Random Heterogeneous Neurochaos Learning Architecture for Data Classification

Remya Ajai A S [ID]*,1 **and Nithin Nagaraj** [ID]α,2

*Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India, αComplex Systems Programme, National Institute of Advanced Studies, Indian Institute of Science Campus, Bengaluru, Karnataka, India.

ABSTRACT Inspired by the human brain's structure and function, Artificial Neural Networks (ANN) were developed for data classification. However, existing Neural Networks, including Deep Neural Networks, do not mimic the brain's rich structure. They lack key features such as randomness and neuron heterogeneity, which are inherently chaotic in their firing behavior. Neurochaos Learning (NL), a chaos-based neural network, recently employed one-dimensional chaotic maps like Generalized Lüroth Series (GLS) and Logistic map as neurons. For the first time, we propose a random heterogeneous extension of NL, where various chaotic neurons are randomly placed in the input layer, mimicking the randomness and heterogeneous nature of human brain networks. We evaluated the performance of the newly proposed Random Heterogeneous Neurochaos Learning (RHNL) architectures combined with traditional Machine Learning (ML) methods. On public datasets, RHNL outperformed both homogeneous NL and fixed heterogeneous NL architectures in nearly all classification tasks. RHNL achieved high F1 scores on the Wine dataset (1.0), Bank Note Authentication dataset (0.99), Breast Cancer Wisconsin dataset (0.99), and Free Spoken Digit Dataset (FSDD) (0.98). These RHNL results are among the best in the literature for these datasets. We investigated RHNL performance on image datasets, where it outperformed stand-alone ML classifiers. In low training sample regimes, RHNL was the best among stand-alone ML. Our architecture bridges the gap between existing ANN architectures and the human brain's chaotic, random, and heterogeneous properties. We foresee the development of several novel learning algorithms centered around Random Heterogeneous Neurochaos Learning in the coming days.

INTRODUCTION

Brain consists of complex networks of enormous number of neurons which are inherently non-linear (Ramachandran *et al.* 1998). Inspired by the human brain in the way biological neurons are signaling to one another, Artificial Neural Networks (ANN) were developed for purposes of information processing and classification. With the rapid growth of Artificial Intelligence (AI) algorithms and easy availability of highly efficient and inexpensive computing hardware, almost all application domains today utilize Machine Learning (ML) algorithms/techniques and Deep Learning (DL) architectures for various tasks.

Applications of AI include (and not limited to) speech processing (Graves *et al.* 2013), cybersecurity (Harikrishnan *et al.* 2018), computer vision (Sebe 2005), and medical diagnosis (Harikrishnan *et al.* 2019; Remya Ajai and Gopalan 2020; Krishna and Ajai 2019; Asif *et al.* 2023). There are also algorithms that were developed to relate with the human brain in terms of learning and memory encoding (Aihara *et al.* 1990). The learning algorithms perform internal weight updates and optimize their hyperparameter values

so that error functions are minimized. Even though ANN is a huge success today, it needs to be greatly improved in order to mimic the human brain in terms of energy-efficient performance of complex tasks. Thus, in recent times, there has been a focus towards developing novel biologically-inspired algorithms and learning architectures by various researchers (Delahunt and Kutz 2019; Balakrishnan *et al.* 2019).

There are fundamental differences between the way the human brain functions at the level of a single neuron (the basic unit that receives, processes and transmits information) and how ANN processes information (see Table 1). Brain neurons are found to exhibit chaotic behaviour (Korn and Faure 2003) whereas neurons in ANNs perform simple weighted addition of input data. Typically, in existing ANNs, homogeneous neurons are used whereas in the central nervous system/brain (Perez-Nieves *et al.* 2021), neurons are known to be heterogeneous. These neurons in biological neural networks are differentiated based on structure and function. Sensory neurons are activated by input of sensory stimuli from the external environment. Motor neurons of the spinal cord connect to the muscle glands and organs throughout the human body. Interneurons connect spinal motor and sensory neurons. Based on structure of the neurons, they are classified as either being unipolar, bipolar, pseudounipolar or multipolar (Weis *et al.* 2019). Based on these observations, we can correctly say that current ANNs are only loosely inspired by the brain.
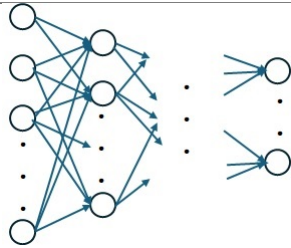
Table 1 A brief comparison of ANNs and biological neural networks.

Artificial Neural Networks (ANN)	Biological Neural Networks
Homogeneous neurons	Heterogeneous neurons
Every neuron performs a weighted linear combination of inputs followed by nonlinear activation	Neurons classified based on structure (unipolar, bipolar, pseudounipolar or multipolar) and function (sensory, motor, interneurons)
Scalar valued output at every neuron	Vector valued output as different neurons fire at different rates and duration
Non-chaotic neurons	Chaotic neurons (spiking and bursting behaviour)
Complexity of network of neurons is achieved through depth	Complexity achieved through depth, heterogeneity, randomness and differentiated processing
Causal structures of input dataset is not preserved internally in ANNs (NB *et al.* 2022)	Internal representation of input stimuli preserves causal structures (NB *et al.* 2022)

Neurochaos Learning (or NL) is a recently developed brain-inspired chaos-based artificial neural network for data classification (Balakrishnan *et al.* 2019; Harikrishnan and Nagaraj 2020). Majority of Machine Learning (ML) algorithms have relied heavily on substantial datasets for acquiring knowledge about the underlying distribution. The first of NL architectures, dubbed `ChaosNet`, has demonstrated state-of-the-art performance in classification tasks with only a fraction of training samples needed for learning. Subsequently, NL was shown to perform equally well on imbalanced datasets, as well as, boost the performance of standard ML classifiers (SVM, kNN and others) (Sethi *et al.* 2023). Moderate levels of noise within the context of neurochaos learning are found to optimize performance in classification tasks (Harikrishnan and Nagaraj 2021). It is no surprise that NL has found to preserve causal structures of input dataset in its internal representation of chaotic neural traces (NB *et al.* 2022) which is completely missing in the internal representation of ANNs.

In our previous study (AS *et al.* 2023), we have proposed an extension of NL architecture to incorporate heterogeneous neurons. We first demonstrated that the NL architecture with homogeneous neurons, but with a different 1D chaotic map (the logistic map) than the one used in `ChaosNet` (1D Generalized Lüroth Series or GLS map) is also equally good at learning tasks. Classification accuracies for *Ionosphere, Statlog (Heart), Bank Note Authentication, Breast Cancer Wisconsin, Haberman's Survival* and *Seeds* increased with the use of one-dimensional (1D) logistic map (with chaotic behaviour regime) as neurons compared with GLS maps as neurons. We then proposed HNL: Heterogeneous Neurochaos Learning which

combined GLS maps as neurons and logistic maps as neurons in a simple odd-even structure of the input layer (AS *et al.* 2023). HNL gave comparable performance to homogeneous NL and in the case of *Seeds* and *Haberman's Survival* datasets, it outperformed. We also studied the effect of degree of chaos on classification accuracies, as characterized by the lyapunov exponent of the chaotic 1D neurons in HNL.

In this work, for the first time, we propose `Random Heterogenous Neurochaos Learning` (RHNL) architecture. As noted in Table 1, the human brain not only has heterogeneous neurons organized in layers, but there is an element of randomness involved. No two human brains have the same topological connectivity of neurons in their networks. The randomness is due to differences in early development that is a function of environment, learning and genetic factors. Inspired by this fact, we incorporate randomness and heterogeneity in NL. Specifically, we have analyzed three different RHNL architectures. The first one consists of 25% of logistic map neurons and the remaining 75% of GLS neurons, all randomly placed in the input layer. The second one consists of 50% − 50% of logistic-GLS neurons (again randomly placed) while the third architecture is composed of 75% − 25% of logistic-GLS neurons. We have rigorously tested these architectures (on classification tasks) in conjunction with different classifiers (cosine similarity and other ML classifiers) on a number of publicly available datasets.

This paper is structured as follows. The proposed RHNL architecture is introduced for the first time (in Section 2). This is followed by a description of datasets in Section 3 and classifiers

used in our study (Section 4). Experiments along with their results follow in Section 5. Section 5.1 gives the results obtained for Time Series Dataset. Discussion on the classification performance of $ChaosFEX_{RHNL}$ for debris and urban images are included in section 5.2. Results and analysis of the classification performance of $ChaosFEX_{RHNL}$ for brain tumor image dataset are included in section 5.3. Performance analysis of $ChaosFEX_{RHNL}$ in comparison with stand-alone ML classifiers in included in section 5.4. Section 5.5 contains the discussion on the performance of RHNL in low training sample regime. The paper then concludes with discussion, followed by potential research directions for the future in Section 6. The appendix contains the complete details of hyperparameter tuning of all the learning architectures used in this study.

PROPOSED ARCHITECTURE

In order to mimic the randomness and heterogeneity of neuronal structures present in our brains, we propose a novel neurochaos learning architecture depicted in Figure 1. The input layer of this *Random Heterogeneous Neurochaos Learning* architecture (RHNL) consists of both chaotic 1D Logistic map and Generalized Lüroth Series (GLS) map neurons, but at randomized locations. Contrast this with the *Heterogeneous Neurochaos Learning* or HNL proposed in (AS *et al.* 2023) where we had employed a simple odd-even structure (odd positions for GLS map neurons and even positions for logistic map neurons). In RHNL, we control the proportion of the randomly placed GLS and logistic neurons in the input layer to yield three distinct RHNL architectures: $25\% - 75\%, 50\% - 50\%$ and $75\% - 25\%$. In each case, the locations of the neurons are chosen uniformly at random. Following the flow in Figure 1, each neuron (either a GLS or a logistic map neuron) starts firing chaotically as soon as it encounters an input stimuli (x_i).

Each input stimuli is a data (text/image/video etc.) sample of a particular class which RHNL is tasked to learn and classify. Each neuron stops firing as soon as it detects the input stimuli (when it lands in an ϵ-neighbourhood). This completes a chaotic neural trace. Since different neurons detect their corresponding stimuli at different times, the chaotic neural traces are of unequal length (similar to the brain). These chaotic neural traces are then analyzed to extract features (*ChaosFEX*) such as entropy, energy, firing time and firing rate. Subsequently, the *ChaosFEX* features are fed to a classifier - that may consist of a basic *Cosine Similarity-based* classifier (Balakrishnan *et al.* 2019) or one of the standard ML classifiers (Sethi *et al.* 2023) such as SVM: Support Vector Machines, DT: Decision Trees, kNN: k-Nearest Neighbour, AB: AdaBoost, RF: Random Forests, or GNB: Gaussian Naive Bayes.

Each neuron (logistic or GLS) starts of with a fixed *initial neural activity* value q which is one of the hyperparameters of the learning algorithm. The other hyperparameters are the value of ϵ (*noise intensity*) that determines the stopping criteria of the neural firing/trajectory (which is chaotic) and the *Discrimination Threshold* (b) which is needed to compute Shannon Entropy (one of the *ChaosFEX* features) from the *symbolic sequence* of the chaotic neural trace (Balakrishnan *et al.* 2019; Harikrishnan and Nagaraj 2020; Sethi *et al.* 2023). These three hyperparameters (q, ϵ, b) are determined by a cross-validation strategy (of five folds). We now describe the GLS and logistic map neurons.

The Generalized Lüroth Series (GLS) Neuron

In (Balakrishnan *et al.* 2019), the one dimensional discrete dynamical system known as the GLS: Generalized Lüroth Series is used as the neuron. Skew-tent/tent maps, skew-binary/binary maps are commonly used among the GLS maps. This class of

one-dimensional systems/maps have demonstrated high effectiveness in various engineering applications (Nagaraj 2022). In our proposed *Random Heterogenous Neurochaos Learning* (RHNL) architecture, skew-tent maps are used as chaotic neurons. The Skew-tent map $C_{Skew-Tent} : [0.0, 1.0) \mapsto [0.0, 1.0)$ is mathematically defined as :

$$C_{Skew-Tent}(z) = \begin{cases} \frac{z}{b} & , \quad 0 \le z < b, \\ \frac{(1-z)}{(1-b)} & , \quad b \le z < 1, \end{cases} \tag{1}$$

where $z \; \epsilon \; [0.0, 1.0)$ and $0.0 < b < 1.0$.

The Logistic Map Neuron

The one-dimensional dynamical system/map known as the *Logistic map* is arguably the simplest example of a chaotic map (Phatak and Rao 1995). We explore the use of this one-dimensional map/dynamical system (in chaotic state) as neurons for RHNL. The equation for this dynamical system/map is:

$$x_{z+1} = rx_z(1 - x_z), \tag{2}$$

where $0.0 \le x_z < 1.0$ and the bifurcation parameter is: $0 < r \le 4.0$. z is the iteration/time step. It is widely recognized that the logistic map displays chaotic behavior for r values that are greater than 3.56995. However, there exist certain regions of r referred to as *islands of stability* where chaotic behavior is lost. Figure 2 shows the first return map of the logistic dynamical system/map with r set to 4.0. The lyapunov exponent is also plotted alongside.

Logistic map displays an infinite number of periodic orbits for each integer value of the period (only for specific values of r), indicating intricate dynamics (Alligood *et al.* 1998). For numerous r values, this map exhibits high sensitivity to initial conditions (*Butterfly Effect*) – a characteristic hallmark/feature of chaos. The degree of sensitivity to initial values can be quantified through the *Lyapunov exponent*. A lyapunov exponent value greater than zero is a symptom of chaotic behavior. Alternatively, a lyapunov exponent $= 0$ or < 0 suggests either a periodic/eventually periodic/quasi-periodic behaviour. For the difference equation (first-order):

$$x_{j+1} = G(x_j), \tag{3}$$

the lyapunov exponent is defined as:

$$\lambda_G(x) = \lim_{k \to \infty} \frac{1}{k} \sum_{j=0}^{k-1} \ln |G'(x_j)|, \tag{4}$$

where $G(\cdot)$ is assumed to be differentiable. The initial value x_0 is randomly chosen (from an uniform distribution) to lie between 0.0 and 1.0. $x_0 \to G(x_0) \to G^2(x_0) \to \ldots$ is the trajectory. $G(x)$ is given by equation 2 for the logistic map/dynamical system.

Feature Transformation, Extraction and Classification

As depicted in Figure 1, in the newly proposed RHNL structure, both 1D chaotic Logistic and GLS neurons are employed in randomly selected locations of input layer. Each neurons transforms the input stimulus (data sample) into respective chaotic neural traces (or chaotic trajectories). From these trajectories, required features are derived for further classification process. The quantity of neurons, denoted as ($C_1, C_2, \ldots C_n$) as illustrated in Figure 1, will precisely match the quantity of features of the input data/samples. All input layer neurons independently start firing (when an input value/stimulus triggers it), say x_i. These start of with an initial
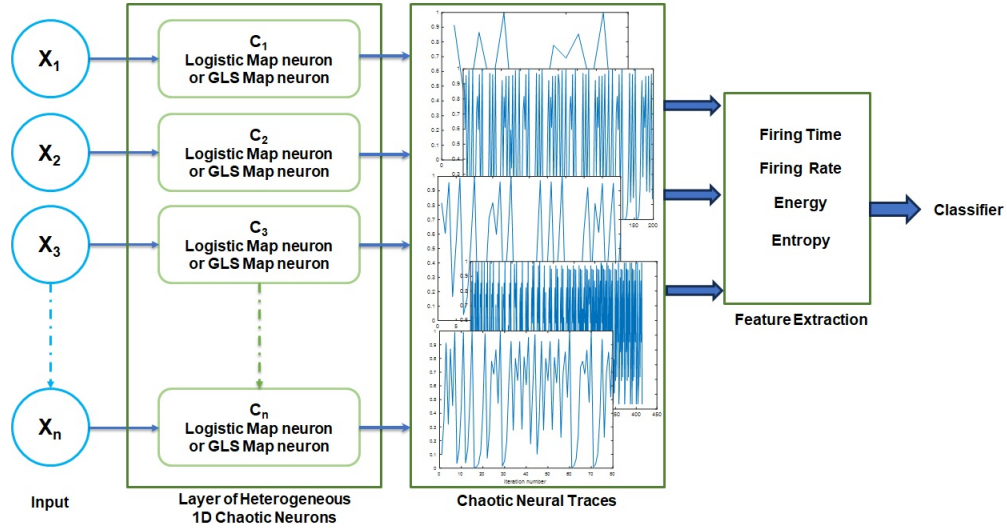
Figure 1 Random Heterogenous Neurochaos Learning (RHNL) Architecture: $(X_1, X_2...X_n)$ are the input stimuli (data sample), $(C_1, C_2, C_3, ..., C_{n-1}, C_n)$ are neurons which can be either 1D Logistic map or GLS map. Each neuron fires chaotically until it detects the input stimuli. From the neural traces of every neuron (which is chaotic), four features namely firing-time, firing-rate, entropy and energy are extracted. These *ChaosFEX* features would now be fed to either a cosine similarity classifier or any of the standard machine learning classifiers. The logistic map and GLS neurons are randomly placed in the input layer with one of the three following proportions: $25\% - 75\%$, $50\% - 50\%$ and $75\% - 25\%$.
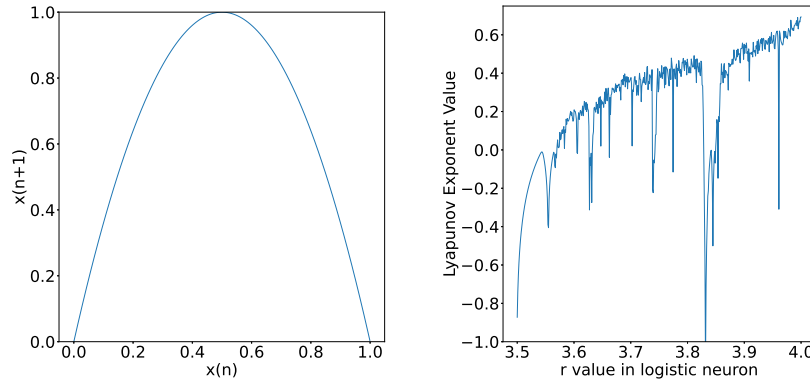
Figure 2 (I) Left: One dimensional logistic map with *r* value set to 4.0. (II) Right: The lyapunov exponent computed for different *r* values (varied from 3.5 to 4.0). The initial neural activity *q* was set to 0.01.

value/neural-activity of q units. Each of the input values are scaled to fall within the range of $[0, 1]$. Upon entering the open ϵ-ball (or neighbourhood) of the input stimulus, the neural trace comes to a halt. In the context of classification using Neurochaos Learning (NL) architecture, a straightforward decision rule is employed, relying on *mean-representation vectors* (Balakrishnan *et al.* 2019). Tuning of the three essential hyper-parameters is necessary: noise intensity (ϵ), initial neural-activity (q) and discrimination-threshold (b). To determine the optimal performance, a cross-validation approach (with five-folds) is utilized for hyperparameter tuning. Following the tuning and stabilization of these for a specific data-set, we generate the *neurochaos* features. These features that are obtained from the neural trajectories/traces (which are chaotic) of the input layer neurons of the NL architecture (could be homogeneous NL or HNL or RHNL) are referred to as *ChaosFEX* (Sethi *et al.* 2023) features. These include firing-rate, firing-time, Shannon entropy and energy (briefly described below).

Firing time is defined as the duration of the chaotic neural trajectory/trace to align with input value (or stimulus) (Sethi *et al.* 2023). This duration is measured in terms of iteration steps. The firing rate is determined by that fraction of time during which the chaotic neural trajectory is greater than the discrimination threshold (Sethi *et al.* 2023). Energy of the chaotic trace/trajectory $c(t)$ is defined as:

$$E_c = \sum_{t=1}^{M} |c(t)|^2, \tag{5}$$

where M = firing-time. Let symbolic sequence (binary) of the trace/trajectory, be denoted by $s(t)$, which is expressed as:

$$s(t_j) = \begin{cases} 0, & c(t_j) < b \\ 1, & b \leqslant c(t_j) < 1, \end{cases} \tag{6}$$

where $j = 1$ to M (the firing-time). The Shannon first-order entropy

for $s(t)$ is determined using the following calculations:

$$H(s) = -\sum_{t=1}^{2} p_i \log_2(p_i) \; bits, \tag{7}$$

where p_1 corresponds to the probability of symbol 0 while p_2 pertains to the probability of symbol 1. Dimensions of these features are similar to the size of the inputs. For an input of dimensions $m \times n$, the output features will have a size of $m \times 4n$. This process involves converting the input data into a feature space characterized by high-dimensional chaos.

In the training phase of the algorithm, for each data sample of each class, a representation vector is formed with the set of *ChaosFEX* features namely energy, entropy, firing time and firing rate. These representation vectors of all the data samples within a particular class are collected and its mean is computed to yield a single *mean representation vector* for each class. Thus every class has its own distinct mean representation vector that is like a signature of the neurochaos features of that class. These mean representation vectors are fed to a classifier to perform the classification. In the testing phase, when a test sample appears at the NL architecture's input layer, it undergoes a similar transformation to yield the *ChaosFEX* features. These features are compared with the mean representation vectors of each class to determine the *closest* one and the class label of the closet one is declared to be the class of input test sample. For the `ChaosNet` version of NL, we employ the cosine similarity measure to determine closeness. Alternatively, we could pass on the representation vectors corresponding to the data samples (within a class) to a traditional ML classifier and learn the decision boundary.

DATASETS

To conduct our analysis, we picked datasets spanning various application domains. Before feeding the data samples to the chaotic neurons of RHNL's input layer, the data samples of the input is normalized to lie within $[0, 1]$. Class labels are assigned numerical names, starting from 0. Further information about the datasets utilized can be found in Table 2. The data-sets are described briefly here.

The *Iris* dataset (Fisher 1936; Dua *et al.* 2017) comprises 150 instances distributed across 3 classes: *Setosa*, *Versicolour*, and *Virginica*. The classification features include sepal length, sepal width, petal length, and petal width. The distribution of data instances into train and test sets for our analysis is presented in Table 2. The *Ionosphere* dataset (Sigillito *et al.* 1989; Dua *et al.* 2017) is divided into 2 classes, labeled as *Good* or *Bad*. Radar signal reflects back if any structure is present in the ionosphere. This state is represented as 'Good'. 'Bad' denotes the condition in which the ionosphere is penetrated by this (radar) signal. The data-set comprises 126 instances labeled as *Good*, 225 instances labeled as *Bad*, and includes 34 attributes. The distribution of instances into train and test sets for our analysis is outlined in Table 2. The *Wine* dataset (Vandeginste 1990; Dua *et al.* 2017) comprises 178 instances categorized into 3 classes labeled as 1, 2, and 3. The chemical constituents of each data are considered for classification. The distribution of instances into train and test sets for our analysis is presented in Table 2.

The *Bank Note Authentication* dataset (Gillich and Lohweg 2010; Dua *et al.* 2017) has two classes:*Genuine* or *Forgery* based on the images of the banknotes. Wavelet transformation is applied on images and the features such as variance, skewness, kurtosis, and entropy are derived. In total, the dataset comprises 1372 instances, with 762 instances classified as *Genuine* and 610 instances as *Forgery*. The distribution of instances into train and test

sets for our analysis is outlined in Table 2. The *Haberman's Survival* dataset (Haberman 1973; Dua *et al.* 2017) encompasses three attributes (collected from those patients who underwent breast cancer surgery). The label1 class represent the patients survived for ≥ 5 years. Class 2 represent the patients who died in a span of 5 years. The train and test sets distribution for our analysis is presented in Table 2. Nine parameters are considered for *Breast Cancer Wisconsin* dataset (Street *et al.* 1993; Dua *et al.* 2017). Data instances are categorized to be either *Malignant* or *Benign*. In total, there are 699 instances, with 241 being classified as *Malignant* and 458 as *Benign*. The distribution of instances into train and test sets for our analysis is outlined in Table 2.

The *Statlog (Heart)* dataset (Dua *et al.* 2017) contains two classes of data: patients having heart problems are classified in *Class*-1 and patients without any heart disease are represented in *Class*-2. The distribution of instances into train and test sets for our analysis is presented in Table 2. The *Seeds* dataset (Dua *et al.* 2017) is employed to distinguish between three types of wheat, namely *Kama*, *Rosa*, and *Canadian*. Wheat kernels are used for identification among the types with seven parameters that represent various features of the kernels. In all, 210 data instances are in consideration, with 70 instances allocated for each class. The distribution of instances into train and test sets for our analysis is outlined in Table 2.

In our analysis, we also included a time series dataset namely *Free Spoken Digit Dataset (FSDD)*. Recordings of six speakers reciting numbers from 0 to 9 (Jackson *et al.* 2018) is contained in this data-set. There are 50 recordings for each number per speaker. The samples in the *FSSD* data set are preprocessed using fast fourier transform (or FFT). We considered *Jackson* (one of the speakers) that has instances of data numbering 500. We filtered 480 data instances for feeding into our proposed algorithms and analysed. For our analysis, the train/test sets of *FSDD* are shown in Table 2.

Around 85 images of debris scars and urban settlements from five Asian countries (India, Nepal, Japan, Taiwan and China) were obtained from Planet labs (Planet Labs Inc. Accessed: 2019-10-09) imagery with $3 - 5m$ resolution for our analysis to identify the classification performance of RHNL, specifically ChaosFEX$_{RH25L75G}$ architecture. For our analysis, we also have considered 100 MRI brain images from *Kaggle* online repository (Chakrabarty Accesed:2019-10-09).

CLASSIFIERS

As mentioned previously, RHNL supports the use of traditional popular ML classifiers. The neurochaos features (*ChaosFEX*) can be fed to one of the many widely available machine learning classifiers to perform classification. Previously, it has been demonstrated that neurochaos features boost the performance of the ML classifiers (Sethi *et al.* 2023). In our study we use the following ML classifiers: Support Vector Machine (SVM) (Boser *et al.* 1992), AdaBoost (Schapire 2013), Decision Tree (Quinlan 1986), Guassian Naive Bayes (Berrar 2018), k-NN (Cover and Hart 1967) and Random Forests (RF) (Breiman 2001). Whenever a traditional ML classifier was used on the neurochaos featuers, the hyperparameters (q, b, ϵ) that were already tuned for the various RHNL architectures (*ChaosFEX*) were maintained and only the ML hyperparameters are further tuned. This reduces the computational burden.

The Adaptive Boosting (AdaBoost) classifier has the following hyperparameters:

- *n_estimator*: The maximum limit on the number of estimators at which boosting is terminated.

Table 2 Datasets (with their details) employed in our study.

Data-set	Num. of Classes	Samples per class (Training)	Samples per class (Testing)	Ref.
Iris	3	(40, 41, 39)	(10, 9, 11)	Fisher (1936); Dua *et al.* (2017)
Ionoshpere	2	(98, 182)	(28, 43)	Sigillito *et al.* (1989); Dua *et al.* (2017)
Wine	3	(45, 57, 40)	(14, 14, 8)	Vandeginste (1990); Dua *et al.* (2017)
Bank Note Authentication	2	(614, 483)	(148, 127)	Gillich and Lohweg (2010); Dua *et al.* (2017)
Haberman's Survival	2	(181, 63)	(44, 18)	Haberman (1973); Dua *et al.* (2017)
Breast Cancer Wisconsin	2	(367, 193)	(91, 48)	Street *et al.* (1993); Dua *et al.* (2017)
Statlog (Heart)	2	(117, 99)	(33, 21)	Dua *et al.* (2017)
Seeds	3	(59, 56, 53)	(59, 56, 53)	Dua *et al.* (2017)
FSDD	10	(40, 35, 44, 42, 38, 34, 37, 44, 33, 37)	(10, 15, 6, 8, 8, 7, 13, 6, 10, 13)	Jackson *et al.* (2018)

All other hyperparameters are maintained at their default values provided by *scikit-learn*. Tuned hyperparameters for all the datasets for RHNL that uses the AdaBoost classifier are given in Tables 21, 22 and 23(Appendix).

The Decision Tree (DT) classifier has the following hyperparameters: *min_samples_leaf* (varied from 1 to 10 in increments of 1), *max_depth* (1 to 10 in increments of 1), and *ccp_alpha*.

All other hyperparameters are maintained at their default values provided by *scikit-learn*. The tuned hyperparameters for all the datasets for RHNL that uses the Decision Trees classifier are given in Tables 24, 25 and 26 (Appendix).

The k-Nearest Neighbours (k-NN) classifier has the value of *k* as a hyperparameter. This is varied from 1.0 to 6.0 (in incremets of 2). All other hyper-parameters are maintained at their default values provided by *scikit-learn*. The tuned hyperparameters for all the datasets for RHNL that uses the kNN classifier are given in Tables 27, 28 and 29 (Appendix).

The Random Forests (RF) classifier has the following hyperparameters: n_estimators (can take values in the set $\{1, 10, 100, 1000, 10000\}$), max_depth (1 to 10 in steps of 1). All other hyperparameters are maintained at their default values provided by *scikit-learn*. The tuned hyperparameters for all the datasets for RHNL that uses the Random Forests classifier are given in Tables 30, 31 and 32 (Appendix).

For Support Vector Machines classifier (SVM), all hyperparameters (offered for linear support vector classification) are maintained at their default values provided by *scikit-learn*. Gaussian Naive Bayes (GNB) classifier calculates the likelihood and prior probabilities for making predictions. GNB assumes that the features follow a Gaussian distribution. Default parameters offered for GNB by scikit-learn are retained for our analysis. For ease of understanding, Table 3 gives a summarized view of all the different neurochaos learning (NL) architectures with the corresponding notations that are used in this paper.

For RHNL, as noted in Table 3, ChaosFEX$_{RH25L50G}$, ChaosFEX$_{RH50L50G}$, and ChaosFEX$_{RH75L25G}$ refers to the three distinct random heterogeneous neurochaos learning architectures with 25% − 75%, 50% − 50% and 75% − 25% proportion of 1D chaotic Logistic neurons and GLS neurons respectively. These chaotic neurons are placed at random locations in the input layer of RHNL.

EXPERIMENTS AND RESULTS

The performance of ChaosFEX$_{RH}$ architectures are analysed for various datasets using Macro F1-score (a function of both macro Recall as well as macro Precision). True-Positive rate (TP) signifies a positive target-value correctly identified as Positive. True-Negative rate (TN) denotes a 'negative target-value' correctly classified as Negative. False Positive rate (FP) accounts those instances when a 'negative target value' is inaccurately deemed/classified as Positive. False-Negative rate (FN) accounts those instances when a 'positive target value' is erroneously deemed/classified as Negative. Mathematically, they are described as:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}, \quad (8)$$

$$Precision = \frac{TP}{(TP + FP)}, \quad (9)$$

$$Recall = \frac{TP}{(TP + FN)}, \quad (10)$$

$$F1 = 2.0 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (11)$$

The Macro F1-score is computed as the average of all F1-scores (for the *m* classes), given by:

$$Macro\ F1 - score = \frac{F1Class_1 + F1Class_2 + \ldots + F1Class_m}{m}. \quad (12)$$

Tuning of the 3 hyper-parameters (q, b, ϵ) are performed across various datasets for the three RHNL architectures proposed namely ChaosFEX$_{RH25L75G}$, ChaosFEX$_{RH50L50G}$ and ChaosFEX$_{RH75L25G}$. Tuned values of hyperparameters for all architectures are given in tables 4, 5 and 6. Cross validation using five folds is utilized to fine tune hyper-parameters and determine the best achieved performance.

We also analysed the performance of our proposed RHNL architectures in combinations with other traditional ML classifiers such as AdaBoost, Decision Trees, Gaussian Naive Bayes, kNN and Random Forests. The ML classifier parameters are tuned for various datasets to obtain the best accuracy possible.

Table 3 Various learning architectures of Neurochaos Learning (NL). These include homogenous NL, heterogeneous NL (HNL) and random heterogeneous NL (RHNL), including combinations with ML classifiers.

No.	NL Architecture	Type of Neurons	Notation	Classifiers	Ref.
1	ChaosNet	Homogeneous, GLS	ChaosFEX	Cosine similarity	Balakrishnan *et al.* (2019)
2	ChaosNet	Homogeneous, Logistic	ChaosFEX$_{logistic}$	Cosine similarity	AS *et al.* (2023)
3	NL	Homogeneous, GLS	ChaosFEX+ML	SVM, AB, DT, kNN, GNB, RF	Sethi *et al.* (2023)
4	NL	Homogeneous, Logistic	ChaosFEX$_{logistic}$+ML	SVM, AB, DT, kNN, GNB, RF	AS *et al.* (2023)
5	HNL: ChaosNet	Heterogeneous, GLS, Logistic in odd-even structure	ChaosFEX$_{Hetero}$	Cosine similarity	AS *et al.* (2023)
6	HNL	Heterogeneous, GLS, Logistic in odd-even structure	ChaosFEX$_{Hetero}$+ML	SVM, AB, DT, kNN, GNB, RF	AS *et al.* (2023)
7	RHNL: ChaosNet	Heterogeneous & Random, GLS, Logistic in randomized locations	ChaosFEX$_{RH25L75G}$, ChaosFEX$_{RH50L50G}$, ChaosFEX$_{RH75L50G}$	Cosine similarity	This work.
8	RHNL	Heterogeneous & Random, GLS, Logistic in randomized locations	ChaosFEX$_{RH25L75G}$+ML, ChaosFEX$_{RH50L50G}$+ML, ChaosFEX$_{RH75L50G}$+ML	SVM, AB, DT, kNN, GNB, RF	This work.

Table 4 Tuned hyperparameters for ChaosFEX$_{RH25L75G}$ for the eight datasets.

Data-set	q	b	ϵ
Iris	0.062	0.185	0.298
Ionosphere	0.010	0.409	0.051
Wine	0.460	0.469	0.141
Bank-Note-Authentication	0.360	0.419	0.121
Haberman's-Survival	0.050	0.269	0.031
Breast-Cancer-Wisconsin	0.170	0.460	0.050
Statlog (Heart)	0.470	0.489	0.030
Seeds	0.050	0.189	0.161

Table 5 Tuned hyperparameters for ChaosFEX$_{RH50L50G}$ for the eight datasets.

Data-set	q	b	ϵ
Iris	0.050	0.359	0.221
Ionosphere	0.099	0.479	0.061
Wine	0.460	0.469	0.131
Bank-Note-Authentication	0.090	0.289	0.041
Haberman's-Survival	0.140	0.489	0.021
Breast-Cancer-Wisconsin	0.069	0.139	0.041
Statlog (Heart)	0.180	0.169	0.011
Seeds	0.050	0.139	0.151

Macro F1 scores obtained for various datasets with ChaosFEX$_{RH25L75G}$ and ChaosFEX$_{RH25L75G}$+SVM are reported in Table 7. For *Haberman's Survival* dataset, we achieved an improved macro F1 score of 0.73 compared to the best F1 score reported in earlier works (Sethi *et al.* (2023); AS *et al.* (2023)). Macro

F1 score of *Statlog (Heart)* dataset is also increased to 0.84 with ChaosFEX$_{RH25L75G}$+SVM classifier.

Table 8 gives the macro F1 score obtained with ChaosFEX$_{RH50L50G}$ and ChaosFEX$_{RH50L50G}$+SVM. Table 9 gives the macro F1 score obtained with ChaosFEX$_{RH75L25G}$ and

Table 6 Tuned hyperparameters for ChaosFEX$_{RH75L25G}$ for the eight datasets.

Data-set	q	b	ϵ
Iris	0.15	0.299	0.231
Ionosphere	0.02	0.219	0.809
Wine	0.47	0.479	0.131
Bank-Note-Authentication	0.01	0.259	0.071
Haberman's-Survival	0.23	0.1	0.011
Breast-Cancer-Wisconsin	0.14	0.489	0.021
Statlog (Heart)	0.13	0.1	0.051
Seeds	0.05	0.189	0.151

ChaosFEX$_{RH75L25G}$+SVM.

Table 7 Macro F1 scores reported for ChaosFEX$_{RH25L75G}$ and ChaosFEX$_{RH25L75G}$+SVM.

Data-set	$ChaosFEX_{RH25L75G}$	$ChaosFEX_{RH25L75G}$+SVM
Iris	1	1
Ionosphere	0.6	0.88
Wine	0.6	0.94
Bank-Note-Authentication	0.75	0.9
Haberman's-Survival	0.73	0.56
Breast-Cancer-Wisconsin	0.85	0.98
Statlog (Heart)	0.77	0.84
Seeds	0.81	0.84

Random Heterogenous Neurochaos Learning architectures which incorporate ChaosFEX features with other ML classifiers such as AdaBoost (AB), Decision Trees (DT), k-NN, Gaussian Naive Bayes (GNB), and Random Forests (RF) are implemented and the results indicate that *randomness* and *heterogeneity* introduced in the NL architectures yields superior performance when compared with homogeneous or fixed heterogeneous structures.

The macro F1 scores obtained for $ChaosFEX_{RH25L75G}$+AdaBoost, $ChaosFEX_{RH50L50G}$+AdaBoost and $ChaosFEX_{RH75L25G}$+AdaBoost structures are given in Table 10. Accuracy of 100% is obtained for *Wine* dataset with $ChaosFEX_{RH50L50G}$+AdaBoost architecture. Macro F1-score = 0.99 is successfully achieved for *Bank Note authentication* data-set for both $ChaosFEX_{RH50L50G}$+AdaBoost and $ChaosFEX_{RH75L25G}$+AdaBoost. High F1-score = 0.99

Table 8 Macro F1 scores reported for ChaosFEX$_{RH50L50G}$ and ChaosFEX$_{RH50L50G}$+SVM.

Data-set Name	$ChaosFEX_{RH50L50G}$	$ChaosFEX_{RH50L50G}$+SVM
Iris	1	1
Ionosphere	0.58	0.9
Wine	0.59	0.94
Bank-Note-Authentication	0.59	0.72
Haberman's-Survival	0.68	0.47
Breast-Cancer-Wisconsin	0.77	0.92
Statlog (Heart)	0.78	0.79
Seeds	0.72	0.81

Table 9 Macro F1 scores reported for ChaosFEX$_{RH75L25G}$ and ChaosFEX$_{RH75L25G}$+SVM.

Data-set Name	$ChaosFEX_{RH75L25G}$	$ChaosFEX_{RH75L25G}$+SVM
Iris	1	0.97
Ionosphere	0.71	0.94
Wine	0.63	0.97
Bank-Note-Authentication	0.65	0.84
Haberman's-Survival	0.6	0.51
Breast-Cancer-Wisconsin	0.79	0.94
Statlog (Heart)	0.65	0.85
Seeds	0.78	0.86

is also achieved for *Breast Cancer Wisconsin* data-set when $ChaosFEX_{RH75L25G}$+AdaBoost is implemented.

The macro F1 scores obtained for $ChaosFEX_{RH25L75G}$+Decision Trees, $ChaosFEX_{RH50L50G}$+Decision Trees and $ChaosFEX_{RH75L25G}$+Decision Trees can be found in Table 11. High F1 score = 0.98 for *Breast Cancer Wisconsin* data-set with $ChaosFEX_{RH25L75G}$+Decision Trees and $ChaosFEX_{RH75L25G}$+Decision Trees has been achieved.

The macro F1 scores obtained for $ChaosFEX_{RH25L75G}$+GNB, $ChaosFEX_{RH50L50G}$+GNB and $ChaosFEX_{RH75L25G}$+GNB can be found in Table 13. The macro F1 scores obtained for $ChaosFEX_{RH25L75G}$+kNN, $ChaosFEX_{RH50L50G}$+kNN and $ChaosFEX_{RH75L25G}$+kNN is seen in Table 12.

The macro F1 scores obtained for $ChaosFEX_{RH25L75G}$+Random Forests, $ChaosFEX_{RH50L50G}$+Random Forests and $ChaosFEX_{RH75L25G}$+Random Forests can be found in Table 14. High performance is obtained for *Breast Cancer Wisconsin* dataset using $ChaosFEX_{RH25L75G}$+Random Forests with F1 score of 0.98.

Table 10 Macro F1 scores obtained for different ChaosFEX$_{RH}$+AdaBoost architectures. **Bold** fonts indicate the highest F1 score achieved for the respective dataset.

Data-set Name	$ChaosFEX_{RH25L75G}$+AB	$ChaosFEX_{RH50L50G}$+AB	$ChaosFEX_{RH75L25G}$+AB
Iris	**1**	**1**	0.967
Ionosphere	**0.97**	**0.97**	**0.97**
Wine	0.97	**1**	0.944
Bank-Note-Authentication	0.93	**0.99**	0.989
Haberman's-Survival	0.5	0.56	**0.66**
Breast-Cancer-Wisconsin	0.98	0.98	**0.99**
Statlog (Heart)	0.81	0.85	**0.88**
Seeds	**0.86**	0.77	0.73

Table 11 Macro F1 score obtained for different ChaosFEX$_{RH}$+Decision Trees architectures. **Bold** fonts indicate the highest F1 score achieved for the respective dataset.

Data-set Name	$ChaosFEX_{RH25L75G}$+DT	$ChaosFEX_{RH50L50G}$+DT	$ChaosFEX_{RH75L25G}$+DT
Iris	**1**	0.97	0.97
Ionosphere	0.92	0.91	**0.97**
Wine	**0.95**	0.94	**0.95**
Bank-Note-Authentication	**0.95**	0.90	0.89
Haberman's-Survival	0.60	**0.65**	0.63
Breast-Cancer-Wisconsin	**0.98**	0.97	**0.98**
Statlog (Heart)	**0.92**	0.84	0.86
Seeds	**0.81**	**0.81**	0.76

Table 12 Macro F1-scores obtained for different ChaosFEX$_{RH}$+kNN architectures. **Bold** fonts indicate the highest F1 score achieved for the respective dataset.

Data-set Name	$ChaosFEX_{RH25L75G}$+kNN	$ChaosFEX_{RH50L50G}$+kNN	$ChaosFEX_{RH75L25G}$+kNN
Iris	**1**	**1**	**1**
Ionosphere	0.74	**0.85**	0.80
Wine	0.66	0.72	**0.77**
Bank-Note-Authentication	**0.93**	0.83	0.89
Haberman's-Survival	**0.64**	0.61	0.61
Breast-Cancer-Wisconsin	**0.98**	0.93	0.94
Statlog (Heart)	0.60	**0.81**	0.78
Seeds	0.76	0.70	**0.79**

When compared with earlier architectures which were either homogeneous NL Sethi *et al.* (2023) or heterogeneous NL but with fixed structure (odd-even) AS *et al.* (2023), we report that RHNL yields either comparable or superior classification performance.

For ease of comparison, we summarize these results in Table 15. Macro F1 scores obtained with RHNL are among the best for the various datasets considered in our study.

Table 13 Macro F1-scores obtained for different ChaosFEX$_{RH}$+GNB architectures. **Bold** fonts indicate the highest F1-score achieved for the respective dataset.

Data-set Name	$ChaosFEX_{RH25L75G}$+GNB	$ChaosFEX_{RH50L50G}$+GNB	$ChaosFEX_{RH75L25G}$+GNB
Iris	**1**	**1** 1	0.97
Ionosphere	0.83	0.83	**0.91**
Wine	**0.94**	**0.94**	**0.94**
Bank-Note-Authentication	**0.73**	0.67	0.70
Haberman's-Survival	**0.62**	0.61	0.52
Breast-Cancer-Wisconsin	**0.94**	0.89	0.91
Statlog (Heart)	0.77	**0.81**	0.74
Seeds	**0.72**	0.63	0.70

Table 14 Macro F1 scores obtained for all datasets using ChaosFEX$_{RH}$+RF architectures. **Bold** fonts indicate the highest F1 score achieved for the respective dataset.

Data-set Name	$ChaosFEX_{RH25L75G}$+RF	$ChaosFEX_{RH50L50G}$+RF	$ChaosFEX_{RH75L25G}$+RF
Iris	**1**	**1**	0.97
Ionosphere	0.96	0.93	**0.97**
Wine	**0.97**	**0.97**	**0.97**
Bank-Note-Authentication	0.93	0.92	**0.94**
Haberman's-Survival	**0.66**	0.57	0.59
Breast-Cancer-Wisconsin	0.98	**0.99**	0.97
Statlog (Heart)	0.86	**0.87**	0.71
Seeds	**0.83**	0.76	0.78

Results Obtained for Time Series Dataset

We analysed the performance of our proposed RHNL architectures with a time series dataset – namely *Free Spoken Digit Dataset* (FSDD). The hyperparameters tuned are given in Tables 33, 34, 35 and 36 (Appendix). Macro F1 scores obtained for various $ChaosFEX_{RHNL}$ architectures are seen in Figures 3, 4 and 5 (respectively).

Classification performance of $ChaosFEX_{RHNL}$ for Debris Scars and Urban Images

Satellites images are processed to detect and estimate vulnerability of human settlements. Machine Learning algorithms are used now a days to identify areas with high risk of landslide Sridharan *et al.* (2020). Around 85 images of debris scars and urban settlements from five Asian countries (India, Nepal, Japan, Taiwan and China) were obtained from Planet labs Planet Labs Inc. (Accessed: 2019-10-09) imagery with $3 - 5m$ resolution for our analysis to identify the classification performance of Neurochaos Learning Architecture (NL), specifically ChaosFEX$_{RH25L75G}$ algorithm. Images are labelled either as "debris" or as "urban" based on visual recognition. We have used 35 debris scar images and 50 urban settlement images for our analysis. Figure 6(a) and Figure 6(b) shows the sample images from class "debris" and class "urban". Out of total 85 satellite images captured, 80% of the data are used for training
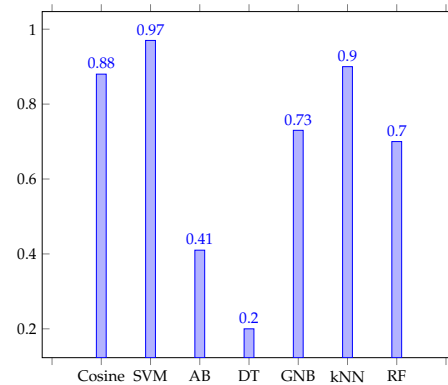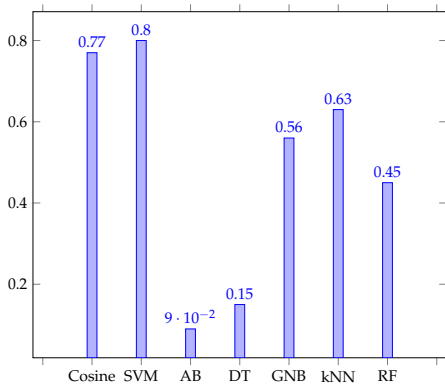

Figure 3 Macro F1 scores obtained for *FSDD* data set for ChaosFEX$_{RH25L75G}$ with various classifiers. Classifiers are labeled along the x-axis.

and the remaining 20% are used for testing. We have used $5-$fold cross validation for our analysis.

The images are initially pre-processed using Otsu global thresholding algorithm and then filtered using Discrete Wavelet Transform(DWT). Daubechies-4 wavelets are used for the DWT im-

Table 15 Comparison of best macro F1-scores obtained for RHNL structures (proposed in this study) with the other architectures reported in AS *et al.* (2023). Best macro F1 scores are highlighted in **bold** font.

Data-set	Best macro F1 score	RHNL architectures with best macro F1-scores		NL with best macro F1 scores reported in (AS *et al.* 2023)
Iris	**1**	$ChaosFEX_{RH25L75G}$, $ChaosFEX_{RH50L50G}$, $ChaosFEX_{RH75L25G}$, $ChaosFEX_{RH25L75G}$+SVM, $ChaosFEX_{RH50L50G}$+SVM, $ChaosFEX_{RH25L75G}$+AB, $ChaosFEX_{RH50L50G}$+AB, $ChaosFEX_{RH25L75G}$+DT, $ChaosFEX_{RH25L75G}$+kNN, $ChaosFEX_{RH25L75G}$+GNB, $ChaosFEX_{RH25L75G}$+RF, $ChaosFEX_{RH50L50G}$+RF, ChaosFEX$_{Hetero}$, ChaosFEX$_{Hetero}$+SVM	1	$ChaosFEX_{Logistic}$, $ChaosFEX_{Logistic}$+SVM,
Ionosphere	**0.97**	$ChaosFEX_{RH25L75G}$+AB, $ChaosFEX_{RH50L50G}$+AB, $ChaosFEX_{RH75L25G}$+AB, $ChaosFEX_{RH75L25G}$+DT, $ChaosFEX_{RH75L25G}$+RF	**0.97**	$ChaosFEX_{Logistic}$+SVM
Wine	**1**	$ChaosFEX_{RH50L50G}$+AB	0.98	$ChaosFEX_{GLS}$
Bank-Note-Authentication	**0.99**	$ChaosFEX_{RH50L50G}$+AB, $ChaosFEX_{RH75L25G}$+AB	0.96	$ChaosFEX_{Logistic}$+SVM
Haberman's-Survival	**0.73**	$ChaosFEX_{RH25L75G}$	0.72	$ChaosFEX_{Hetero}$
Breast-Cancer-Wisconsin	**0.99**	$ChaosFEX_{RH75L25G}$, $ChaosFEX_{RH50L50G}$+RF	0.97	$ChaosFEX_{Logistic}$+SVM
Statlog (Heart)	**0.92**	$ChaosFEX_{RH25L75G}$+DT	0.89	$ChaosFEX_{Logistic}$+SVM
Seeds	0.86	$ChaosFEX_{RH25L75G}$+AdaBoost, $ChaosFEX_{RH75L25G}$+SVM	**0.86**	$ChaosFEX_{Hetero}$

Figure 4 Macro F1 scores obtained for *FSDD* data set for ChaosFEX$_{RH50L50G}$ with various classifiers. Classifiers are labeled along the x-axis.
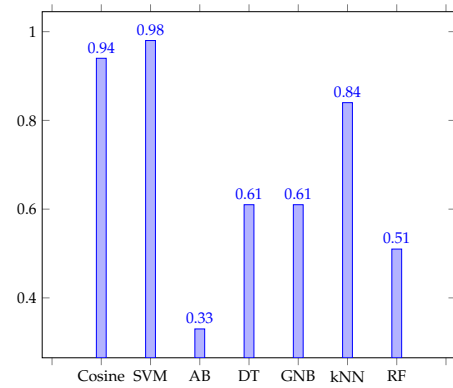
Figure 5 Macro F1 scores obtained for *FSDD* data set for ChaosFEX$_{RH75L25G}$ with various classifiers. Classifiers are labeled along the x-axis.

Figure 6 (a) Debris scar image (class "debris"). (b) Urban settlement image (class "urban").

plementation. Grey-level co-occurrence matrix (GLCM) is then created which analyses the pairs of horizontally adjacent pixels in a scaled version of the image. From the GLCM matrix, 12 features namely Contrast, Correlation, Energy, Homogeneity, Mean, Standard Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis and Skewness are extracted. These features are fed to various $ChaosFEX_{RHNL}$.

Parameters tuned for various structures considered for analysis with debris-urban dataset are shown in Appendix Tables 37, 38 and 39.

Performance analysis is done and the macro F1 score obtained for the various architectures considered are given in Table 16. We analysed the structures and found that high F1 score of 0.94 is obtained with $ChaosFEX_{RH25L75G}$, $ChaosFEX_{RH25L75G}$ +SVM, $ChaosFEX_{RH25L75G}$ +kNN, $ChaosFEX_{RH25L75G}$ +DT and $ChaosFEX_{RH25L75G}$+RF for the debris-urban dataset considered.

■ **Table 16** Macro F1 scores for debris-urban dataset using $ChaosFEX_{RHNL}$ with various classifiers (cosine similarity and other ML classifiers).

Classifier	CosSim	SVM	k-NN	AB	DT	GNB	RF
$ChaosFEX_{RH25L75G}$	0.94	0.94	0.94	0.70	0.94	0.88	0.94
$ChaosFEX_{RH50L50G}$	0.70	0.70	0.74	0.81	0.82	0.80	0.88
$ChaosFEX_{RH75L25G}$	0.66	0.94	0.70	0.88	0.88	0.87	0.88

Classification performance of $ChaosFEX_{RHNL}$ for Brain Tumor Dataset

Brain tumors are the leading cause of cancer death in children. They are caused by the abnormal and uncontrolled growth of cells inside the brain or spinal canal. Classification of brain tumors using machine learning technology is very relevant for radiologists to confirm their analysis more effectively and quickly.

For our analysis, we have considered 100 MRI brain images from *Kaggle* online repository Chakrabarty (Accesed:2019-10-09). Images are labelled as "malignant" or "benign". We considered 40 malignant and benign images for our analysis. We split 80% of data for training and the remaining 20% for testing. Five-fold cross validation is adopted in this analysis. Figure 7(a) and Figure 7(b) shows sample images from each of the two classes (malignant and benign).

Preprocessing is done by applying anisotropic filtering of all images. Grey-level co-occurrence matrix (GLCM) is then created which analyses the pairs of horizontally adjacent pixels in a scaled version of the image. From the GLCM matrix, we can calculate the features needed for classification. The twelve features generated are *Contrast, Correlation, Energy, Homogeneity, Mean, Standard Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis* and
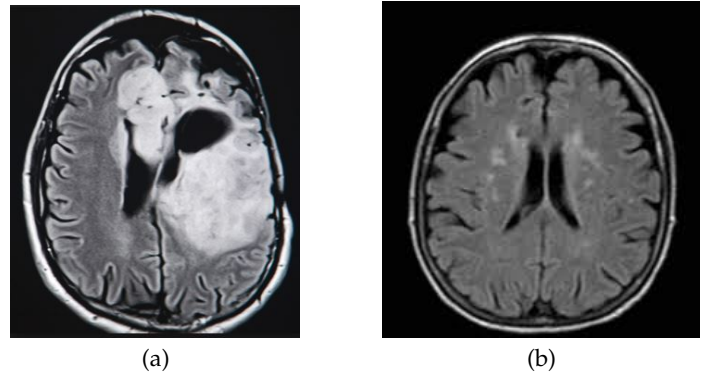
Figure 7 (a) MRI image showing malignant brain tumour (class "malignant"). (b) MRI image showing benign brain tumour (class "benign").

Skewness. These extracted features are subsequently fed to stand-alone SVM, stand-alone k-NN, and $ChaosFEX_{RH25L75G}$ +SVM and $ChaosFEX_{RH25L75G}$ +k-NN. The hyperparameters tuned for various architectures under analysis for brain tumor dataset is given in Appendix Tables 40, 41 and 42.

■ **Table 17** Macro F1 scores for MRI brain tumor dataset using $ChaosFEX_{RHNL}$ with various classifiers (cosine similarity and other ML classifiers).

Classifier	CosSim	SVM	k-NN	AB	DT	GNB	RF
$ChaosFEX_{RH25L75G}$	0.88	0.81	0.72	0.76	0.81	0.48	0.83
$ChaosFEX_{RH50L50G}$	0.44	0.50	0.69	0.78	0.73	0.78	0.73
$ChaosFEX_{RH75L25G}$	0.83	0.78	0.76	0.82	0.60	0.50	0.73

The results (Table 17) show classification performance is better with $ChaosFEX_{RH25L75G}$ (F1 score = 0.881) for brain tumor dataset. We may further improve the classification performance with other $ChaosFEX_{RHNL}$ architectures with properly tuned hyperparameters.

Performance of $ChaosFEX_{RHNL}$ compared with Stand-alone ML Classifiers

When compared with either homogeneous NL or heterogeneous NL but with fixed structure (odd-even placement of GLS and Logistic neurons), we have reported that RHNL yields either comparable or superior classification performance. Table 18 compares the highest F1 score obtained with various stand-alone ML classifiers (SA-ML) and RHNL architectures for all the 11 datasets in this study. For *Iris, Wine* and *Bank Note Authentication* datasets, RHNL architectures perform equally well with some of the stand-alone ML classifiers. It is interesting to note that a 16% increase in performance is achieved for *Haberman's Survival* dataset with our proposed RHNL structure when compared with GNB which gives the best F1 score among all the stand-alone ML classifiers (F1-score = 0.57). Also for *Ionosphere, Breast Cancer Wisconsin* and *Statlog (Heart)* datasets, our newly proposed RHNL structures outperform stand-alone classifiers significantly well. However for *Seeds* dataset alone, RHNL gives a lower performance. Table 19 shows that $ChaosFEX_{RH25L75G}$, $ChaosFEX_{RH25L75G}$+SVM, $ChaosFEX_{RH25L75G}$+kNN, $ChaosFEX_{RH25L75G}$+DT and $ChaosFEX_{RH25L75G}$+RF give higher performance (F1 Score

= 0.94) than any of the stand-alone ML classifiers for the debris-urban dataset. For brain tumor dataset also $ChaosFEX_{RH25L75G}$ outperforms all the stand-alone ML classifiers.

Dataset	Best F1 (SA-ML)	SA-ML	Best F1 (RHNL)
Iris	1.00	RF, k-NN	1.00
Ionosphere	0.96	SVM	0.97
Wine	1.00	GNB	1.00
Bank Note	0.99	SVM, k-NN	0.99
Haberman's	0.57	GNB	0.73
Breast Cancer	0.95	k-NN	0.99
Statlog (Heart)	0.84	k-NN, SVM	0.92
Seeds	0.92	k-NN	0.86
FSDD	0.97	RF	0.98

Dataset	SA-SVM	SA-kNN	SA-AB	SA-DT	SA-GNB	SA-RF	$ChaosFEX_{RH25L75G}$
Debris-Urban	0.71	0.71	0.71	0.71	0.71	0.82	**0.94**
Brain Tumor	0.70	0.60	0.70	0.70	0.55	0.75	**0.88**

Table 20 shows that highest macro F1 score is obtained for $ChaosFEX_{RHNL}$ structures for all dataset except *Seed*.

Performance of RHNL in low training sample regime

One of the major significance of Neurochaos Learning architectures is that they perform well in the low training sample regime (Balakrishnan *et al.* 2019). We analysed the performance of ChaosFEX$_{RH25L75G}$ for the *MRI brain tumor* dataset in the low training sample regime and compared its performance with the standalone ML classifiers. The 12 features generated from the selected 100 brain tumor MRI images from *Kaggle* dataset online repository (Chakrabarty Accesed:2019-10-09) are used for our analysis. Analysis is done starting with one sample per class in the training set and the remaining samples were used for testing. The process is repeated with 2 to 15 samples per class in train data set and the remaining samples in test dataset. In every case, we did 10 independent random trials. The average F1 score of these 10 trials are reported in each case and compared with the results obtained for stand-alone classifiers namely Decision Tree (DT), Random Forest (RF), AdaBoost (AB), SVM, k-NN, Guassian Naive Bayes (GNB) and ChaosFEX$_{RH25L75G}$. In ChaosFEX$_{RH25L75G}$, 25% of the locations are randomly allotted to logistic map neurons and the remaining locations with GLS neurons. Cosine Similarity classifier is used in ChaosFEX$_{RH25L75G}$ architecture. Figure 8 shows that

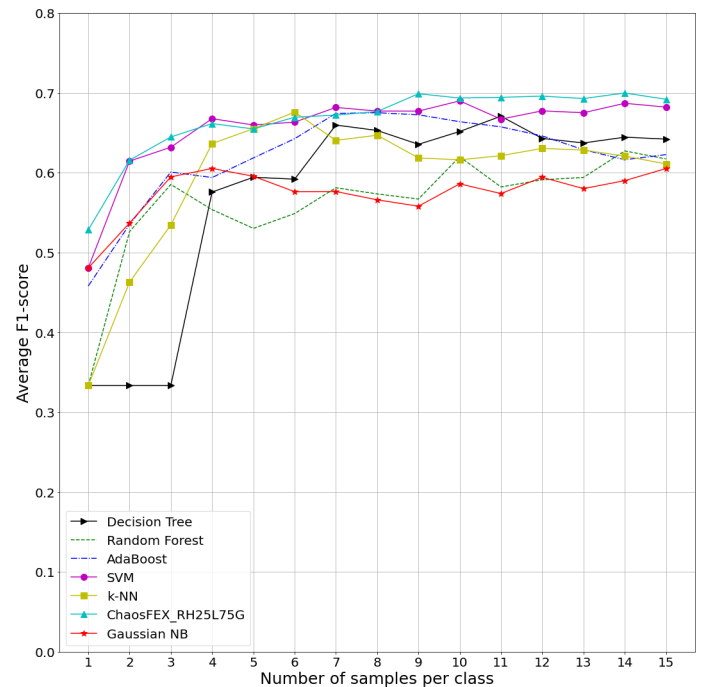| Dataset	Best F1 Score	SA-ML	RHNL+ML
Iris	1.00	✓	✓
Ionosphere	0.96		✓
Wine	1.00	✓	✓
Bank Note	0.99	✓	✓
Haberman's	0.73		✓
Breast Cancer	0.99		✓
Statlog (Heart)	0.92		✓
Seed	0.86	✓	
FSDD	0.98		✓
Debris-Urban	0.94		✓
Brain Tumor	0.83		✓

Figure 8 Comparative performance of ChaosFEX$_{RHNL25L75G}$ with stand-alone ML classifiers in the regime of low number of training instances for the brain tumour image dataset.

for low training samples, ChaosFEX$_{RH25L75G}$ consistently high performance with respect to other standalone ML algorithms.

For training dataset with one sample per class, $ChaosFEX_{RHNL25L75G}$ gives high performance and Decision Tree and K-NN gives low performance. For training dataset with 4 and 5 samples per class, SVM outperforms $ChaosFEX_{RH25L75G}$. k-NN gives high performance with 6 samples per class in training dataset. However from 8 samples per class in train dataset onwards, $ChaosFEX_{RHNL25L75G}$ performed well again and continues to outperform other classifiers. This shows that RHNL architecture is able to learn with very few training samples per class making it very desirable in practical applications where there is a paucity of training data.

CONCLUSION

Incorporation of both *heterogeneity* and *randomness* into Neurochaos Learning is one step in the right direction of mimicking the complex neural organization of the human brain. In this study, we find that such Random Heterogeneous Neurochaos Learning (RHNL) architectures perform very well. When compared with earlier architectures which were either homogeneous NL (Sethi *et al.* 2023) or heterogeneous NL but with fixed structure (odd-even) (AS *et al.* 2023), we report that RHNL yields either comparable or superior classification performance. Table 15 shows that the macro F1 scores obtained with RHNL are among the best for all the datasets considered in our study. RHNL architectures beat both homogeneous NL and fixed heterogeneous NL architectures on nearly all classification tasks. RHNL achieved a high level of F1 score for *Wine* dataset (1.0), *Bank Note Authentication* dataset (0.99), *Breast Cancer Wisconsin* dataset (0.99) and *FSDD* dataset (0.98). RHNL is also superior on time series dataset. For the *FSDD* dataset, $ChaosFEX_{RH75L25G}$+SVM achieves the highest macro F1 score of 0.98. This is infact higher than the best F1 score obtained with homogeneous NL architecture reported in (Sethi *et al.* 2023). These results of RHNL are clearly among the best in the literature on the same datasets. For the debris-urban image data, $ChaosFEX_{RH25L75G}$ gave a high F1 score of 0.94 with cosine similarity, SVM, k-NN, Decision Tree and Random Forest classifiers. Brain tumor image dataset gives high F1 score of 0.881 with $ChaosFEX_{RH25L75G}$. RHNL also yields best F1 score when compared with standalone-ML classifiers except for the *Seeds* dataset. In the case of low training sample regime, RHNL outperforms almost all traditional ML classifiers.

This line of work can be applied to classification of other datasets in various application domains. We will undertake extensive evaluation of the proposed architectures for image datasets in the future. For future research, we will explore incorporating other chaotic maps such as Standard map, Circle map, Ikeda map, Hénon map, Gumowski-Mira map, Arnold's cat map, Lorenz system, Baker's map, Lozi map, Hindmarsh-Rose neuronal model, Rössler system and other such dynamical systems as neurons in RHNL.

APPENDIX

This section contains the additional supplementary details related to the main manuscript. It contains the hyperparameter tuned values for each dataset used in this study, for different classifiers namely AdaBoost, Decision Trees, kNN and Random Forests that were used in $ChaosFEX_{RH}$ architectures. The hyperparameters tuned for *FSSD* dataset for various $ChaosFEX_{RH}$ architectures are also included in this section.

■ **Table 21** Tuned hyperparameters for $ChaosFEX_{RH25L75G}$ + AdaBoost.

Dataset	q	b	ϵ	$n\_estimators$
Iris	0.062	0.185	0.298	10
Ionosphere	0.01	0.409	0.051	50
Wine	0.46	0.469	0.141	10
Bank Note	0.36	0.419	0.121	10
Haberman's	0.05	0.269	0.031	5000
Breast Cancer	0.17	0.46	0.05	100
Statlog (Heart)	0.47	0.489	0.0309	5000
Seeds	0.05	0.189	0.161	50

■ **Table 22** Tuned hyperparameters for $ChaosFEX_{RH50L50G}$ + AdaBoost.

Dataset	q	b	ϵ	$n\_estimators$
Iris	0.05	0.359	0.221	10
Ionosphere	0.099	0.479	0.061	1000
Wine	0.46	0.469	0.131	10
Bank Note	0.09	0.289	0.041	1000
Haberman's	0.14	0.489	0.021	1000
Breast Cancer	0.069	0.139	0.041	10
Statlog (Heart)	0.18	0.169	0.011	100
Seeds	0.05	0.139	0.151	10

■ **Table 23** Tuned hyperparameters for $ChaosFEX_{RH75L25G}$ + AdaBoost.

Dataset	q	b	ϵ	$n\_estimators$
Iris	0.15	0.299	0.231	10
Ionosphere	0.02	0.219	0.809	50
Wine	0.47	0.479	0.131	10
Bank Note	0.01	0.259	0.071	5000
Haberman's	0.23	0.1	0.011	5000
Breast Cancer	0.14	0.489	0.021	1000
Statlog (Heart)	0.13	0.1	0.051	10
Seeds	0.05	0.189	0.151	50

ACKNOWLEDGEMENTS

Table 24 Tuned hyperparameters for $ChaosFEX_{RH25L75G}$ + Decision Trees.

Dataset	q	b	ϵ	min_samples_leaf	max_depth	ccp_alpha
Iris	0.062	0.185	0.298	1	2	0.0
Ionosphere	0.01	0.409	0.051	3	7	0.0
Wine	0.46	0.469	0.141	1	4	0.0
Bank Note	0.36	0.419	0.121	1	6	0.0
Haberman's	0.05	0.269	0.031	7	4	0.0
Breast Cancer	0.17	0.46	0.05	10	4	0.0022
Statlog (Heart)	0.47	0.489	0.0309	1	3	0.0
Seeds	0.05	0.189	0.161	2	3	0.0

Table 25 Tuned hyperparameters for $ChaosFEX_{RH50L50G}$ + Decision Trees.

Dataset	q	b	ϵ	min_samples_leaf	max_depth	ccp_alpha
Iris	0.05	0.359	0.221	1	2	0.0
Ionosphere	0.099	0.479	0.061	3	4	0.0
Wine	0.46	0.469	0.131	1	4	0.0
Bank Note	0.09	0.289	0.041	1	8	0.0
Haberman's	0.14	0.489	0.021	4	5	0.0
Breast Cancer	0.069	0.139	0.041	3	3	0.0
Statlog (Heart)	0.18	0.169	0.011	1	4	0.0
Seeds	0.05	0.139	0.151	1	3	0.0

Table 26 Tuned hyperparameters for $ChaosFEX_{RH75L25G}$ + Decision Trees.

Dataset	q	b	ϵ	min_samples_leaf	max_depth	ccp_alpha
Iris	0.15	0.299	0.231	1	2	0.0
Ionosphere	0.02	0.219	0.809	1	6	0.0
Wine	0.47	0.479	0.131	2	4	0.0
Bank Note	0.01	0.259	0.071	8	7	0.0
Haberman's	0.23	0.1	0.011	5	6	0.0
Breast Cancer	0.14	0.489	0.021	6	9	0.00218
Statlog (Heart)	0.13	0.1	0.051	1	3	0.004
Seeds	0.05	0.189	0.151	1	4	0.0

Table 27 Tuned hyperparameters for $ChaosFEX_{RH25L75G}$ + kNN.

Dataset	q	b	ϵ	k
Iris	0.062	0.185	0.298	3
Ionosphere	0.01	0.409	0.051	3
Wine	0.46	0.469	0.141	5
Bank Note	0.36	0.419	0.121	1
Haberman's	0.05	0.269	0.031	5
Breast Cancer	0.17	0.46	0.05	5
Statlog (Heart)	0.47	0.489	0.0309	3
Seeds	0.05	0.189	0.161	3

Table 28 Tuned hyperparameters for $ChaosFEX_{RH50L50G}$ + kNN.

Dataset	q	b	ϵ	k
Iris	0.05	0.359	0.221	3
Ionosphere	0.099	0.479	0.061	3
Wine	0.46	0.469	0.131	1
Bank Note	0.09	0.289	0.041	1
Haberman's	0.14	0.489	0.021	5
Breast Cancer	0.069	0.139	0.041	3
Statlog (Heart)	0.18	0.169	0.011	3
Seeds	0.05	0.139	0.151	3

Table 29 Tuned hyperparameters for $ChaosFEX_{RH75L25G}$ + kNN.

Dataset	q	b	ϵ	k
Iris	0.15	0.299	0.231	5
Ionosphere	0.02	0.219	0.809	5
Wine	0.47	0.479	0.131	5
Bank Note	0.01	0.259	0.071	1
Haberman's	0.23	0.1	0.011	1
Breast Cancer	0.14	0.489	0.021	1
Statlog (Heart)	0.13	0.1	0.051	5
Seeds	0.05	0.189	0.151	3

Table 30 Tuned hyperparameters for $ChaosFEX_{RH25L75G}$ + Random Forests.

Dataset	q	b	ϵ	$n\_estimators$	max_depth
Iris	0.062	0.185	0.298	10	3
Ionosphere	0.01	0.409	0.051	100	5
Wine	0.46	0.469	0.141	100	3
Bank Note	0.36	0.419	0.121	10	4
Haberman's	0.05	0.269	0.031	10	6
Breast Cancer	0.17	0.46	0.05	10	5
Statlog (Heart)	0.47	0.489	0.0309	10	4
Seeds	0.05	0.189	0.161	100	4

Table 31 Tuned hyperparameters for $ChaosFEX_{RH50L50G}$ + Random Forests.

Dataset	q	b	ϵ	$n\_estimators$	max_depth
Iris	0.05	0.359	0.221	100	3
Ionosphere	0.099	0.479	0.061	1000	8
Wine	0.46	0.469	0.131	1000	5
Bank Note	0.09	0.289	0.041	100	7
Haberman's	0.14	0.489	0.021	1000	4
Breast Cancer	0.069	0.139	0.041	10	8
Statlog (Heart)	0.18	0.169	0.011	100	5
Seeds	0.05	0.139	0.151	100	4

Table 32 Tuned hyperparameters for $ChaosFEX_{RH75L25G}$ + Random Forests.

Dataset	q	b	ϵ	$n\_estimators$	max_depth
Iris	0.15	0.299	0.231	10	2
Ionosphere	0.02	0.219	0.809	10	5
Wine	0.47	0.479	0.131	100	6
Bank Note	0.01	0.259	0.071	100	7
Haberman's	0.23	0.1	0.011	10	5
Breast Cancer	0.14	0.489	0.021	10	10
Statlog (Heart)	0.13	0.1	0.051	10	2
Seeds	0.05	0.189	0.151	100	4

■ **Table 33** Tuned hyperparameters for ChaosFEX$_{RH25L75G}$ for *FSDD*.

Hyper-parameter	Tuned Value
q	.086
b	.303
ϵ	.055

■ **Table 34** Tuned hyperparameters for ChaosFEX$_{RH50L50G}$ for *FSDD*.

Hyper-parameter	Tuned Value
q	.106
b	.032
ϵ	.104

■ **Table 35** Tuned hyperparameters for ChaosFEX$_{RH75L25G}$ for *FSDD*.

Hyper-parameter	Tuned Value
q	.40
b	.20
ϵ	.15

■ **Table 36** Tuned hyperparameters for various classifiers for *FSDD*.

Classifiers	ChaosFEX$_{RH25L75G}$	ChaosFEX$_{RH50L50G}$	ChaosFEX$_{RH75L25G}$
AdaBoost	$n\_estimators = 50$	$n\_estimators = 1$	$n\_estimators = 10$
Decision Trees	$min\_samples\_leaf = 1$	$min\_samples\_leaf = 1$	$min\_samples\_leaf = 4$
	$max\_depth = 2$	$max\_depth = 2$	$max\_depth = 7$
	$ccp\_alpha = 0.0074$	$ccp\_alpha = 0.0$	$ccp\_alpha = 0.00723$
k-NN	$k = 3$	$k = 5$	$k = 1$
Random Forests	$n\_estimators = 2$	$n\_estimators = 2$	$n\_estimators = 2$
	$max\_depth = 1000$	$max\_depth = 1000$	$max\_depth = 10$

■ **Table 37** Parameters tuned for various $ChaosFEX_{RH25L75G}$ structures considered for analysis with debris-urban dataset

Algorithm	Hyper Parameters
$ChaosFEX_{RH25L75G}$	q=.3649
	b=.430
	ϵ=.259
$ChaosFEX_{RH25L75G}$+SVM	q=.3649
	b=.430
	ϵ=.259
$ChaosFEX_{RH25L75G}$+k-NN	q=.3649
	b=.430
	ϵ=.259
	k=3
$ChaosFEX_{RH25L75G}$+AdaBoost	q=.3649
	b=.430
	ϵ=.259
	$n\_estimator$=1
$ChaosFEX_{RH25L75G}$+Decision Tree	q=.3649
	b=.430
	ϵ=.259
	$min\_samples\_leaf = 1$
	$random\_state = 42$
	$max\_depth = 6$
	$ccp\_alpha = 0$
$ChaosFEX_{RH25L75G}$+GNB	q=.3649
	b=.430
	ϵ=.259
$ChaosFEX_{RH25L75G}$+RF	q=.3649
	b=.430
	ϵ=.259
	$n\_estimators = 1000,$
	max_depth= 8

■ **Table 38** Parameters tuned for various $ChaosFEX_{RH50L50G}$ structures considered for analysis with debris-urban dataset

Algorithm	Hyper Parameters
$ChaosFEX_{RHNL50L50G}$	$q = .121$
	$b = .0041$
	$\epsilon = .015$
$ChaosFEX_{RHNL50L50G}$+SVM	$q = .121$
	$b = .0041$
	$\epsilon = .015$
$ChaosFEX_{RHNL50L50G}$+k-NN	$q = .121$
	$b = .0041$
	$\epsilon = .015$
	$k = 1$
$ChaosFEX_{RHNL50L50G}$+Decision Tree	$q = .121$
	$b = .0041$
	$\epsilon = .015$
	$min\_samples\_leaf = 1$
	$max\_depth = 4$
	$ccp\_alpha = 0$
$ChaosFEX_{RHNL50L50G}$+GNB	$q = .121$
	$b = .0041$
	$\epsilon = .015$
$ChaosFEX_{RHNL50L50G}$+AdaBoost	$q = .121$
	$b = .0041$
	$\epsilon = .015$
	$n\_estimators = 3$
$ChaosFEX_{RHNL50L50G}$+Random Forest	$q = .121$
	$b = .0041$
	$\epsilon = .015$
	$n\_estimators = 1000$
	$max\_depth = 5$

■ **Table 39** Parameters tuned for various $ChaosFEX_{RH75L25G}$ structures considered for analysis with debris-urban dataset

Algorithm	Hyper Parameters
$ChaosFEX_{RHNL75L25G}$	$q = .491$
	$b = .010$
	$\epsilon = .0856$
$ChaosFEX_{RHNL75L25G}$+SVM	$q = .491$
	$b = .010$
	$\epsilon = .0856$
$ChaosFEX_{RHNL75L25G}$+k-NN	$q = .491$
	$b = .010$
	$\epsilon = .0856$
	$k = 5$
$ChaosFEX_{RHNL75L25G}$+Decision Tree	$q = .491$
	$b = .010$
	$\epsilon = .0856$
	$random\_state = 42$
	$min\_samples\_leaf = 1$
	$max\_depth = 4$
	$ccp\_alpha = 0$
$ChaosFEX_{RHNL75L25G}$+GNB	$q = .491$
	$b = .010$
	$\epsilon = .0856$
$ChaosFEX_{RHNL75L25G}$+AdaBoost	$q = .491$
	$b = .010$
	$\epsilon = .0856$
	$n\_estimators = 100$
$ChaosFEX_{RHNL75L25G}$+Random Forest	$q = .491$
	$b = .010$
	$\epsilon = .0856$
	$n\_estimators = 100$
	$max\_depth = 4$

Table 40 Parameters tuned for various $ChaosFEX_{RH25L50G}$ structures considered for analysis with Brain Tumor dataset

Algorithm	Parameters
$ChaosFEX_{RH25L75G}$	$q=.01$
	$b=.36$
	$\epsilon=.090$
$ChaosFEX_{RH25L75G}$+SVM	$q=.01$
	$b=.36$
	$\epsilon=.090$
$ChaosFEX_{RH25L75G}$+k-NN	$q=.01$
	$b=.430$
	$\epsilon=.230$
	$k=3$
$ChaosFEX_{RH25L75G}$+Decision Tree	$q=.01$
	$b=.430$
	$\epsilon=.230$
	$min\_samples\_leaf = 1$
	$max\_depth = 6$
	$ccp\_alpha = 0$
$ChaosFEX_{RH25L75G}$+GNB	$q=.01$
	$b=.430$
	$\epsilon=.230$
$ChaosFEX_{RH25L75G}$+AdaBoost	$q=.01$
	$b=.430$
	$\epsilon=.230$
	$n\_estimator= 1000$
$ChaosFEX_{RH25L75G}$+Random Forest	$q=.01$
	$b=.430$
	$\epsilon=0.230$
	$n\_estimator= 1000$
	$max\_depth = 8$

Table 41 Parameters tuned for various $ChaosFEX_{RH50L50G}$ structures considered for analysis with Brain Tumor dataset

Algorithm	Hyper Parameters
$ChaosFEX_{RH50L50G}$	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
$ChaosFEX_{RH50L50G}$+SVM	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
$ChaosFEX_{RH50L50G}$+k-NN	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
	$k=1$
$ChaosFEX_{RH50L50G}$+Decision Tree	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
	$min\_samples\_leaf = 1$
	$max\_depth = 4$
	$ccp\_alpha = 0$
$ChaosFEX_{RH50L50G}$+GNB	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
$ChaosFEX_{RH50L50G}$+AdaBoost	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
	$n\_estimator= 3$
$ChaosFEX_{RH50L50G}$+Random Forest	$q = .094$
	$b = .0065$
	$\epsilon = .0092$
	$n\_estimator= 1000$
	$max\_depth = 5$

Table 42 Parameters tuned for various $ChaosFEX_{RH75L25G}$ structures considered for analysis with Brain Tumor dataset

Algorithm	Parameters
$ChaosFEX_{RH75L25G}$	$q = .156$
	$b = .107$
	$\epsilon = .059$
$ChaosFEX_{RH75L25G}$+SVM	$q = .156$
	$b = .107$
	$\epsilon = .059$
$ChaosFEX_{RH75L25G}$+k-NN	$q = .156$
	$b = .107$
	$\epsilon = .059$
	$k=5$
$ChaosFEX_{RH75L25G}$+Decision Tree	$q = .156$
	$b = .107$
	$\epsilon = .059$
	$min\_samples\_leaf = 10$
	$max\_depth = 4$
	$ccp\_alpha = 0.0$
$ChaosFEX_{RH75L25G}$+GNB	$q = .156$
	$b = .107$
	$\epsilon = .059$
$ChaosFEX_{RH75L25G}$+AdaBoost	$q = .156$
	$b = .107$
	$\epsilon = .059$
	$n\_estimator = 10$
$ChaosFEX_{RH75L25G}$+Random Forest	$q = .156$
	$b = .107$
	$\epsilon = .059$
	$n\_estimator = 100$
	$max\_depth = 4$

Vidyapeetham (Amritapuri Campus).

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

LITERATURE CITED

Aihara, K., T. Takabe, and M. Toyoda, 1990 Chaotic neural networks. Physics letters A **144**: 333–340.

Alligood, K. T., T. D. Sauer, J. A. Yorke, and D. Chillingworth, 1998 Chaos: an introduction to dynamical systems. SIAM Review **40**: 732–732.

AS, R. A., N. Harikrishnan, and N. Nagaraj, 2023 Analysis of logistic map based neurons in neurochaos learning architectures for data classification. Chaos, Solitons & Fractals **170**: 113347.

Asif, S., M. Zhao, F. Tang, and Y. Zhu, 2023 An enhanced deep learning method for multi-class brain tumor classification using deep transfer learning. Multimedia Tools and Applications pp. 1–28.

Balakrishnan, H. N., A. Kathpalia, S. Saha, and N. Nagaraj, 2019 Chaosnet: A chaos based artificial neural network architecture for classification. Chaos: An Interdisciplinary Journal of Nonlinear Science **29**: 113125.

Berrar, D., 2018 Bayes' theorem and naive bayes classifier. Encyclopedia of bioinformatics and computational biology: ABC of bioinformatics **403**: 412.

Boser, B. E., I. M. Guyon, and V. N. Vapnik, 1992 A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.

Breiman, L., 2001 Random forests. Machine learning **45**: 5–32.

Chakrabarty, N., Accesed:2019-10-09 Brain mri images for brain tumor detection. https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection/code.

Cover, T. and P. Hart, 1967 Nearest neighbor pattern classification. IEEE transactions on information theory **13**: 21–27.

Delahunt, C. B. and J. N. Kutz, 2019 Putting a bug in ml: The moth olfactory network learns to read mnist. Neural Networks **118**: 54–64.

Dua, D., C. Graff, *et al.*, 2017 Uci machine learning repository .

Fisher, R. A., 1936 The use of multiple measurements in taxonomic problems. Annals of eugenics **7**: 179–188.

Gillich, E. and V. Lohweg, 2010 Banknote authentication. 1. Jahreskolloquium Bild. Der Autom pp. 1–8.

Graves, A., A.-r. Mohamed, and G. Hinton, 2013 Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee.

Haberman, S. J., 1973 The analysis of residuals in cross-classified tables. Biometrics pp. 205–220.

Harikrishnan, J., A. Sudarsan, A. Sadashiv, and R. A. Ajai, 2019 Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision. In *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)*, pp. 1–5, IEEE.

Harikrishnan, N. and N. Nagaraj, 2020 Neurochaos inspired hybrid machine learning architecture for classification. In *2020 International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–5, IEEE.

Harikrishnan, N. and N. Nagaraj, 2021 When noise meets chaos: Stochastic resonance in neurochaos learning. Neural Networks **143**: 425–435.

Harikrishnan, N., R. Vinayakumar, and K. Soman, 2018 A machine learning approach towards phishing email detection. In *Proceedings of the Anti-Phishing Pilot at ACM International Work-*

shop on Security and Privacy Analytics (IWSPA AP), volume 2013, pp. 455–468.

Jackson, Z., C. Souza, J. Flaks, Y. Pan, H. Nicolas, *et al.*, 2018 Jakobovski/free-spoken-digit-dataset: v1. 0.8. Zenodo, August .

Korn, H. and P. Faure, 2003 Is there chaos in the brain? ii. experimental evidence and related models. Comptes rendus biologies **326**: 787–840.

Krishna, S. and A. R. Ajai, 2019 Analysis of three point checklist and abcd methods for the feature extraction of dermoscopic images to detect melanoma. In *2019 9th International Symposium on Embedded Computing and System Design (ISED)*, pp. 1–5, IEEE.

Nagaraj, N., 2022 The unreasonable effectiveness of the chaotic tent map in engineering applications. Chaos Theory and Applications **4**: 197–204.

NB, H., A. Kathpalia, and N. Nagaraj, 2022 Causality preserving chaotic transformation and classification using neurochaos learning. Advances in Neural Information Processing Systems **35**: 2046–2058.

Perez-Nieves, N., V. C. Leung, P. L. Dragotti, and D. F. Goodman, 2021 Neural heterogeneity promotes robust learning. Nature communications **12**: 5791.

Phatak, S. and S. S. Rao, 1995 Logistic map: A possible random-number generator. Physical review E **51**: 3670.

Planet Labs Inc., Accessed: 2019-10-09 Planet Imagery and Archive. https://www.planet.com.

Quinlan, J. R., 1986 Induction of decision trees. Machine learning **1**: 81–106.

Ramachandran, V., S. Blakeslee, and R. J. Dolan, 1998 Phantoms in the brain probing the mysteries of the human mind. Nature **396**: 639–640.

Remya Ajai, A. and S. Gopalan, 2020 Analysis of active contours without edge-based segmentation technique for brain tumor classification using svm and knn classifiers. In *Advances in Communication Systems and Networks: Select Proceedings of ComNet 2019*, pp. 1–10, Springer.

Schapire, R. E., 2013 Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pp. 37–52, Springer.

Sebe, N., 2005 *Machine learning in computer vision*, volume 29. Springer Science & Business Media.

Sethi, D., N. Nagaraj, and N. Harikrishnan, 2023 Neurochaos feature transformation for machine learning. Integration .

Sigillito, V. G., S. P. Wing, L. V. Hutton, and K. B. Baker, 1989 Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest **10**: 262–266.

Sridharan, A., R. A. AS, and S. Gopalan, 2020 A novel methodology for the classification of debris scars using discrete wavelet transform and support vector machine. Procedia computer science **171**: 609–616.

Street, W. N., W. H. Wolberg, and O. L. Mangasarian, 1993 Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pp. 861–870, SPIE.

Vandeginste, B., 1990 Parvus: An extendable package of programs for data exploration, classification and correlation, m. forina, r. leardi, c. armanino and s. lanteri, elsevier, amsterdam, 1988, price: Us $$$645 isbn 0-444-43012-1. Journal of Chemometrics **4**: 191–193.

Weis, S., M. Sonnberger, A. Dunzinger, E. Voglmayr, M. Aichholzer, *et al.*, 2019 pp. 225–265 in *Histological Constituents of the Nervous System*, Springer Vienna.