Research Article

Solving First-Order Differential Equations of Z-Numbers' Initial Value Using Radial Basic Function

Leila Qalehe,[1] Mozhdeh Afshar Kermani ⓘ,[1] and Tofigh Allahviranloo[2]

[1]*Department of Applied Mathematics, Tehran North Branch, Islamic Azad University, Tehran, Iran*
[2]*Faculty of Engineering and Natural Sciences, Bahcesehir University, Istanbul, Turkey*

Correspondence should be addressed to Mozhdeh Afshar Kermani; mog_afshar@yahoo.com

In this paper, a method was proposed based on RBF for numerical solution of first-order differential equations with initial values that are valued by Z-numbers. The proposed method consists of two parts. The first part has stated the amount of limitation of the fragmentation solution, while the second part has described the assurance of the first part. The limitation section also has two parts. The first part has included the initial condition of the problem, while the second part has included the RBF network. The confidence interval was also considered as a function based on the probability function, which has calculated the confidence level of the first part (limitation). The RBF network or the radial-base grid network has three distinct layers: the input layer that is the set of elementary nodes (sensory units); the second layer is the hidden layers with high dimensions, in which the output layer that has responded to the network response and the activation patterns used in the input layer. The advantage of using RBF is that the use of this technique does not require sufficient information. It only relies on the domain and the boundary. In an example, we have showed that our proposed approach could approximate the problem with acceptable confidence.

1. Introduction

The emergence of a new concept of mathematics called fuzzy sets, the study of fuzzy differential equations, provides a suitable basis for mathematical modeling of real-world issues in which there is ambiguity or some uncertainty. For example, in Science and Engineering, many problems are limited to a set of fuzzy differential equations (FDE) through the process of mathematical modeling. As it is not simple to achieve an exact solution, numerical methods should be utilized [1–8]. One of these methods is the radial basis function (RBF) method.

Generally, radial basis functions are methods, which are based on the location method for interpolation of discrete data while have a high convergence rate [9]. This method is one of the most widely used methods for approximating the functions in the theory of modern approximation [10].

RBFs are first used by Broomhead and Lowe [11]. Much of its use is in the theory, design, and applications of RBF networks [12, 13]. In this paper [14], the use of regulation theory for this group of neural networks is presented as a way to enhance the generalization of the new data. In RBF methods, data are interpolated by linear combinations of functions of a function. This method has been very much considered due to high precision and flexibility versus problem geometry, dimensional independence, and ease of implementation. Today, RBFs are used in some cases such as estimation, modeling, prediction, and classification in various fields, including geosciences [15–19]. Besides, these methods are utilized to solve the numerical differential equations with partial derivatives [20]. The main advantage of numerical methods that use RBF is their nonnetwork characteristic. In nonnetworked methods, it is not necessary to produce a regular network in the domain of the problem, which, due to the high computational cost of network production, is the main advantage of these methods to finite difference methods and finite elements, and so on. The geometric feature used in the RBF approximation is the distance between points. The distance in each space dimension is easily calculated, resulting in higher

dimensioning and does not increase the complexity of RBF methods. As stated above, a grid-independent grid function does not require a grid and in spite of our data connections between points; only spatial points are used. Thus, several studies have been used based on RBF to solve various equations. For instance, in [21], the radial function of RBF along with the time-dimensional discretization by the BKM boundary node method and the AEM analogue equation method to solve the time-dependent hyperbolic equations in [22] of RBF to solve time-dependent elliptic equations in [23] of RBF along with the MOL lines in the time dimension to solve the time-dependent nonlinear equations in [24] of RBF using the Kansa idea based on Hermitian interpolation for Fokker–Planck equation. In [25], RBF and quasi-spatial methods to solve the sin-Gordon equation are also used in [26] to use the RBF time-independent functions.

As mentioned, the theory of fuzzy sets is a powerful method to model the uncertainties and processing ambiguity and information dependent on mathematical models. But, they need to be trusted to make this information useful. Human beings have a clear capacity to make rational decisions based on obscure, inaccurate, or incomplete information. Formalization of this capacity is at least somewhat difficult to predict. The author proposed a theorem, called a Z-number, which is a regular pair of fuzzy numbers (A, B) [27]. The first component, A is a limitation for values that have an uncertain variable X that has a value or a real value. The second component, B is a measure of reliability for the first component. Typically, A and B are described with natural disadvantages. For example, about 45 minutes, very sure. Yager used Z-number to calculate the waiting time for a bus [28]. Kang Wang used Z-number to make decisions in an ambiguous environment in 2012 [29]. Ezadi and Allahviranloo initially introduced the Z-based generalized neural network and then estimated the Z-based regression [30]. They also presented methods to rank the Z-numbers in 2017 and 2018 [31, 32]. Also, they, in 2019, introduced the Z-Advanced number process [1]. Further investigations have been carried out on Z-numbers by researchers [33–36]. However, a few studies have been conducted in the field of Z-differential equations [37].

In this paper, we try to provide a numerical method to solve the differential equations with an initial value based on Z-numbers. In Section 2, basic concepts and theorems are presented as well. In Section 3, the network of generalized RBF, radial base functions, is introduced. In Section 4, a method for approximating first-order differential equations with a Z value based on the generalized RBF network is presented. In Section 5, numerical examples are presented and ultimately referenced.

2. Preliminaries

This section provides the necessary definitions and required theorems, which are used to propose the model.

Definition 1 (definition of Z-number).
This valuation for Z according to proposal of Zadeh is observed as a restriction in x and is interpreted as follows [27]:

$$P(y \text{ is } A) \text{ is } B. \tag{1}$$

Indeed,   it   means   that $R(y)$: y is $A \longrightarrow$ $Poss(y = u) = \mu_A(u)$.

$$P(y \, i \, sA) = \int_R \mu_A(u)P_y(u)du \text{ is } B, \tag{2}$$

where μ_A is the membership function of fuzzy set A and u is a part of y. $P_y(u)$ is the probability density function of y, and $P(y = u)$ is the probability function of y, where as we do not know the basic probability distribution. It is clear from this information that probability distribution function is itself a fuzzy number.

Definition 2 (parametric form of Z-numbers).
Assuming that Z^* is set of $Z - numbers$, an arbitrary Z-number, $Z = (A, B) \in Z^*$, in the parametric form is represented by an ordered pair of functions:

$$Z = ((\underline{A}(r), \overline{A}(r)), B), r \, \epsilon \, [0, 1], \tag{3}$$

and its components satisfy the following requirements:

$$
\begin{aligned}
&1. \underline{A}(r) \le \overline{A}(r) \quad \text{for} \quad r\epsilon [0, 1], \\
&2. \underline{A}(r) \text{ is bounded left} - \text{continuous non} - \text{decreasing functions over} [0, 1], \\
&3. \overline{A}(r) \text{ is bounded left} - \text{continuous non} - \text{increasing functions over} [0, 1], \\
&4. B = f\left(\underline{A}(r), \overline{A}(r)\right).
\end{aligned}
\tag{4}
$$

Definition 3 (normal $Z - numbers$).
Let us consider Z to be a $Z - numbers$, where $Z = (A, B)$. We say that Z is a normal Z-number if $h(B) = 1$, where $h(B)$ is the height of B [37].

Definition 4 (Z-number initial value problem (ZIVP)).

In real world, most of the phenomena are based on doubt and the information, which we have from various subjects such as economic, political, and physics , which have been evaluated according to verbal valuables. Here, we try to formulate and investigate the mentioned information to the initial value problem, while our initial data are Z-numbers. For instance, evaluate the population growth issue with the

Z-number data (population growth, very high, usually) in the starting moment t_0 and consider how is the information in the next time like in t1. For this purpose, first, we consider a ZIVP as (5), and then we study the existence and uniqueness condition of solution of this problem [37]. Assuming that Z^* is set of Z-numbers, so, the general form of differential equation initial value based on $Z - numbers$ is defined as follows:

$$\begin{cases} x'(t) = f(t, x(t)), \\ x(t_0) = x_0 \in Z^*, \end{cases} \tag{5}$$

where $t \in [t_0, T] \subseteq R_+$, f is a continuous mapping from $[t_0; T] \times Z^*$ in to Z^*, x_0 is a Z-number in Z^*, and $x(t) = (A_x, B_x)$.

Suppose x is a Z-valuation; then, according to [28],

$$x'(t) = (A_{x'}, B_{x'}), \tag{6}$$

where $A_{x'}$ is a limitation for values and $B_{x'}$ is a measure of reliability for the first component, $A_{x'}$. For $(A_{x'}, B_{x'})$, assuming there is a Hukuhara difference between $A_x(t + h)$ and $A_x(t)$, we define

$$A_{x'} = \lim_{h \longrightarrow 0} \frac{A_x(t + h) -_h A_x(t)}{h},$$
$$B_{x'} = p(x'(t) \, is \, A_{x'}), \tag{7}$$

where $p(x'(t) \, is \, A_{x'})$ is the probability distribution function.

2.1. Gaussian Function Definition.
The Gaussian function is a function of the form defined as

$$f(x) = \lambda e^{-\left((x-c)^2/2\sigma^2\right)}, \tag{8}$$

where λ, c, and σ are real constant coefficients and e is the Euler number. The shape of this function is symmetric that quickly drops to zero. The constant λ defines the height of the peak of the curve, c determines the location of the center of the peak, and σ is the standard deviation.

3. RBF Networks

The RBF networks, as shown in Figure 1, are of the type of leading networks with an intermediate layer, first introduced by Broomhead and Low [11] (Figure 1). In this method, the transfer function in the intermediate layer of the Gaussian function and in the output layer of the transfer function is linear [38, 39]. Generally, the RBF network training is divided into two parts. The first part is primarily non-monitoring-type learning. Using clustering methods, the parameters of the basic functions (centers and latitudes) are determined using input information, and in the second part, learning from type is monitored. The weights between the middle and the output layer are determined using slope reduction and linear regression methods. The intermediate neuron of RBF is connected to each of the input neurons with the weight parameters. These parameters are centers of neurons. The output of each intermediate neural is a

function of the distance between the input vector $X = [x_1, x_2, \ldots, x_n]$ and the radial center vector $r_j = [r_{1j}, r_{2j}, \ldots, r_{nj}]$, which is calculated as follows:

$$\delta = \sqrt{\sum_{i=1}^{n}(x_i - r_{ij})}. \tag{9}$$

The average neuron output can be calculated in different ways. The main transfer function for this aim is the Gaussian function given as follows [39]:

$$\phi(\delta_i) = e^{-\lambda \delta_i^2}. \tag{10}$$

In this case, λ is a constant coefficient. Finally, outputs of the output layer are calculated from the following equation:

$$y_k = \sum_{j=1}^{n} v_{jk}\phi_j. \tag{11}$$

In this regard, w_{jk} is the weight coefficient between j middle neurons and the k neuron of the output layer and ϕ_j, the output of j is the middle layer neuron. The results of the outcomes were evaluated using root mean square error statistics, RMSE, and determination coefficient R^2. One of the important points in the RBF network is the choice of the type of transfer function, which is the responsibility of the modeling person. One of the important issues is the attention to the characteristics and statistical variables of the data. For example, if the RBF type is Gaussian, the width of the function is important, which should be chosen so that the distance between the data points is larger, and the dimension of the data is smaller. One of the other issues in designing RBF networks is the number of centers used in the middle layer. As the selection of these centers is directly related to the accuracy and complexity of the network, these centers should be chosen to balance the desired accuracy and complexity.

4. Solving First-Order Differential Equations of Z-Numbers' Initial Value Using Radial Basic Function (RBF)

Assuming Z^* is the set of Z-numbers, the first-order differential equations of Z-numbers' initial value are defined as follows:

$$\begin{cases} [x'(t)]^Z = f(t, [x(t)]^Z), \\ [x(t_0)]^Z = x_0 \in Z^*. \end{cases} \tag{12}$$

Equation (12) can be rewritten in the form of the parameter defined in [37]:

$$\begin{cases} [x'(t)]^Z = (A_{x'}, B_{x'}), \\ [x(t_0)]^Z = (A_{x_0}, B_{x_0}), \end{cases} \tag{13}$$

in which in (13), $A_{x'}$ is a fuzzy function and $B_{x'}$ is a real function. A_{x_0} and B_{x_0}, respectively, have a certain fuzzy and a real amount, $A_{x'}$ and A_{x_0} have the role of limitation, and $B_{x'}$

and B_{x_0} have the role of reliability for $[x'(t)]$ and $[x(t_0)]^Z$. $A_{x'}$ is introduced as follows:

$$A_{x'} = f(t, \tilde{x}(t)). \tag{14}$$

With the following parametric form,

$$[A_{x'}]^r = \begin{cases} \overline{A}_{x'}(t) = \overline{f}(t, x) = G(t, \underline{x}, \overline{x}), & \overline{x}(t_0) = \overline{x}_0, \\ \underline{A}'_x(t) = \underline{f}(t, x) = F(t, \underline{x}, \overline{x}), & \underline{x}(t_0) = \underline{x}_0, \end{cases} \tag{15}$$

where $[.]^r$ is the symbol of r-cut, and

$$\begin{cases} G(t, \underline{x}, \overline{x}) = \max\{f(t, u) \mid u \in [\underline{x}, \overline{x}]\}, \\ F(t, \underline{x}, \overline{x}) = \min\{f(t, u) \mid u \in [\underline{x}, \overline{x}]\}. \end{cases} \tag{16}$$

Theorem 1. *ZIVP (12) has a unique Z-process solution.*

Proof. Based on [27], relation (12) and relation (13) are equivalent. In (13), $A_{x'}$ is a fuzzy function, and it is introduced on relation (14). For each α-cut, relation (14) and relation (15) are equivalent. On the other hand, the fuzzy initial value problem in (15) has a unique solution (based on Theorem (3.2) in [40]). Similarly $B_{x'}$. Therefore, it is obvious that ZIVP (12) has a unique solution.

In the general process for obtaining the solution of equation (12) with nonnetworked physical domain methods, we present the problem as a set of scattered points in the domain. These points can be regular or irregular, but they must cover the entire domain in any case. We then approximate the unknown function at any point in the domain. In this way, we consider a base for the solution space and a solution as a linear combination of the base members. In this process, this solution is expanded in terms of base functions. Therefore, the coefficients of the basic sentences are unknown to us. These coefficients must hold in the equations. In the cases where basic radial functions for inwardness are used, the number of centers with the dimension of input data is considered equal in order to achieve high precision. In this paper, we propose a method using the RBF of the generalized RBF-based Z-based numbers in such a way that the initial condition of the main problem is to be established as follows:

$$[x_T(t)]^Z = \left(A_{x_T}(t), B_{x_T}(t)\right), \tag{17}$$

in which, $A_{x_T}(x)$ is a function with a fuzzy value and $B_{x_T}(x)$ can depend on the type of the problem of function with the real or fuzzy value, in which we consider the function with the real value. The values of $A_{x_T}(x)$ and $B_{x_T}(x)$, respectively, represent the limiting function and the confidence scale for $[x_T(t)]^Z$. With regard to the definition of the Z-number, the interpretation of (17) is described as follows.

If x_T is A_{x_T}, then possibility (x_T is A_{x_T}) is B_{x_T}.

Here, we define $A_{x_T}(x)$ as follows ($A_{x_T}(x)$ is a derivative function):

$$A_{x_T}(t) = A_{x_T}(t_0) + t \sum_{j=1}^{n} w_j \phi_j(\|t - r\|), \tag{18}$$

which is a method that computes the approximate solution of the equation constraint section, taking into account the initial condition of the problem and is based on the generalized RBF network with fuzzy coefficients. Here, r is the same cut–r that is considered as the center of the RBF network, and $\phi_j(\|t - r_j\|)$ is defined as follows:

$$\phi_j\left(\|t - r_j\|\right) = e^{-\left(\|\varepsilon t - r_j\|\right)^2}, \tag{19}$$

where φ is a nonlinear function in the space $R^+ \longrightarrow R$, which refers to the transfer function. Some radial base transmission functions φ, in addition to the dependence of the distance function, depend on the factor (ε), which depends on the elongation and extension of the function, whose relation to the standard deviation of the normal distribution (σ) is as follows:

$$\varepsilon^2 = \frac{1}{2\sigma^2}, \tag{20}$$

in which the value of σ^2 can be determined according to the type of problem. We also utilize the Gaussian radial base transfer function introduced in (15). Some other RBFs with their mathematical relations can be seen in Table 1; r_j are neural network centers, and W is the matrix of the weight coefficients of the output from the intermediate layer to the output layer (w_j). There is a value with fuzzy-value. In fact, when the centers (points) $r_1, r_2, \ldots, r_n$ are constant, the network is trained using the data to find the weight matrix. This process can be done in one form and by data, or by increasing the number of RBF cores, the weight function can be updated in several steps. Training operations begin by dividing the data into training and testing sets. After doing so, the number and position of the RBF centers and the transfer function φ are decided. Training with the implementation of the linear algebra in a system of equations is formed by weights and bias using a pair of input x and each output y. □

Moreover, $B_{x_T}(x)$ is a function that specifies the confidence level of the constraint section and is based on the exponential function that we define in terms of the initial condition of the problem in the form as follows:

$$B_{x_T}(t) = 1 - te^{-\lambda A_{x_T}}. \tag{21}$$

Now, with respect to relations (17) and (21), we will have

$$[x_T(t)]^Z = \left(A_{x_T}(x), 1 - te^{-\lambda y}|_{A_{x_T}}\right). \tag{22}$$

As A_{x_T} has a fuzzy value according to the Z value definition, it can be rewritten in the following form:

$$\begin{cases} \underline{A}_{x_T}(t) = \underline{x}(t_0) + t \sum_{j=1}^{n} \underline{w}_j \phi_j\left(\|t - r_j\|\right), \\ \overline{A}_{x_T}(t) = \overline{x}(t_0) + t \sum_{j=1}^{n} \overline{w}_j \phi_j\left(\|t - r_j\|\right). \end{cases} \tag{23}$$

In this case, equation (22) can be rewritten in the following form:

Table 1: Transition functions used in the radial base function ($r = \|x - r_j\|\|$).

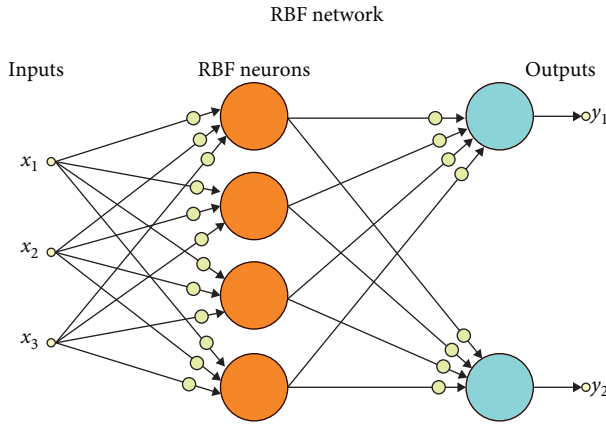| Function name	Mathematical relation
Linear	$\phi(r) = r$
Cubic	$\phi(r) = |r|^3$
Narrow page	$\phi(r) = r^2 ln|r|$
Making	$\phi(r, \varepsilon) = (1/1 + \varepsilon r)$
Robey inversion	$\phi(r, \varepsilon) = (1/1 + (\varepsilon r)^2)$
A few rebates	$\phi(r, \varepsilon) = \sqrt{1 + (\varepsilon r)^2}$
Multiple reverses	$\phi(r, \varepsilon) = (1/\sqrt{1 + (\varepsilon r)^2})$

RBF network

Figure 1: RBF network diagram.

$$B_{x_T}(t) = 1 - t\left(e^{-\lambda \underline{A}_{x_T}(t)} - e^{-\lambda \overline{A}_{x_T}(t)}\right). \tag{24}$$

The value of $B_{x_T}(t)$ is also a fuzzy number, which must be computed.

We now want to derive the derivative of the function $[x_T(t)]^Z$ i.e. $[x_T'(t)]^Z$ for this purpose. We derive from the functions of (23), so we have

$$\begin{cases} \underline{A}'_{x_T}(t) = \sum_{j=1}^{n} \underline{w}_j \phi_j\left(\|t - r_j\|\right) + t \sum_{j=1}^{n} \underline{w}_j \frac{\partial \phi_j}{\partial t}\left(\|t - r_j\|\right), \\ \overline{A}'_{x_T}(t) = \sum_{j=1}^{n} \overline{w}_j \phi_j\left(\|t - r_j\|\right) + t \sum_{j=1}^{n} \overline{w}_j \frac{\partial \phi_j}{\partial t}\left(\|t - r_j\|\right). \end{cases} \tag{25}$$

In this case, $B'_{x_T}(t)$ is calculated as follows:

$$B'_{x_T}(t) = 1 - te^{-\lambda' y'}|_{\underline{A}'_{x_T}}^{\overline{A}'_{x_T}} = 1 - t\left(e^{-\lambda^{\underline{A}'_{x_T}(t)}} - e^{-\lambda^{\overline{A}'_{x_T}}}\right). \tag{26}$$

To optimize the weights $\underline{w}_j$ and $\overline{w}_j$, we use the minimization of the sum-squared error function:

$$e = \min \sum_{i=1}^{n} \left([x_T'(t_i)]^Z - [x'(t_i)]^Z\right)^2 \longrightarrow 0. \tag{27}$$

So, we define it as follows:

$$\begin{cases} \underline{A}_e = \min \sum_{i=1}^{n} \left[\underline{A}'_{x_T}(t) - F\right]^2 \longrightarrow 0, \\ \overline{A}_e = \min \sum_{i=1}^{n} \left[\overline{A}'_{x_T}(t) - G\right]^2 \longrightarrow 0, \end{cases} \tag{28}$$

$$B_e(t) = 1 - te^{-\lambda' y'}|_{\underline{A}_e}^{\overline{A}_e} \longrightarrow 1.$$

It means that

$$\min \sum_{i=1}^{n} \left[e^{-\lambda' \underline{A}'_{x_T}(t)} - e^{-\lambda F}\right]^2 + \left[e^{-\lambda' \overline{A}'_{x_T}(t)} - e^{-\lambda G}\right]^2 \longrightarrow 0. \tag{29}$$

4.1. The Numerical Modeling of the Grid Radial Base Function. The RBF algorithm is a function of the distance between the input and some fixed point called center; the function φ is called a radial base function if $\varphi(x) = \varphi(\|x\|)$. Therefore, this function φ can be applied to the vector in space R^n but is expressed only in the form of distance. This means that φ can be considered as a numerical function. In these discussions, the application of the Euclidean distance matrix is expressed using the function φ [41]:

$$EDM = \begin{bmatrix} \|x_1 - x_1\| & \|x_1 - x_2\| & \cdots & \|x_1 - x_n\| \\ \|x_2 - x_1\| & \|x_2 - x_2\| & \cdots & \|x_2 - x_n\| \\ \vdots & \vdots & \vdots & \vdots \\ \|x_n - x_1\| & \|x_n - x_2\| & \cdots & \|x_n - x_n\| \end{bmatrix}, \tag{30}$$

$$f(x) = a_1 \phi\left(\|x - x_1\|\right) + \cdots a_n \phi\left(\|x - x_n\|\right),$$

where $f(x)$ is the estimator function, $[a_1, \ldots, a_n]$ are the constant coefficients of the function, and r_i, $(i = 1, 2, \ldots, n)$ are the same r-slices. To convert the numerical function φ to the φ matrix, we can use the formulas in the form $\phi(A)_{ij} = \phi(A_{ij})$. The function φ is applied to the Euclidean distance matrix and is called the transfer matrix. When the type of transfer function is determined, by using the number n equation, on the input data $x(x_1, x_2, \ldots, x_n)$, we can obtain the coefficients $a_1, \ldots, a_n$:

$$\begin{aligned} a_1 \phi(x_1 - x_1) + \cdots a_n \phi(x_1 - x_n) &= y_1, \\ a_1 \phi(x_2 - x_1) + \cdots a_n \phi(x_2 - x_n) &= y_2, \\ &\vdots \\ a_1 \phi(x_n - x_1) + \cdots a_n \phi(x_n - x_n) &= y_m. \end{aligned} \tag{31}$$

This set holds until vectors $x_i \neq x_j$, for $i \neq j$ and the matrix $n \times n$ derived from the set of equations, are inverse. In order to balance the complexity accuracy, we can use all the n data points in this model or that the number of k points, $c_1, \ldots, c_k$, is considered for the RBF center, which is $k \ll n$:

$$f_n(x) = \sum_{j=1}^{n} a_j \phi\left(\|x - x_j\|\right),$$

$$f_k(x) = \sum_{j=1}^{k} a_j \phi\left(\|x - x_j\|\right), \tag{32}$$

with assuming the output multiplicity, the generalization function will be given, where α is the vector of coefficients and v is the coefficient matrix. The RBF network, slightly different from the display mode, is a layer of two mappings: first, the mapping space dimensions from $R^n n$ to R^k and then mapped to the output layer R^m [42, 43]:

$$x \longrightarrow \begin{bmatrix} \|x - r_1\| \\ \|x - r_2\| \\ \vdots \\ \|x - r_k\| \end{bmatrix} \longrightarrow \phi \begin{bmatrix} \|x - r_1\| \\ \|x - r_2\| \\ \vdots \\ \|x - r_k\| \end{bmatrix} \longrightarrow w \begin{bmatrix} \phi(\|x - r_1\|) \\ \phi(\|x - r_2\|) \\ \vdots \\ \phi(\|x - r_k\|) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix},$$

$$y_h(x) = \sum_{j=1}^{k} w_{jh} \phi_j \left(\|x - r_j\| \right),$$

(33)

where r_j is considered for the RBF center, and in this paper, r_j are cut-r_j and w_{jh} are output layer weights. In the design of RBF networks, in most cases, the number of central layer units is significantly less than the number of input data, but in the case of interpolation using a radial base function, the number of middle layer centers with dimensions is equivalent to the input data, and its inputs correspond to input data. In the cases where the transfer function is dependent on the variable, for example, the Gaussian function (also depends on the variable σ), and because in many models, it is assumed to be Gaussian (normal), the determination of the width of the function is important. Here, the width factor of the function controls the softness and elongation of the transfer function. When the width of the function is small, the related space and representation of the function are small as well. As a result, a large number of centers are needed for training. In contrast, the estimation accuracy decreases if the space is relevant and represents the broad function. In interpolation methods, it is common practice to use the radial base function to consider the width of all centers.

5. Numerical Examples

In this section, to show the behavior and properties of this new method, we discuss the simulation results of one example. The simulation is conducted on Matlab12, and the objective function in (12) minimizer engaged is fminunc. The initial weights were randomly selected.

Example 1. Consider the following first-order FDE:

$$\begin{cases} y' = y(t), & t \in [0, 1], \\ y(0) = (0.75 + 0.25r, 1.125 - 0.125r). \end{cases}$$

(34)

The exact solution of the FDE is

$$y(1) = ((0.75 + 0.25r)e, (1.125 - 0.125r)e) \quad r \in [0, 1].$$

(35)

Assume that the initial value based on Z-numbers is as follows:

$$y(0) = ((0.75 + 0.25r, 1.125 - 0.125r), (1)).$$

(36)

In this case, differential equation (34) can be rewritten as follows:

$$\begin{cases} y' = y(t), & t \in [0, 1], \\ y(0) = ((0.75 + 0.25r, 1.125 - 0.125r), (1)). \end{cases}$$

(37)

And assume that the exact solution of the ZDE is

$$y(1, r) = (((0.75 + 0.25r)e, (1.125 - 0.125r)e),$$

$$1 - \left(e^{-0.216((0.75+0.25r)e)} - e^{-0.216((1.125-0.125r)e)} \right))$$

$$r \in [0, 1].$$

(38)

Our proposed method is as follows:

$$y_T(t) = \left(\left(\underline{A}_T(t), \overline{A}_T(t) \right), B_T(t) \right),$$

(39)

where

$$\begin{cases} \underline{A}_T(t) = 0.75 + 0.25r + t \sum_{j=1}^{n} \underline{w}_j \phi_j \left(\|t - r_j\| \right), \\ \overline{A}_T(t) = 1.125 - 0.125r + t \sum_{j=1}^{n} \overline{w}_j \phi_j \left(\|t - r_j\| \right), \end{cases}$$

(40)

$$B_T(t) = 1 - t \left(e^{-\lambda \left(\underline{A}_T(t) \right)} - e^{-\lambda \left(\overline{A}_T \right)} \right).$$

Comparison of the approximate solution with the RBF network training for 50 points and with a choice of $\lambda = 1$ is shown in Table 2. The convergence of the neural network weights for each r-cut is shown in Figure 2. Comparison of the approximate method with the exact method is shown in Figure 3. The comparison of the confidence level of the actual answer and the approximate solution is shown in Figure 4.

Example 2. Consider the following first-order ZDE:

$$\begin{cases} y' = 3t^2 y(t), & t \in [0, 1], \\ y(0) = ((0.5\sqrt{r}, 0.2\sqrt{1-r} + 0.5), (1).) \end{cases}$$

(41)

The exact solution of the FDE is

$$y(1) = (0.5\sqrt{r} \, e, (0.2\sqrt{1-r} + 0.5)e), \quad r \in [0, 1].$$

(42)

TABLE 2: Comparison of the exact Z and Z_T approximated solutions for $\lambda = 1$.

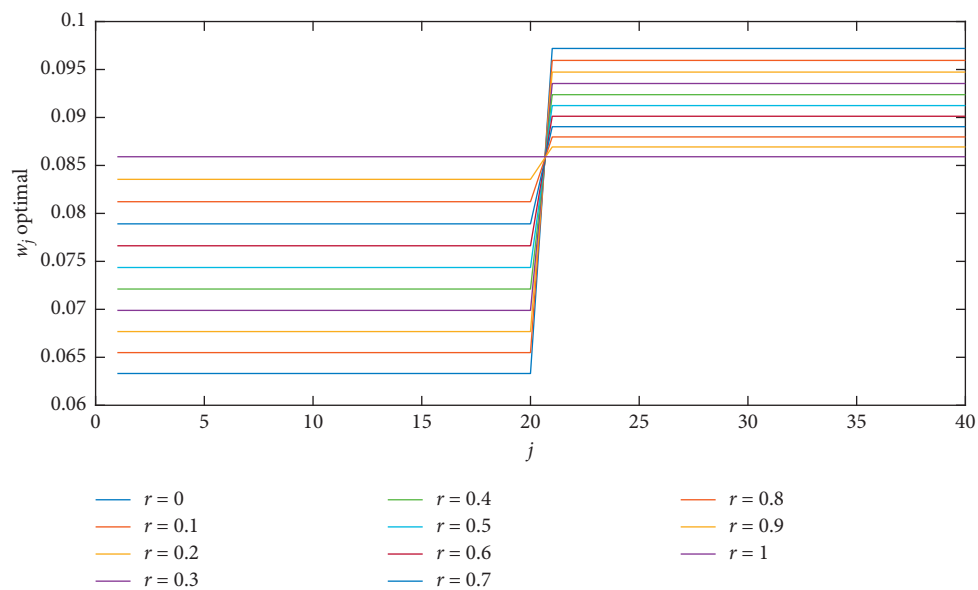| r	$\underline{A}_T(t)$	$\underline{A}(t)$	$\overline{A}_T(t)$	$\overline{A}(t)$	$B_T(t)$	$B(t)$
0	2.00	2.03	3.04	3.05	0.874	0.872
0.1	2.07	2.10	3.02	3.02	0.885	0.885
0.2	2.14	2.17	2.99	2.99	0.899	0.898
0.3	2.21	2.24	2.95	2.95	0.911	0.912
0.4	2.28	2.31	2.92	2.92	0.925	0.925
0.5	2.35	2.37	2.88	2.88	0.937	0.937
0.6	2.42	2.44	2.84	2.85	0.950	0.949
0.7	2.50	2.51	2.81	2.82	0.962	0.962
0.8	2.57	2.58	2.78	2.78	0.975	0.975
0.9	2.64	2.65	2.75	2.75	0.987	0.987
1	2.71	2.71	2.71	2.71	1	1

FIGURE 2: The convergence of the neural network weights for each r-cut.
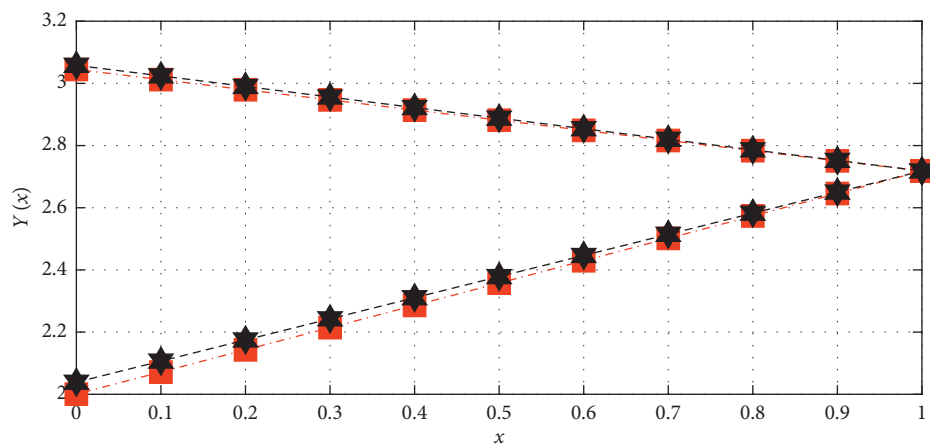
FIGURE 3: The exact and computed solution for Example 1. Yellow chart indicates real data and black chart indicates approximate data.
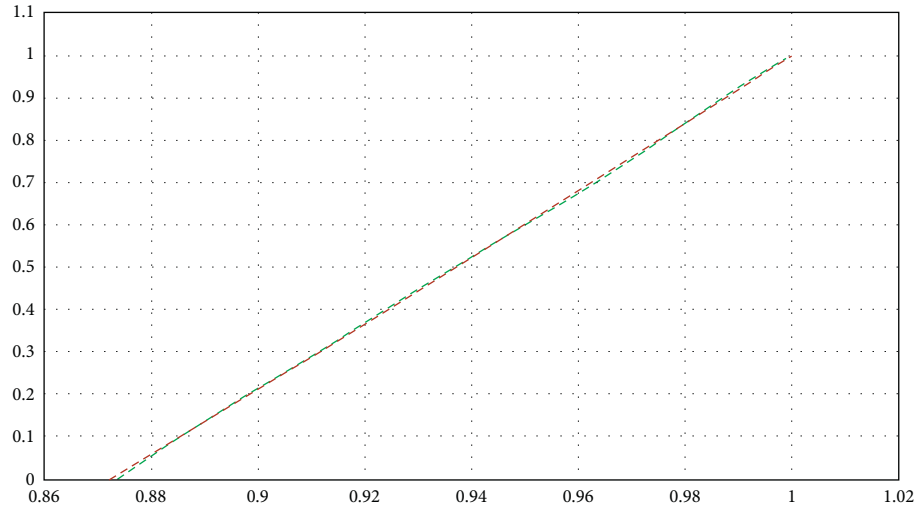
FIGURE 4: Comparison of the confidence level of the actual answer and the approximate solution for Example 1. Red color indicates confidence level in exact solution, and green color indicates confidence level in approximate solution.

TABLE 3: Comparison of the exact Z and Z_T approximated solutions for $\lambda = 1$.

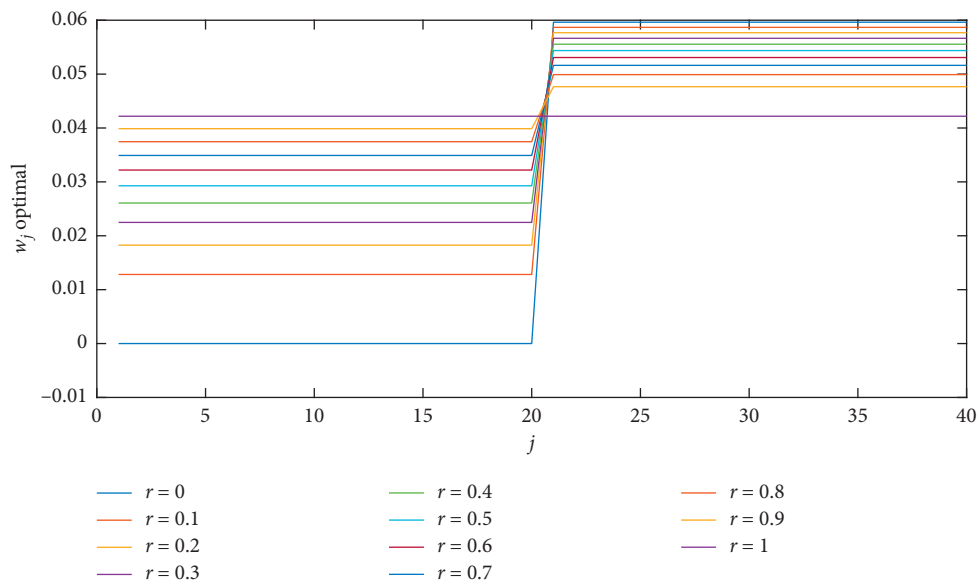| r	$\underline{A}_T(t)$	$\underline{A}(t)$	$\overline{A}_T(t)$	$\overline{A}(t)$	$B_T(t)$	$B(t)$
0	−0.000000002	0	1.87	1.90	0.687	0.663
0.1	0.41	0.42	1.85	1.87	0.769	0.754
0.2	0.58	0.60	1.82	1.84	0.804	0.793
0.3	0.72	074	1.79	1.81	0.833	0.824
0.4	0.83	0.85	1.76	1.78	0.856	0.848
0.5	0.93	0.96	1.72	1.74	0.878	0.873
0.6	1.02	1.05	1.68	1.90	0.899	0.866
0.7	1.11	1.13	1.64	1.65	0.919	0.916
0.8	1.19	1.21	1.58	1.60	0.940	0.937
0.9	1.27	1.28	1.51	1.53	0.963	0.960
1	1.34	1.35	1.34	1.35	1	1

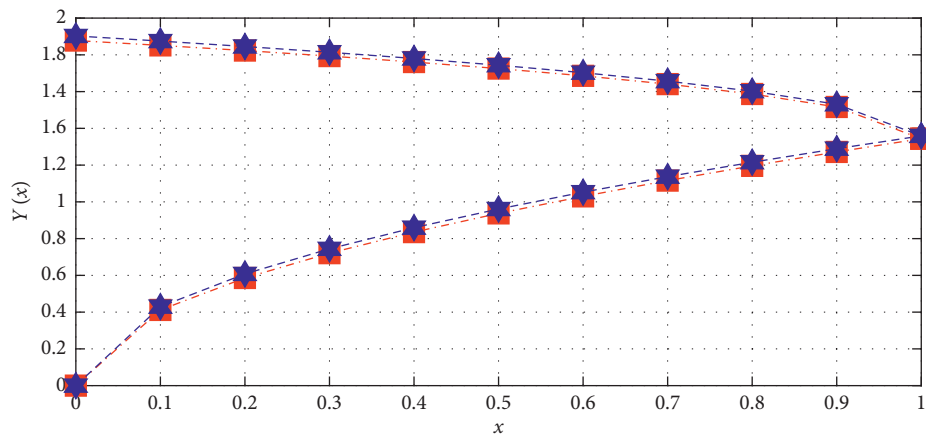FIGURE 5: The convergence of the neural network weights for each r-cut for Example 2.

FIGURE 6: The exact and computed solution for Example 2. Red chart indicates real data, and blue chart indicated approximate data.
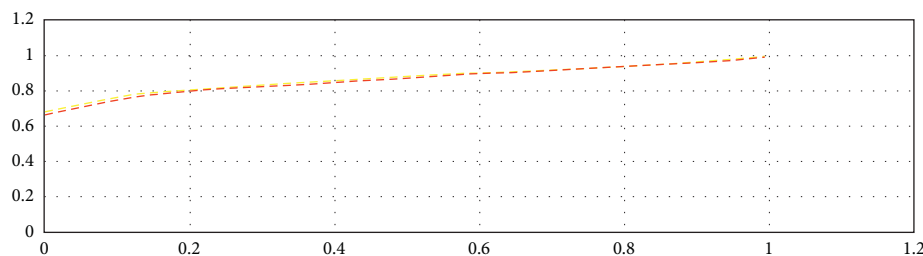

FIGURE 7: Comparison of the confidence level of the actual answer and the approximate solution for Example 2. Red color indicates confidence level exact solution and yellow color indicates confidence level approximate solution.

Assume that the initial value based on Z-numbers is as follows:

$$y(0) = ((0.5\sqrt{r}\,e, 0.2\sqrt{1-r} + 0.5), (1)). \quad (43)$$

In this case, differential equation (34) can be rewritten as follows:

$$\begin{cases} y' = 3t^2 y(t), & t \in [0,1], \\ y(0) = ((0.5\sqrt{r}, 0.2\sqrt{1-r} + 0.5), \quad (1).) \end{cases} \quad (44)$$

And assume that the exact solution of the ZDE is

$$y(1, r) = ((0.5\sqrt{r}\,e, (0.2\sqrt{1-r} + 0.5)e),$$
$$1 - \left(e^{-0.216(0.5\sqrt{r}\,e)} - e^{-0.216((0.2\sqrt{1-r} + 0.5)e)}\right),$$
$$r \in [0,1]. \quad (45)$$

Our proposed method is as follows:

$$y_T(t) = ((\underline{A}_T(t), \overline{A}_T(t)), B_T(t)), \quad (46)$$

where

$$\begin{cases} \underline{A}_T(t) = 0.5\sqrt{r} + t\sum_{j=1}^{n} \underline{w}_j \phi_j\left(\|t - r_j\|\right), \\ 0.2\sqrt{1-r} + 0.5 + t\sum_{j=1}^{n} \underline{w}_j \phi_j\left(\|t - r_j\|\right), \end{cases} \quad (47)$$

$$B_T(t) = 1 - t\left(e^{-\lambda(\underline{A}_T(t))} - e^{-\lambda(\overline{A}_T)}\right).$$

Comparison of the approximate solution with the RBF network training for 50 points and with a choice of $\lambda = 1$ is shown in Table 3. The convergence of the neural network weights for each r-cut is shown in Figure 5. The comparison of the approximate method with the exact method is shown in Figure 6. The comparison of the confidence level of the actual answer and the approximate solution is shown in Figure 7.

6. Concluding Remarks

In this paper, we proposed a new approach for solving first-order differential equations of Z-numbers' initial value under uncertainty using radial basic function under generalized H-differentiability. At first, the problem was divided into two parts: the first part of the limitation and the second part of the reliability of the first part. We employed the RBF method for finding upper and lower solutions of the equation of the problem limitation section. The main advantage of this approach is that the fuzzy equation was reduced to the problem of solving two systems of linear equations. Then, we use the information obtained of the proposed method of the first part to calculate the confidence level, and an exponential function was proposed to calculate the reliability of a function. The numerical investigation presented in this paper shows that excellent accuracy can be obtained even when few nodes are used in analysis. In contrast, many more nodes are needed to achieve relatively good accuracy in other methods. Numerical example is

included to demonstrate the validity and applicability of the technique and is performed on a computer using a code written in Matlab. The method can be implemented for solving linear and nonlinear equations in higher dimensions.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Abbasbandy and T. AllahViranloo, "Numerical solution of fuzzy differential equation by RungeKutta method," *Nonlinear Studies*, vol. 11, no. 1, pp. 117–129, 2004.

[2] E. Babolian, H. Sadeghi, and S. Javadi, "Numerically solution of fuzzy differential equations by Adomian method," *Applied Mathematics and Computation*, vol. 149, no. 2, pp. 547–557, 2004.

[3] S. S. L. Chang and L. A. Zadeh, "On fuzzy mapping and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 2, no. 1, pp. 30–34, 1972.

[4] D. Dubois and H. Prade, "Towards fuzzy differential calculus part 3: differentiation," *Fuzzy Sets and Systems*, vol. 8, no. 3, pp. 225–233, 1982.

[5] S. Effati and M. Pakdaman, "Artificial neural network approach for solving fuzzy differential equations," *Information Sciences*, vol. 180, no. 8, pp. 1434–1457, 2010.

[6] O. Kaleva, "Fuzzy differential equations," *Fuzzy Sets and Systems*, vol. 24, no. 3, pp. 301–317, 1987.

[7] M. L. Puri and D. A. Ralescu, "Fuzzy random variables," *Journal of Mathematical Analysis and Applications*, vol. 114, no. 2, pp. 409–422, 1986.

[8] L. Stefanini, L. Sorini, and M. L. Guerra, "Parametric representation of fuzzy numbers and application to fuzzy calculus," *Fuzzy Sets and Systems*, vol. 157, no. 18, pp. 2423–2455, 2006.

[9] M. Esmaeilbeig, M. Paripour, and G. Garmanjani, "Approximate solution of the fuzzy fractional Bagley-Torvik equation by the RBF collocation method," *Computational Methods for Differential Equations*, vol. 56, pp. 86–214, 2018.

[10] W. Pedrycz and S. Member, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, 1998.

[11] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[12] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[13] S. Renals, "Radial basis function network for speech pattern classification," *Electronics Letters*, vol. 25, no. 7, pp. 437–439, 1989.

[14] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[15] N. Flyer, G. B. Wright, and B. Fornberg, "Radial basis function-generated finite differences: a mesh-free method for computational geosciences," *Handbook of Geomathematics*, Springer, Berlin, Germany, 2014.

[16] M. J. Hillier, E. M. Schetselaar, E. A. de Kemp, and G. Perron, "Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions," *Mathematical Geosciences*, vol. 67, pp. 1–23, 2014.

[17] M. R. Mustafa, R. B. Rezaur, H. Rahardjo, and M. H. Isa, "Prediction of pore-water pressure using radial basis function neural network," *Engineering Geology*, vol. 135–136, pp. 40–47, 2012.

[18] I. Ostermann, "Modeling heat transport in deep geothermal systems by radial basis functions," Ph.D. thesis, University of Kaiserslautern, Geomathematics Group, Verlag Dr. Hut GmbH, Munich, Germany, 2011.

[19] M. Zhang, K. Wang, C. Zhang et al., "Using the radial basis function network model to assess rocky desertification in northwest Guangxi, China," *Environmental Earth Sciences*, vol. 62, no. 1, pp. 69–76, 2011.

[20] B. Fornberg and C. Piret, "On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere," *Journal of Computational Physics*, vol. 227, no. 5, pp. 2758–2780, 2008.

[21] M. Dehghan and R. Salehi, "A method based on meshless approach for the numerical solution of the two-space dimensional hyperbolic telegraph equation," *Mathematical Methods in the Applied Sciences*, vol. 35, no. 10, pp. 1220–1233, 2012.

[22] E. Larsson and B. Fornberg, "A numerical study of some radial basis function based solution methods for elliptic PDEs," *Computers & Mathematics with Applications*, vol. 46, no. 5–6, pp. 891–902, 2003.

[23] M. Uddin, "RBF meshless method of lines for the numerical solution of nonlinear sine-gordon equation," *Walailak Journal of Science and Technology (WJST)*, vol. 11, no. 4, pp. 349–360, 2013.

[24] S. Kazem, J. A. Rad, and K. Parand, "Radial basis functions methods for solving Fokker-Planck equation," *Engineering Analysis with Boundary Elements*, vol. 36, no. 2, pp. 181–189, 2012.

[25] A. Mohebbi, M. Abbaszadeh, and M. Dehghan, "The use of a meshless technique based on collocation and radial basis functions for solving the time fractional nonlinear Schrödinger equation arising in quantum mechanics," *Engineering Analysis with Boundary Elements*, vol. 37, no. 2, pp. 475–485, 2013.

[26] S.-J. Lai, B.-Z. Wang, and Y. Duan, "Solving Helmholtz equation by meshless radial basis functions method," *Progress in Electromagnetics Research B*, vol. 24, pp. 351–367, 2010.

[27] L. A. Zadeh, "A note on Z-numbers," *Information Sciences*, vol. 181, no. 14, pp. 2923–2932, 2011.

[28] R. R. Yager, "On Z-valuations using Zadeh's Z-numbers," *International Journal of Intelligent Systems*, vol. 27, no. 3, pp. 259–278, 2012.

[29] B. Kang, D. WEI, Y. LI, and Y. DENG, "Decision making using Z-numbers under uncertain environment," *Journal of Computational Information Systems*, vol. 7, pp. 2807–2814, 2012.

[30] S. Ezadi and N. Allahviranloo, "Numerical solution of linear regression based on Z-numbers by improved neural network," *IntellIgent AutomAtIon and Soft ComputIng*, vol. 34, pp. 1–11, 2017.

[31] S. Ezadi and T. Allahviranloo, "New multi-layer method for Z-number ranking using Hyperbolic Tangent function and convex combination," *Intelligent Automation Soft Computing*, vol. 24, pp. 1–7, 2017.

[32] S. Ezadi and N. Allahviranloo, "Two new methods for ranking of Z-numbers based on sigmoid function and sign method," *International Journal of Intelligent Systems*, vol. 56, pp. 1–12, 2018.

[33] R. A. Alive, A. V. Alizadeh, and O. H. Huseynov, "The arithmetic of discrete Z-numbers," *Information Sciences*, vol. 290, pp. 134–155, 2015.

[34] B. Kang, D. Wei, Y. Li, and Y. Deng, "A method of converting Z-number to classical fuzzy number," *Journal of Information and Computational Scienc*, vol. 3, pp. 703–709, 2012.

[35] D. Mohamad, S. A. Shaharani, and N. H. Kamis, "A Z-number based decision making procedure with ranking fuzzy numbers method," *AIP Conference Proceedings*, vol. 1635, pp. 160–166, 2014.

[36] A. A. Rafik, H. Oleg, R. Aliyev, and A. Alizadeh, *The Arithmetic of Z-Numbers*, World Scientific Publishing Company, Singapore, 2015.

[37] S. Pirmuhammadi, T. Allahviranloo, and M. Keshavarz, "The parametric form of Z-number and its application in Z-number initial value problem," 2017.

[38] Y. B. Dibike, D. Solomatine, and M. B. Abbott, "On the encapsulation of numerical-hydraulic models in artificial neural network," *Journal of Hydraulic Research*, vol. 37, no. 2, pp. 147–161, 1999.

[39] J. C. Mason, R. K. Price, and A. Tem'Me, "A neural network model of rainfall-runoff using radial basis functions," *Journal of Hydraulic Research*, vol. 34, no. 4, pp. 537–548, 1996.

[40] S. Seikkala, "On the fuzzy initial value problem," *Fuzzy Sets and Systems*, vol. 24, no. 3, pp. 319–330, 1987.

[41] B. J. C. Baxter, "Conditionally positive functions and p-norm distance matrices," *Constructive Approximation*, vol. 7, no. 1, pp. 427–440, 1991.

[42] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, New Jersey, NJ, USA, 2nd edition, 1999.

[43] A. Zhang and L. Zhang, "RBF neural networks for the prediction of building interference effects," *Computers & Structures*, vol. 82, no. 27, pp. 2333–2339, 2004.