CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.7 / No.2 / 2025 / pp.107-116
https://doi.org/10.51537/chaos.1601947

Neurochaos Learning for Classification using Composition of Chaotic Maps

Akhila Henry [*,1] **and Nithin Nagaraj** [α,2]

[*]Amrita Vishwa Vidyapeetham, Amritapuri Campus, 690525, India, [α]Complex Systems Programme, National Institute of Advanced Studies, Bengaluru, 560012, India.

ABSTRACT

In the age of increasing data availability, there is a pressing need for fast and precise algorithms that can classify datasets. Traditional methods like Support Vector Machines, Random Forest, and Neural Networks are commonly used, but a novel approach known as Neurochaos Learning (NL) has demonstrated strong classification performance across various datasets by incorporating chaos theory. However, the original NL algorithm requires tuning three hyperparameters and involves extraction of multiple features, leading to significant training time. In this study, we propose a modified NL algorithm with only a single hyperparameter and a single feature, using two distinct compositions of 1D chaotic maps, the Skew Tent map with the Logistic map, and the Skew Tent map with $sin(\pi x)$, thereby drastically reducing training time while maintaining classification performance. This study also analyses the 1D chaotic properties of composition of these chaotic maps including Lyapunov Exponent and the stability of fixed points. Testing on ten datasets including *Iris*, *Penguin*, *Haberman*, and *Bank Note Authentication*, our method yields very competitive F1 scores. The composition of the Logistic Map and Skew Tent Map yields an F1 score of 0.569 for the *Haberman* dataset and an impressive 0.968 for the *Penguin* dataset using cosine similarity. Utilizing the composition of $sin(\pi x)$ and Skew Tent Map, the *Ionosphere* dataset achieves an F1 score of 0.876. Our method's versatility is further demonstrated with the Random Forest Algorithm, achieving a perfect F1 score of 1.0 on the *Iris* dataset with the Skew Tent and Logistic Map composition and the same score on the *Penguin* dataset using the $sin(\pi x)$ and Skew Tent Map composition. This streamlined approach meets the demand for faster and more efficient classification algorithms, offering reliable performance in data-rich environments.

KEYWORDS

Neurochaos learning
Skew tent map
Logistic map
Tracemean
Composition

INTRODUCTION

In recent years, Artificial Intelligence (AI), particularly machine learning (ML), has seen rapid advancement, significantly improving data analysis and intelligent computing applications (Sarker *et al.* 2021). The digital era is characterised by an abundance of data from several areas, including mobile technology, commerce, social media, and healthcare. Comprehensive analysis of this data and the development of intelligent, automated systems require a profound understanding of AI, especially machine learning. This domain includes many machine learning approaches, including supervised, unsupervised, semi-supervised, and reinforcement learning. Furthermore, deep learning, a subset of machine learning, has exhibited considerable effectiveness in extensive analysis of data (Sarker 2021). However, machine learning models frequently require substantial training, and as the dataset size increases, the computational resources needed for training correspondingly escalate (Niel 2023).

Contrary to prevalent misunderstandings, "Chaos" in mathematics does not denote disorder or confusion (Faure and Korn 2001). The investigation of *Deterministic Chaos* (Devaney 2018) has emerged as a prominent research domain across multiple disciplines. Nonlinear dynamical systems (for continuous flows) with more than two degrees of freedom can demonstrate chaotic behaviour, rendering their long-term evolution uncertain in spite of the dynamics being completely deterministic. For discrete-time nonlinear dynamical systems, chaos is exhibited at 1-dimension itself. Here *Chaos* refers to the unpredictable outcomes (often random-like) from such simple deterministic systems. The human brain is a distinctly nonlinear system (Kowalik *et al.* 1996). In contrast to other systems that often stabilise following transient

Manuscript received: 15 December 2024,
Revised: 17 February 2025,
Accepted: 25 February 2025.

[1]akhilahenryu@am.amrita.edu (**Corresponding author**)
[2]nithin@nias.res.in

states, the brain perpetually shifts between several states. Evidence indicates that chaos exists in numerous biological systems, especially in the brain, where chaotic dynamics are observable in electroencephalogram (EEG) signals. Despite the seemingly random nature of these signals, they possess intrinsic patterns (Aram *et al.* 2017).

Conventional machine learning and deep neural network frameworks are only marginally influenced by the internal workings of the human brain (Harikrishnan and Nagaraj 2020). In (Harikrishnan and Nagaraj 2020), a novel brain-inspired learning system called *Neurochaos Learning* (NL) is proposed for classification tasks. NL basically utilizes chaos at the level of individual neurons, unlike artificial neural networks (ANNs). The proposed learning paradigm comprises of two distinct architectures: (a) ChaosNet, and (b) Chaos-based features or ChaosFEX (CFX) combined with classical machine learning (ML) models. Input data is transmitted to the Feature Extraction block, where properties derived from the chaotic firing of 1-dimensional chaotic Generalized Lüroth Series (GLS) neurons, specifically firing rate, energy, firing time, and entropy, are retrieved and classified either using cosine similarity (ChaosNet) or via machine learning classifiers (CFX+ML). ChaosNet and CFX+ML (Sethi *et al.* 2023) are designed to harness the advantageous characteristics of biological neural networks, stemming from the complex chaotic behaviour of individual neurons, and have demonstrated the ability to perform challenging classification tasks on par with or superior to traditional artificial neural networks, while necessitating significantly fewer training samples. However, a limitation of the proposed algorithm is the presence of hyperparameters, which require significant time for tuning. Additionally, another drawback is that the transformed features in ChaosNet and ChaosFEX exhibit dependency.

The research presented here aims to improve the current Neurochaos Learning architectures by employing compositions of chaotic maps (as neurons in the input layer of NL) rather than one specific map, thus minimising the number of features and hyperparameters while preserving method efficacy. We propose employing four distinct combinations of chaotic maps: (i) Skew Tent (Harikrishnan and Nagaraj 2020) and Logistic Map (AS *et al.* 2023), (ii) Logistic and Skew Tent Map, (iii) Skew Tent and $sin(\pi x)$ (Palacios-Luengas *et al.* 2021), and (iv) $sin(\pi x)$ and Skew Tent Map for feature extraction. The efficacy of the proposed approach is assessed on classification tasks for 10 different datasets: *Iris* (Fisher 1936), *Haberman* (Haberman 1973), *Seeds* (Dua *et al.* 2017), *Statlog* (Dua *et al.* 2017), *Bank* (Gillich and Lohweg 2010), *Cancer* (Street *et al.* 1993), *Ionosphere* (Sigillito *et al.* 1989), *Wine* (Forina *et al.* 1988), *Sonar* (Horst *et al.* 2020), and *Penguin* (Gorman and Sejnowski 1988).

The subsequent sections of the paper are organised as follows: Section 2 presents the properties of the 1D chaotic maps and their compositions used in this study. Section 3 delineates the suggested algorithm. Section 4 presents the findings from the algorithm using various compositions of chaotic maps. Section 5 compares the F1 scores of the proposed algorithm to those of ChaosNet. Ultimately, Section 6 concludes with prospective avenues for further research.

1D CHAOTIC MAPS AND THEIR COMPOSITIONS

Chaotic maps are iterative mathematical functions that exhibit highly sensitive dependence on initial conditions, leading to seemingly random behavior despite being deterministic in nature. In this work, we focus on three well-known 1D chaotic maps: Logistic map, Skew Tent map, and $sin(\pi x)$ map. Each map has distinct characteristics that make it suitable for various applications in chaos theory and feature extraction for machine learning.

In the following subsections, we briefly define these chaotic maps and subsequently introduce the composition of chaotic maps, which plays a crucial role in the algorithm proposed in this study. The Skew Tent Map, $sin(\pi x)$, and Logistic Map are one-dimensional nonlinear dynamic systems characterised by a single degree of freedom. They are extensively examined in chaos theory and have been previously delineated in many studies (Palacios-Luengas *et al.* 2021; AS *et al.* 2023; Nagaraj 2022).

A skew tent map $T_{skew-tent}(x) : [0,1) \rightarrow [0,1)$ is defined as:

$$T_{\text{skew-tent}}(x) = \begin{cases} \frac{x}{b}, & 0 \leq x < b, \\ \frac{1-x}{1-b}, & b \leq x < 1, \end{cases}$$

where $x \in [0,1)$ and $0 < b < 1$. In this work, we consider $b = 0.499$. The graph of $T_{\text{skew-tent}}(x)$ for $b = 0.499$ is illustrated in Figure 1a. It intersects the line $y = x$ at two distinct positions. Consequently, the fixed points are 0 and 0.666 (Kuijpers 2021). Both the fixed points are unstable. If we start with a value slightly greater than zero, the map will push the trajectory away from zero, making $x = 0$ an unstable fixed point. Likewise, the other fixed point, 0.666, is also unstable. Generalised Luröth Series (GLS) maps preserve the Lebesgue measure and exhibit uniform distribution on the interval $[0,1)$ as the invariant distribution (Nagaraj 2022). Every GLS map on the interval [0,1) exhibits ergodicity. Therefore, the skew tent map is also ergodic (Dajani and Kraaikamp 2002). The Lyapunov Exponent for this skew tent map, with a skew value of 0.499, is 0.6931.

The logistic map is defined by the following equation:

$$x_{n+1} = rx_n(1 - x_n),$$

where x_n takes the value in the interval $(0,1)$ and r is the bifurcation parameter which lies in the interval $(0,4]$. For the Logistic Map, we consider the parameter $r = 4$, which results in fully chaotic behavior (Chen *et al.* 2021). The corresponding graph for the Logistic Map with $r = 4$ is shown in Figure 1b. The logistic map with $r = 4$ intersect the line $y = x$ at two distinct points, resulting in two fixed points are 0 and $\frac{3}{4}$. Both the fixed points are unstable. The logistic map with $r = 4$ has an invariant distribution given by: (Ayers and Radunskaya 2024)

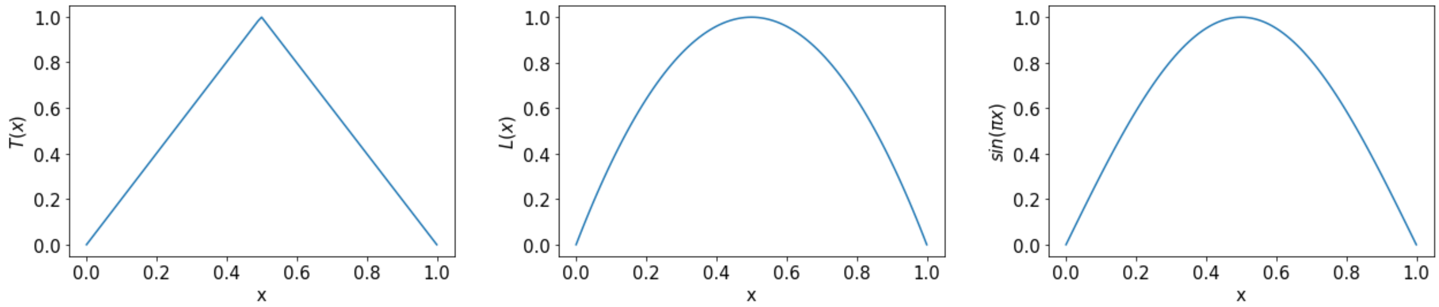
$$f(x) = \begin{cases} \frac{1}{x(1-x)} & \text{for } 0 < x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Also the map is ergodic with a Lyapunov exponent of 0.6724 (Machicao *et al.* 2019; Naanaa 2015).

The $sin(\pi x)$ map on $(0,1)$ is shown in the Figure 1c. It is a simple dynamical system, similar to logistic map, exhibiting complex chaotic behaviour with a lyapunov exponent of 0.6889. Here the fixed points are $x = 0$ and $x = 0.7365$, both of which are unstable (Griffin 2013; Alzaidi *et al.* 2018).

Composition of 1D Chaotic Maps

Let f and g be two functions. Then, by definition, the composition (Muthuvel *et al.* 2000; Suryadi *et al.* 2020) of two functions f and g defined by $fog(x) = f(g(x))$ where g is applied first followed by f. In this work, we study the following compositions of maps:



(a) Skew-tent map

(b) Logistic map

(c) $sin(\pi x)$ map

Figure 1 1D Chaotic Maps: Skew Tent Map $T(x)$, Logistic Map $L(x)$, $sin(\pi x)$ Map $S(x)$.

- Skew Tent map and Logistic map ($T \circ L$ map)
- Logistic map and Skew Tent map ($L \circ T$ map)
- Skew Tent map and $sin(\pi x)$ ($T \circ S$ map)
- $sin(\pi x)$ and Skew Tent map ($S \circ T$ map)

The composition of these maps are continuous, non-commutative, indicating that the sequence of compositions is distinct. The sequence of applying the Skew Tent Map followed by the Logistic Map yields different outcomes/trajectories when compared to the reverse sequence (Logistic Map followed by the Skew Tent Map). This significantly influences the algorithm's performance across the ten analysed datasets. The $sin(\pi x)$ map exhibits certain similarities to the Logistic Map (Zhu *et al.* 2019). The composition of the Skew Tent Map followed by the $sin(\pi x)$ map also differs from the reverse order ($sin(\pi x)$ map followed by the Skew Tent Map). This distinction is essential for evaluating the impact of various map combinations on the algorithm's efficacy.

The Figure 2 illustrate the compositions of the maps and shows that each composition has four distinct fixed points. To determine the approximate values of fixed points of composition of chaotic maps $f \circ g$, it is enough to compute the values of x such that $(f \circ g)(x) - x = 0$ using *fsolve* function in Python with ten equally spaced guesses in $(0, 1)$ within an error of 10^{-5}. Table 1 summarizes the fixed points, their stability, and Lyapunov exponent values of the composition of chaotic maps. A point x_0 is said to be stable if $|f'(x_0)| < 1$ (Alligood *et al.* 1998). Thus, to evaluate the stability of fixed points, the condition

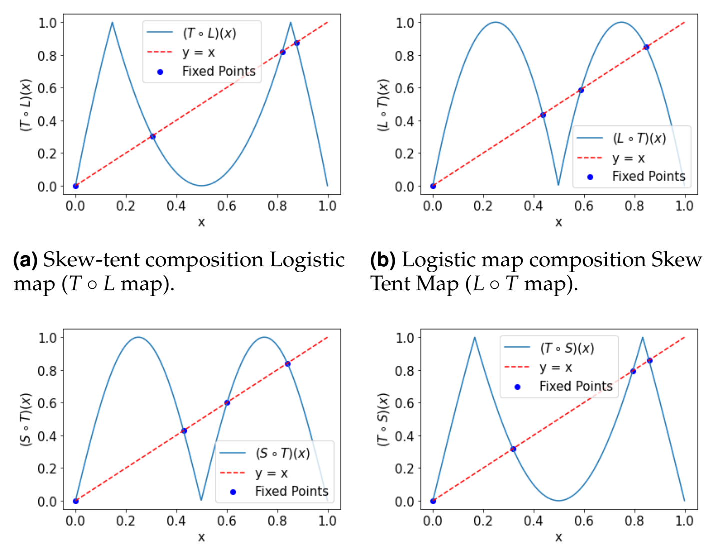
$$\frac{|f(x+h) - f(x-h)|}{2h} < 1,$$

where $h = 10^{-5}$ is verified.

The Lyapunov exponent $h(x_1)$ (Alligood *et al.* 1998) for a given orbit $x_1, x_2, ...$ is defined by:

$$h(x_1) = \lim_{n \to \infty} \frac{ln|f'(x_1)| + ln|f'(x_2)| + \ldots + ln|f'(x_n)|}{n}.$$

The values of the Lyapunov exponent for $T \circ L$, $L \circ T$, $S \circ T$ and $T \circ S$ are all greater than one, implying that the divergence of nearby trajectories happens at an exponentially fast rate. Here the limiting value for Lyapunov exponent is approximated after 100,000 iterations. These compositions are highly chaotic and its sensitivity to initial conditions is extreme. The distributions of points in the trajectory generated by the composition of maps (in each of the four cases) with an initial point 0.01 are shown in Figure 3.



(a) Skew-tent composition Logistic map ($T \circ L$ map).

(b) Logistic map composition Skew Tent Map ($L \circ T$ map).

(c) $sin(\pi x)$ map composition Skew Tent Map ($S \circ T$ map).

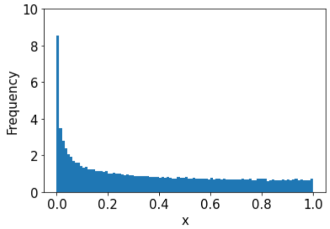
(d) Skew Tent map composition $sin(\pi x)$ map ($T \circ S$ map).

Figure 2 First return maps and fixed points for the composition of 1D chaotic maps (Skew Tent map, Logistic Map and $sin(\pi x)$).

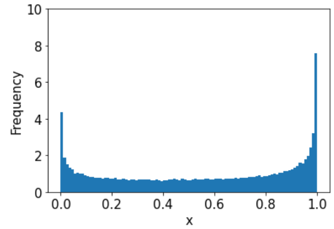
The following section introduces the modified Neurochaos Learning Algorithm using the composition of chaotic maps - $T \circ L$, $L \circ T$, $S \circ T$, $T \circ S$.

Table 1 Fixed Points, Stability and Lyapunov Exponents of 1D Chaotic Map Compositions

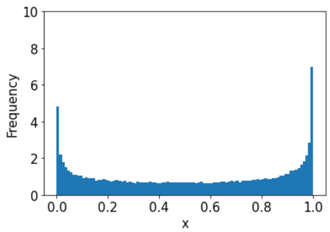
Composition	Fixed Points (approx.)	Stability	Lyap. Exp.
ToL	0.0, 0.3046, 0.8206, 0.8752	Unstable	1.2840
LoT	0.0, 0.4367, 0.5889, 0.8470	Unstable	1.2838
ToS	0.0, 0.3179, 0.8590, 0.7944	Unstable	1.2957
SoT	0.0, 0.6020, 0.4286, 0.8407	Unstable	1.2961



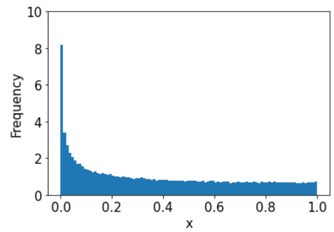
(a) Skew-tent composition Logistic map ($T \circ L$ map).



(b) Logistic map composition Skew Tent Map ($L \circ T$ map).



(c) $sin(\pi x)$ map composition Skew Tent Map ($S \circ T$ map).



(d) Skew Tent map composition $sin(\pi x)$ map ($T \circ S$ map).

Figure 3 Distribution of points from the trajectories generated by $T \circ L$, $L \circ T$, $S \circ T$ and $T \circ S$ maps.

PROPOSED ALGORITHM

The proposed algorithm in Figure 4 leverages the composition of chaotic maps to enhance the feature extraction process in Neurochaos Learning (Sethi *et al.* 2023; Balakrishnan *et al.* 2019). The motivation behind this approach is to improve the classification performance by reducing the number of hyperparameters and utilising a single feature – the *mean* of the neural trace, which is computationally efficient and robust across various datasets. This section outlines the key steps involved in the algorithm.

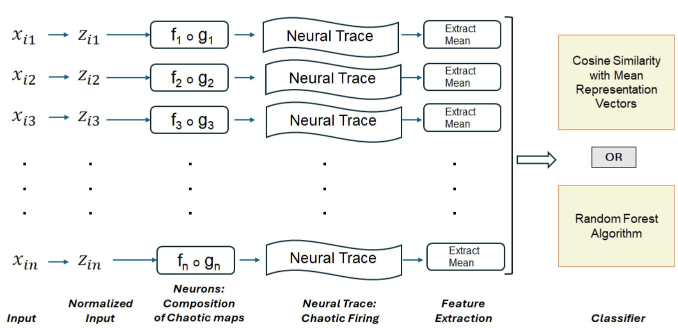


Figure 4 Block diagram depicting the proposed algorithm in this study.

- **Step 1 : Normalisation**
  Consider a dataset of m samples, each with n features, represented as:

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

  Each feature attribute x_{ij} can be normalised using min-max normalisation technique. Specifically, the normalised value z_{ij} can be computed as:

  $$z_{ij} = \frac{x_{ij} - min\{x_{ij} : 1 \leq i \leq m\}}{max\{x_{ij} : 1 \leq i \leq m\} - min\{x_{ij} : 1 \leq i \leq m\}}$$

  for $1 \leq j \leq n$. Thus the resulting normalised dataset is:

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

  This transformation ensures that all features are scaled within the range $[0, 1]$, thereby making them suitable for subsequent steps in the algorithm.

- **Step 2 : Neural Trace Generation**
  For each normalised feature value z_{ij}, a corresponding neural trace is generated by applying a composition of chaotic maps. The generation process begins with an initial neural activity, denoted as q. Here, q serves as a hyperparameter, and its value is tuned between 0.01 to 0.99 with a step value 0.01 during training to optimise performance. The dataset was split with 20% allocated for testing and remaining 80% for training. The chaotic maps are iteratively composed, transforming the initial neural activity based on the value of z_{ij}, to produce a unique neural trace for each feature. Suppose f and g are the chaotic maps and using $f \circ g$ for generating neural trace, the neural trace can be mathematically represented as

  $$N = \{q, f \circ g(q), (f \circ g)^2(q) = f \circ g \circ f \circ g(q), \ldots, (f \circ g)^T(q)\}.$$

  This neural trace (with firing time T) serves as the foundation for the feature extraction process in the subsequent steps of the algorithm.

- **Step 3 : Feature Extraction**
  In the next phase, feature extraction is performed by analysing the generated neural trace corresponding to each z_{ij}. Given n features, we require n instances of the composition of chaotic maps f and g, denoted as $f_1 \circ g_1, f_2 \circ g_2 \ldots, f_n \circ g_n$, corresponding to each $z_{i1}, z_{i2}, \ldots, z_{in}$.

  The neural trace

  $$N_j = \{q, f_j \circ g_j(q), (f_j \circ g_j)^2(q) = f_j \circ g_j \circ f_j \circ g_j(q), \ldots\}$$

  where $j = 1, 2, \ldots, n$, evolves under the influence of the chaotic map composition $f_j \circ g_j$ until it reaches an ϵ-neighborhood of the corresponding stimulus, z_{ij}. For this algorithm, the noise ϵ is set to a value of 0.25, meaning the neural trace halts when its trajectory comes within 0.25 units of the feature z_{ij}. Once this condition is met, the mean value of the neural trace up to this point, t_{ij} is computed. This mean serves as a summary statistic for the chaotic behavior of the neural trace, capturing essential information about the feature z_{ij}. The resulting mean value for each trajectory

  $$\{(t_{i1}, t_{i2}, \ldots, t_{in}) : i = 1, 2, \ldots, m\}$$

  will be utilised in the subsequent classification step. This computation of mean is performed for each feature of each training instance of each class.

- **Step 4 : Classification**
  Once the mean of the neural trace is computed for each z_{ij}, they can be either classified using cosine similarity or Random Forest Algorithm (Breiman 2001).

  1. *Cosine Similarity(Cos) classifier*: In this approach, the transformed features are classified by computing the cosine similarity between the feature vectors and the mean representation

vectors of each class.
Let the given m data samples belongs to k classes.

$$\{(x_{l_1 1}, x_{l_1 2}, \ldots, x_{l_1 n}), (x_{l_2 1}, x_{l_2 2}, \ldots, x_{l_2 n}), \ldots, (x_{l_r 1}, x_{l_r 2}, \ldots, x_{l_r n})\}$$

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

$$\{(t_{l_1 1}, t_{l_1 2}, \ldots, t_{l_1 n}), (t_{l_2 1}, t_{l_2 2}, \ldots, t_{l_2 n}), \ldots, (t_{l_r 1}, t_{l_r 2}, \ldots, t_{l_r n})\}.$$

Then the mean representation vector corresponding to the class l can be defined as:

$$M^{(l)} = \left(\frac{\sum_{i=l_1}^{l_r} t_{i1}}{r}, \frac{\sum_{i=l_1}^{l_r} t_{i2}}{r}, \ldots, \frac{\sum_{i=l_1}^{l_r} t_{in}}{r} \right).$$

In order to classify a particular *test* data instance $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, calculate the cosine similarity of extracted feature vector of the data instance $T_i = (t_{i1}, t_{i2}, \ldots, t_{in})$ with the mean representation vectors of each class $\{M^{(1)}, M^{(2)}, \ldots, M^{(l)}, \ldots, M^{(k)}\}$. Cosine similarity is defined as follows :

$$\cos\theta = \frac{M^{(j)} \cdot T_i}{||M^{(j)}|| \; ||T_i||}$$

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

2. *Random Forest (RF) classifier*: Random Forest (Breiman 2001) is a powerful machine learning algorithm that employs technique of *bagging* with numerous decision trees and consolidates their predictions to attain a high degree of classification accuracy. The RF classifier can adeptly manage the non-linear interactions inherent in the chaotic dynamics of the neural trace by utilising the diversity of decision trees. Instead of classifying using cosine similarity, the transformed dataset

$$\{(t_{l_1 1}, t_{l_1 2}, \ldots, t_{l_1 n}), (t_{l_2 1}, t_{l_2 2}, \ldots, t_{l_2 n}), \ldots, (t_{l_r 1}, t_{l_r 2}, \ldots, t_{l_r n})\}$$

is classified using Random Forest Algorithm. The number of estimators for Random Forest Algorithm was optimised through 5-fold cross-validation selecting from the values $1, 10, 100, 1000, 10000$.

While extracting features for classification using Random Forest Algorithm, instead of tuning q value again for neural trace, the value of q tuned for each dataset in Cos Classifier is reused. Thus both Cos and RF classifiers used same values of q, which is indicated in Table 3, Table 4, Table 5 and Table 6. The flowchart (Figure 4) depicts the proposed algorithm. Both methods offer distinct advantages, and the choice between them depends on the specific characteristics of the dataset and the performance requirements of the task.

RESULTS AND ANALYSIS

In this section, we present the F1 Score (Christen *et al.* 2023) results for the proposed algorithm using four different compositions of chaotic maps. The classification was performed on 10 different datasets, and we report the F1 scores for two classification methods: Cosine Similarity and Random Forest (RF). The F1 Score, which balances precision and recall, provides a robust measure of the classifier's performance across varying data characteristics.

Table 2 Description of datasets used in this study.

Dataset	Features	Classes	Samples
Iris	4	3	150
Haberman's Survival	3	2	306
Seeds	7	3	210
Statlog (Heart)	13	2	270
Ionosphere	34	2	351
Bank Note Auth.	4	2	1372
Breast Cancer Wis.	31	2	569
Wine	13	3	178
Penguin	4	3	342
Sonar	60	2	208

The datasets used in this evaluation are as follows: *Iris*, *Haberman*, *Seeds*, *Statlog*, *Ionosphere*, *Bank*, *Breast Cancer*, *Wine*, *Penguin*, and *Sonar*. Table 2 shows the number of features and samples in each dataset used in this study. For each dataset, the F1 scores for both Cosine Similarity and Random Forest classifiers are presented, allowing for a comprehensive comparison of the algorithm's effectiveness across different domains (Sethi *et al.* 2023).

Tables 3 and 4 display the training and testing F1 scores of the proposed Neurochaos Learning algorithm utilising the composition of the Skew Tent Map and the Logistic Map. These tables illustrate the algorithm's performance over the chosen ten unique datasets. Similarly, tables 5 and 6 substitute the Logistic Map with the $sin(\pi x)$ map, which possesses certain dynamical characteristics akin to those of the Logistic Map. The performance of the Neurochaos Learning algorithm is rigorously analysed and compared utilising these compositions of chaotic maps across identical datasets.

The performance of custom Random Forest (RF) algorithms, where features are extracted using various compositions of 1D chaotic maps and subsequently passed to the Random Forest algorithm, was compared against standalone RF algorithm across multiple datasets:

- *Iris*: The composition of the Skew Tent and Logistic Maps (denoted as ToL and LoT) followed by classification using Random Forest resulted in a perfect F1 Score of 1.0, achieving 100% classification accuracy. This demonstrates the powerful synergy between chaotic feature extraction and the RF classifier for this dataset.
- *Penguin*: A similar performance boost was observed when using features from the composition of the $sin(\pi x)$ and Skew Tent Maps (denoted as ToS and SoT) with RF classification, also reaching an F1 Score of 1.0, marking complete accuracy in classifying the *Penguin* dataset.
- *Cancer*: The combination of the Skew Tent and $sin(\pi x)$ maps (ToS) followed by RF classification delivered a notable improvement of 4.79% over standalone RF. Additionally, the composition of the Skew Tent and Logistic Maps (ToL) with RF also exhibited a 3.81% increase in performance.
- *Wine*: Minor yet consistent improvements were observed, with the LoT RF and ToS RF algorithms outperforming stan-

Table 3 Training & Testing F1 scores: $L \circ T$ **Map -Based NL Algorithm.**

Sl No	Dataset	Initial Neural Activity(q)	F1 Score using Cosine Similarity		F1 Score using Random Forest	
			Training	Testing	Training	Testing
1	*Iris*	0.98	0.912	0.916	0.948	1.000
2	*Haberman*	0.27	0.584	0.569	0.559	0.456
3	*Seeds*	0.78	0.831	0.652	0.889	0.926
4	*Statlog*	0.78	0.810	0.792	0.843	0.774
5	*Bank Note Authentication*	0.28	0.858	0.774	0.938	0.911
6	*Breast Cancer Wisconsin*	0.74	0.931	0.903	0.949	0.918
7	*Ionosphere*	0.75	0.803	0.727	0.923	0.893
8	*Wine*	0.63	0.950	0.862	0.977	0.976
9	*Sonar*	0.45	0.757	0.703	0.818	0.775
10	*Penguin*	0.98	0.957	0.968	0.918	0.920

Table 4 Training & Testing F1 scores: $T \circ L$ **Map -Based NL Algorithm.**

Sl No	Dataset	Initial Neural Activity (q)	F1 Score using Cosine Similarity		F1 Score using Random Forest	
			Training	Testing	Training	Testing
1	*Iris*	0.98	0.920	0.910	0.946	1.000
2	*Haberman*	0.27	0.594	0.557	0.573	0.483
3	*Seeds*	0.03	0.876	0.783	0.893	0.822
4	*Statlog*	0.23	0.807	0.710	0.842	0.797
5	*Bank Note Authentication*	0.95	0.858	0.836	0.863	0.843
6	*Breast Cancer Wisconsin*	0.54	0.930	0.849	0.952	0.954
7	*Ionosphere*	0.50	0.808	0.691	0.917	0.909
8	*Wine*	0.63	0.957	0.943	0.973	0.968
9	*Sonar*	0.46	0.750	0.734	0.813	0.798
10	*Penguin*	0.25	0.959	0.960	0.911	0.917

dalone RF by 1.04% and 3.73%, respectively. This highlights the advantage of feature extraction using chaotic maps for this dataset.

The custom algorithms using Cosine Similarity classifier (Cos) for classification, after feature extraction from 1D chaotic maps, were also compared against the original ChaosNet algorithm (Sethi *et al.* 2023):

- *Sonar*: ToL Cos and SoT Cos significantly outperformed ChaosNet, showing improvements of 14.15% and 12.80%, respectively. Additionally, ToS Cos exhibited a solid 10.08% increase, underscoring the effectiveness of these map compositions in enhancing classification accuracy for the *Sonar* dataset.
- *Cancer*: The LoT Cos and SoT Cos algorithms outperformed ChaosNet by 6.86% and 9.41%, respectively, indicating their superior performance for this dataset, especially in capturing complex patterns.
- *Penguin*: The performance of LoT Cos and SoT Cos were nearly equivalent to ChaosNet, with only marginal improvements.

Table 5 Training & Testing F1 scores: $T \circ S$ **Map -Based NL Algorithm.**

Sl No	Dataset	Initial Neural Activity (q)	F1 Score using Cosine Similarity		F1 Score using Random Forest	
			Training	Testing	Training	Testing
1	*Iris*	0.98	0.884	0.916	0.966	0.889
2	*Haberman*	0.28	0.594	0.557	0.546	0.431
3	*Seeds*	0.04	0.898	0.783	0.931	0.902
4	*Statlog*	0.56	0.806	0.671	0.837	0.791
5	*Bank Note Authentication*	0.95	0.852	0.850	0.941	0.911
6	*Breast Cancer Wisconsin*	0.54	0.928	0.849	0.946	0.963
7	*Ionosphere*	0.50	0.808	0.691	0.899	0.876
8	*Wine*	0.63	0.957	0.943	0.973	0.968
9	*Sonar*	0.46	0.755	0.708	0.808	0.798
10	*Penguin*	0.04	0.950	0.965	0.949	1.000

Table 6 Training & Testing F1 scores: $S \circ T$ **Map -Based NL Algorithm.**

Sl No	Dataset	Initial Neural Activity (q)	F1 Score using Cosine Similarity		F1 Score using Random Forest	
			Training	Testing	Training	Testing
1	*Iris*	0.98	0.894	0.917	0.948	0.928
2	*Haberman*	0.95	0.591	0.490	0.540	0.447
3	*Seeds*	0.04	0.878	0.749	0.915	0.897
4	*Statlog*	0.01	0.811	0.735	0.824	0.810
5	*Bank Note Authentication*	0.28	0.858	0.771	0.924	0.904
6	*Breast Cancer Wisconsin*	0.70	0.927	0.924	0.961	0.901
7	*Ionosphere*	0.76	0.803	0.876	0.873	0.807
8	*Wine*	0.63	0.949	0.888	0.971	0.918
9	*Sonar*	0.46	0.761	0.725	0.813	0.775
10	*Penguin*	0.76	0.954	0.964	0.946	1.000

- *Statlog*: LoT Cos demonstrated superiority over ChaosNet, with a 7.32% improvement. This further emphasizes the strength of the proposed algorithm in specific datasets where chaotic features provide additional discriminative power.
- *Haberman*: In this dataset, LoT Cos showed a modest improvement of 1.61% over ChaosNet, indicating enhanced classification performance even in more challenging datasets.

Table 7 compares the best F1 scores of our proposed algorithm with that of ChaosNet and standalone Random Forest algorithms.

Figures 5 and 6 gives a comparison of F1 scores of proposed algorithm with ChaosNet and Random Forest Algorithm on different datasets.

Overall, the analysis highlights that the proposed Neurochaos Learning algorithms using different 1D chaotic map compositions for feature extraction outperform classical stand-alone classifiers like Random Forest and ChaosNet on various datasets. The non-commutative nature of map compositions plays a significant role in the performance variations observed, with different combinations producing distinct improvements depending on the dataset and

Table 7 Comparison of best F1 scores achieved by the proposed algorithms, Random Forest and ChaosNet. 'Cos' stands for Cosine Similarity based classifier used in NL.

Dataset	ChaosNet F1 Score	Standalone RF F1 Score	Best F1 Score -Cos Algorithms	Best F1 Score - Custom RF
Iris	1.000	1.000	0.917 ($S \circ T$ Cos)	1 ($L \circ T$ RF, $T \circ L$ RF)
Haberman	0.560	0.560	0.569 ($L \circ T$ Cos)	0.483 ($T \circ L$ RF)
Seeds	0.845	0.877	0.783 ($T \circ S$ Cos, $T \circ L$ Cos)	0.926($L \circ T$ RF)
Statlog	0.738	0.838	0.792 ($L \circ T$ Cos)	0.810 ($S \circ TRF$)
Bank Note Authentication	0.845	0.974	0.85 ($T \circ S$ Cos)	0.911 ($L \circ T$ RF, $T \circ S$ RF)
Breast Cancer Wisconsin	0.927	0.919	0.924($S \circ T$ Cos)	0.963($T \circ S$ RF)
Ionosphere	0.860	0.909	0.876($S \circ T$ Cos)	0.909($T \circ L$ RF)
Wine	0.976	0.966	0.943 ($T \circ L$ Cos, $T \circ S$ Cos)	0.976 ($L \circ T$ RF)
Sonar	0.643	0.798	0.734 ($T \circ L$ Cos)	0.798($T \circ L$ RF, $T \circ S$ RF)
Penguin	0.964	0.929	0.968($L \circ T$ Cos)	1.000 ($T \circ S$ RF , $S \circ T$ RF)

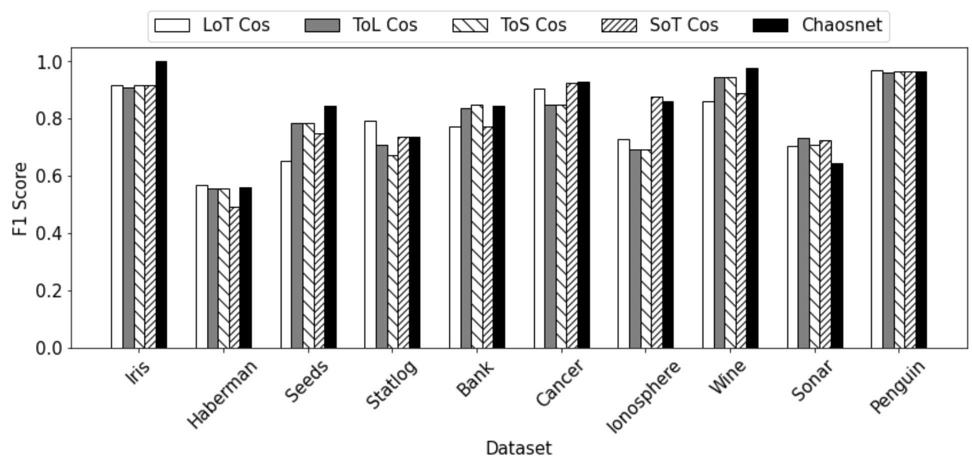


Figure 5 Comparison of F1 scores of Algorithms computed using cosine similarity on different datasets.

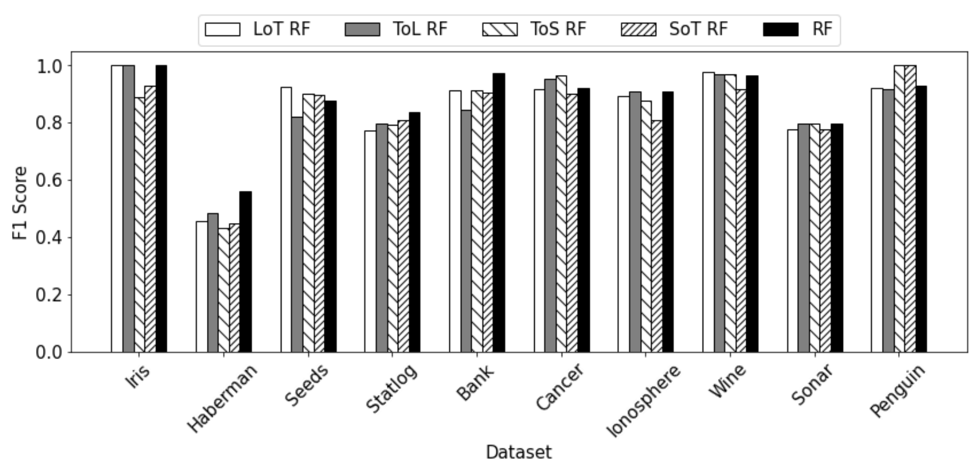


Figure 6 Comparison of F1 scores of custom Random Forest algorithms on different datasets.

classification method used.

CONCLUSION

The results demonstrate the robust performance of the proposed neurochaos-based algorithms on diverse datasets. Across multiple datasets, custom algorithms that use the composition of 1D chaotic maps, such as Skew Tent, Logistic, and $sin(\pi x)$ maps, consistently outperform traditional models like standalone Random Forest and ChaosNet. Notably, the improvements are especially pronounced in datasets such as *Cancer*, *Penguin*, and *Sonar*, indicating that the chaotic map-based feature extraction technique offers substantial advantages in these scenarios. The Lyapunov exponent for each of these composition of 1D chaotic maps is greater than 1.0, and their fixed points are unstable, indicating they are highly chaotic. Building on the promising results of the current study, there are several avenues for future research that can further enhance the performance and applicability of the proposed neurochaos-based algorithms:

- Exploring New Compositions of 1D Chaotic Maps: We plan to investigate different new combinations of 1D chaotic maps to uncover compositions that yield even better F1 scores. The non-commutative nature of these compositions offers a rich space to explore for improving classification results across diverse datasets.
- Application to Varied Datasets: Expanding the analysis to a broader range of datasets, particularly those with more complex or noisy structures, will help assess the generalisability and robustness of the custom algorithms in different real-world scenarios.
- Impact of Chaotic Map Compositions on Performance: A deeper investigation into how different chaotic map compositions affect algorithm performance can reveal valuable insights. Understanding the influence of map properties, such as their sensitivity to initial conditions and chaotic behavior, could lead to optimised feature extraction techniques for specific types of data.
- Incorporating Noise Robustness: Analysing the performance of these algorithms under varying noise levels can offer insights into their stability and resilience. Future work could involve introducing noise into the datasets and observing how different compositions of chaotic maps handle this, with the goal of developing more noise-tolerant algorithms.

These research directions not only offer opportunities to improve the current algorithms but also contribute to advancing the broader field of chaos theory-based machine learning.

Acknowledgments

Nithin Nagaraj would like to acknowledge the financial support of Anusandhan National Research Foundation (ANRF), Department of Science & Technology, Govt. of India (Grant No. TAR/2021/000206) towards this research.

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.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

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

Alzaidi, A. A., M. Ahmad, H. S. Ahmed, and E. A. Solami, 2018 Sine-cosine optimization-based bijective substitution-boxes construction using enhanced dynamics of chaotic map. Complexity **2018**: 9389065.

Aram, Z., S. Jafari, J. Ma, J. C. Sprott, S. Zendehrouh, *et al.*, 2017 Using chaotic artificial neural networks to model memory in the brain. Communications in Nonlinear Science and Numerical Simulation **44**: 449–459.

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

Ayers, K. and A. Radunskaya, 2024 Noisy fixed points: Stability of the invariant distribution of the random logistic map. arXiv preprint arXiv:2403.13116 .

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**.

Breiman, L., 2001 Statistics department university of california berkeley, ca 94720. 2001. Random Forests .

Chen, S., S. Feng, W. Fu, and Y. Zhang, 2021 Logistic map: Stability and entrance to chaos. In *Journal of Physics: Conference Series*, volume 2014, p. 012009, IOP Publishing.

Christen, P., D. J. Hand, and N. Kirielle, 2023 A review of the f-measure: its history, properties, criticism, and alternatives. ACM Computing Surveys **56**: 1–24.

Dajani, K. and C. Kraaikamp, 2002 *Ergodic theory of numbers*, volume 29. American Mathematical Soc.

Devaney, R. L., 2018 *A first course in chaotic dynamical systems: theory and experiment*. CRC Press.

Dua, D., C. Graff, *et al.*, 2017 Uci machine learning repository, 2017. URL http://archive. ics. uci. edu/ml **7**: 62.

Faure, P. and H. Korn, 2001 Is there chaos in the brain? i. concepts of nonlinear dynamics and methods of investigation. Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie **324**: 773–793.

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

Forina, M., R. Leardi, C. Armanino, S. Lanteri, P. Conti, *et al.*, 1988 Parvus an extendable package of programs for data exploration, classification and correlation .

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

Gorman, R. P. and T. J. Sejnowski, 1988 Analysis of hidden units in a layered network trained to classify sonar targets. Neural networks **1**: 75–89.

Griffin, J., 2013 The sine map. Retrieved May **4**: 2018.

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

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.

Horst, A. M., A. P. Hill, and K. B. Gorman, 2020 palmerpenguins: Palmer archipelago (antarctica) penguin data. r package version 0.1. 0.

Kowalik, Z. J., A. Wróbel, and A. Rydz, 1996 Why does the human brain need to be a nonlinear system? Behavioral and Brain Sciences **19**: 302–303.

Kuijpers, B., 2021 Deciding the point-to-fixed-point problem for skew tent maps on an interval. Journal of Computer and System Sciences **115**: 113–120.

Machicao, J., M. Alves, M. S. Baptista, and O. M. Bruno, 2019 Exploiting ergodicity of the logistic map using deep-zoom to improve security of chaos-based cryptosystems. International Journal of Modern Physics C **30**: 1950033.

Muthuvel, K. *et al.*, 2000 Composition of functions. International Journal of Mathematics and Mathematical Sciences **24**: 213–216.

Naanaa, A., 2015 Fast chaotic optimization algorithm based on spatiotemporal maps for global optimization. Applied Mathematics and Computation **269**: 402–411.

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

Niel, O., 2023 A novel algorithm can generate data to train machine learning models in conditions of extreme scarcity of real world data. arXiv preprint arXiv:2305.00987 .

Palacios-Luengas, L., R. Marcelín-Jiménez, E. Rodriguez-Colina, M. Pascoe-Chalke, O. Jiménez-Ramírez, *et al.*, 2021 Function composition from sine function and skew tent map and its application to pseudorandom number generators. Applied Sciences **11**: 5769.

Sarker, I. H., 2021 Machine learning: Algorithms, real-world applications and research directions. SN computer science **2**: 160.

Sarker, I. H., M. H. Furhad, and R. Nowrozy, 2021 Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. SN Computer Science **2**: 173.

Sethi, D., N. Nagaraj, and N. B. Harikrishnan, 2023 Neurochaos feature transformation for machine learning. Integration **90**: 157–162.

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.

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.

Suryadi, M., Y. Satria, V. Melvina, L. N. Prawadika, and I. M. Sholihat, 2020 A new chaotic map development through the composition of the ms map and the dyadic transformation map. In *Journal of Physics: Conference Series*, volume 1490, p. 012024, IOP Publishing.

Zhu, S., G. Wang, and C. Zhu, 2019 A secure and fast image encryption scheme based on double chaotic s-boxes. Entropy **21**: 790.

How to cite this article: Henry, A., and Nagaraj, N. Neurochaos Learning for Classification using Composition of Chaotic Maps. *Chaos Theory and Applications*, 7(2), 107-116, 2025.