DOI:10.53992/njns.v9i1.154 **NUST Journal of Natural Sciences, Vol. 9, Issue 1, 2024**

Modeling a fractal-based C Beizer function for predicting underlying data pattern

Tayba Arooj[1], Farheen Ibraheem[2], Faira Kanwal Janjua [2]

[1]Lahore College for Women University, Lahore, Pakistan
[2]Forman Christian College, A Chartered University (FCCU), Lahore-Pakistan.

***Corresponding author's email**: tayba.arooj@lcwu.edu.pk

Abstract

As artificial intelligence advances, more and more tasks that formerly required human discretion can be automated. The proposed study aims to create an automated supervised, hybrid computing algorithm for the synthesis and analysis of engineering and scientific data and gaining insightful knowledge about the data. A novel iterated function approach has been designed by integrating rational C Bezier function with classical fractal function. Sufficient conditions on the scaling and shape factors have been calculated to obtain various simulating patterns occurring in data. The developed model is trained and tested on a set of data values to predict prevalent shape characteristics of the data. Numerical examples validate the suggested approach.

Keywords: Machine Learning, Artificial Intelligence, Fractals, Computing Algorithm, Prediction

Article History: **Received:** 21 September 2023, **Revised:** 24 March 2024, **Accepted**: 25 April 2024, **Published:** 30 May 2024.

Introduction

In the current, Fourth Industrial Revolution (4IR or Industry 4.0) era, the digital world is inundated with data, such as mobile data, Internet of Things (IoT) data, cybersecurity data, business data, health data, social media data, etc. [1]. A variety of intelligent applications can be built within the pertinent domains by utilizing the insights that can be extracted from this data. For example, cybersecurity data can be used to create an automated and intelligent cybersecurity system [2]. To perform proficient analysis of this data, to construct intelligent automated applications artificial intelligence (AI) techniques are extensively applied. Artificial intelligence (AI) has evolved significantly in the past few years in the realms of data analysis and computing, which often enables the applications to perform intelligently [3].

Artificial intelligence (AI) and Pattern recognition (PR) technologies have a significant impact on many facets of our daily lives, and their importance is rapidly expanding. These technologies have real-world applications in a variety of fields, including but not limited to Chatbots, image recognition, self-driving cars, are creation, gaming, and predictive analysis [4]. This demonstrates the diverse range of

applications where PR and AI technologies make a significant impact, enhances efficiency, accuracy, and personalization in numerous industrial and everyday aspects of life.

Predicting patterns is a vital aspect of diverse domains, including cognitive psychology, computer science and machine learning. It allows us to understand complex systems and make informed decisions based on available data. It is a key component of many machines learning algorithm that seek to find patterns in big data sets. These algorithms use machine learning techniques to identify patterns and make predictions based on those algorithms. Several machine learning methods have been used over the years to fit collections of training datasets with attributes to predict such characteristics on sets of unknown data. There is no optimal method for every optimization issue, according to the "No Free Lunch theorem," which was put forth in 1997 by Wolpert and Macready [5]. An approach that excels on one problem may utterly fails on another. There are numerous machine learning methods available currently for predicting a function *f(x)*. One of the earliest techniques was polynomial based technique, which is still widely used for scientific data and in industries like digital photography and image resampling.

There is a long and successful history to the fundamental challenge of predicting patterns in data. For instance, Tycho Brahe's comprehensive astronomical observations throughout the 16th century enabled Johannes Kepler to deduce the empirical laws governing planetary motion, which later served as a foundation for the invention of classical mechanics. At the turn of twentieth century analysis of irregularities of atomic spectra resulted in development of quantum physics. The automatic detection of regularities in data using computer algorithms is the primary concern of the field of pattern recognition and using these patterns to perform actions like categorizing the data into various groups or making predictions among few of these activities.

Although Gaussian processes (GPs) were developed in the 1940s [6], it wasn't until 1978 that they were used to specify prior distributions over functions. Gaussian processes have begun to be employed for regression problems [7] and supervised machine learning more recently because of the advent and rising popularity of neural networks with back propagation. Many attempts have been made in recent years to enhance established methods, most notably by the University of Chicago team led by Robert B. Gramacy, who introduced treed Gaussian processes [8] and dynamic trees [9]. In 1996, Radford Neal demonstrated that a Bayesian neural network with an infinite hidden node count and a Gaussian prior on individual weights converges to a GP [10].

A rational spline with cubic polynomial as numerator and quadratic polynomial as denominator of the fractal form using iterative functions have been developed by Balasubramani [13]. The uniform error limit was calculated to be of C^2.

In the realms of Computer aided geometric design, curves with minimum bending energy or with minimum arc length are considered fair. The “Circle transition problem” for Quadratic Bezier functions have been investigated in [12] and expressions for arc length and minimum bending energy were developed.

Shape conserving Iterative system [13] has been built using a cubic/quadratic rational spline and conditions on shape and scaling parameter have been computed. Infinite numbers of Fractal interpolants can be generated by varying shape and scaling parameter. An approximation technique for self-similar wiggly functions has been

developed in [14]. Pattern identification and preservation is attained by adjusting scaling factors. Barnsley [14] introduced iterated function system on continuous function $F: I \to R$, which interpolates the data a, $F(e_i) = f_i$ for $i \in \{0,1,2,..,n\}$ where $I = [e_0, e_n]$ and $f_i \in R$. The attractors of iterative function were the graph of this continuous function. Some important results for fractal function were proposed like existence, coding, theory, and functional equation. By fluctuating functions, the geometrical applications of

nature was described. Some kind of self-similarity under magnification such as in cloud tops and mountain ranges were described. The problem of fractal interpolation has been solved by using harmonic functions in [15].

The renowned subdivision algorithm namely de Casteljau has been used in [16] to establish that Bezier curves are fractals and new techniques to render these curves have been developed. In [17], Beizer curves of rational form have been used for interpolation and shape modification. Beizer curves with trigonometric basis functions and two shape parameters have been developed in [18] for the problem of interpolation.

The problem of presenting inherent data characteristic has been dealt in [19] with a piece wise rational cubic technique. The desired shape characteristics are achieved by four shape control parameters in the description of rational function. The author in [20] also worked on shape conserving interpolating function and established a rational technique to visualize positive data set.

Three main shape features of data that is convexity, monotonicity and positivity has been discussed in [21] and interpolation techniques based on trigonometric quadratic function following geometric continuity have been developed and implemented successfully. Here again authors used two shape parameters to modify and control the shape of the curve.

In Kocić L.M et.al [22], self-similarity embedded in subdivision property of Bernstein polynomial proved to be faster to evaluate Bezier curves. The more general type of classical interpolating function has been replaced by iterated fractal interpolation function in [23]. A more general form of Mazel approach has been developed in [24]. A rational Beizer curve with affine invariant characteristic meeting generalized monotonicity has been purposed in [25].

Section 2 thoroughly describes the process underwent for the development of supervised computing algorithm to detect and predict hidden data patterns. A Cubic C Beizer fractal function with four shape parameters has been constructed to identify underlying data patterns and make predictions which satisfies data attributes. Algorithms to identify underlying shape characteristics have been developed in Section 3 and 4. Four types of different data sets, in Section 5 have been chosen to apply the proposed approach. Section 6 discusses conclusion and future directions. A typical supervised learning flow diagram concludes the current section.

Materials and Methods

In this section, the novel model construction is presented with the help of rational cubic C Bezier function. and

classical fractal theory. Four new shape control parameters are introduced in this method to simulate the patterns actively. Furthermore, scale parameters are used to adjust the irregularity.

Model Formulation

Consider the data points $\{(e_i, f_i),\ i = 0,1,2,\dots,n\}$ defined over the closed interval $[a, b]$ where $a = e_0 < e_1 < e_2 < \cdots < e_n = b$. The rational cubic C-Bezier curve defined over each sub-interval $I = [e_i, e_{i+1}]$, $i = 0,1,2,3,\dots,n-1$ with four shape parameters is represented as follows.

$$r_i(e) = \frac{p(t)}{q(t)} \qquad (2.1)$$

where,

$$p(t) = \sum_{i=0}^{3} V_i U_i(t) Q_i \qquad (2.2)$$

$$q(t) = \sum_{i=0}^{3} V_i U_i(t) \qquad (2.3)$$

and

$$t = \frac{\pi}{2}\left(\frac{e-e_i}{h_i}\right), t \in \left[0, \frac{\pi}{2}\right], \quad h_i = e_{i+1} - e_i,$$

then equation (2.1) becomes

$$r_i(e) = \frac{\sum_{i=0}^{3} V_i U_i(t) Q_i}{\sum_{i=0}^{3} V_i U_i\,(t)} \qquad (2.4)$$

where the basic functions are defined as:

$$U_0(t) = \frac{2}{(\pi-2)}\left(-cost - t + \left(\frac{\pi}{2}\right)\right),$$
$$U_1(t) = \frac{2}{(\pi-2)(4-\pi)}\big((2-\pi)sint + 2cost + 2t - 2\big),$$
$$U_2(t) = \frac{2}{(\pi-2)(4-\pi)}\big(2sint + (2-\pi)cost - 2t + (\pi-2)\big),$$
$$U_3(t) = \frac{2}{(\pi-2)}(-sint + t),$$

$Q_i^{'}$s are the control points and $V_i^{'}$s, i = 0, 1, 2, 3 are weight parameters.

The fractal interpolator function is,

$$\Psi\big(T_i(e)\big) = \alpha_i \Psi(e) + r_i(e) \qquad (2.5)$$

And derivative of fractal interpolation function is,

$$a_i \Psi'\big(T_i(e)\big) = \alpha_i \Psi'(e) + r_i'(e) \qquad (2.6)$$

Also, following interpolating properties hold

$$\Psi(e_i) = f_i \qquad (2.7)$$
$$\Psi(e_{i+1}) = f_{i+1} \qquad (2.8)$$
$$\Psi^{(1)}(e_i) = d_i \qquad (2.9)$$
$$\Psi^{(1)}(e_{i+1}) d_{i+1} \qquad (2.10)$$

Where $\Psi^{(1)}$ represents the first derivative with respect to e and d_i denotes the derivative value at the knots e_i.

The interpolator conditions (2.7), (2.8), (2.9), (2.10) yields the following results.

$$\Psi\big(T_i(e_1)\big) = \alpha_i \Psi(e_1) + \frac{p_i(0)}{q_i(0)} \qquad (2.11)$$
$$\Rightarrow U_0 = f_i - \alpha_i f_1$$
$$\Psi\big(T_i(e_n)\big) = \alpha_i \Psi(e_n) + \frac{p_i(1)}{q_i(1)} \qquad (2.12)$$
$$\Rightarrow\ U_3 = f_{i+1} - \alpha_i f_n,$$

$$a_i = \alpha_i \Psi'(e_1) + \frac{q_i(0)p_i'(0) - q_i'(0)p_i(0)}{(q_i(0))^2} \qquad (2.13) \Rightarrow$$
$$U_1 = (f_i - \alpha_i f_1) + \frac{V_0(\pi-2)}{\pi V_1}(h_i d_i - \alpha_i(e_n - e_1)d_1)$$

$$a_i = \alpha_i \Psi'(e_n) + \frac{q_i(1)p_i'(1) - q_i'(1)p_i(1)}{(q_i(1))^2} \qquad (2.14)$$

$$\Rightarrow U_2 = (f_{i+1} - \alpha_i f_n) - \frac{V_3(\pi-2)}{\pi V_2}\big(h_i d_{i+1} - (e_n - e_1)\big)$$

Hence, the fractal interpolation function is.

$$\Psi(T_i(e)) = \alpha_i \Psi(e) + \frac{p(t)}{q(t)} \quad (2.15)$$

Where

$$p(t) = V_0U_0(f_i - \alpha_i f_1) + V_1U_1\left((f_i - \alpha_i f_1) + \frac{V_0(\pi-2)}{\pi V_1}(h_i d_i - \alpha_i(e_n - e_1)d_1)\right) + V_2U_2\left((f_{i+1} - \alpha_i f_n) - \frac{V_3(\pi-2)}{\pi V_2}(h_i d_{i+1} - \alpha_i(e_n - e_1)d_n)\right) \quad (2.16)$$

$$q(t) = V_0U_0 + V_1U_1 + V_2U_2 + V_3U_3\,,$$

$$t = \frac{\pi}{2}\left(\frac{e-e_1}{e_n-e_1}\right).$$

Generating Algorithm for Computational Model 1 (Positive Data Set)

The data set, which is positive $\{(e_i, f_i)\ with\ e_0 < e_1 < e_2 < \cdots < e_n\}$ has been considered for the problem under consideration. The positivity of cubic function only depends upon positivity of $p(t)$ because $q(t)$ is always positive for all values of parameters.

Consider,

$$p(t) = \sum_{i=0}^{3} V_iU_i(t)Q_i = V_0U_0Q_0 + V_1U_1Q_1 + V_2U_2Q_2 + V_3U_3Q_3$$

$$p(t) = A_0U_0 + A_1U_1 + A_2U_2 + A_3U_3$$

Where,

$$A_0 = V_0Q_0 = V_0(f_i - \alpha_i f_1),$$

$$A_1 = V_1Q_1 = V_1\left((f_i - s_i f_1) + \frac{V_0(\pi-2)}{\pi V_1}(h_i d_i - \alpha_i(e_n - e_1)d_1)\right),$$

$$A_2 = V_2Q_2 = V_2\left((f_{i+1} - \alpha_i f_n) - \frac{V_3(\pi-2)}{\pi V_2}(h_i d_{i+1} - (e_n - e_1)\alpha_i d_n)\right),$$

$$A_3 = V_3Q_3 = V_3(f_{i+1} - \alpha_i f_n),$$

$p(t)$ will be positive if A_0, A_1, A_2, A_3 will be positive. Now,

$A_0 = V_0(f_i - \alpha_i f_1) > 0$
implies $\alpha_i < \frac{f_i}{f_1}$
Also, if

$$A_1 = V_1\left((f_i - \alpha_i f_1) + \frac{V_0(\pi-2)}{\pi V_1}(h_i d_i - \alpha_i(e_n - e_1)d_1)\right) > 0$$

Then

$$V_1 > \frac{\frac{-V_0(\pi-2)}{\pi}(h_i d_i - \alpha_i(e_n-e_1)d_1)}{f_i - \alpha_i f_1}$$

Likewise, $A_2 = V_2\left((f_{i+1} - \alpha_i f_n) - \frac{V_3(\pi-2)}{\pi V_2}(h_i d_{i+1} - (e_n - e_1)\alpha_i d_n)\right) > 0$

$$V_2 > \frac{\frac{V_3(\pi-2)}{\pi}(h_i d_{i+1} - (e_n-e_1)\alpha_i d_n)}{f_{i+1} - \alpha_i f_n}$$

And $A_3 = V_3(f_{i+1} - \alpha_i f_n) > 0$

implies $\alpha_i < \frac{f_{i+1}}{f_n}$ and

$$0 < \alpha_i < \min\left\{\frac{f_i}{f_1}, \frac{f_{i+1}}{f_n}\right\}$$

The above construction returns the following condition on shape parameters.

The positivity of rational cubic C-Bezier curve holds over the interval $[a, b]$ if it fulfils the subsequent sufficient conditions in each subinterval $I_i = [e_i, e_{i+1}]$.

$$V_0 > 0\,, V_3 > 0$$

$$V_1 > \max\left\{0, \frac{\frac{-V_0(\pi-2)}{\pi}[h_i d_i - \alpha_i(e_n-e_1)d_1]}{f_i - \alpha_i f_1}\right\},$$

$$V_2 > max\left\{0, \frac{\frac{V_3(\pi-2)}{\pi}(h_i d_{i+1} - (e_n-e_1)\alpha_i d_n)}{f_{i+1} - \alpha_i f_n}\right\}$$

and $0 < \alpha_i < \min\left\{\frac{f_i}{f_1}, \frac{f_{i+1}}{f_n}\right\}$.

Equivalently, shape constraints can be reorganized as follows:

$$V_0 > 0\,, V_3 > 0,$$

$$V_1 = \gamma_i \max\left\{0, \frac{\frac{-V_0(\pi-2)}{\pi}[h_i d_i - \alpha_i(e_n - e_1)d_1]}{f_i - \alpha_i f_1}\right\},$$

$$\gamma_i > 0$$

$$V_2 = \sigma_i + max\left\{0, \frac{\frac{V_3(\pi-2)}{\pi}(h_i d_{i+1} - (e_n - e_1)\alpha_i d_n)}{f_{i+1} - \alpha_i f_n}\right\},$$

$$\sigma_i > 0$$

$$0 < \alpha_i < \min\left\{\frac{f_i}{f_1}, \frac{f_{i+1}}{f_n}\right\}$$

For preserving the shape of the curve, the parameters $V_1 and\ V_2$ to be used. While to adjust and amend the shape of the curves, parameters $V_0, V_3\ and\ \alpha_i$ will be used.

Generating Algorithm for Computational Model 2 (Monotone Data Set)

Consider the data set points $\{(e_0, f_0), (e_1, f_1), (e_2, f_2), \dots, (e_n, f_n)\}$ where $e_0 < e_1 < e_2 < \cdots < e_n$ and $f_i > 0$ also $f_i < f_{i+1},\ \Delta_i > 0,\ d_i > 0$

Since,

$$a_i \Psi'\left(T_i(e)\right) = \alpha_i \Psi'(e) + r_i'(e) \quad (4.1)$$

where,

$$(e) = \frac{p^{(1)}(t)q(t) - p(t)q^{(1)}(t)}{(q(t))^2} \quad (4.2)$$

The given function is monotonicity increasing if $\Psi'\left(T_i(e)\right) > 0$, for this we will show that $r_i'(e) > 0$.

Where,

$$p(t) = \sum_{i=0}^{3} V_i U_i(t) Q_i\,,$$

$$q(t) = \sum_{i=0}^{3} V_i U_i\,(t),$$

$$p^{(1)}(t) = \sum_{i=0}^{3} V_i U_i^{(1)}(t) A_i,$$

$$q^{(1)}(t) = \sum_{i=0}^{3} V_i U_i^{(1)}\,(t)$$

And

$$U_0^{(1)}(t) = \frac{2}{(\pi-2)}(\sin t - 1),$$

$$U_1^{(1)}(t) = \frac{2}{(\pi-2)(4-\pi)}((2-\pi)\cos t - 2\sin t + 2),$$

$$U_2^{(1)}(t) = \frac{2(2\cos t - (2-\pi)\sin t - 2)}{(\pi-2)(4-\pi)},$$

$$U_3^{(1)}(t) = \frac{2}{(\pi-2)}(-\cos t + 1),$$

Replacing these values in (4.2) and simplifying, the following equation is obtained.

$$s^{(1)}(e) = \left(\frac{\pi}{2h_i}\right)\frac{\begin{pmatrix} V_0V_1(Q_1-Q_0)(U_0U_1' - U_0'U_1) + \\ V_0V_2(Q_2-Q_0)(U_0U_2' - U_0'U_2) + \\ V_0V_3(Q_3-Q_0)(U_0U_3' - U_3U_0') + \\ V_1V_2(Q_2-Q_1)(U_1U_2' - U_2U_1') + \\ V_1V_3(Q_3-Q_1)(U_1U_3' - U_1'U_3) + \\ V_2V_3(Q_3-Q_2)(U_2U_3' - U_3U_2') \end{pmatrix}}{(V_0U_0 + V_1U_1 + V_2U_2 + V_3U_3)^2} = \frac{\begin{pmatrix} D_0\left(U_0U_1^{(1)} - U_0^{(1)}U_1\right) + D_1\left(U_0U_2^{(1)} - U_0^{(1)}U_2\right) + \\ D_2\left(U_0U_3^{(1)} - U_0^{(1)}U_3\right) + D_3\left(U_1U_2^{(1)} - U_2U_1^{(1)}\right) + \\ D_4\left(U_3^{(1)}U_1 - U_3U_1^{(1)}\right) + D_5(U_2U_3^{(1)} - U_2^{(1)}U_3) \end{pmatrix}}{(V_0U_0 + V_1U_1 + V_2U_2 + V_3U_3)^2}$$

Where,

$$D_0 = \frac{\pi}{2h_i}V_0V_1(Q_1 - Q_0),$$
$$D_1 = \frac{\pi}{2h_i}V_0V_2(Q_2 - Q_0),$$

$$D_2 = \frac{\pi}{2h_i} V_0 V_3 (Q_3 - Q_0)$$
$$D_3 = \frac{\pi}{2h_i} V_1 V_2 (Q_2 - Q_1),$$
$$D_4 = \frac{\pi}{2h_i} V_1 V_3 (Q_3 - Q_1),$$
$$D_5 = \frac{\pi}{2h_i} V_2 V_3 (Q_3 - Q_2)$$

Putting the values of $Q_i^{'}s$ in $D_i^{'}s$ we get,

$$D_0 = V_0^2 \left(\frac{\pi-2}{2}\right)\left(d_i - \frac{\alpha_i(e_n-e_1)d_1}{h_i}\right),$$

$$D_1 = \frac{\pi}{2h_i} V_0 V_2 \big((f_{i+1} - f_i) - \alpha_i (f_n - f_1)\big) - V_0 V_3 \left(\frac{\pi-2}{2}\right)\left(d_{i+1} - \frac{(e_n-e_1)\alpha_i d_n}{h_i}\right),$$

$$D_2 = \frac{\pi}{2h_i} V_0 V_3 \big((f_{i+1} - f_i) - \alpha_i (f_n - f_1)\big),$$

$$D_3 = \frac{\pi}{2h_i} V_1 V_2 \big((f_{i+1} - f_i) - \alpha_i (f_n - f_1)\big) - V_1 V_3 \left(\frac{\pi-2}{2}\right)\left(d_{i+1} - \frac{(e_n-e_1)\alpha_i d_n}{h_i}\right) - V_0 V_2 \left(\frac{\pi-2}{2}\right)\left(d_i - \frac{\alpha_i(e_n-e_1)d_1}{h_i}\right),$$

$$D_4 = \frac{\pi}{2h_i} V_1 V_3 \big((f_{i+1} - f_i) - \alpha_i (f_n - f_1)\big) - V_0 V_3 \left(\frac{\pi-2}{2}\right)\left(d_i - \frac{\alpha_i(e_n-e_1)d_1}{h_i}\right),$$

$$D_5 = \frac{\pi}{2h_i} V_3^2 (\pi - 2)(h_i d_{i+1} - (e_n - e_1)\alpha_i d_n,$$

Let

$$a_i = \frac{e_{i+1}-e_i}{e_n-e_1} = \frac{h_i}{e_n-e_1} \text{ , and}$$

$$\Delta_i = \frac{f_{i+1}-f_i}{h_i},\ \Delta_i^* = \Delta_i - \alpha_i \left(\frac{f_n-f_1}{h_i}\right),$$

$$d_{i+1}^* = d_{i+1} - \alpha_i \frac{d_n}{a_i},$$

$$d_i^* = d_i - \alpha_i \frac{d_1}{a_i}$$

Now, $D_0 > 0$,

implies,

$$\alpha_i < \frac{h_i d_i}{(e_n - e)d_1} \Rightarrow \alpha_i < a_i \frac{d_i}{d_1}$$

Similarly,

$$D_1 > 0 \Rightarrow V_2 > \left(\frac{\pi-2}{\pi}\right) V_3 \frac{d_{i+1}^*}{\Delta_i^*},$$

$$D_2 > 0, \Rightarrow \alpha_i < \frac{(f_{i+1}-f_i)}{(f_n-f_1)},$$

$$D_3 > 0, \Rightarrow V_2 > V_3 \left(\frac{\pi-2}{\pi}\right)\frac{d_{i+1}^*}{\Delta_i^*}$$

$$D_4 > 0, \Rightarrow V_1 > V_0 \left(\frac{\pi-2}{\pi}\right)\frac{d_i^*}{\Delta_i^*},$$

And

$$D_5 > 0, \Rightarrow \alpha_i < a_i \frac{d_{i+1}}{d_n}$$

So that Ψ will preserve monotonicity on the interval $I_i = [e_i, e_{i+1}]$

if the following constrained is true.

$$0 < \alpha_i < \left\{a_i \frac{d_i}{d_1}, a_i \frac{d_{i+1}}{d_n}, \frac{(f_{i+1} - f_i)}{(f_n - f_1)}\right\},$$

Equivalently,

$$V_0 > 0\ , V_3 > 0$$

$$V_1 > \max\left\{0, V_0 \left(\frac{\pi-2}{\pi}\right)\frac{d_i^*}{\Delta_i^*}\right\},$$

$$V_2 > \max\left\{0, V_3 \left(\frac{\pi-2}{\pi}\right)\frac{d_{i+1}^*}{\Delta_i^*}\right\}$$

Where,

$$\Delta_i^* = \Delta_i - \alpha_i \left(\frac{f_n - f_1}{h_i}\right),$$
$$d_{i+1}^* = d_{i+1} - \alpha_i \frac{d_n}{a_i},$$
$$d_i^* = d_i - \alpha_i \frac{d_1}{a_i}$$

And

$$0 < \alpha_i < \min\left\{\frac{a_i d_i}{d_1}, \frac{a_i d_{i+1}}{d_n}, \frac{f_{i+1} - f_i}{f_n - f_1}\right\}$$

These can be rearranged as

$$V_0 > 0\,, V_3 > 0$$

$$V_1 = \gamma_i + \max\left\{0, V_0\left(\frac{\pi-2}{\pi}\right)\frac{d_i^*}{\Delta_i^*}\right\},$$

$$V_2 = \sigma_i + \max\left\{0, V_3\left(\frac{\pi-2}{\pi}\right)\frac{d_{i+1}^*}{\Delta_i^*}\right\}$$

Where,

$$\Delta_i^* = \Delta_i - \alpha_i\left(\frac{f_n - f_1}{h_i}\right),$$

$$d_{i+1}^* = d_{i+1} - \alpha_i\frac{d_n}{a_i}, \quad d_i^* = d_i - \alpha_i\frac{d_1}{a_i}$$

$$a_i = \frac{e_{i+1} - e_i}{e_n - e_1} = \frac{h_i}{e_n - e_1},$$

$$\Delta_i = \frac{f_{i+1} - f_i}{h_i}$$

And

$$0 < \alpha_i < \min\left\{\frac{a_i d_i}{d_1}, \frac{a_i d_{i+1}}{d_n}, \frac{f_{i+1} - f_i}{f_n - f_1}\right\}$$

For preserving the shape of the curve these parameters $V_1 and\, V_2$ are to be used. To adjust and alter the shape of the curves, parameters $V_0, V_3\ and\ \alpha_i$ are used.

Validation of Proposed Algorithm

The proposed algorithm for predicting data values has been tested and trained on two types of data sets in this section.

Simulation of Model 1

The data set of type positive is presented in Table 1 and Table 2. This data has been tested against some randomly chosen values of shape parameter and scaling factors which are tabulated in Table 3 and Table 4 respectively.

The type of positive data can be seen easily in data set 1 and 2, yet the curves in Figure 5.1 and Figure 5.5 show the opposite negative behavior. So, the prediction pattern misguides the observer. To overcome this type of crucial problem, the proposed technique constructed in Section 3, then, has been applied to test and train the same data sets. Figures 5.2, 5.3, 5.4, 5.6, 5.7 and 5.8 predict the real scenario hidden in the data.

Table 1: Data Set 1

i	0	1	2	3	4	5	6	7	8
e_i	0.05	0.03	0.05	1.05	1.55	2.05	2.55	3.05	4.05
f_i	2.05	0.65	0.15	0.85	1.05	0.55	1.55	0.35	0.25

Table 2: Data Set 2

i	0	1	2	3	4	5	6
e_i	2	3	7	8	9	13	14
f_i	10	2	3	7	2	3	10

Table 3: Random Values of Parameters and Scaling Factors for Data Set 1

α_i	$V_0(i)$	$V_1(i)$	$V_2(i)$	$V_3(i)$
0.2	0.002	25	12	0.1

Table 4: Random Values of Parameters and Scaling Factors for Data Set 2

α_i	$V_0(i)$	$V_1(i)$	$V_2(i)$	$V_3(i)$
0.4	0.3	2.5	1.5	0.1

The Figures (5.2, 5.3, 5.4, 5.6, 5.7 and 5.8) further exemplify how the proposed method allows for the freedom to obtain various results even while considering the nature of the data.

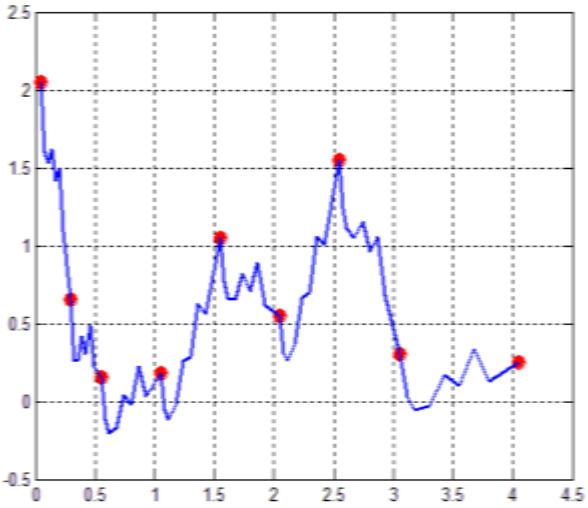

Figure 5.1. 2D plot of data set

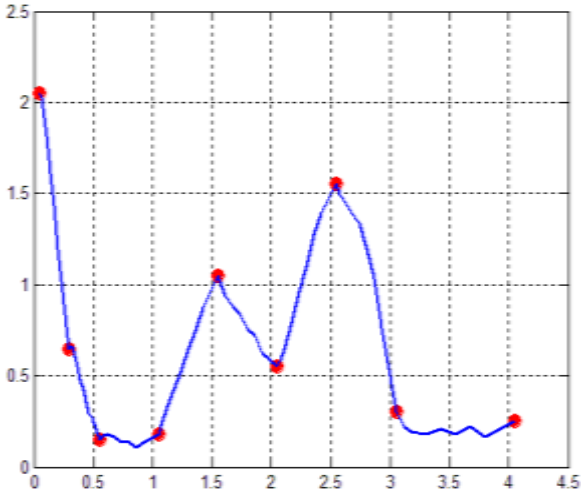

Figure 5.2. Proposed supervised computing algorithm for data set 1 with $V_0 = 0.002, V_3 = 0.$

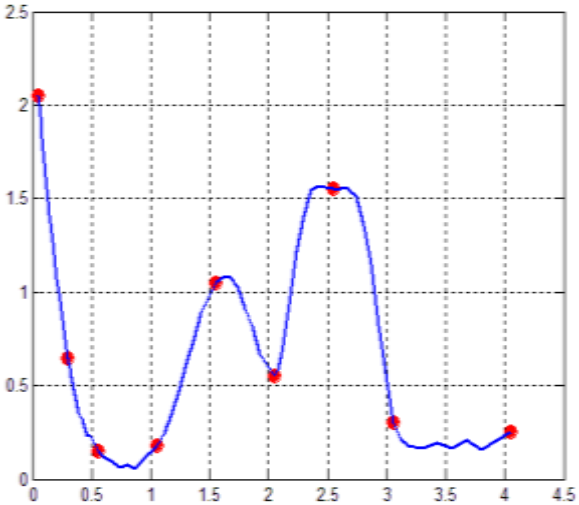

Figure 5.3. Proposed supervised computing algorithm with $V_0 = 0.2,\ V_3 = 0.1$

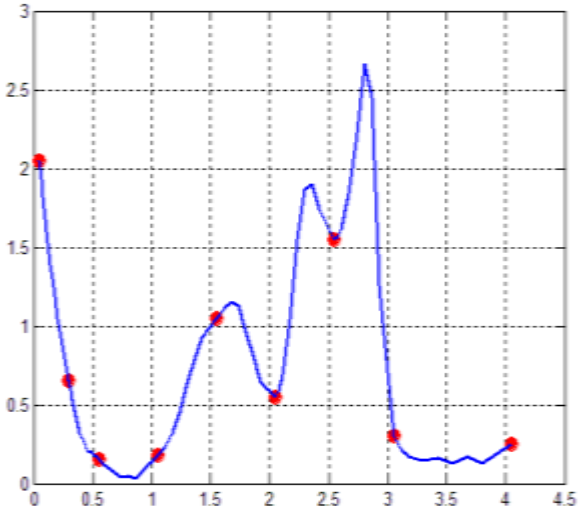

Figure 5.4. Proposed supervised computing algorithm with $V_0 = 0.4,\ V_3 = 0.3$

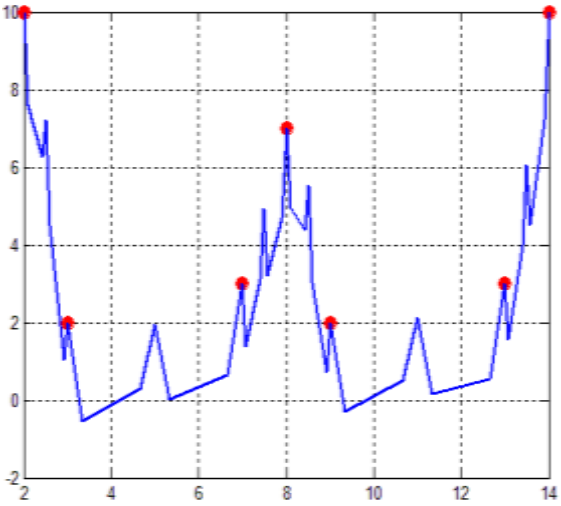

Figure 5.5. 2D plot of data set 2.

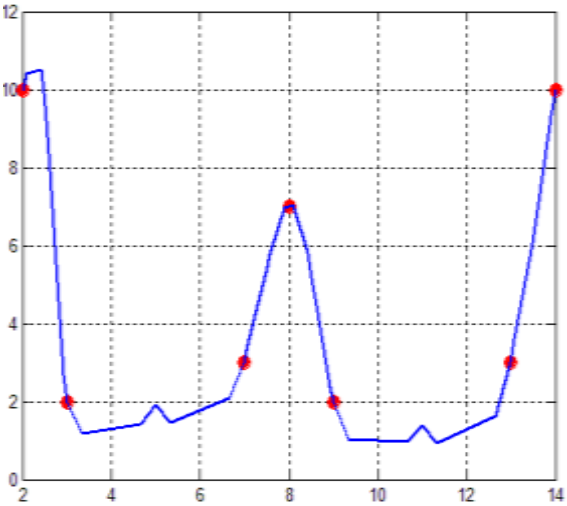

Figure 5.6. Proposed supervised computing algorithm for data set 2 with $V_0 = 0.003,\ V_3 = 0.3$

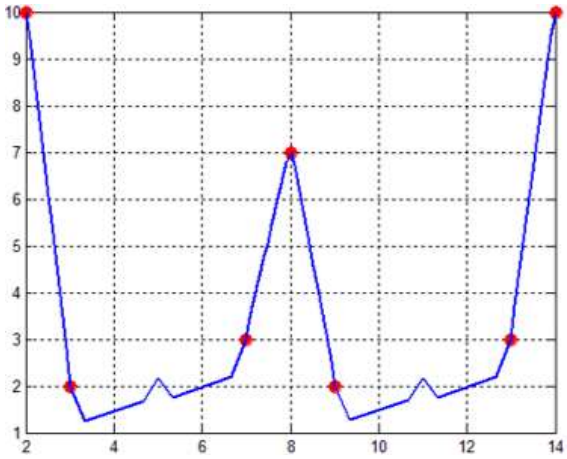

Figure 5.7. Proposed supervised computing algorithm for data set 2 with $V_0 = 0.002$, $V_3 = 0.001$

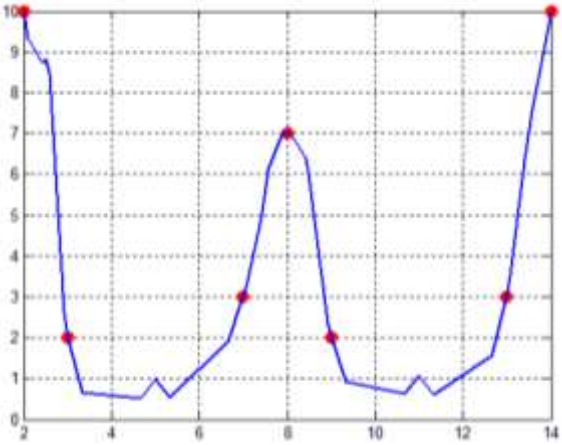

Figure 5.8. Proposed supervised computing algorithm for data set 2 with $V_0 = 0.9$, $V_3 = 0.9$

Simulation of Model 2

The monotone type data set is considered in Table 5 and Table 6. This data has been tested against some randomly chosen values of shape parameter and scaling factors which are tabulated in Table 7 and Table 8 respectively.

Table 5: Data Set 3

i	0	1	2	3
e_i	0	0.5	2.2	3.3
f_i	124	331	379	835

Table 6: Data Set 4

i	0	1	2	3	4	5	6	7	8	9	10
e_i	0.1	4	6.52	10	15	25	40	50	62	65	66
f_i	11	1	2	3.5	5.5	5.5	10	10	12.5	18	20

Table 7: Random Values of Parameters and Scaling Factors for Data Set 3

s_i	$V_0(i)$	$V_1(i)$	$V_2(i)$	$V_3(i)$
0.7	0.002	15	1.2	0.2

Table 8: Random Values of Parameters and Scaling Factors for Data Set 4

s_i	$V_0(i)$	$V_1(i)$	$V_2(i)$	$V_3(i)$
0.5	3	5	15	1

The type of monotone data can be seen easily in data set 5 and 6, however, the curves in Figure 5.9 and Figure 5.13 show the false pattern. To resolve the intrinsic pattern prediction issue, the proposed technique constructed in Section 4, then, has been applied to test and train the same data sets. Figures 5.10, 5.11, 5.12, 5.14, 5.15 and 5.16 overcome this misleading trend and real scenario hidden in the data can clearly be witnessed. Moreover, the Figures (5.10, 5.11, 5.12, 5.14, 5.15 and (5.16) depict how the proposed method provides freedom to obtain various results even while considering the nature of the data.

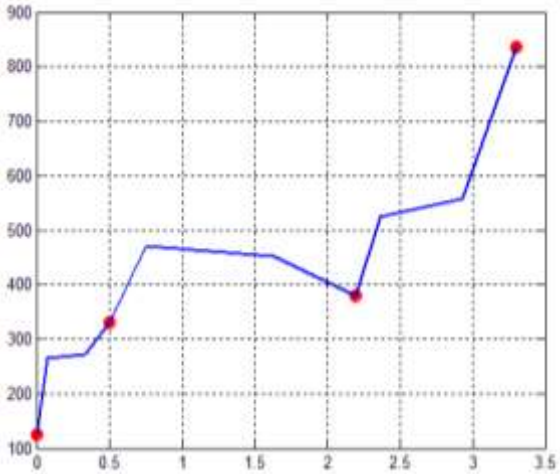

Figure 5.9. 2D plot of data set 3.

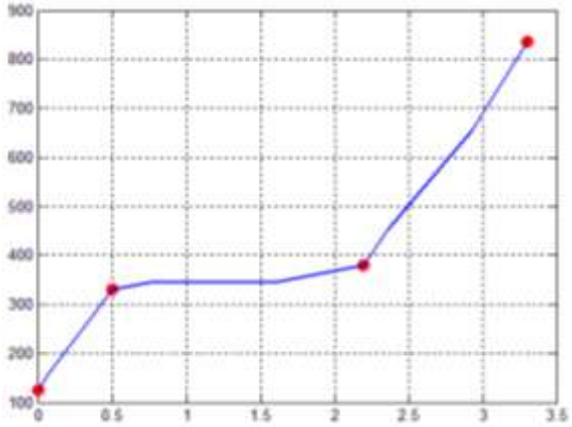

Figure 5.10. Proposed Supervised Computing Algorithm for Data Set 3 with $V_0 = 0.001,\ V_3 = 0.1$

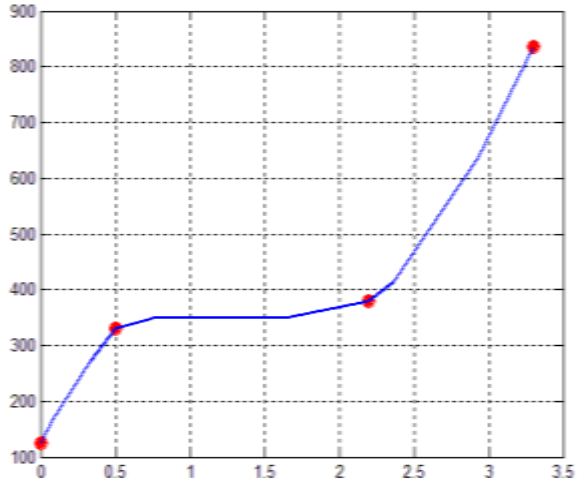

Figure 5.11. Proposed supervised computing algorithm for data set 3 with $V_0 = 0.3,\ V_3 = 0.3$

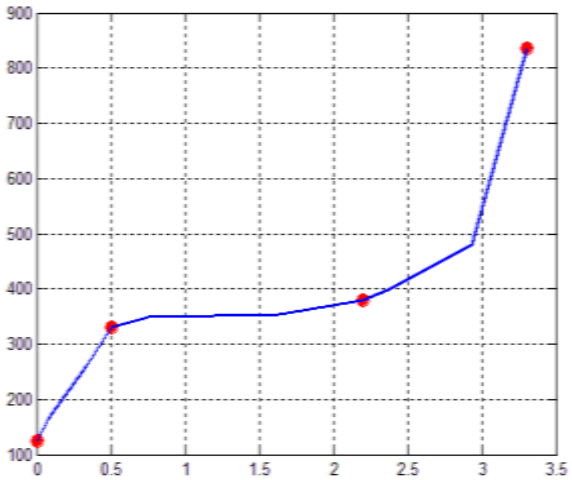

Figure 5.12. Proposed Supervised Computing Algorithm for Data Set 3 with $V_0 = 11,\ V_3 = 0.003$

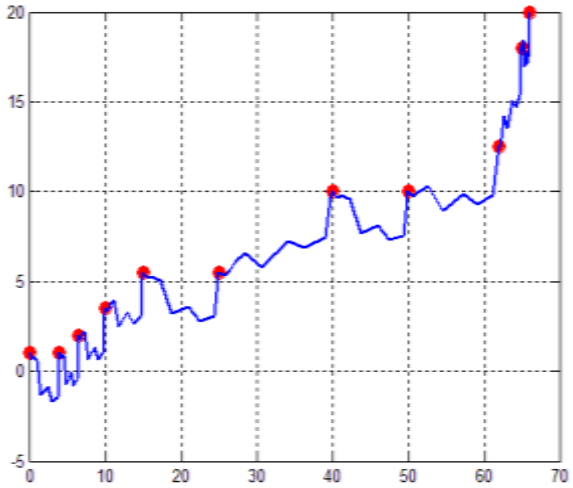

Figure 5.13. 2D plot of Data Set 4

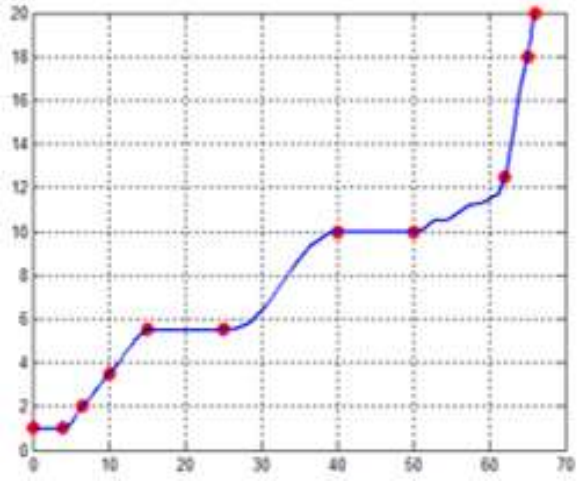

Figure 5.14. Proposed Supervised Computing Algorithm for Data Set 4 with $V0$=0.5, $V3$=0.4

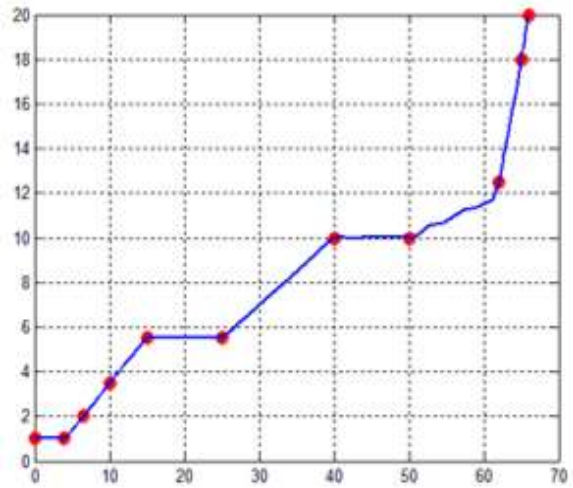

Figure 5.15. Proposed Supervised Computing Algorithm for Data Set 4 with $V_0 = 0.004,\ V_3 = 0.001$

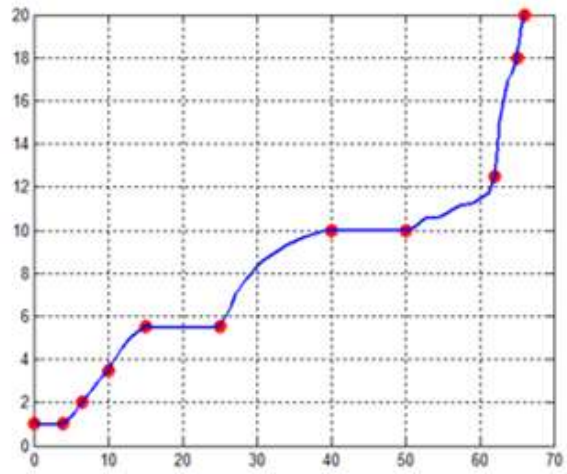

Figure 5.16. Proposed Supervised Computing Algorithm for Data Set 4 with $V_0 = 0.001, V_3 = 21$

Conclusion

In this research, a supervised, hybrid computing algorithm based on rational C cubic Bezier and fractal function has been proposed to solve the problem of predicting intrinsic data attributes. The proposed algorithm introduces four shape parameters and a scale parameter to provide a variety of simulated results. It has been tested and trained in two scenarios, Scenario 1 –Type positive data set and Scenario 2 –Type monotone data set. Four different data sets, (two for each of the scenarios) have been trained and tested to validate the authenticity of the developed algorithm. Simulation results for shape control and scale parameter V_0, V_3 and α_i against each data set have been demonstrated which depict fruitful implementation to have insight of intrinsic data attributes. The suggested algorithm also handles irregularities of data successfully by adjusting values of scale parameter.

The proposed model is a supervised learning model in which two main attributes are supplied to generate simulating effect. the first one plays a crucial role in determining the vertical magnification or compression of the pattern being generated. This factor essentially controls the amplitude of the pattern along the vertical axis.

Mainly, controlling the gives you greater control over the visual appearance of the generated underlying pattern. Depending on the specific requirements or aesthetic preferences, you can adjust the factor to produce visually pleasing patterns with the desired characteristics, such as smoothness, roughness, or intricacy. The ability to adjust it allows you to adapt the generated pattern to better represent specific characteristics, ensuring more accurate and meaningful visualizations or analyses.

The second important attribute of the prop-

-posed method is their random weights that are exogenous factors. This enables functions to adapt to the variability and complexity of the underlying data while minimizing deviations. By optimizing the weights, the curve can capture both global trends and local variations in the data, resulting in a more accurate representation. Random weight can be subject to regularization techniques or constraints to impose specific properties on the curve, such as monotonicity, convexity, or boundedness. This allows for the creation of customized patterns that adhere to desired criteria of the underlying data.

Abbreviations

IOT: Internet of Things
AI: Artificial Intelligence
PR: Pattern Recognition
GP: Gaussian Processor

Availability of data and materials

All datasets have been made available within the manuscript.

Competing Interests

The authors have no competing interests to

declare.

Contributions

Conceptualization, Analysis, Modeling and Algorithms: Tayba Arooj, Farheen Ibraheem Designing manuscript write up: Farheen Ibraheem. Reviewing manuscript: Faira Kanwal Janjua

References

1. Sarker I H, Machine Learning: Algorithms, Real- World Applications and Research Directions, SN Computer Science 2021, 2 (160), 1-21.
2. Sarker I. H., Kayes A.S.M, Badshah S, Alqahtani H, Watters P, Ng A, Cybersecurity data science: An Overview from Machine Learning Perspective, Journal of Big Data, 2020, 7(1), 1-29.
3. Sarker I. H, Machine Learning for Intelligent Data Analysis and Automation in Cybersecurity: Current and Future Prospects. *Annals of Data Science* 2022.
4. F. Fontanella, F. Colace, M. Molinara, A. Scotto Di Freca, F. Stanco, Pattern recognition and artificial intelligence techniques for cultural heritage 2020, Pattern Recognition Letters, 138, 23-29.
5. Wolpert D, W. Macready, No free lunch theorems for optimization 1997, IEEE Transactions on Evolutionary Computations, 1 (1), 67-82.
6. Rasmussen C.E, Gaussian Processes for Machine Learning, MIT Press, 2006.
7. Rumelhart D.E, Hinton G.E, Williams R.J, Learning representations by back-propagating errors Nature, 323(9), (198), 533-536.
8. Rasmussen C.E, Williams C.K.I, Gaussian Processes for Machine Learning, MIT Press, 2006. ISBN 026218253X. c 2006 Massachusetts Institute of Technology.
9. Gramacy R.B, Lee H.K.H, Bayesian Treed Gaussian Process Models with an Application to Computer Modeling, 2008, Journal of the American Statistical Association, 103 (483) , 1119-1130.
10. Matthew A. T, Gramacy R. B, Nicholas G. P, Dynamic Trees for Learning and Design, 2011, Journal of the American Statistical Association, 106(493),109-123.
11. Neal R, Bayesian Learning for Neural Networks 1996, Lecture Notes in Statistics, Springer, New York.
12. Ahn Y. J, Hoffmann C, Rosen P, Geometric constraints on quadratic Bézier curves using minimal length and energy, 2014, Journal of Computational and Applied Mathematics, 255, 887-897.
13. Balasubramani N, Shape preserving rational cubic fractal interpolation function, Journal of Computational and Applied Mathematics, 319, 2017, 277-297.
14. Barnsley M.F, Fractal functions and interpolation, 1986, Constructive Approximation, 2, 303-327.
15. De Amo E, Carrillo M. D, Sánchez J. F, PCF self-similar sets and fractal interpolation, 2013, Mathematics and Computers in Simulation, *92*,28-29.
16. Goldman R, The fractal nature of Bezier curves, In Geometric Modeling and Processing, Proceedings, Beijing, China, 2004, 3-11.
17. Gangdou T, Shape control and modification of rational Bezier curve and surface, 1991Journal of Systems Engineering and Electronics, 2(2), ,65-72.
18. Han X, Ma Y, Huang Y, X, the cubic trigonometric Bezier curve with two shape parameters 2009, Applied Mathematics Letters, 22, 226-231.
19. Hussain M. Z, Sarfraz M, Monotone piecewise rational cubic interpolation 2009, international journal of Computer Mathematics, 86(3), 423-430.
20. Hussain M. Z, Sarfraz M, Positivity preserving interpolation of positive data by rational cubic 2008, Journal of Computational and Applied Mathematics, 218, 446-458.

21. Hussain M. Z, Hussain M, Waseem A, Shape-preserving trigonometric functions 2014, Computational and Applied Mathematics, *33*(2), 411-431.
22. Kocić L. M, Fractals and Bernstein polynomials 1996, Periodica Mathematica Hungarica, 33(3),185-195
23. Navascués M. A, Chand A. K. B, Fundamental sets of fractal functions 2008, Acta Applicandae Mathmaticae, 100(3), 247-261.
24. Prasad B, Sing B, and Katiyar K, Modeling curves via fractal interpolation with VSFF, 2014, International Journal of Computer Applications, 0975-8887.
25. Shen W. Q, Wang G. Z, Huang F, Direction monotonicity for a rational Bézier curve 2016, Applied Mathematics-A Journal of Chinese Universities, 31(1), ,1-20.
26. Tariq Z, Ibraheem Z, F, Hussain M. Z, M. Sarfraz, Monotone data modelling using rational functions 2019, Turkish Journal of Electrical Engineering and Computer Science, 27(3), 2331-2343.