017.

- [9] Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.
- [10] Eugen Gillich and Volker Lohweg. Banknote authentication. *1. Jahreskolloquium Bild. Der Autom*, pages 1–8, 2010.
- [11] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. SPIE, 1993.
- [12] Michele Forina, Riccardo Leardi, C Armanino, Sergio Lanteri, Paolo Conti, P Princi, et al. Parvus an extendable package of programs for data exploration, classification and correlation. 1988.
- [13] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. palmerpenguins: Palmer archipelago (antarctica) penguin data. *R package version 0.1.0*, 2020.
- [14] R Paul Gorman and Terrence J Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural networks*, 1(1):75–89, 1988.