Eur. Phys. J. Spec. Top.
https://doi.org/10.1140/epjs/s11734-021-00346-z

THE EUROPEAN PHYSICAL JOURNAL SPECIAL TOPICS

Regular Article

Machine learning-based classification of time series of chaotic systems

Süleyman Uzun[a]

Department of Computer Engineering, Faculty of Technology, Sakarya University of Applied Sciences, 54050 Sakarya, Turkey

Received 7 April 2021 / Accepted 19 November 2021

Abstract In this study, the classification of time series belonging to three different chaotic systems has been proposed using machine learning methods. For this purpose, the time series of Lorenz, Chen, and Rossler systems, three of the well-known chaotic systems, are classified using machine learning methods. In the study, the classification of chaotic systems has been made with 18 sub-methods of Naive Bayes, Support Vector Machines, K-Nearest Neighborhood, and Tree methods. As a result, the K-Nearest Neighborhood method has classified time series belonging to chaotic systems with very high accuracy of 99.2%. In this way, it has become possible to associate the chaotic-random signals with a mathematical system.

1 Introduction

Chaotic systems are nonlinear mathematical models which originated from the rules of defining chaotic behaviors. In recent years, chaos theory and chaotic systems have been used in various engineering fields such as cryptography, image and voice encryption, secure communication, data security, random number generators, digital signature applications, weak signal detection, DC–DC converters, and neurophysiology, among others. Similarly, machine learning has been one of the most popular subjects widely used in recent years. Although many studies on machine learning mainly focus on the classification processes in different fields. Therefore, in this study, chaotic systems and machine learning are used together.

Machine learning is a branch of artificial intelligence that enables computers to learn using existing data from complex and large data sets [1]. Due to the high classification performance of machine learning algorithms, they have been used in many different areas [2]. Because of its classification ability, it can also be used to classify chaotic systems that have similar characteristics to each other.

According to the literature review, there are various studies on the classification of chaotic signals using machine learning algorithms. The radial basis function approach and standard numerical techniques have been widely used in estimating the time series of chaotic systems [3–6]. It is very successful in classifying and predicting machine learning and deep learning techniques [7]. This performance reveals that it can also be used to estimate and classify time series [8,10]. Pathak et al. [11], [12] used machine learning to perform model-free prediction of chaotic dynamical systems. Boulle et al. [13] classified the univariate time series of a dynamic system using machine learning algorithms. They trained neural networks on a more complex and larger dimensional data set than test data to assess the ability to generalize with respect to the low-dimensional phase space.

Machine learning algorithms have been used in various fields. Zhang et al. [14] designed a machine learning-based classifier to differentiate between schizophrenia patients and healthy constructs using features extracted from electroencephalograph (EEG) signals based on time-dependent EEG activity (ERP). Sayilgan et al. [15] classified seven different hand gestures using a brain–computer interface based on steady-state visually evoked potential (SSVEP). They used Naive Bayes (NB), Support Vector Machines (SVM), and Extreme Learning Machine as classification algorithms. Shimpi et al. [16] classified 16 types of arrhythmias with electrocardiogram (ECG) data using machine learning algorithms. In the classification, they used the SVM, Random Forest, Logistic Regression, and K-Nearest Neighbor (KNN) algorithms.

In this study, we classified the signals presenting chaotic behaviors and random characteristics using a new approach. According to experimental results, time series of chaotic systems have been classified with high accuracy using machine learning methods such as NB, SVM, KNN, and Tree. The rest of the study has been organized as follows. Section 2 introduces the chaotic systems used in the study and the data set obtained

[a] e-mail: suleymanuzun@subu.edu.tr (corresponding author)

from them. Section 3 briefly explains the machine learning methods used for the classification. Section 4 presents the simulation and performance results of the classification processes. Section 5 give the conclusions.

2 Chaotic systems and data set used in the study

This section presents three different systems used for generating the time series to be classified and the data set belonging to the time series obtained from these systems. There are numerous chaotic systems in the literature, such as Lorenz, Chen, and Rossler systems. For that reason, some criteria were taken into account while selecting the systems used in this study. Based on this, the systems that are among the most common chaotic systems, 3-dimensional, and whose mathematical models contain similar nonlinear terms were determined. Additionally, these systems can be used for real physical system modeling such as atmospheric, electrical, chemical systems. Moreover, since the time series and phase portraits of the Lorenz and Chen systems are similar and the Rossler system's time series and phase portraits are different from the others, it is deemed appropriate to examine the classification performance.

2.1 Lorenz system

Lorenz system was developed in 1963 by Edward Lorenz as a simplified mathematical model for atmospheric convection [17]. Lorenz system consists of three ordinary differential equations, as seen in Eq. (1). In Eq. (1), the constants of a, b and $c$are system parameters, and x, y, and z are the system's state variables.

$$\begin{aligned} \dot{x} &= a(y - x) \\ \dot{y} &= x(c - z) - y \\ \dot{z} &= xy - bz. \end{aligned} \tag{1}$$

The Lorenz system has also been used in simplified mathematical models of thermosiphons [18], lasers [19], electric circuits [20], brushless DC motors [21], dynamos [22], and chemical reactions [20]. In Fig. 1, the time series and the phase portraits of the Lorenz System are plotted for parameter values of $a = 10$, $b = 8/3$, $c = 28$. And the initials values are set to $x_0 = 10$, $y_0 = -10$, $z_0 = 15$.

2.2 Chen system

A double scroll chaotic attractor, similar but nonequivalent to Lorenz System, is named Chen System or Chen Attractor proposed by Guanrong Chen and Ueta in 1999 [23]. Chen System also consists of three ordinary differential equations, as seen in Eq. (2).

$$\begin{aligned} \dot{x} &= a(y - x) \\ \dot{y} &= (c - a)x - xz + cy \\ \dot{z} &= xy - bz, \end{aligned} \tag{2}$$

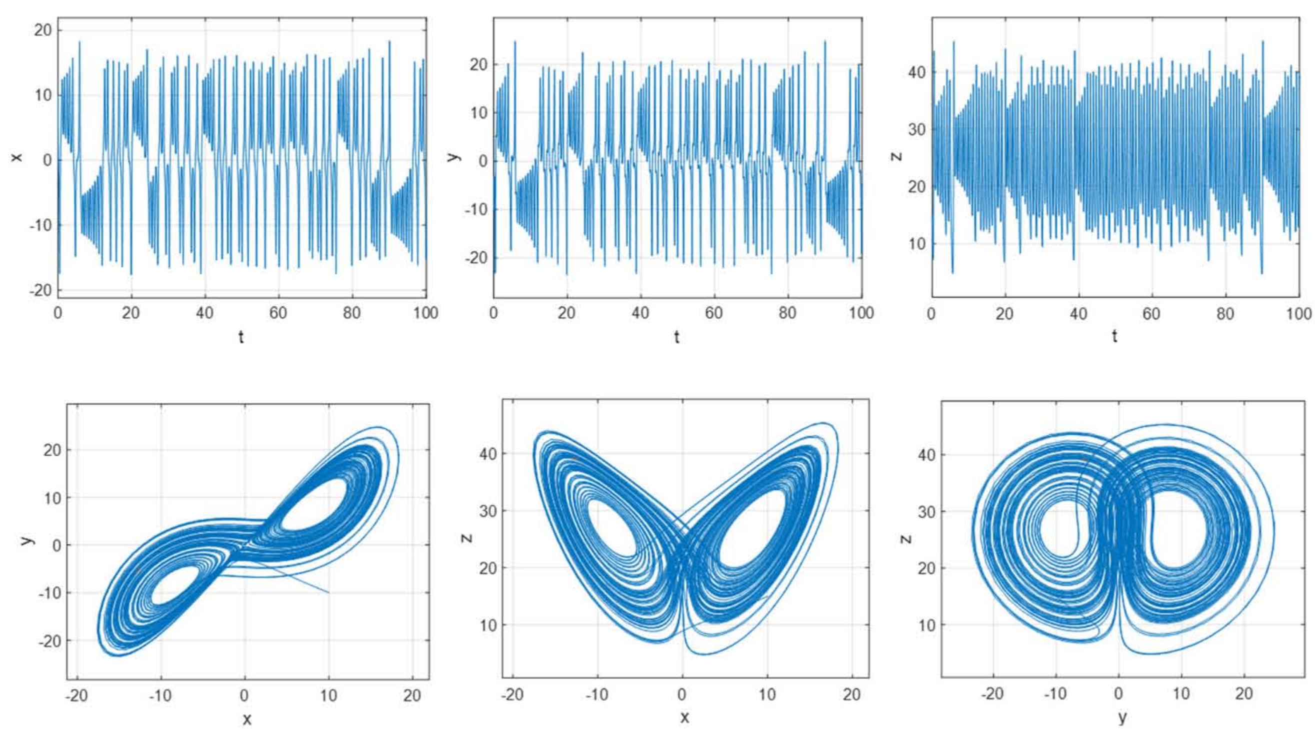

Fig. 1 Time series and phase portraits of Lorenz system for a = 10, b = 8/3, c = 28, initials = {10, −10, 15}

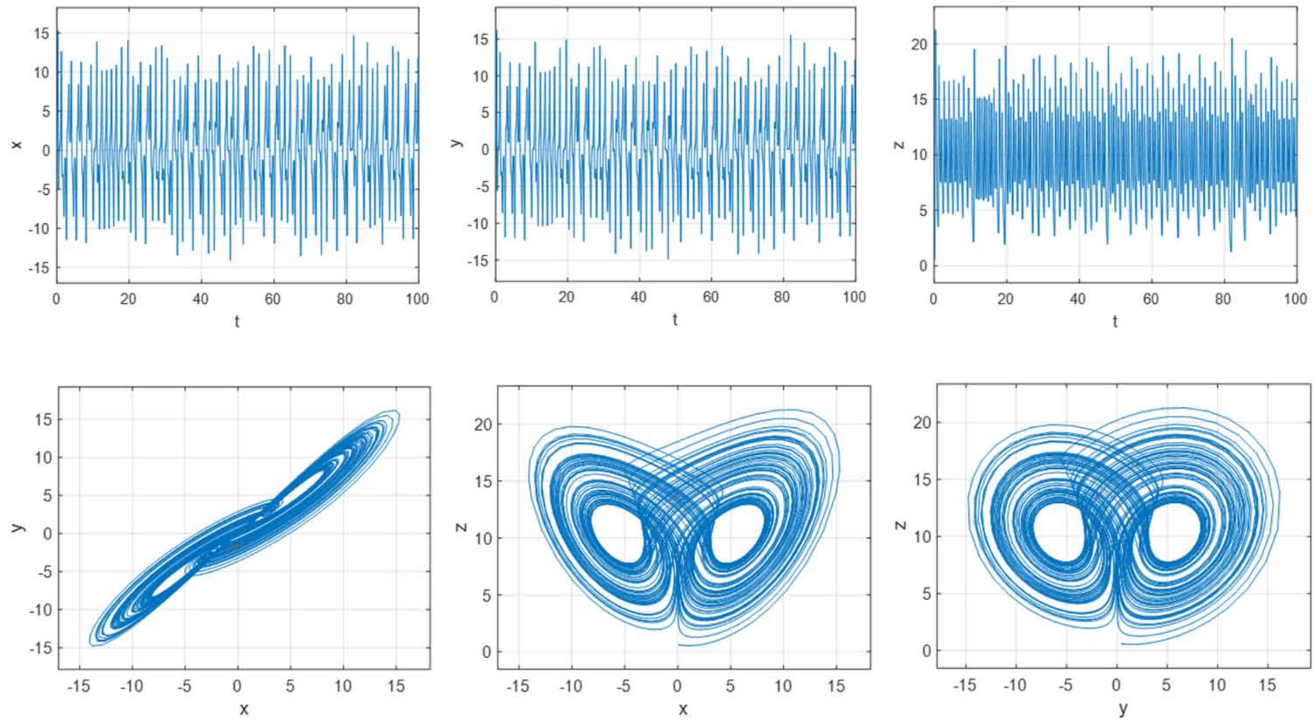

Fig. 2 Time series and phase portraits of Chen system for a = 40, b = 3, c = 25, initials = $\{0.1, 0.5, 0.6\}$

where a, b, and c are system parameters and x, y and z are the system's state variables. The time series plots and the phase portraits of the Chen System can be seen in Fig. 2 for parameter values of $a = 40$, $b = 3$, $c = 25$, and the initials of $x_0 = 0.1$, $y_0 = 0.5$, $z_0 = 0.6$. When the time series and phase portraits in Figs. 1 and 2 and Eqs. 1 and 2 are examined, it is seen that these systems have similar properties, which poses a challenge that should be overcome in the classification of time series.

2.3 Rossler system

The Rossler system, developed by Otto Rossler in 1976, is an advantageous model for modeling chemical reactions [24]. The Rossler System, like the other two systems used in the study, consists of three continuous-time nonlinear ordinary differential equations that present chaotic behavior (Eq. 3).

$$\begin{aligned}\dot{x} &= -y - z\\ \dot{y} &= x + ay\\ \dot{z} &= b + z(x - c),\end{aligned} \tag{3}$$

where a, b, and c are system parameters and x, y and z are the system's state variables. Although Rossler System is similar to the Lorenz system in terms of equations, it is a system that can be analyzed more straightforwardly and has a single scroll. Figure 3 shows the time series and phase portraits of the Rossler System obtained with parameter values of $a = 0.1$, $b = 0.1$, $c = 14$, and the initials of $x_0 = 10$, $y_0 = 10$, $z_0 = 0$. Examining these time series and phase portraits show that they are different from the other two systems, and it will be easier to classify them.

2.4 Data set

In this section, the process followed in obtaining the data set used in the study is explained. While creating the data set, the Runge-Kutta 4 (RK4) algorithm has been used to calculate all systems' state variables and obtain time series. In addition, many results have been obtained by making changes in calculation and systems parameters to diversify the time series data. When calculating the time series, 750 different results for each system were calculated by changing the length of the time series, the step interval of the RK4 algorithm, the system parameter values, and the initial values. Since all systems are 3-dimensional, 2250 time series for each system and 6750 different time series data sets have been obtained for three systems. The calculation and the system parameters used in the study are shown in Table 1.

3 The machine learning methods used in the study

Time series are classified using machine learning algorithms of Lorenz, Chen, and Rossler chaotic systems. In this study, Support Vector Machines (SVM), Naive Bayes (NB), K-Nearest Neighbor (KNN), and Tree algorithms are used to classify the system type by

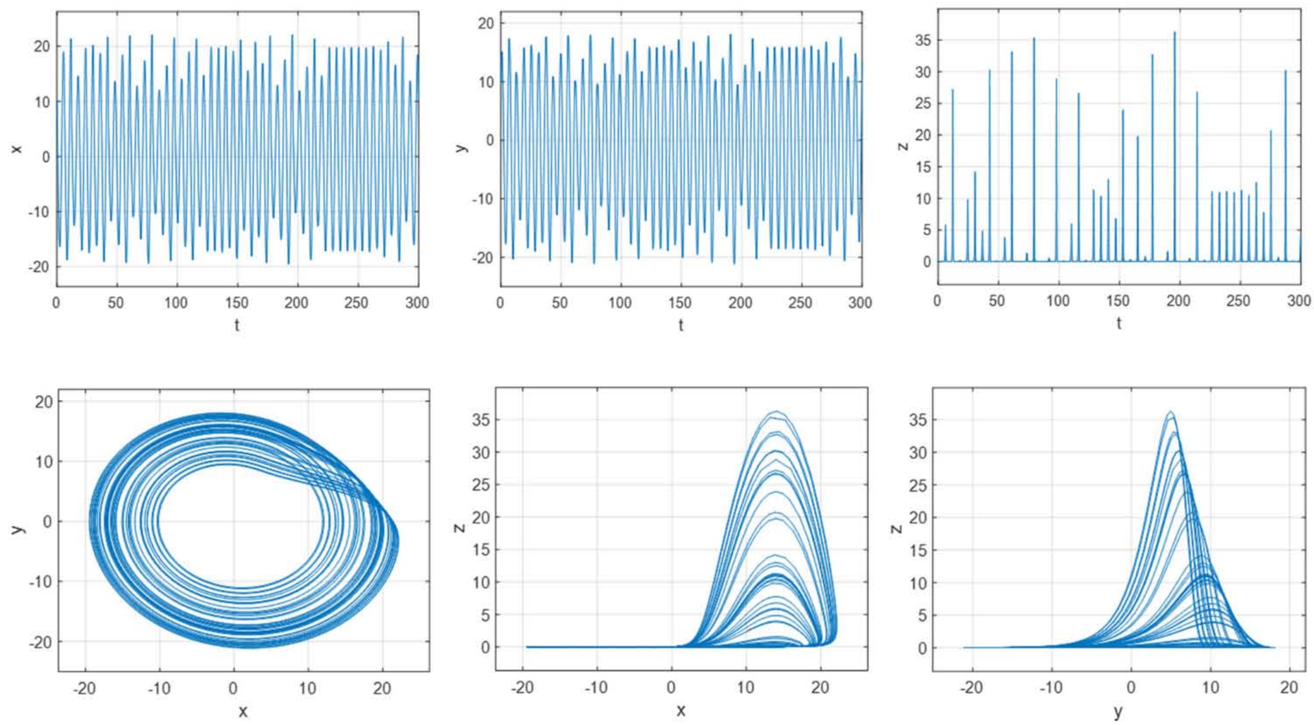

Fig. 3 Time series and phase portraits of Rossler system for a=0.1, b=0.1, c=14, initials={10, 10, 0}

Table 1 The calculation and the system parameters

System	Length of Time Series (The number of calculated values)	Step Interval of RK4 Algorithm	System Parameters (Values of Parameter b for all systems)	Initials
Lorenz	10000, 12500, 15000, 17500, 20000	0.01	b = 2.5, 3, 3.5, 4, 4.5, 5	X_0 = 0.1, 0.2, 0.3, 0.4, 0.5 Y_0 = 0.3, 0.4, 0.5, 0.6,0.7
		0.02		Z_0 = 0.4, 0.5, 0.6, 0.7, 0.8
Chen	20000, 25000, 30000, 35000, 40000			X_0 = 8, 9, 10, 11, 12
		0.05	b = 0.05, 0.1, 0.15, 0.2, 0.25, 0.3	Y_0 = 8, 9, 10, 11, 12 Z_0 = 0, 1, 2, 3, 4
		0.1		X_0 = 8, 9, 10, 11, 12
Rossler	2000, 4000, 6000, 8000, 1000	0.2	b = 2.5, 2.55, 2.6, 2.65, 2.7, 2.75	Y_0 = − 8, − 9, − 10, − 11, − 12 Z_0 = 13, 14, 15, 16, 17

checking the time series data. These algorithms used are briefly explained below.

3.1 Support vector machine

Support vector machine was developed by Vapnik et al. in the 1990s. SVM is used to separate data belonging to two basic classes. It is also an algorithm used to perform classification and regression operations [25]. It allows drawing the decision boundary with the quadratic optimization method on a plane where the data are located and the farthest from the members of the two classes, as shown in Fig. 4 [26].

Each of the data shown in Fig. 4 is computed using the equation in Eq. 4 as follows:

$$\left\{(m_i, n_i)\,|m_i \in R^d, \quad n_i \in \{-1, 1\}\right\}_{i=1}^{a}, \qquad (4)$$

Where m is the input data, and n is a class represented between − *1* and *1*.

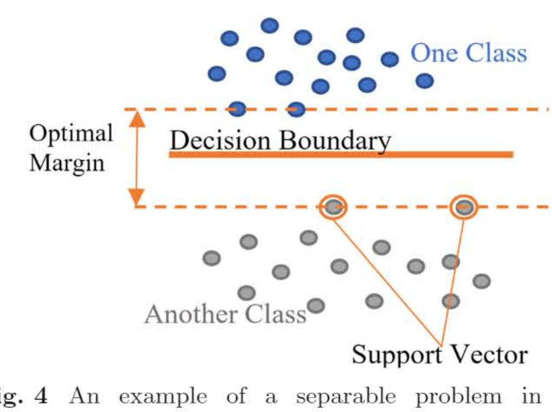

Fig. 4 An example of a separable problem in a 2-dimensional space

3.2 Naive Bayes

It is an algorithm developed by British mathematician Thomas Bayes [27]. NB is one of the most essential and successful learning algorithms in machine learning [28]. It is a probabilistic statistical inference and is used to identify previously created classes [29]. For this purpose, Chen, Lorenz, and Rossler are used in the study to divide the time series of chaotic systems into 3 different classes. NB, the theorem of the algorithm, is shown in Eq. 5. The conditional probability of $P(C_x|I)$ is calculated according to Eq. 5.

$$P(C_x|I) = \frac{P(C_x)\, P(I|C_x)}{P(I)} \tag{5}$$

where $P(C_x|I)$ expresses the probability that the input I sample belongs to the C class. $P(C_x)$ indicates the probability of observing class x, $P(I|C_x)$ is the probability of observing the class C_x, and $P(I)$ refers to the probability of input independent of classes.

3.3 K-nearest neighbor

KNN is a sample-based classifier that allows classification without knowing the probability distributions of classes [30]. Therefore, it is a model that enables data classification to be classified according to the closest point by the K number determined from the previously classified data. The closest points are calculated from the Euclidean distance. The ideal K value differs according to the data that are being studied [1]. The process of determining the closest neighbors can be accelerated using traditional indexing methods.

3.4 Decision tree

It is a data mining classification technique that is frequently used in classification problems [31]. In Decision trees, a decision tree is first created, then the rules generated from the decision trees are classified [32]. Decision trees have a structure consisting of roots, branches, and leaves. It looks like a tree due to its structure. Decision trees start with the root node and split large data sets into smaller groups moving through the leaves. In this tree structure, each element is called a node. Heterogeneous nodes created in these trees are called child nodes, and homogeneous ones are called terminal nodes [33]. Data classification in decision trees consists of two primary stages: learning and classification stages [34]. In the learning phase, the training data set is analyzed by the classification algorithm to create a model, and the learned model is shown as classification rules. In the classification stage, the accuracy of the classification rules is measured using test data. If the accuracy measured is within the acceptable limits, these rules are used to classify new data sets [35].

Table 2 Confusion matrix

Predicted Class	Actual Class	
	Positive	Negative
True	True Positive	True Negative
False	False Positive	False Negative

4 Simulation results and performance evaluation

A data set consisting of time series belonging to Chen, Lorenz, and Rossler chaotic systems and containing x, y, z, and time parameters in each system has been created. Chen dataset with 19831 pieces of data, Lorenz dataset with 17827 pieces of data, and Rossler dataset with 18167 pieces of data, and therefore, a total of 55825 pieces of data have been used. 80% of the created data set has been used for training and 20% for testing and verification. Test and training data have been determined randomly. After the training process, the developed classifier has been tested by the test and verification data. Moreover, the classification achievements have been calculated due to the classification, accuracy, precision, sensitivity, specificity, recall, and F1-score values. Then the results of these measurements were presented with a confusion matrix, scatter plot graphics, and ROC (receiver operating characteristic) curves were drawn to measure the model's success.

The determined classes from the test data and the number of true classes of the systems were compared. The accurate estimation ratio was calculated using the confusion matrix (Table 2) and Eq. 6.

$$Accuracy = \frac{True\,Positive + True\,Negative}{True\,Positive + True\,Negative + False\,Positive + False\,Negative} \tag{6}$$

This ratio shows the general classification accuracy of the classifier. In addition, calculations of the other measurements are given as follows. As shown in Eq. 7, precision is expressed as the ratio of true positive values to the sum of true positive and false positives.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \tag{7}$$

As shown in Eq. 8, sensitivity is expressed as the ratio of true positive values to the sum of true positive and

Table 3 Three-class confusion matrix

Actual Class	Predicted class		
	Chen	Lorenz	Rossler
Chen	True Chen (TC)	False Chen-1(FC-1)	False Chen-2(FC-2)
Lorenz	False Lorenz-1(FL-1)	True Lorenz (TL)	False Lorenz-2(FL-2)
Rossler	False Rossler-1(FR-1)	False Rossler-2(FR-2)	True Rossler(TR)

Table 4 Conversion table of three-class confusion matrix to 2-class confusion matrix

Parameters	Classes		
	Chen	Lorenz	Rossler
True Positive	TC	TL	TR
False Positive	FC-1 + FC-2	FL-1 + FL-2	FR-1 + FR-2
True Negative	TL + TR + FL-2 + FR-2	TC + TR + FC-2 + FR-1	TC + TL + FC-1 + FL-1
False Negative	FL-1 + FR-1	FC-1 + FR-2	FC-2 + FL-2

Fig. 5 Confusion matrix of three classes, **a** KNN Algorithm, **b** SVM Algorithm, **c** Tree Algorithm, **d** Naive Bayes Algorithm

(a)

True Class	Predicted Class		
	Chen	Lorenz	Rossler
Chen	**1949**	21	3
Lorenz	17	**1765**	1
Rossler	4	0	**1812**

(b)

True Class	Predicted Class		
	Chen	Lorenz	Rossler
Chen	**1953**	20	0
Lorenz	55	**1728**	0
Rossler	5	0	**1811**

(c)

True Class	Predicted Class		
	Chen	Lorenz	Rossler
Chen	**1857**	111	5
Lorenz	115	**1665**	3
Rossler	11	2	**1803**

(d)

True Class	Predicted Class		
	Chen	Lorenz	Rossler
Chen	**1823**	147	3
Lorenz	371	**1409**	3
Rossler	46	44	**1726**

false negatives.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{False Negative} + \text{True Positive}}. \quad (8)$$

As shown in Eq. 9, specificity is expressed as the ratio of true negative values to the sum of true negative and false positives.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}}. \quad (9)$$

As shown in Eq. 10, recall is expressed as the ratio of true positive values to the sum of true positive and false negatives.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \quad (10)$$

The F1-score, shown in Eq. 11, is expressed as the ratio of *2*True positive* values to the sum of false positives, false negatives, and *2*True positive*.

$$F1 - \text{score} = \frac{2 * \text{True Positive}}{2 * \text{True Positive} + \text{False Positive} + \text{False Negative}}. \quad (11)$$

Table 5 Conversion table of three-parameter confusion matrix to two-parameter confusion matrix

Parameters	Chen				Lorenz				Rossler			
	KNN	SVM	TREE	NB	KNN	SVM	TREE	NB	KNN	SVM	TREE	NB
True Positive	1949	1953	1857	1823	1765	1728	1665	1409	1812	1811	1803	1726
True Negative	24	20	116	150	18	55	118	374	4	5	13	90
False Positive	3578	3539	3473	3182	3768	3769	3676	3598	3752	3756	3748	3750
False Negative	21	60	126	417	21	20	113	191	4	0	8	6

Table 6 Classification achievements of models, a) KNN Algorithm, b) SVM Algorithm, c) Tree Algorithm, d) Naive Bayes Algorithm

Class	Accuracy	precision	sensitivity	specificity	Re-call	F1-Score
(a)						
Chen	0.9919	0.9878	0.9893	0.9933	0.9893	0.9886
Lorenz	0.9930	0.9899	0.9892	0.9952	0.9882	0.9891
Rossler	0.9986	0.9978	0.9978	0.9989	0.9978	0.9978
(b)						
Chen	0.9856	0.9899	0.9702	0.9944	0.9702	0.9799
Lorenz	0.9865	0.9692	0.9886	0.9856	0.9886	0.9788
Rossler	0.9986	0.9972	1.0000	0.9987	1.0000	0.9986
(c)						
Chen	0.9566	0.9412	0.9365	0.9677	0.9365	0.9388
Lorenz	0.9585	0.9338	0.9364	0.9689	0.9364	0.9351
Rossler	0.9962	0.9928	0.9956	0.9965	0.9956	0.9942
(d)						
Chen	0.8982	0.9240	0.8138	0.9550	0.8138	0.8654
Lorenz	0.8986	0.7902	0.8806	0.9058	0.8806	0.8330
Rossler	0.9828	0.9504	0.9965	0.9766	0.9965	0.9729

Table 7 Total performance results of the three-class system

Classifier	Accuracy %
Naive Bayes	89.0
SVM	98.6
KNN	**99.2**
Tree	95.6

Bold letter indicates the best performance result

Since different classes have been used in the study, the confusion matrix given in Table 2 was recreated as shown in Table 3.

Table 3 was created to convert a three-class confusion matrix into a two-class confusion matrix [36]. Table 4 was obtained using Table 3 to calculate the accuracy, precision, sensitivity, specificity, recall, and F1-score classification performance of each class.

The confusion matrices obtained as a result of the study are shown in Fig. 5. The confusion matrix obtained with the KNN algorithm is shown in Fig. 5a. The confusion matrix obtained by the SVM algorithm is given in Fig. 5b, the confusion matrix obtained by the Tree algorithm is shown in Fig. 5c, and the confusion matrix obtained by the NB algorithm is presented in Fig. 5d.

A conversion table of the three-class confusion matrix to 2-class confusion matrix, presented in Table 4, was used to calculate the performance metrics easily. Additionally, using Tables 4, 5 was obtained.

Table 8 Accuracy rates according to classification types

Classifier	Accuracy %
Naive Bayes	
Gaussian Naive Bayes	88.3
Kernel Naive Bayes	**89.0**
Support Vector Machine	
Linear SVM	89.0
Quadratic SVM	97.2
Cubic SVM	93.0
Fine Gaussian SVM	**98.6**
Medium Gaussian SVM	97.2
Coarse Gaussian SVM	94.3
K-Nearest Neighbor	
Fine KNN	**99.2**
Medium KNN	98.6
Coarse KNN	96.7
Cosine KNN	96.0
Cubic KNN	98.6
Weighted KNN	98.8
Decision Treee	
Fine Tree	**95.6**
Medium Tree	91.7
Coarse Tree	89.2

Bold letter indicates the best performance result

The values in Table 6 were obtained using accuracy, precision, sensitivity, specificity, recall, and F1-score performance results calculated according to the equations given in Eqs. 6, 7, 8, 9, 10, and 11 according to the values given in Table 5. The total performance of the classification algorithms used in the study for three classes is generally calculated using Eq. 12.

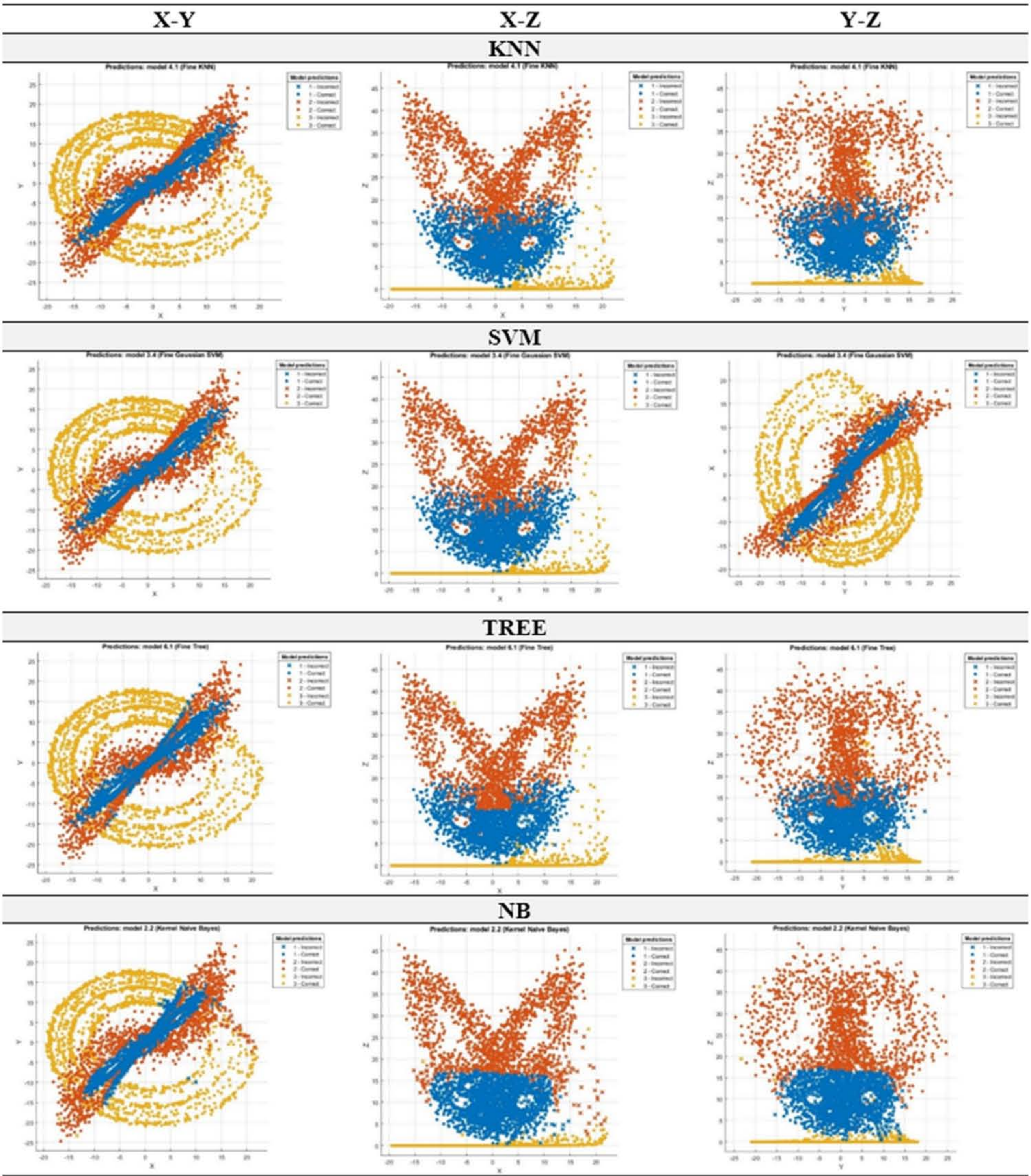

Fig. 6 Graphs of data classified as true and false by algorithms

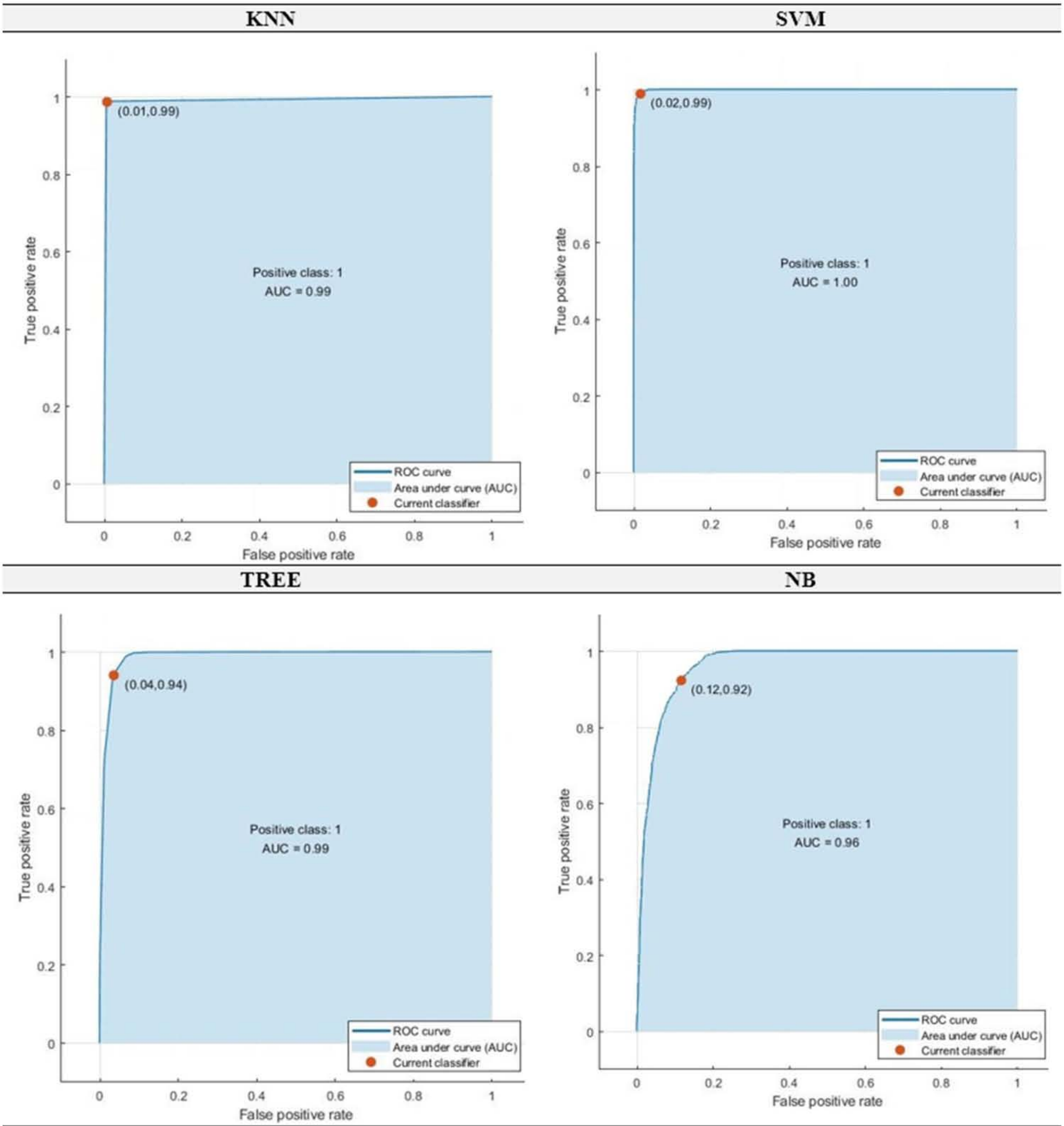

Fig. 7 ROC curves of the classification algorithms

$$\text{Accuracy} = \frac{\text{TC} + \text{TL} + \text{TR}}{\text{TC} + \text{TL} + \text{TR} + \text{FC} - 1 + \text{FC} - 2 + \text{FL} - 1 + \text{FL} - 2 + \text{FR} - 1 + \text{FR} - 2} \tag{12}$$

The achievements of the system for the three classifications obtained using Eq. 12 are presented in Table 7.

The results obtained in Table 7 are the best in terms of the methods explained above. In addition, there are different sub-methods among the four used methods. The results obtained with these sub-methods are presented in Table 8. When Table 8 is examined, it is seen that the best results obtained are consistent with Table 7.

Figure 6 shows the graphics of the data classified as true and false by the algorithms. The graphics of correctly and incorrectly classified data for three classes obtained using the KNN algorithm, SVM algorithm, NB algorithm, and Tree algorithm are shown in Fig. 6a–d, respectively.

In addition, ROC curves were also used to show the performance of the study with different graphics. ROC curves are calculated as the ratio of sensitivity to precision. Classification algorithms try to strike a balance between sensitivity and precision. Therefore, ROC curves are often used to evaluate the balance between sensitivity and precision.

The area under the ROC curve is called Area Under Curve (AUC). The value of this field gives the ROC score and when this value approaches to 1, the positives are successfully separated from the negatives.

The approach of the ROC curve to the upper left corner indicates that the correct positive rate is high and the area under the curve is significant. It can be used to see whether the positives are separated from the negatives successfully.

The ROC curves given in Fig. 7 show that the ROC curve is closer to the upper left corner than the others, according to the result obtained with the KNN algorithm, so the correct positive rate is high and the area under the curve is high. As a result, positives do not seem to be successfully separated from negatives. It is seen that SVM, TREE, and NB have the most significant area under the ROC curve, respectively.

5 Conclusions

In this study, a novel approach was presented, and time series of three different chaotic systems were classified with high accuracy using machine learning methods, which are frequently used today. Lorenz, Chen, and Rossler systems, three of the most widely known systems in the literature, were used for this study. The data set used in the study was obtained by solving these systems with the RK4 method. This data set was obtained for different step intervals, initial values, system parameters, and time intervals, and therefore, a variety of data was created. In particular, systems with similar dynamic properties and time series, such as Lorenz and Chen, were correctly classified. Along with achieving relatively high accuracy in each method, the highest accuracy rate was achieved as 99.2% with the Fine KNN method. In addition, this study shows a way for classifying signals and data with chaotic or random characters and associating them with a mathematical model.

References

1. O. Sevli, Performance Comparison of Different Machine Learning Techniquesin Diagnosis of Breast Cancer. Eur. J. Sci. Technol. **16**, 176–185 (2019)
2. J. A. Cruz and D. S. Wishart, “Applications of machine learning in cancer prediction and prognosis,” *Cancer Informatics*, vol. 2. Libertas Academica Ltd., pp. 59–77, 2006
3. Nonlinear forecasting for the classification of natural time series, *Philos. Trans. R. Soc. London. Ser. A Phys. Eng. Sci.*, vol. 348, no. 1688, pp. 477–495, Sep. 1994
4. M. Casdagli, Nonlinear prediction of chaotic time series. Phys. D Nonlinear Phenom. **35**(3), 335–356 (1989)
5. Y. C. Lai and N. Ye, Recent developments in chaotic time series analysis, *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 13, no. 6. World Scientific Publishing Co. Pte Ltd, pp. 1383–1422, 2003
6. S. Mukherjee, E. Osuna, and F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop*, 1997, pp. 511–520
7. K. He, X. Zhang, S. Ren, and J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, vol. 2015 Inter, pp. 1026–1034
8. J.B. Elsner, A.A. Tsonis, Nonlinear Prediction, Chaos, and Noise. Bull. Am. Meteorol. Soc. **73**(1), 49–60 (1992)
9. T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi, Time series forecasting using a deep belief network with restricted Boltzmann machines. Neurocomputing **137**, 47–56 (2014)
10. Z. Wang, W. Yan, and T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, vol. 2017-May, pp. 1578–1585
11. J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, *Phys. Rev. Lett.*, vol. 120, no. 2, Jan. 2018
12. J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, *Chaos*, vol. 27, no. 12, Dec. 2017
13. N. Boullé, V. Dallas, Y. Nakatsukasa, D. Samaddar, Classification of chaotic time series with deep learning. Phys. D Nonlinear Phenom. **403**, 132261 (2020)
14. L. Zhang, EEG Signals Classification Using Machine Learning for the Identification and Diagnosis of Schizophrenia, in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2019, pp. 4521–4524
15. E. Sayilgan, Y. Kemal, Y.Y. Isler, Classification of Hand Movements from EEG Signals using Machine Learning Techniques, in Proceedings -, Innovations in Intelligent

Systems and Applications Conference. ASYU **2019**, 2019 (2019)
16. P. Shimpi, S. Shah, M. Shroff, and A. Godbole, A machine learning approach for the classification of cardiac arrhythmia, in *Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017*, 2018, vol. 2018-January, pp. 603–607
17. E. N. Lorenz, Deterministic nonperiodic flow, in *Universality in Chaos, Second Edition*, 2017, pp. 367–378
18. M. Gorman, P.J. Widmann, K.A. Robbins, Nonlinear dynamics of a convection loop: A quantitative comparison of experiment with theory. Phys. D Nonlinear Phenom. **19**(2), 255–267 (1986)
19. H. Haken, Analogy between higher instabilities in fluids and lasers. Phys. Lett. A **53**(1), 77–78 (1975)
20. K.M. Cuomo, A.V. Oppenheim, Circuit implementation of synchronized chaos with applications to communications. Phys. Rev. Lett. **71**(1), 65–68 (1993)
21. N. Hemati, Strange Attractors in Brushless DC Motors, IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **41**(1), 40–45 (1994)
22. E. Knobloch, Chaos in the segmented disc dynamo. Phys. Lett. A **82**(9), 439–440 (1981)
23. G. Chen, T. Ueta, Yet another chaotic attractor. Int. J. Bifurc. Chaos **9**(7), 1465–1466 (1999)
24. O.E. Rössler, An equation for continuous chaos. Phys. Lett. A **57**(5), 397–398 (1976)
25. C. Cortes, V. Vapnik, Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
26. L. Shen et al., Evolving support vector machines using fruit fly optimization for medical data classification. Knowl.-Based Syst. **96**, 61–75 (2016)
27. A. Wood, V. Shpilrain, K. Najarian, and D. Kahrobaei, Private naive bayes classification of personal biomedical data: Application in cancer data analysis, *Artic. Comput. Biol. Med.*, 2018
28. A. Tekerek, Support vector machine based spam SMS detection. J. Polytech. **22**(2), 779–784 (2018)
29. W.A. Awad, S.M. Elseuofi, Machine Learning methods for E-mail Classification. Int. J. Comput. Appl. **16**(1), 39–45 (2011)
30. S. Theodoridis, K. Koutroumbas, *Pattern Recognition and Neural Networks, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2049 (Springer Verlag, LNAI, 2001), pp. 169–195
31. M.Ç. Aksu, E. Karaman, Karar Ağaçlarıile Bir Web Sitesinde Link Analizi ve Tespiti. Acta Infologica **1**(2), 84–91 (2017)
32. G. Silahtaroğlu, *Veri madenciliği* (Papatya Yayıncılık Eğitim A.Ş, İstanbul, 2013)
33. G. Pehlivan, *CHAID Analizi ve Bir Uygulama* (Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, FBE İstatistik Anabilimdalı, Yüksek Lisans Tezi, 2006)
34. Ş Demirel, S.G. Yakut, Decision Tree Algorithms and an Application on Child Labor. Sos. Bilim. Araştırma Derg. **8**(4), 52–65 (2019)
35. S. Özekes, Veri Madenciliği Modelleri ve Uygulama Alanları. İstanbul Ticaret Üniversitesi Derg. **3**, 65–82 (2003)
36. M. Felkin, Comparing classification results between N-ary and binary problems. Stud. Comput. Intell. **43**, 277–301 (2007)