

Research Article

A Generalized Bridge Regression in Fuzzy Environment and Its Numerical Solution by a Capable Recurrent Neural Network

Delara Karbasi , Mohammad Reza Rabiei , and Alireza Nazemi

Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran

Correspondence should be addressed to Mohammad Reza Rabiei; rabie1354@yahoo.com

Received 6 September 2020; Revised 18 October 2020; Accepted 26 October 2020; Published 26 November 2020

Academic Editor: Heng Liu

Copyright © 2020 Delara Karbasi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bridge regression is a special family of penalized regressions using a penalty function $\sum |A_j|^{\gamma}$ with $\gamma \geq 1$ that for $\gamma = 1$ and $\gamma = 2$, it concludes lasso and ridge regression, respectively. In case where the output variable in the regression model was imprecise, we developed a bridge regression model in a fuzzy environment. We also exhibited penalized fuzzy estimates for this model when the input variables were crisp. So, we perform the presented optimization problem for the model that leads to a multiobjective program. Also, we try to determine the shrinkage parameter and the tuning parameter from the same optimization problem. In order to estimate fuzzy coefficients of the proposed model, we introduce a hybrid scheme based on recurrent neural networks. The suggested neural network model is constructed based on some concepts of convex optimization and stability theory which guarantees to find the approximate parameters of the proposed model. We use a simulation study to depict the performance of the proposed bridge technique in the presence of multicollinear data. Furthermore, real data analysis is used to show the performance of the proposed method. A comparison between the fuzzy bridge regression model and several other shrinkage models is made with three different well-known fuzzy criteria. In this study, we visualize the performance of the model by Taylor's diagram and Bubble plot. Also, we examine the predictive ability of the model, thus, obtained by cross validation. The numerical results clearly showed higher accuracy of the proposed fuzzy bridge method compared to the other existing fuzzy regression models: fuzzy bridge regression model, multiobjective optimization, recurrent neural network, stability convergence, and goodness-of-fit measure.

1. Introduction

In regression analysis, researchers often face the problem of multicollinearity defined as the existence of nearly linear dependency between the columns of the design matrix leading to wide confidence intervals for individual parameters or a linear combination of the parameters [1–3]. To overcome the multicollinearity problem, the ridge regression methodology has been widely studied in the literature [1, 4]. The proposed ridge estimator performs better than other estimators that rely on ordinary least-squares. Moreover, when multicollinearity happens, the least-squares methods may lead to poor prediction. In such cases, the ridge regression method often gives better accuracies compared to the least square technique. As an extension, Frank and Friedman [3] introduced bridge regression which includes ridge regression with $\gamma = 2$ and subset selection with $\gamma = 1$ as

special cases. The authors in [3] did not solve the estimator of bridge regression for any given $\gamma > 0$; however, they indicated that it is more appropriate to optimize the parameter γ . As a special case of the bridge regression (with $\gamma = 1$), the authors in [5] introduced the lasso regression model. The authors in [6] extended the lasso method, called fused lasso, in cases where the number of features is much greater than the sample size. Furthermore, the lasso method was modified by Zou [7] who proposed the adaptive lasso with adaptive weights to penalize different coefficients in the l_1 -penalized. Park and Casella [8] proposed the Bayesian lasso which provides interval estimates that can guide variable selection.

Bridge regression has been successfully applied to solve problems of wide-ranging scope and complexity, starting from the pioneering works in [3, 9]. Zou and Yuan [10] imposed the F_{∞} norm on support vector machines in the

classification context. Huang et al. [11] studied the asymptotic properties of bridge regression. The authors in [12] suggested an adaptive elastic net combining the strengths of the quadratic regularization and the adaptively weighted lasso shrinkage. The authors in [13] showed that bridge regression possesses oracle, sparsity, and unbiasedness. In [14], the authors provided a bridge regression that adaptively selects the penalty order from data and produces flexible solutions in various settings. Polson et al. [15] proposed a Bayesian bridge estimator for regularized regression. Mallick and Yi [16] considered Bayesian bridge regression and provided sufficient conditions for strong posterior consistency under a sparsity assumption on a high-dimensional parameter. Finally, in [17], the authors consider a bridge-randomized penalty of regression coefficients by incorporating uncertainty penalty into Bayesian bridge quantile regression.

In traditional regression models, the observed values of the dependent and independent variables are supposed to be exact numbers. However, in real-world applications, because of parameter uncertainty, human errors data are usually not exact but have uncertainty. As a result, the classical regression models experience some restriction, which need to be adjusted accordingly to deal with fuzzy data (see [18–32]). As in regression models in traditional statistical theory, multicollinearity is also an important issue in fuzzy regression models. Accordingly, the fuzzy bridge regression model which combines bridge regression with the fuzzy regression model in order to reduce the effect of multicollinearity for the first time in this paper is considered. To solve such a problem, in this paper, the fuzzy bridge regression model is converted to a nonlinear multiobjective problem. The proposed multiobjective problem is solved by the scalarization method making the multiobjective function create a single solution, and the weight is determined before the optimization process (see [33]). A large weight that is determined to an objective function shows that the considered function has a higher priority compared to the ones with a smaller weight. There are several approaches to determining the weight of scalarization, and the one that is used in this paper is introduced in [34]. For a more detailed presentation of the methods treated here, as well as other related topics, refer to [35] and the references therein.

The fuzzy regression problems can be solved by applying artificial neural networks and evolutionary algorithms (see [36–45]). For example, Ishibuchi et al. [39] proposed a learning algorithm of fuzzy neural networks with triangular fuzzy weights, and Hayashi et al. [40] fuzzified the delta rule. Neural network solutions were examined to fuzzy problems by Buckley and Eslami in [41]. The authors in [46] extended the risk-neutral model suggested in [38, 47] by adopting a multilayered feed-forward neural network where weights, biases, input, and output variables were assumed to be LR-fuzzy. Furthermore, Coppi et al. [48] studied on FLR by using the least-squares approach with LR-fuzzy numbers. In [49], a class of fuzzy clusterwise regression model which integrates fuzzy cluster analysis into the fuzzy regression model was introduced. The authors in [50] also used triangular fuzzy numbers for fuzzy modeling with a robust

framework through the least-squares approach. The topic of the numerical solution of fuzzy polynomials by a fuzzy neural network was investigated in [42]. Mosleh et al. [44, 51, 52] and Otadi [53] provide a hybrid approach based on a feed-forward neural network to approximate fuzzy coefficients of fuzzy linear, nonlinear, and polynomial regression models. Roh et al. [54] concerned an FLR model based on the design approach of polynomial neural networks. The authors in [55, 56] used a random weight network to develop a fuzzy nonlinear regression model. A fuzzified radial basis function network for obtaining estimations of fuzzy regression models is utilized in [57]. For further papers on this topic, see [19, 20, 46, 58–61].

The aforesaid references in the main refer to a learning algorithm which is used to adjust the parameters of the feed-forward fuzzy neural network. The although authors in [62] presented a hybrid scheme based on recurrent neural networks, and there is no record for solving the fuzzy bridge regression model by a neural network with stability and convergence properties. Furthermore, to obtain weighting coefficients, the authors use a recurrent procedure where the inverse of a matrix in each iteration needs to be computed explicitly. Obviously, in high-dimensional cases, the computational complexity is extremely high. Besides, the convergence of the proposed neural network methods depends on choosing a suitable initial point. Consequently, suggesting a recurrent neural network to deal with fuzzy bridge regression problems without penalty parameter with rigorous stability and convergence analysis is completely necessary and relevant. Proposing a capable neural network for analog hardware implementation which gives a good solution of the fuzzy bridge regression problem is the main advantage of this work. The optimality conditions of convex programming for solving the related quadratic formulation of the fuzzy bridge regression problem is studied according to the Karush–Kuhn–Tucker (KKT). It is proved that the equilibrium point of the proposed neural network is equivalent to the KKT point of the related quadratic programming formulation. The existence and uniqueness of an equilibrium point of the network are also analyzed. An adequate condition to ensure the existence and global asymptotic stability for the unique equilibrium point of the neural network are obtained by constructing an appropriate Lyapunov function. Additionally, the global convergence of the model is analyzed. The proposed recurrent neural networks in this paper are substantially different from the traditional ones for two important reasons: Firstly, the neural network method has its advantage in dealing with real-time optimization problems which are usually imperative in many applications. However, traditional optimization methods may not be competent due to the problem's stringent requirement for computational time. As a result, many continuous-time neural networks for solving real-time optimization have been widely expanded (see [63–67]). Secondly, neural networks for solving optimization problems are hardware implementable. In other words, neural networks can be implemented by using integrated circuits.

The remainder of this paper is organized as follows. In Section 2, some preliminaries of the fuzzy sets and fuzzy

calculus are briefly described. In Section 3, the fuzzy bridge regression problem is introduced. In Section 4, an optimization model to solve the fuzzy bridge linear regression model with LR-fuzzy-dependent variable and nonfuzzy explanatory variables is proposed, and it is transformed into the equivalent quadratic form. In Section 5, based on the (KKT) optimality conditions a neural network model for solving the related quadratic formulation with stability and convergence properties is studied. The goodness-of-fit indices are also studied in Section 6. A numerical experiment and a simulation study to illustrate the effectiveness of the model in the presence of multicollinearity are provided in Section 7. Some concluding remarks are given in Section 8. Proofs of technical statements are stated in Appendix, to better focus on the computational part in the body of paper.

2. Preliminaries

In this section, first, we summarize some basic concepts of fuzzy set and fuzzy arithmetic which are used throughout the paper (see [68]); then, some definitions and a basic theorem in multiobjective optimization are briefly discussed.

2.1. Fuzzy Set and Fuzzy Arithmetic. A fuzzy set A is denoted by $A = \{(x, \mu_A(x)) | x \in X\}$ where the function $\mu_A(x): X \rightarrow [0, 1]$, is the degree of membership of element x in A . The h -level set of a fuzzy set A is a crisp subset of X denoted by $A(h)$ that is defined by $A(h) = \{x \in X | \mu_A(x) \geq h\}$, where $h \in [0, 1]$.

A fuzzy number A is a normal and convex fuzzy subset of the real line \mathbb{R} with bounded support. In a special case, a class of fuzzy numbers, so-called LR-fuzzy numbers, is used as a parametric form:

$$\mu_A(x) = \begin{cases} L\left(\frac{a_m - x}{a_l}\right), & x \leq a_m, \\ R\left(\frac{x - a_m}{a_u}\right), & x > a_m, \end{cases} \quad (1)$$

where a_m is the mode of A and $a_l, a_r > 0$ are the left and right spreads of A , respectively. $L, R: \mathbb{R}^+ \rightarrow [0, 1]$ are decreasing shape functions S , with $S(0) = 1, S(z) < 1$ for all $z > 0, S(z) > 0$ for all $z < 1$ and $S(1) = 0$ (or $S(z) > 0$ for all z and $S(+\infty) = 0$). As an abbreviated notation, we can define an LR-fuzzy number A with the membership function $\mu_A(x)$ in (1) by $A = (a_m, a_l, a_u)_{LR}$. If $L(x)$ and $R(x)$ are both linear functions on the domains $\{x | 0 < L(x), R(x) < 1\}$, the corresponding LR fuzzy number is a triangular fuzzy number, which is symbolically written as $A = (a_m, a_l, a_u)_T$. Moreover, if $a_l = a_u$, A becomes a symmetric triangular fuzzy number which is denoted by $A = (a_m, a_l)_T$.

Suppose two fuzzy numbers A and B are represented by LR-fuzzy numbers of the form $A = (a_m, a_l, a_u)_{LR}$ and $B = (b_m, b_l, b_u)_{LR}$. The sum of A and B is again an LR-fuzzy number of the form

$$A + B = (a_m + b_m, a_l + b_l, a_u + b_u)_{LR}. \quad (2)$$

The scalar multiplication kA is defined as

$$kA = \begin{cases} (ka_m, ka_l, ka_u)_{LR}, & k > 0, \\ (ka_m, -ka_u, -ka_l)_{RL}, & k < 0. \end{cases} \quad (3)$$

2.2. MultiObjective Optimization. A multiobjective optimization problem (MOP) can be defined by the following problem:

$$\text{minimize } F(\mathbf{X}), \quad (4)$$

subject to

$$g_q(\mathbf{X}) \leq 0, \quad q = 1, \dots, m, \quad (5)$$

$$\mathbf{X} \in \mathbb{R}^l, \quad (6)$$

where $F(\mathbf{X}) = [f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_k(\mathbf{X})]^T: \mathbf{X} \rightarrow \mathbb{R}^k$ is a vector with the values of objective functions to be minimized, \mathbf{X} is the vector containing the decision variables, defined in the design space \mathbb{R}^l , and $g_i(\mathbf{X})$ represents the i^{th} inequality constraint function. Equations (5) and (6) define the region of feasible solutions, S , in the design space \mathbb{R}^l .

Definition 1 (See [69]). A decision vector $\mathbf{X}^* \in S$ is Pareto optimal if there does not exist another $\mathbf{X} \in S$ such that $f_i(\mathbf{X}) \leq f_i(\mathbf{X}^*)$ for all $i = 1, 2, \dots, k$ and $f_j(\mathbf{X}) < f_j(\mathbf{X}^*)$ for, at least, one index j .

Definitions 1 introduces global optimality. However, it may be difficult to generate globally optimal solutions if the problem is not convex (that is, if some of the objective functions or the feasible region are not convex). In that case, the solutions obtained may be only locally optimal unless a global solver is available. Note that, under the assumptions mentioned in the problem formulation, the Pareto optimal solutions exist (see [70]).

Classically, multiobjective optimization problems are often solved using scalarization techniques. The scalarization method makes the multiobjective function create a single solution, and the weight is determined before the optimization process. A large weight that is given to an objective function shows that the said function has a higher priority compared to the ones with a smaller weight.

One of the most intuitive scalarization methods used to obtain a single unique solution for MOP is the weighted sum method. In this approach, the MOP (4)–(6) is converted into a scalar preference function using a linear weighted sum function of the form

$$\text{minimize } \sum_{s=1}^k (w_s f_s(\mathbf{X})), \quad (7)$$

subject to

$$g_q(\mathbf{X}) \leq 0, \quad q = 1, \dots, m, \quad (8)$$

$$\sum_{s=1}^k w_s = 1, \quad w_s \geq 0, \quad (9)$$

where $\mathbf{X} \in \mathbb{R}^l$.

Theorem 1 (see [69]). *The solution of the MOP (4)–(6) generated by the weighted sum method (problem (7)–(9)) is Pareto optimal if $w_s > 0$ for $s = 1, 2, \dots, k$.*

According to Theorem 1, for positive weights and convex problem, the optimal solutions of the substitute problem (7)–(9) are Pareto optimal.

3. Fuzzy Bridge Regression Model

Consider a set of data $\{(x_{i0}, \dots, x_{ip}, Y_i) | i = 1, \dots, n\}$, in which x_{ij} ($i = 1, \dots, n$ and $j = 0, 1, \dots, p$) is the value of j^{th} independent variable and Y_i ($i = 1, \dots, n$) is the corresponding value of dependent variable Y in the i^{th} case. The purpose of the linear regression model is to fit a linear model to the given data. This model can be considered as follows:

$$\hat{Y}_i = A_0 + A_1x_{i1} + \dots + A_px_{ip}, \quad i = 1, \dots, n, \quad (10)$$

where A_0, A_1, \dots, A_p denote the list of regression coefficients (parameters). In some cases, we may need to consider that the relationship expressed in (10) may be fuzzy, and then, the following different cases are obtained:

- (a) The case where the predictor variable is fuzzy, but the parameters are crisp
- (b) The case of a crisp predictor and fuzzy parameters
- (c) The case of a fuzzy predictor and fuzzy parameters

It should be noted that the observed responses in all cases are naturally fuzzy. As in LR models in traditional statistical theory, in most of the empirical works, we often encounter the multicollinearity phenomenon in FLR models, as a result of highly intercorrelated explanatory variables. In the last

two decades, different shrinkage methods have been proposed for estimating regression coefficients; among them, the bridge shrinkage method has received considerable attention. The authors in [3] proposed the following optimization problem to solve the bridge regression for any given $\gamma \geq 0$:

$$\text{minimize } \sum_{i=1}^n \left(Y_i \sum_{j=0}^p A_j x_{ij}^2 \right), \quad (11)$$

subject to

$$\sum_{j=0}^p |A_j|^\gamma \leq t, \quad (12)$$

where the positive parameter t represents the tuning constant. Also, the optimal γ selection will increase the efficiency of the model. It is shown that for $\gamma \geq 1$, the bridge regression model is a convex problem (see [9]). Moreover, by setting $\gamma = 0$, $\gamma = 1$ and $\gamma = 2$ in optimization problem (11) and (12), the least-squares regression model, the lasso regression model, and the ridge regression model are, respectively, obtained.

In the following section, in order to develop the bridge regression model for fuzzy numbers, the aforementioned dataset (b) is considered. Then, a possible approach for solving the problem connected to multicollinearity in an FLR with LR-fuzzy numbers is discussed.

3.1. Proposed Model. Under the assumption of model (10), suppose $A_j = (a_{jm}, a_{jl}, a_{ju})_{\text{LR}}$ and $Y_i = (y_{im}, y_{il}, y_{iu})_{\text{LR}}$ ($i = 1, \dots, n$ and $j = 0, 1, \dots, p$). Let I_{ij} be the identification function which only equals to 1 when $x_{ij} \geq 0$. Using (2) and (3), the LR-fuzzy number \hat{Y}_i is defined as

$$\hat{Y}_i = \left(\sum_{j=0}^p a_{jm} x_{ij}, \sum_{j=0}^p (I_{ij} a_{jl} x_{ij} - (1 - I_{ij}) a_{ju} x_{ij}), \sum_{j=0}^p (I_{ij} a_{ju} x_{ij} - (1 - I_{ij}) a_{jl} x_{ij})_{\text{LR}} \right), \quad \text{for } i = 1, \dots, n \text{ and } j = 0, 1, \dots, p. \quad (13)$$

Now, the proposed MOP for solving the fuzzy bridge regression model consists of three objective functions via the least squares method on the center, and the spreads is presented as follows:

$$\text{minimize } \sum_{i=1}^n \left(y_{im} - \sum_{j=0}^p a_{jm} x_{ij}^2 \right), \quad (14)$$

$$\text{minimize } \sum_{i=1}^n \left(y_{il} - \sum_{j=0}^p (I_{ij} a_{jl} x_{ij} - (1 - I_{ij}) a_{ju} x_{ij}) \right)^2, \quad (15)$$

$$\text{minimize } \sum_{i=1}^n \left(y_{iu} - \sum_{j=0}^p (I_{ij} a_{ju} x_{ij} - (1 - I_{ij}) a_{jl} x_{ij}) \right)^2, \quad (16)$$

subject to

$$\sum_{j=0}^p |a_{jm}|^\gamma \leq t_1, \quad (17)$$

$$\sum_{j=0}^p (a_{jl})^\gamma \leq t_2, \quad a_{jl} \geq 0, \quad (18)$$

$$\sum_{j=0}^p (a_{ju})^\gamma \leq t_3, \quad a_{ju} \geq 0, \quad (19)$$

where t_1, t_2 , and t_3 represent three tuning constants. It should be pointed out that the use of different t_s ($s = 1, 2, 3$) will lead to better results rather than constraining them to be the same throughout the computations [71].

4. An Optimization Model

In order to obtain the estimate for coefficients A_0, A_1, \dots, A_p of the fuzzy bridge regression model, it is necessary to minimize objective functions (14)–(16) simultaneously under constraints (17)–(19). A scalarization problem related to (14)–(19) is stated as

$$\begin{aligned} \text{minimize } C(A) = & w_1 \sum_{i=1}^n \left(y_{im} - \sum_{j=0}^p a_{jm} x_{ij} \right)^2 \\ & + w_2 \sum_{i=1}^n \left(y_{il} - \sum_{j=0}^p (I_{ij} a_{jl} x_{ij} - (1 - I_{ij}) a_{ju} x_{ij}) \right)^2 \\ & + w_3 \sum_{i=1}^n \left(y_{iu} - \sum_{j=0}^p (I_{ij} a_{ju} x_{ij} - (1 - I_{ij}) a_{jl} x_{ij}) \right)^2, \end{aligned} \quad (20)$$

subject to

$$\sum_{j=0}^p |a_{jm}|^\gamma \leq t_1, \quad (21)$$

$$\sum_{j=0}^p (a_{jl})^\gamma \leq t_2, \quad a_{jl} \geq 0, \quad (22)$$

$$\sum_{j=0}^p (a_{ju})^\gamma \leq t_3, \quad a_{ju} \geq 0, \quad (23)$$

where

$$A = (a_{0m}, \dots, a_{pm}, a_{0l}, \dots, a_{pl}, a_{0u}, \dots, a_{pu})^\top, \quad (24)$$

is a $(3p + 3)$ -dimensional column vector, $w_s \geq 0$, ($s = 1, 2, 3$), and $\sum_{s=1}^3 w_s = 1$. To solve problem (20)–(23) we first use the following proposition [72]. Let $u, v \in \mathbb{R}^n$ be auxiliary variables such that

$$x = u - v, \quad u \geq 0, v \geq 0, \quad (25)$$

where $u_i = (x_i)^+$, $v_i = (-x_i)^+$, and $(\cdot)^+$ denotes the positive part defined as $(x)^+ = \max\{0, x\}$. Therefore, in order to model the absolute values in optimization problem (20)–(23), a_{jm} should be split into positive and negative parts such that $a_{jm} = a'_{jm} - a''_{jm}$ with $a'_{jm}, a''_{jm} \geq 0$. Thus, the following quadratic optimization problems for $\gamma \geq 1$ are proposed:

$$\text{minimize } C_{\text{bridge}}(\bar{A}) = \bar{A}^\top \bar{Q} \bar{A} + \bar{B}^\top \bar{A} + M, \quad (26)$$

subject to

$$\bar{g}_q(\bar{A}) \leq 0, \quad q = 1, 2, 3, \quad (27)$$

$$\bar{G} \bar{A} \leq 0, \quad (28)$$

where

$$= (a'_{0m}, \dots, a'_{pm}, a''_{0m}, \dots, a''_{pm}, a_{0l}, \dots, a_{pl}, a_{0u}, \dots, a_{pu})^\top, \quad (29)$$

is a $(4p + 4)$ -dimensional column vector, $w_s \geq 0$, ($s = 1, 2, 3$), and $\sum_{s=1}^3 w_s = 1$. Here,

$$\bar{Q} = \begin{bmatrix} w_1 \mathbf{P}_1 & -2w_1 \mathbf{P}_1 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & w_1 \mathbf{P}_1 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & (w_3 \mathbf{P}_3 + w_2 \mathbf{P}_2) & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & (w_2 \mathbf{P}_3 + w_3 \mathbf{P}_2) \end{bmatrix}, \quad (30)$$

with

$$\mathbf{P}_1 = \begin{bmatrix} \sum_{i=1}^n x_{i0} & \sum_{i=1}^n x_{i1} & \dots & \sum_{i=1}^n x_{ip} \\ \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i1}^2 & \dots & \sum_{i=1}^n x_{i1} x_{ip} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^n x_{ip} & \sum_{i=1}^n x_{ip} x_{i1} & \dots & \sum_{i=1}^n x_{ip}^2 \end{bmatrix}, \quad (31)$$

$$\mathbf{P}_2 = \begin{bmatrix} \sum_{i=1}^n I_{i0}x_{i0} & \sum_{i=1}^n I_{i1}x_{i1} & \cdots & \sum_{i=1}^n I_{ip}x_{ip} \\ \sum_{i=1}^n I_{i1}x_{i1} & \sum_{i=1}^n I_{i1}x_{i1}^2 & \cdots & \sum_{i=1}^n I_{i1}x_{i1}I_{ip}x_{ip} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^n I_{ip}x_{ip} & \sum_{i=1}^n I_{ip}x_{ip}I_{i1}x_{i1} & \cdots & \sum_{i=1}^n I_{ip}x_{ip}^2 \end{bmatrix}, \quad (32)$$

$$\mathbf{P}_3 = \begin{bmatrix} \sum_{i=1}^n (1-I_{i0})x_{i0} & \sum_{i=1}^n (1-I_{i1})x_{i1} & \cdots & \sum_{i=1}^n (1-I_{ip})x_{ip} \\ \sum_{i=1}^n (1-I_{i1})x_{i1} & \sum_{i=1}^n (1-I_{i1})x_{i1}^2 & \cdots & \sum_{i=1}^n (1-I_{i1})x_{i1}(1-I_{ip})x_{ip} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^n (1-I_{ip})x_{ip} & \sum_{i=1}^n (1-I_{ip})x_{ip}(1-I_{i1})x_{i1} & \cdots & \sum_{i=1}^n (1-I_{ip})x_{ip}^2 \end{bmatrix}, \quad (33)$$

where \mathbf{O} is a $(p+1) \times (p+1)$ zeros matrix. $\bar{\mathbf{G}}$ is a $(4p+4) \times (4p+4)$ diagonal matrix with diagonal element (-1) , and $\bar{\mathbf{B}}$ is a column vector:

$$\bar{\mathbf{B}} = (b'_{10}, b'_{11}, \dots, b'_{1p}, b''_{10}, \dots, b''_{1p}, b_{20}, \dots, b_{2p}, b_{30}, \dots, b_{3p})^\top, \quad (34)$$

where

$$\begin{aligned} b'_{1j} &= -2w_1 \sum_{i=1}^n x_{ij}y_{im}, \\ b''_{1j} &= 2w_1 \sum_{i=1}^n x_{ij}y_{im}, \\ b_{2j} &= -2w_2 \sum_{i=1}^n I_{ij}x_{ij}y_{il} + 2w_3 \sum_{i=1}^n (1-I_{ij})x_{ij}y_{iu}, \\ b_{3j} &= -2w_3 \sum_{i=1}^n I_{ij}x_{ij}y_{iu} + 2w_2 \sum_{i=1}^n (1-I_{ij})x_{ij}y_{il}. \end{aligned} \quad (35)$$

Also,

$$\bar{g}_1 = \sum_{j=0}^p (a_{jm}' - a_{jm}'')^y - t_1, \bar{g}_2 = \sum_{j=0}^p (a_{jl})^y - t_2, \bar{g}_3 = \sum_{j=0}^p (a_{ju})^y - t_3, \quad (36)$$

$$M = w_1 \sum_{i=1}^n y_{im}^2 + w_2 \sum_{i=1}^n y_{il}^2 + w_3 \sum_{i=1}^n y_{iu}^2. \quad (37)$$

It can be easily verified that the Hessian matrix of $C_{\text{bridge}}(\bar{\mathbf{A}})$ is positive definite matrix because

$$\nabla^2 C_{\text{bridge}}(\bar{\mathbf{A}}) = \bar{\mathbf{Q}}. \quad (38)$$

Then, problems (26)–(28) are convex quadratic problems with inequality constraints.

4.1. Fuzzy Lasso Regression and Fuzzy Ridge Regression.

In this section, to achieve the two well-known forms of the fuzzy bridge regression model, let $\gamma = 1$ and $\gamma = 2$. Thus, the following quadratic optimization problems are, respectively, obtained as

$$\text{minimize } C_{\text{lasso}}(\bar{\mathbf{A}}) = \bar{\mathbf{A}}^\top \bar{\mathbf{Q}} \bar{\mathbf{A}} + \bar{\mathbf{B}}^\top \bar{\mathbf{A}} + M, \quad (39)$$

subject to

$$\bar{g}_q(\bar{\mathbf{A}}) \leq 0, \quad q = 1, 2, 3, \quad (40)$$

$$\bar{\mathbf{G}} \bar{\mathbf{A}} \leq 0, \quad (41)$$

$$\text{minimize } C_{\text{ridge}}(\mathbf{A}) = \mathbf{A}^\top \mathbf{Q} \mathbf{A} + \mathbf{B}^\top \mathbf{A} + M, \quad (42)$$

subject to

$$g_q(\mathbf{A}) \leq 0, \quad q = 1, 2, 3, \quad (43)$$

$$\mathbf{G} \mathbf{A} \leq 0, \quad (44)$$

where \mathbf{Q} is the following diagonal block matrix:

$$\mathbf{Q} = \begin{bmatrix} w_1 \mathbf{P}_1 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & (w_3 \mathbf{P}_3 + w_2 \mathbf{P}_2) & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & (w_2 \mathbf{P}_3 + w_3 \mathbf{P}_2) \end{bmatrix}, \quad (45)$$

where \mathbf{G} is a $(3p+3) \times (3p+3)$ -dimensional matrix defined as

$$\mathbf{G} = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & -1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & -1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & -1 \end{bmatrix}, \quad (46)$$

\mathbf{O} is a $(p+1) \times (p+1)$ zeros matrix, and $\bar{\mathbf{G}}$ is a $(4p+4) \times (4p+4)$ -diagonal matrix with diagonal element (-1) . B is a column vector:

$$B = (b_{10}, b_{11}, \dots, b_{1p}, b_{20}, \dots, b_{2p}, b_{30}, \dots, b_{3p})^\top, \quad (47)$$

with

$$\begin{aligned} b_{1j} &= -2w_1 \sum_{i=1}^n x_{ij} y_{im}, \\ b_{2j} &= -2w_2 \sum_{i=1}^n I_{ij} x_{ij} y_{il} + 2w_3 \sum_{i=1}^n (1 - I_{ij}) x_{ij} y_{iu}, \\ b_{3j} &= -2w_3 \sum_{i=1}^n I_{ij} x_{ij} y_{iu} + 2w_2 \sum_{i=1}^n (1 - I_{ij}) x_{ij} y_{il}. \end{aligned} \quad (48)$$

Also,

$$\begin{aligned} \bar{g}_1 &= \sum_{j=0}^p (a'_{jm} - a''_{jm}) - t_1, \\ \bar{g}_2 &= \sum_{j=0}^p (a_{jl}) - t_2, \\ \bar{g}_3 &= \sum_{j=0}^p (a_{ju}) - t_3, \\ g_1 &= \sum_{j=0}^p (a_{jm})^2 - t_1, \\ g_2 &= \sum_{j=0}^p (a_{jl})^2 - t_2, \\ g_3 &= \sum_{j=0}^p (a_{ju})^2 - t_3, \end{aligned} \quad (49)$$

where \bar{A} , A , $\bar{\mathbf{Q}}$, \mathbf{P}_1 , \mathbf{P}_2 , \mathbf{P}_3 , \bar{B} , and M are the same as defined in (24), (29)–(34), and (37), respectively.

It can be easily verified that the Hessian matrix of $C_{\text{lasso}}(\bar{A})$ and $C_{\text{ridge}}(A)$ is a positive definite matrix because

$$\nabla^2 C_{\text{lasso}}(\bar{A}) = \bar{\mathbf{Q}}, \quad \nabla^2 C_{\text{ridge}}(A) = \mathbf{Q}. \quad (50)$$

Then, problems (39)–(41) and (42)–(44) are convex quadratic problems with inequality constraints.

There are several optimization methods for solving the convex quadratic problems with inequality constraints that each one of these methods has its weaknesses and strengths. Following Section 5, we proposed a neural network that can approximate coefficients A_0, A_1, \dots, A_p of the fuzzy bridge optimization model.

5. A Neural Network Scheme

In this section, using standard optimization techniques, we transform quadratic programming problem (26)–(28) into the equivalent nonlinear dynamical system model. From [73], $\bar{A}^* \in \mathbb{R}^{4p+4}$ is an optimal solution of (26)–(28) if and only if there exists $\bar{U}^* \in \mathbb{R}^{4p+4}$ such that $((\bar{A}^*)^\top, (\bar{U}^*)^\top)^\top$ satisfies the following KKT system:

$$\begin{cases} \bar{U}^* \geq 0, \\ \bar{\mathbf{G}} \bar{A}^* \leq 0, \quad \bar{g}_q(\bar{A}^*) \leq 0, \quad q = 1, 2, 3, \\ (\bar{U}^*)^\top \bar{\mathbf{G}} \bar{A}^* = 0, \quad (\bar{U}^*)^\top \bar{g}_q(\bar{A}^*) = 0, \quad q = 1, 2, 3, \\ 2\bar{\mathbf{Q}} \bar{A}^* + \bar{B} + \bar{\mathbf{G}}^\top \bar{U}^* + \bar{g}_q^\top(\bar{A}^*) \bar{U}^* = 0, \quad q = 1, 2, 3, \end{cases} \quad (51)$$

where \bar{A}^* is called a KKT point of (26)–(28) and \bar{U}^* is called the Lagrangian multiplier vector corresponding to \bar{A}^* .

Theorem 2 (see [73]). \bar{A}^* is an optimal solution of (26)–(28), if and only if \bar{A}^* is a KKT point of (24)–(26).

Lemma 1. $(\bar{U} + \bar{\mathbf{G}}\bar{A})^+ = \bar{U}$ if and only if $\bar{U} \geq 0, \bar{\mathbf{G}}\bar{A} \leq 0, \bar{U}^\top (\bar{\mathbf{G}}\bar{A}) = 0$, where

$$(\bar{U} + \bar{\mathbf{G}}\bar{A})^+ = ([(\bar{U} + \bar{\mathbf{G}}\bar{A})_1]^+, [(\bar{U} + \bar{\mathbf{G}}\bar{A})_2]^+, \dots, [(\bar{U} + \bar{\mathbf{G}}\bar{A})_{4p+4}]^+)^\top, \quad [(\bar{U} + \bar{\mathbf{G}}\bar{A})_k]^+ = \max\{(\bar{U} + \bar{\mathbf{G}}\bar{A})_k, 0\}, \quad k = 1, 2, \dots, 4p+4. \quad (52)$$

Now, let $\bar{A}(\cdot)$ and $\bar{U}(\cdot)$ be some time-dependent variables. The aim is to construct a continuous-time dynamical system that will settle down to the KKT point of problem (26)–(28). We propose a neural network as

$$\frac{dy}{dt} = \tau\phi(y), \quad (53)$$

$$y(t_0) = y_0 = (\bar{A}(t_0)^\top, \bar{U}(t_0)^\top)^\top \in \mathbb{R}^{8p+8}, \quad \tau > 0, \quad (54)$$

where for $q = 1, 2, 3$,

$$\phi(y) = \left(-\left(2\bar{Q}\bar{A} + \bar{B} + \bar{G}^\top (\bar{U} + \bar{G}\bar{A})^+ + \nabla\bar{g}_q(\bar{A})^\top (\bar{U} + \bar{g}_q(\bar{A}))^+ \right), (\bar{U} + \bar{G}\bar{A})^+ - \bar{U}, (\bar{U} + \bar{g}_q(\bar{A}))^+ - \bar{U} \right), \quad (55)$$

and τ is a tuning parameter. For simplicity of the analysis, let $\tau = 1$.

Theorem 3. Let $y^* = (\bar{A}^{*\top}, \bar{U}^{*\top})^\top$ be the equilibrium point of neural network (53) and (54). Then, A^* is a KKT point of problem (26)–(28). On the other hand, if $\bar{A}^* \in \mathbb{R}^{4p+4}$ is an optimal solution of problem (26)–(28), then there exists $\bar{U}^* \in \mathbb{R}^{4p+4}$ such that $y^* = (\bar{A}^{*\top}, \bar{U}^{*\top})^\top$ is an equilibrium point of network (53) and (54).

In the following, it is proved that the proposed neural network (53) and (54) is stable in the sense of Lyapunov.

Lemma 2. Let $\bar{G} \in \mathbb{R}^{(4p+4) \times (4p+4)}$ be of full rank. The Jacobian matrix $\nabla\phi(y)$ of the mapping ϕ defined in (53) is a negative semidefinite matrix.

Lemma 3. The function $\|(\bar{U} + \bar{G}\bar{A})^+\|^2$ of $(\bar{U} + \bar{G}\bar{A})^+$ defined in (53) is convex and continuously differentiable on $\mathbb{R}^{4p+4} \times \mathbb{R}^{4p+4}$.

Theorem 4. Let the assumption of Lemma 2 be satisfied. Then, the neural network in (53) and (54) is stable in the sense of Lyapunov.

To show the global convergence of the model, the following lemma and theorem are considered.

Lemma 4

- (i) For any initial point $y(t_0) = (\bar{A}(t_0)^\top, \bar{U}(t_0)^\top)^\top$, there exists a unique continuous solution $y(t) = (\bar{A}(t)^\top, \bar{U}(t)^\top)^\top$ for system (53) and (54).
- (ii) Let $y(t) = (\bar{A}(t)^\top, \bar{U}(t)^\top)^\top$ be the state trajectory of (53) and (54) with the initial point $y(t_0) = (\bar{A}(t_0)^\top, \bar{U}(t_0)^\top)^\top$. If $\bar{U}(t_0) \geq 0$, then $\bar{U}(t) \geq 0$.

Theorem 5. The state trajectory of neural network (53) and (54) converges to an equilibrium point for any initial point $y_0 = (\bar{A}(t_0)^\top, \bar{U}(t_0)^\top)^\top \in \mathbb{R}^{8p+8}$. In particular, neural network (53) and (54) with any initial point $y_0 = (\bar{A}(t_0)^\top, \bar{U}(t_0)^\top)^\top \in \mathbb{R}^{8p+8}$ is globally asymptotically stable when D^* (the optimal point set of the original problem and its dual) has a unique equilibrium point.

Theorem 6. The convergence rate of the neural network in (53) and (54) increases as τ increases.

6. The Goodness of Fit

In this section, before focusing on the measurement of goodness of fit for a bridge fuzzy regression model, some popularly used measures are introduced to define the distance between fuzzy numbers. Then, some goodness-of-fit indices are employed, which are represented in [74–77].

Definition 2. Xu [78] presented the following distance between two fuzzy numbers A and B as

$$D_1(A, B) = \left[\int_0^1 f(h) d^2(A(h), B(h)) dh \right]^{1/2}, \quad (56)$$

where

$$\begin{aligned} d^2(A(h), B(h)) &= (A^L(h) - B^L(h))^2 + (A^U(h) - B^U(h))^2, \\ A(h) &= [A^L(h), A^U(h)], \\ B(h) &= [B^L(h), B^U(h)], \end{aligned} \quad (57)$$

and $f(h)$ is an increasing function on $[0, 1]$ with $f(0) = 0$ and $\int_0^1 f(h) dh = 1/2$.

The advantage of this distance is that it can let different h -level cuts play different parts and can be applied to general fuzzy number space.

Definition 3. authors in [48] defined the distance between two LR-fuzzy numbers A and B as follows:

$$\begin{aligned} D_2^2(A, B) &= (a_m - b_m)^2 + ((a_m - \gamma a_l) - (b_m - \gamma b_l))^2 \\ &\quad + ((a_m + \rho a_u) - (b_m + \rho b_u))^2, \end{aligned} \quad (58)$$

where parameters $\gamma = \int_0^1 L^{-1}(\omega) d\omega$ and $\rho = \int_0^1 R^{-1}(\omega) d\omega$ have the twofold role of taking into account the variability of the membership function and decreasing the emphasis on the spreads (see [50]).

Hassanpour et al. [79] defined the distance between two triangular fuzzy numbers as follows.

Definition 4. For two triangular fuzzy numbers A and B , D_3 is called the Hassanpour distance defined by

$$D_3(A, B) = |a_m - b_m| + |a_l - b_l| + |a_u - b_u|. \quad (59)$$

The most important topic in fuzzy regression models is the ability to describe a model. Accordingly, the following goodness-of-fit indices are used in this paper to evaluate the goodness of fit of the proposed model.

6.1. Mean Square of Prediction Error. One of the indicators of goodness-of-fit assessment in the statistical regression model is the mean square of prediction error (MSE) between the predicted and the observed values. In the following, this criterion is presented as a definition.

Definition 5. For regression model (10), the MSE between the estimated and the observed values is defined by

$$MSE = \frac{1}{n} \sum_{i=1}^n D_j^2(\hat{Y}_i, Y_i), \quad j = 1, 2, 3, \quad (60)$$

where $D_j(\hat{Y}_i, Y_i)$ is the distance between two fuzzy numbers \hat{Y}_i and Y_i (see Definitions 2– 4). Smaller values of MSE indicate that the model fits better the data (see [74]).

6.2. Similarity Measure. Several similarity measures are introduced in the literature to measure the similarity between fuzzy sets. In this work, we use the index that has certain advantages over other similarity measures. Such a similarity measure has been used for evaluating the performance of a fuzzy regression model in [75, 76].

Definition 6. Consider fuzzy regression model (10), with fuzzy observed and fuzzy estimated output. The mean of similarity measures (MSM) is defined as

$$MSM = \frac{1}{n} \sum_{i=1}^n S_{UI}(\hat{Y}_i, Y_i), \quad (61)$$

where

$$S_{UI}(\hat{Y}_i, Y_i) = \frac{\text{Card}(\hat{Y}_i \cap Y_i)}{\text{Card}(\hat{Y}_i \cup Y_i)}, \quad (62)$$

and $\text{Card}(A) = \int_x \mu_A(x) dx$ where A is a fuzzy number.

The similarity measure ranges from 0 to 1, and the model with a higher value of MSM provides a better fit to the data.

6.3. Cross Validation. To further investigate the performance of the model obtained in Section 4, we apply another index based on the cross-validation method to examine the predictive ability of the model (see [80]). To this end, each time, the i^{th} observation is left out from the dataset, while the remaining observations are used to develop a fuzzy regression model. Then, the obtained model is used to predict the response value of the i^{th} observation ($\hat{Y}_{(-i)}(x_i)$). Finally, to compare the i^{th} observed Y_i and the predicted value $\hat{Y}_{(-i)}(x_i)$, we calculate the distance D between Y_i and $\hat{Y}_{(-i)}(x_i)$ which is called MDC.

Definition 7. For regression model (10), the index of MDC is defined by

$$MDC = \frac{1}{n} \sum_{i=1}^n D_j^2(\hat{Y}_{(-i)}(x_i), Y_i), \quad j = 1, 2, 3, \quad (63)$$

where $\hat{Y}_{(-i)}(x_i)$ is the fuzzy response prediction by omitting the i^{th} pair (x_i, Y_i) and $D_j(\hat{Y}_i, Y_i)$ is defined in Definitions 2– 4.

To compare performances of different models, the following well-known diagrams can be used to summarize the power of the proposed model to represent the actual values graphically.

6.4. Taylor’s Diagrams. Further to the common performance evaluation criteria, Taylor’s diagrams are depicted to convey information about pattern similarity between estimates and data. This similarity is quantified in terms of the correlation between estimates and data, their centered Root Mean Square (RMS) differences, and their standard deviations. This diagram is suitable for evaluating multiple aspects of complex models. In addition, it is possible to compare the relative performance among different models. The observed dataset is represented by a point on the x -axis as “Actual data.” The estimates are positioned according to three statistics: the radial distance of the point labelled “actual data” from the origin of the plot that is proportional to the standard deviation of each dataset; the RMS difference between data and estimates that is proportional to their distance; and their correlation is given by the azimuthal position of the estimates, i.e. the angle from the x -axis. The formulas for calculating the centered Root Mean Square Error (RMS), the correlation (R), the standard deviation of the observed data (σ_Y), and the standard deviation of the estimated data ($\sigma_{\hat{Y}}$) are given as follows:

$$R = \frac{(1/n) \sum_{i=1}^n (\hat{Y}_i - \bar{\hat{Y}})(Y_i - \bar{Y})}{\sigma_{\hat{Y}} \sigma_Y},$$

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2},$$

$$\sigma_{\hat{Y}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - \bar{\hat{Y}})^2},$$

$$\sigma_Y = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2}, \quad (64)$$

where $\bar{\hat{Y}}$ and \bar{Y} are the calculated mean values of \hat{Y}_i and Y_i , respectively. In Taylor’s diagrams, when the distance to the point representing the actual data is relatively short, there is a good agreement between the estimated and observed data.

6.5. Fuzzy Bubble Plots. Coppi et al. [48] made use of bubble plots to evaluate the fit of the fuzzy regression model with fuzzy output and crisp inputs. The x -axis and the y -axis

denote, respectively, the observed and estimated centers, the circles are centered in (y_{im}, \hat{y}_{im}) , and their diameters are given by $(|y_{il} - \hat{y}_{il}| + |y_{iu} - \hat{y}_{iu}|)^{1/2}$. In bubble plots, the closeness of these circles to the bisector of the plane shows that the model correctly estimates the centers of the fuzzy data, and the smaller diameters of the circles indicate that the model correctly estimates the spreads of the fuzzy data.

7. Numerical Examples

In order to demonstrate the effectiveness of the proposed scheme, in this section, we test some applicable examples by neural network (51) and (52). For testing problems, we also compare the numerical performance of neural network (53) and (54) with various initial points. To illustrate further the effectiveness of the proposed neural network, we make some comparisons with the existing models in the work of Hesamian and Akbari [71], Taheri and Kelkinnama [76], Zeng et al. [81], Choi and Yoon [82], Choi and Buckley [83], and Kula and Apaydin [84] with different distances explained in Section 6 applied to calculate the goodness-of-fit criteria. It must be noted that each method claimed to exhibit better performance compared to other fuzzy multiple regression models. In addition, we present Taylor’s diagrams [77] which provide information on multiple statistical indices to evaluate the accuracy of neural network (53) and (54). The simulation is conducted on Matlab R2017b, and the ordinary differential equation solver engaged is ode45s.

Example 1 (A Simulation Study). To further illustrate the performance of the proposed procedure, we generated a 30 simulated dataset of size $n = 20$ with fuzzy output and crisp inputs. Consider the following linear regression model:

$$\hat{Y}_i = A_0 + A_1x_{i1} + \dots + A_{25}x_{i25} + \varepsilon_i, \tag{65}$$

where for $i = 1, 2, \dots, 20$,

$$x_{ij} = \begin{cases} z_1 + e_1, & \text{for } j = 1, 2, \dots, 5, \\ z_2 + e_2, & \text{for } j = 6, 7, \dots, 10, \\ z_3 + e_3, & \text{for } j = 11, 12, \dots, 15, \\ z_4, & \text{for } j = 16, 17, \dots, 25, \end{cases}$$

$$A_j = \begin{cases} (0, 0, 0)_T, & \text{for } j = 0, \\ (1.5, 0.5, 0.35)_T, & \text{for } j = 1, 2, \dots, 5, \\ (0, 0.5, 0.35)_T, & \text{for } j = 6, 7, \dots, 10, \\ (0, 0.35, 0.5)_T, & \text{for } j = 11, 12, \dots, 15, \\ (0, 0, 0.5)_T, & \text{for } j = 16, 17, \dots, 20, \\ (0, 0, 0)_T, & \text{for } j = 21, 22, \dots, 25, \end{cases} \tag{66}$$

$$\varepsilon_i = (\varepsilon_{im}, \varepsilon_{il}, \varepsilon_{iu})_T, \tag{67}$$

$$\varepsilon_{il} = \varepsilon_{iu} = |\varepsilon_{im}|,$$

with $z_1 \sim N(0, 0.5)$, $z_2 \sim N(0, 0.8)$, $z_3 \sim N(0, 1.2)$, $z_4 = (0, \Sigma)$, $e_1 \sim N(0, 0.01)$, $e_2 \sim N(0, 0.02)$, $e_3 \sim N(0, 0.03)$, $\varepsilon_{im} = N(0, 1.5)$, and

$$[\Sigma]_{lk} = \text{Cov}(x_{jl}, x_{jk}) = 0.5^{|l-k|}. \tag{68}$$

Using the proposed methods for every 30 simulated dataset, the mean of all the employed goodness-of-fit measures (MSM, MSE, and MDC) with respect to the distances D_1 , D_2 , and D_3 is shown in Table 1.

To compare the performance of the proposed models with the existing models in [71, 76, 81, 83], the estimated errors of the observed responses are calculated based on all 30 simulated datasets of size $n = 20$. The numerical results are summarized in Table 1. It can be seen that the proposed fuzzy bridge regression model with different values of γ provides more accurate results.

We utilize the proposed neural network method to the 15th simulated dataset. According to Section 3, the optimal selection of γ increases the efficiency of the model. Thus, to estimate the best value of \hat{Y} using (26)–(28), we need to find the optimal bridge parameter γ . Table 2 shows the corresponding values of MSM, MSE, and MDC for the proposed fuzzy bridge regression model with different values of γ .

The total errors of the proposed fuzzy bridge regression model with $\gamma = 3.7$ are obviously better than the total errors calculated from the methods with $\gamma = 1, 2$, and 3 (see Figure 1).

In addition, Table 3 shows the estimated fuzzy coefficients of the fuzzy bridge regression model for $\gamma = 3.7$, $t_1 = 1669$, $t_2 = 41$, and $t_3 = 38$.

Figures 2–4 show that the trajectories of neural network model (53) and (54) for solving the fuzzy bridge regression model ($\gamma = 3.7$) are convergent to the optimal solution of the problem. Simulation results show the trajectories of (53) and (54) with any initial point are always convergent to the optimal solution of the problem. Taking A_{24} for example, Figure 5 displays the transient behaviors of the fuzzy bridge regression model based on (53) and (54) with 20 random initial points.

The model functioning in the graphical state is illustrated in Figures 6–8, through Taylor’s diagrams.

Figure 6 shows proposed neural network (53) and (54) for the bridge regression model of the centers with $\gamma = 1$, $\gamma = 2$, $\gamma = 3$ and $\gamma = 3.7$, depicted by brown, green, black, and magenta circle, respectively. It is clear that the proposed model with $\gamma = 3.7$ lies very close to the actual dataset. This result holds also for the left and right bounds.

In addition, to visualize how the models fit the observed fuzzy data, fuzzy bubble plots are used (see Figure 9). The fuzzy bubble plots show that observations fit well to the proposed models because of the closeness of the circles to the bisector of the plane and the short radius of each circle.

Therefore, it is concluded that the proposed model used in this paper has certain merits in practice.

Example 2. In this example, the dataset is shown in Table 4 is taken from [85], with crisp inputs and a symmetric triangular fuzzy output.

Using these data, we develop an estimated fuzzy regression equation:

$$\hat{Y} = A_0 + A_1x_1 + A_2x_2 + A_3x_3 + A_4x_4 + A_5x_5. \tag{69}$$

TABLE 1: Mean of 30 simulated performance measures of the proposed fuzzy bridge regression model with different values of γ and some common fuzzy regression techniques in Example 1.

Performance measures	$\overline{\text{MSM}}$			$\overline{\text{MSE}}$			$\overline{\text{MDC}}$		
	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
Proposed fuzzy lasso ($\gamma = 1$)	0.920	0.039	0.146	0.734	1.349	3.046	3.184		
Proposed fuzzy bridge ($\gamma = 1.5$)	0.922	0.039	0.144	0.783	1.318	3.148	3.184		
Proposed fuzzy ridge ($\gamma = 2$)	0.911	0.037	0.156	0.845	1.329	3.136	3.189		
Proposed fuzzy bridge ($\gamma = 3$)	0.922	0.039	0.153	0.782	1.318	3.148	3.184		
The model in [71]	0.805	0.204	3.009	1.225	1.522	3.344	3.976		
The model in [76]	0.607	0.504	5.233	2.130	3.661	5.912	5.059		
The model in [81]	0.654	0.592	5.997	2.363	2.325	5.734	4.370		
The model in [83]	0.721	0.484	4.768	1.760	2.922	4.740	4.446		

TABLE 2: Performance measures of the proposed fuzzy bridge regression model to the 15th simulated dataset with different values of γ in Example 1.

Fuzzy bridge regression model	MSM			MSE			MDC		
γ	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
1	0.93928	0.04310	0.12929	0.79921	1.34901	3.11175	3.18970		
2	0.93821	0.04318	0.12952	0.79921	1.32912	3.13748	3.18974		
3	0.93921	0.04311	0.12952	0.79921	1.31882	3.11100	3.18892		
3.7	0.93940	0.04215	0.12679	0.79862	1.31855	3.11085	3.18892		

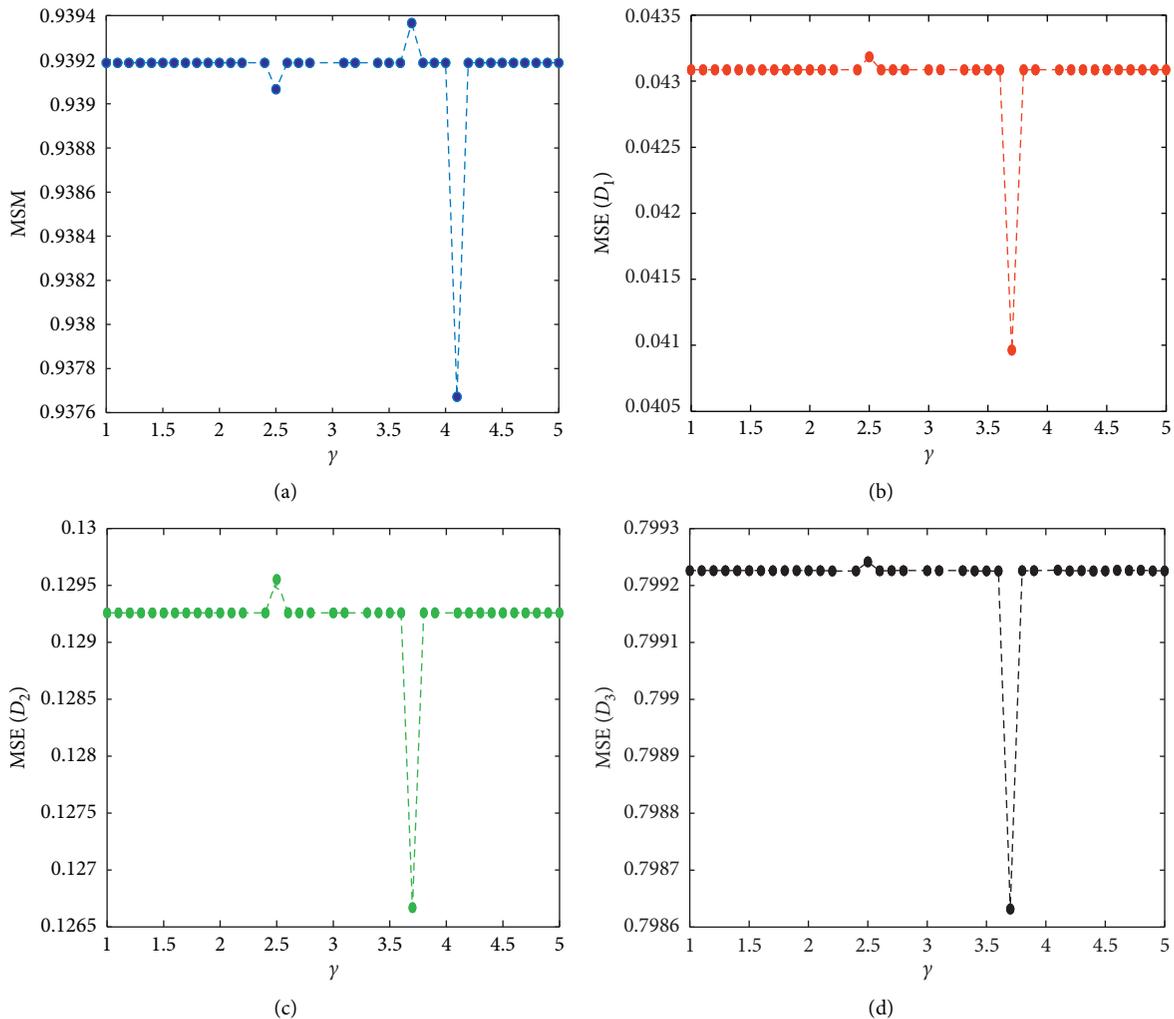


FIGURE 1: Goodness-of-fit measures for different values of γ with three different distances $D_1, D_2,$ and D_3 in Example 1. (a) MSM index, (b) MSE index with distance $D_1,$ (c) MSE index with distance $D_2,$ and (d) MSE index with distance $D_3.$

TABLE 3: Coefficients of fuzzy bridge regression models with $\gamma = 3.7$, corresponding to the 15th simulated dataset in Example 1.

Fuzzy Coefficient	Proposed fuzzy bridge Regression model ($\gamma = 3.7$)
A_0	(-2.2756, 0, 0)
A_1	(-2.8291, 0.6433, 0.9030)
A_2	(3.5032, 0, 0)
A_3	(2.2692, 0, 0)
A_4	(-1.1800, 0, 0)
A_5	(0.2767, 0, 0)
A_6	(-0.3055, 0.3847, 0.2925)
A_7	(0.1711, 0.2676, 0.2585)
A_8	(-1.4283, 0.9777, 0.7293)
A_9	(0.1526, 0.6975, 0.3733)
A_{10}	(0.2140, 2.4168, 2.6125)
A_{11}	(1.4807, 0.3650, 0.3956)
A_{12}	(-0.5725, 0.8036, 0.9055)
A_{13}	(-2.2556, 0.6890, 1.0281)
A_{14}	(1.9892, 0, 0.1928)
A_{15}	(4.2537, 0, 0)
A_{16}	(0.2557, 0.0792, 0.4831)
A_{17}	(0.4384, 0, 0)
A_{18}	(0.7506, 0, 0)
A_{19}	(0.8426, 0, 0.4403)
A_{20}	(2.2863, 0, 0.4810)
A_{21}	(0.0551, 0.0341, 0.1301)
A_{22}	(-0.4957, 0.8162, 0.7252)
A_{23}	(-2.2926, 0, 0)
A_{24}	(0.6620, 0, 0)
A_{25}	(0.0260, 0.0953, 0.5637)

In order to estimate the value of \hat{Y} using (26)–(28), we need to find the optimal bridge parameter γ . Figure 10, considering two different goodness-of-fit measures and distances D_1 , D_2 , and D_3 , illustrates a dramatic rise in the MSM index when $\gamma = 2$. The MSE index with three different distances D_1 , D_2 , and D_3 , on the other hand, shows a different trend. The numbers fall significantly by $\gamma = 2$. Therefore, the optimal fuzzy bridge regression model is obtained using (26) and (28) as

$$\begin{aligned} \hat{Y} = & (-1831.984, 201.7570)_T + (2624.075, 0.7776 \times 10^{-5})_T x_1 \\ & + (82.16977, 0.4976 \times 10^{-7})_T x_2 + (72.51824, 3.6588)_T x_3 \\ & + (-45.344, 0)_T x_4 + (-39.2779, 70.6566)_T x_5, \end{aligned} \quad (70)$$

with $\gamma = 2$, $r_1 = 0.1139 \times 10^8$, $r_2 = 0.3893 \times 10^6$.

Figures 11 and 12 show that the trajectories of model (53) and (54) for solving the fuzzy bridge regression model with $\gamma = 2$ are convergent to the optimal solution of the problem. Simulation results show that the trajectories of (53) and (54) with any initial point are always convergent to the optimal solution of the problem. Figure 13 displays the transient behaviors of fuzzy bridge regression coefficient A_2 based on (53) and (54) with 20 random initial points. Furthermore, the results of several common fuzzy regression models are presented in Table 5.

Comparing the different methods in terms of the goodness-of-fit criteria (MSM, MSE, and MDC, concerning

three distances D_1 , D_2 , and D_3), it can be seen that the proposed method exhibits better performance compared to the other methods. This is clarified in addition in Figures 13–16. In Taylor’s diagrams, a comparison of the proposed fuzzy bridge regression model with the existing models in [71, 76, 81–84] is shown.

The position of each circle on the graph represents a different model result and is determined by the values of the correlation coefficient and standard deviation. It is clear that the proposed model (53) and (54) has a better capability and the most appropriate compared to the other exiting models.

The fuzzy bubble plots show that observations fit well to the selected model because of the closeness of the circles to the bisector of the plane and the short radius of each circle. Therefore, it is concluded that the proposed model used in this paper has a good fitting effect.

In general terms, all figures and the goodness-of-fit criterion in this example indicate that the fuzzy bridge regression model with neural network (53) and (54) has good fitting performance and this model can more accurately describe the original data than the others models.

To end this section, we summarize some advantages of the proposed method as follows:

A suitable neural network for analog hardware implementation is suggested which gives a good solution of the fuzzy bridge regression problems.

The proposed recurrent neural network does not require any penalty parameter for solving the fuzzy regression problems.

The proposed neural network method convergence does not depend on choosing a suitable initial point, since it is globally convergent.

The computational burden can be greatly reduced using the proposed approach compared with some existing methods because it is not needed to calculate the inverse of a matrix (see, for example, [31, 50]) in each iteration in order to obtain weighting coefficients.

In order to evaluate the proposed model, three different distances and goodness-of-fit measures are used. However, some manuscripts use only one distance. For example, one can see [44, 51, 53].

We use a statistical well-known method called “cross validation” and make a statistical comparison with observation through Taylor’s diagram and bubble plots to show the capability of the model.

We provide a simulation and an applicable example through our methodology as Examples 1 and 2. It has been shown that the proposed method is completely capable for solving real-world problems in the fuzzy bridge regression.

8. Conclusions and Future Work

In this paper, a neural network model is presented for solving the fuzzy bridge regression model with fuzzy coefficients and crisp input. To appraise the efficiency of the presented model, a dataset generates through a

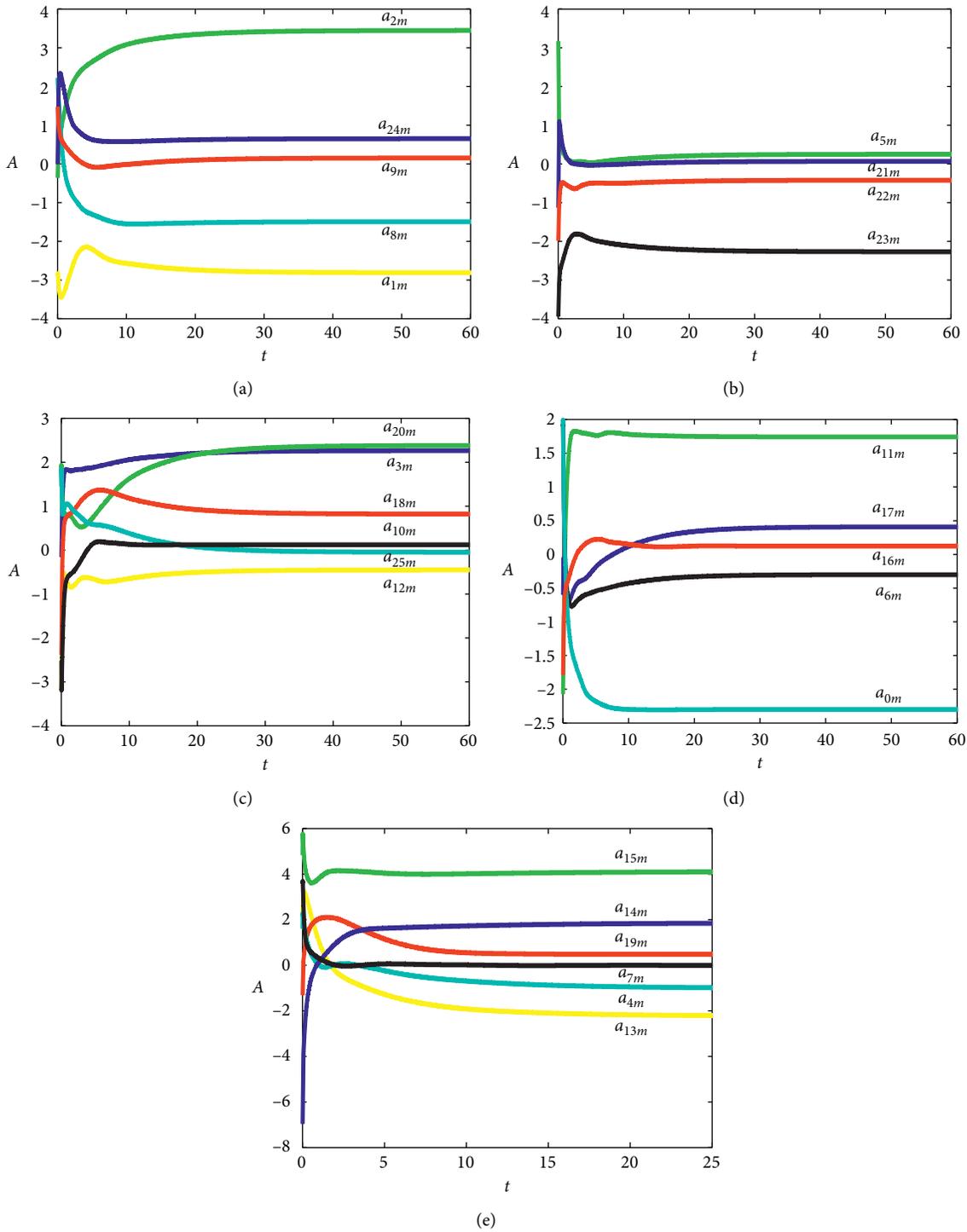


FIGURE 2: Transient behaviors of $a_{0m}, a_{1m}, \dots, a_{25m}$ for neural network (53) and (54) in Example 1.

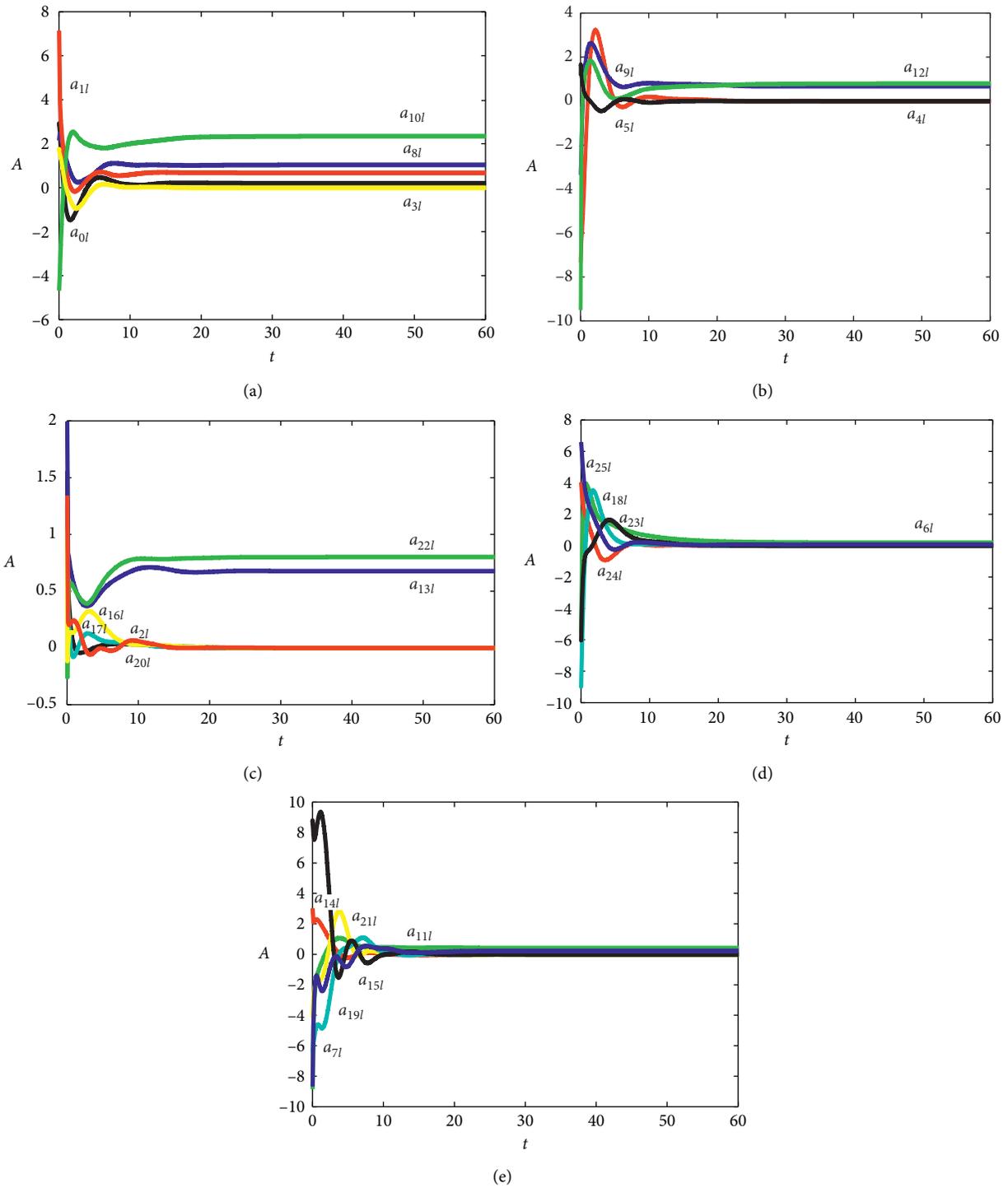


FIGURE 3: Transient behaviors of $a_{0l}, a_{1l}, \dots, a_{25l}$ for neural network (53) and (54) in Example 1.

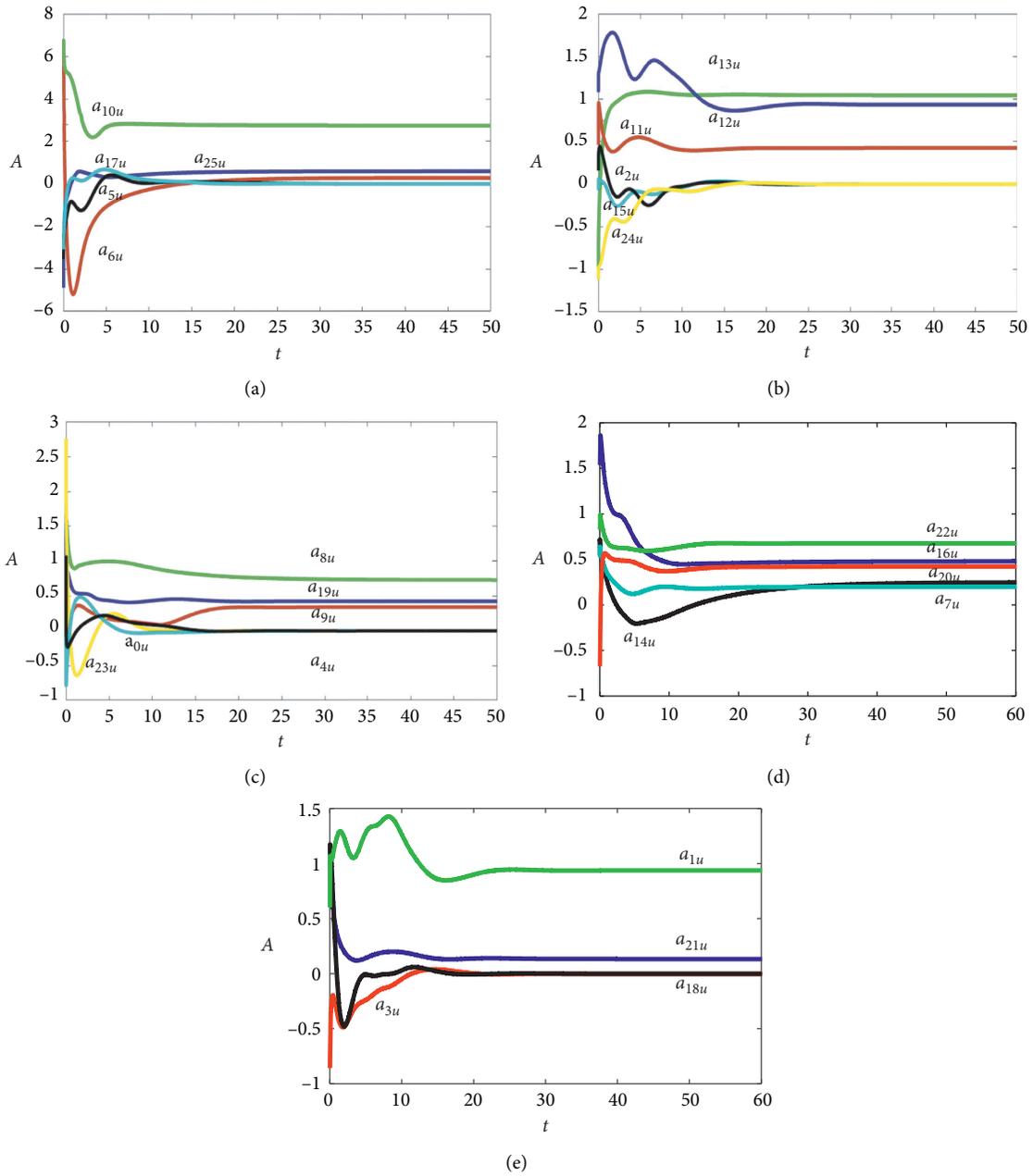


FIGURE 4: Transient behaviors of $a_{0u}, a_{1u}, \dots, a_{25u}$ for neural network (53) and (54) in Example 1.

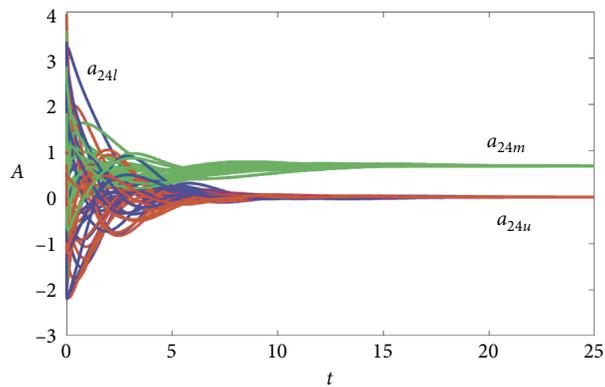


FIGURE 5: Transient behaviors of of fuzzy bridge regression coefficient A_{24} ($\gamma = 3.7$) for neural network (53) and (54) with 20 various initial points in Example 1.

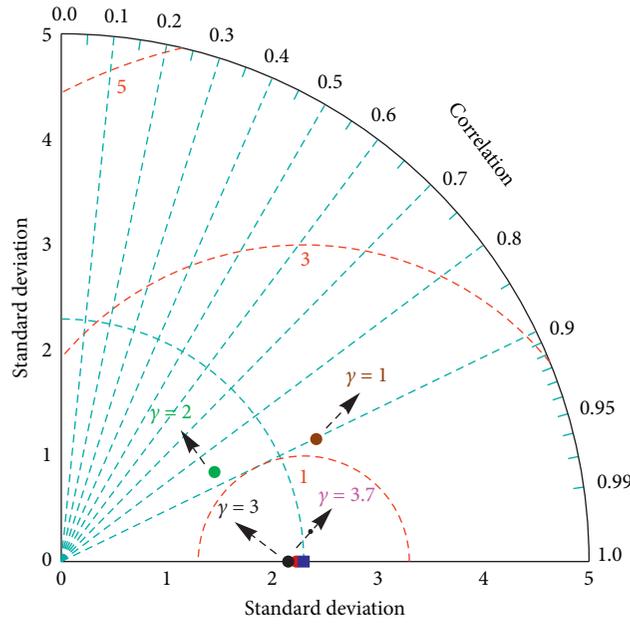


FIGURE 6: Taylor's diagram for the model of the centers in Example 1. The rectangle on the x -axis represents the observations, and the red-dashed contours indicate the centered Root Mean Square Errors.

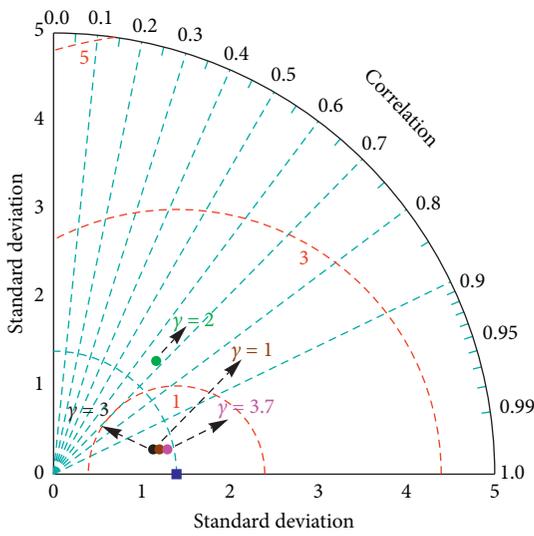


FIGURE 7: Taylor's diagram for the model of the lower bounds in Example 1. The rectangle on the x -axis represents the observations, and the red-dashed contours indicate the centered Root Mean Square Errors.

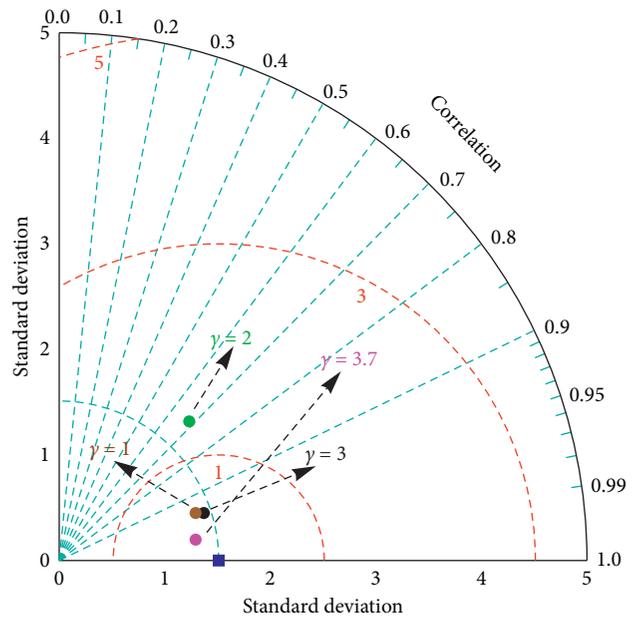


FIGURE 8: Taylor's diagram for the model of the upper bounds in Example 1. The rectangle on the x -axis represents the observations, and the red-dashed contours indicate the centered Root Mean Square Errors.

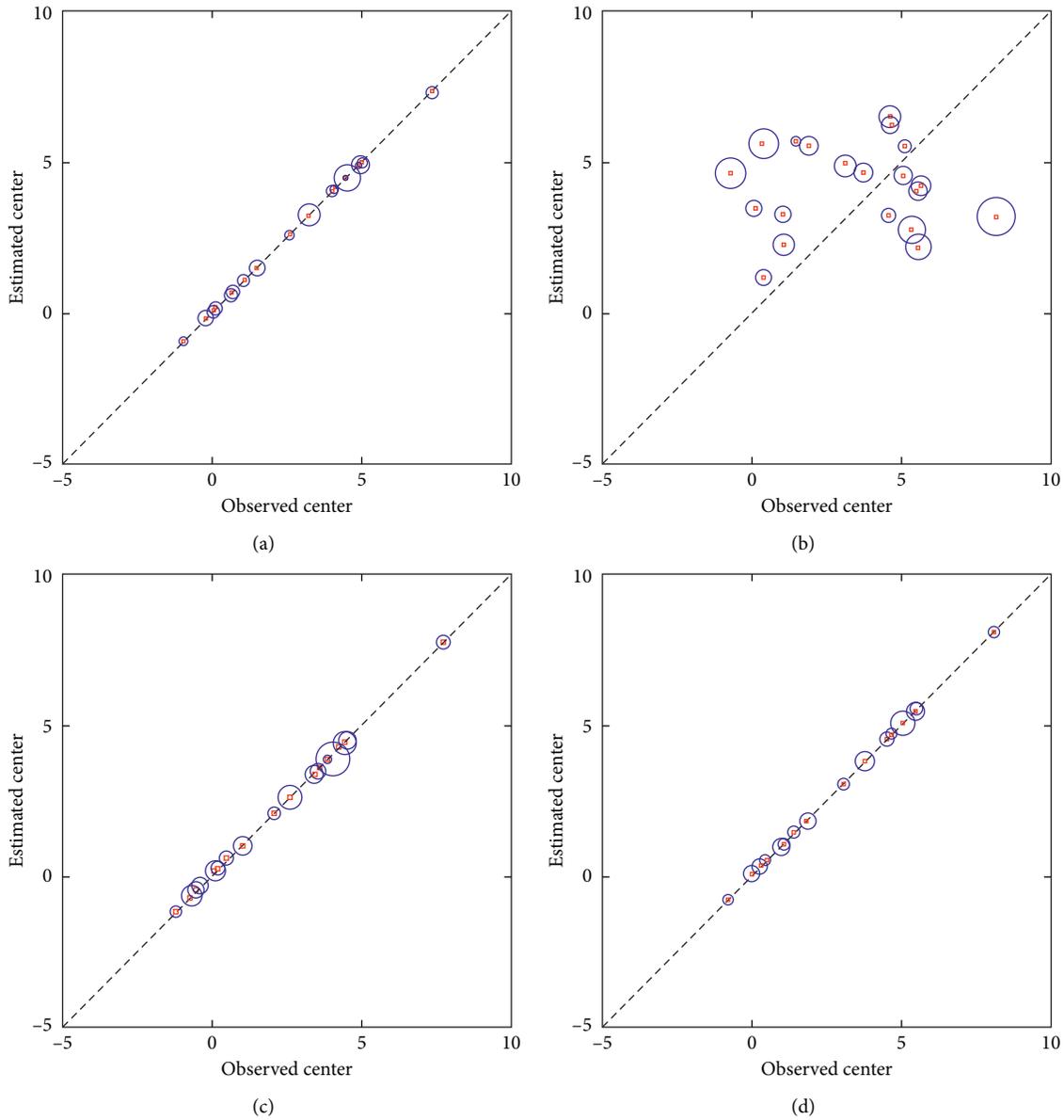


FIGURE 9: Fuzzy bubble plots in Example 1. (a) Proposed fuzzy bridge regression model with $\gamma = 1$, (b) proposed fuzzy bridge regression model with $\gamma = 2$, (c) proposed fuzzy bridge regression model with $\gamma = 3$, and (d) proposed fuzzy bridge regression model with $\gamma = 3.7$.

TABLE 4: Observed and estimated data in Example 2.

No.	x_1	x_2	x_3	x_4	x_5	Y
1	1	38.09	36.43	5	1	(6060, 550)
2	1	62.10	26.50	6	1	(7100, 50)
3	1	63.70	44.71	7	1	(8080, 400)
4	1	74.52	38.09	8	1	(8260, 150)
5	1	75.38	41.40	7	2	(8650, 750)
6	2	52.99	26.49	4	2	(8520, 450)
7	2	62.93	26.49	5	2	(9170, 700)
8	2	72.04	33.12	6	3	(10310, 200)
9	2	76.12	43.06	6	2	(10920, 600)
10	2	90.26	42.64	7	2	(12030, 100)
11	3	85.70	31.33	7	3	(13940, 350)
12	3	95.27	27.64	6	3	(14200, 250)
13	3	105.98	27.64	6	3	(16010, 300)
14	3	79.25	66.81	6	3	(16320, 500)
15	3	120.5	32.25	6	3	(16990, 650)

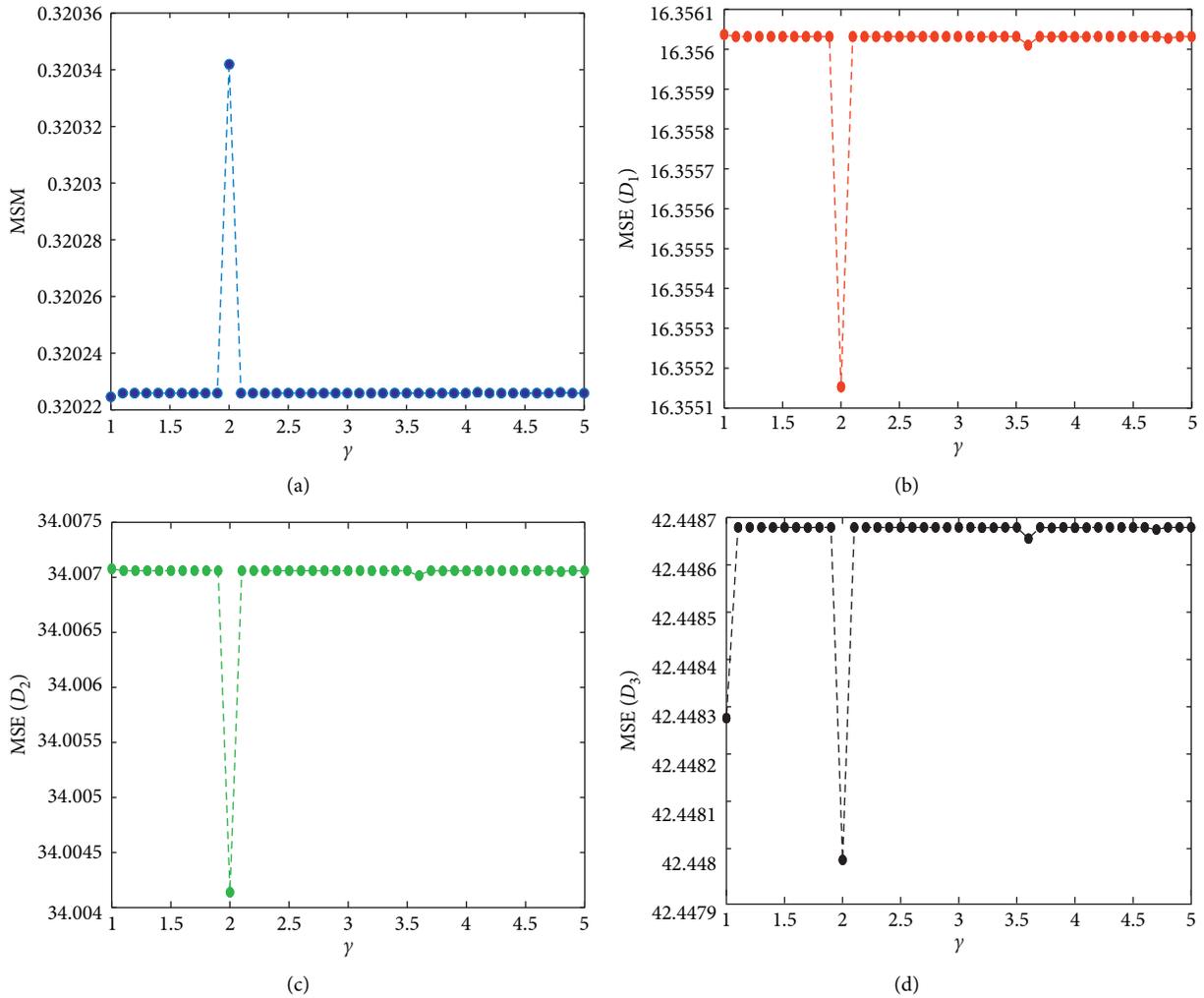


FIGURE 10: Goodness-of-fit measures for different values of γ with three different distances D_1, D_2 , and D_3 in Example 7. (a) MSM index, (b) MSE index with distance D_1 , (c) MSE index with distance D_2 , and (d) MSE index with distance D_3 .

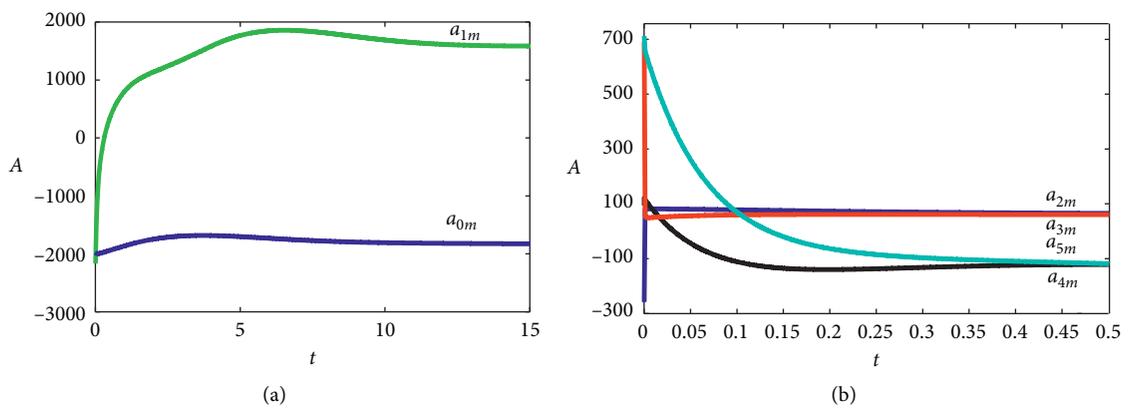


FIGURE 11: Transient behaviors of $a_{0m}, a_{1m}, a_{2m}, a_{3m}, a_{4m}$ and a_{5m} for neural network (53) and (54) in Example 9.

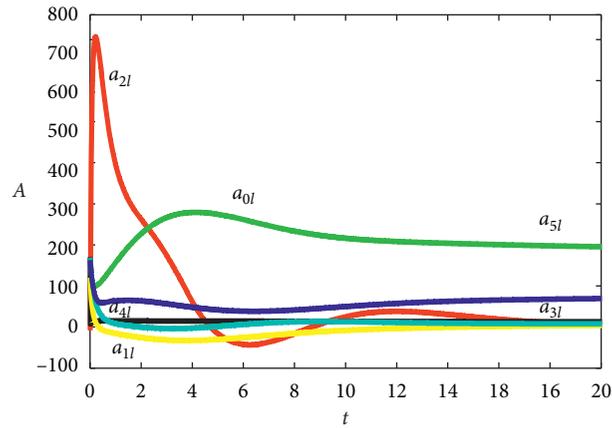


FIGURE 12: Transient behaviors of $a_{0l}, a_{1l}, a_{2l}, a_{3l}, a_{4l}$ and a_{5l} for neural network (53) and (54) in Example 9.

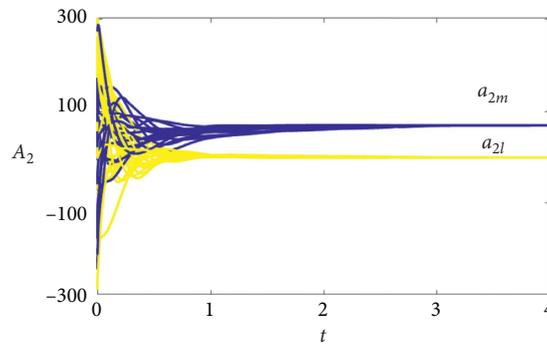


FIGURE 13: Transient behaviors of A_2 for neural network (53) and (54) with 20 various initial points in Example 9.

TABLE 5: Performance measures of the proposed fuzzy bridge regression model for $\gamma = 2$ and some common fuzzy regression models in Example 9.

Performance measures	MSM	MSE			MDC		
		D_1	D_2	D_3	D_1	D_2	D_3
Fuzzy bridge ($\gamma = 2$)	0.3203	16.3551	34.0042	42.4479	27.1788	58.0488	59.8414
The model in [71]	0.1439	67.9434	143.0799	108.8617	46.6452	102.2077	74.0827
The model in [81]	0.2355	33.7791	71.1343	74.2606	29.4994	72.5741	75.7516
The model in [76]	0.1344	44.2797	100.7872	118.9439	53.2842	115.0091	75.5325
The model in [82]	0.2083	33.0820	69.9909	65.2126	38.2911	86.8299	76.1910
The model in [83]	0.1931	44.3149	96.4879	86.6276	39.9299	87.6634	64.3055
The model in [84]	0.1132	42.8039	93.5259	82.4803	59.3054	144.3003	113.2669

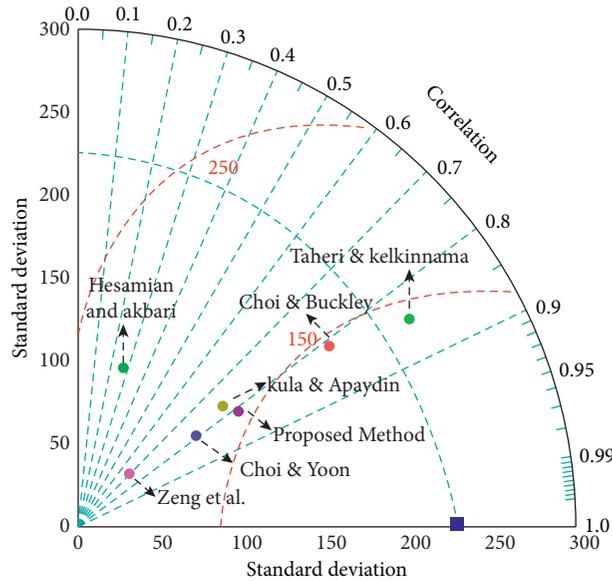


FIGURE 14: Comparison of performances of all models with Taylor’s diagram for the centers in Example 9. The rectangle on the x -axis represents the observations, and the red-dashed contours indicate the centered Root Mean Square Errors.

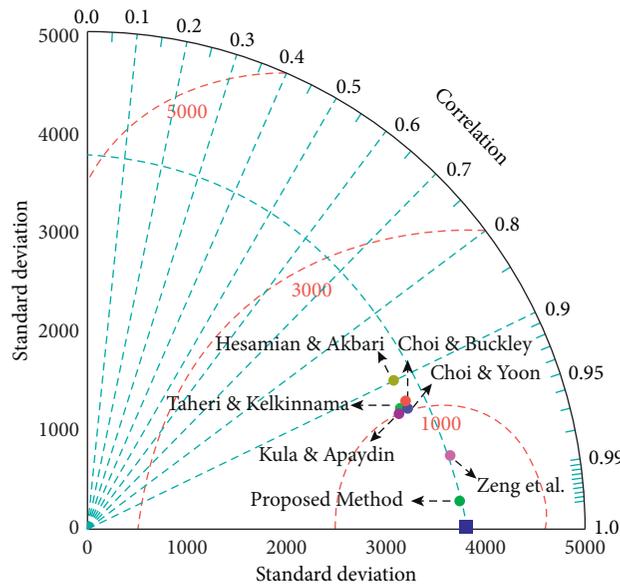


FIGURE 15: Comparison of performances of all models with Taylor’s diagram for the lower bounds in Example 9. The rectangle on the x -axis represents the observations, and the red-dashed contours indicate the centered Root Mean Square Errors.

simulation study in Example 1. The suggested neural network performance is, then, evaluated using the MSM, MSE, and MDC with three different fuzzy distance measures compared to some exiting techniques. As a result, we obtained lesser values for these three measures. To further investigate, and the performance of the given model is achieved using another applicable dataset. These numerical results also show that the model is superior to some

other exiting techniques. In addition, we present the bubble plots to evaluate the accuracy of the introduced model. Also, Taylor’s diagrams which provide information on multiple statistical indices are presented to evaluate the accuracy of the introduced model. As future work, one can apply the proposed scheme to solving fully fuzzy regression, interval-valued regression, and fuzzy type-2 regression problems.

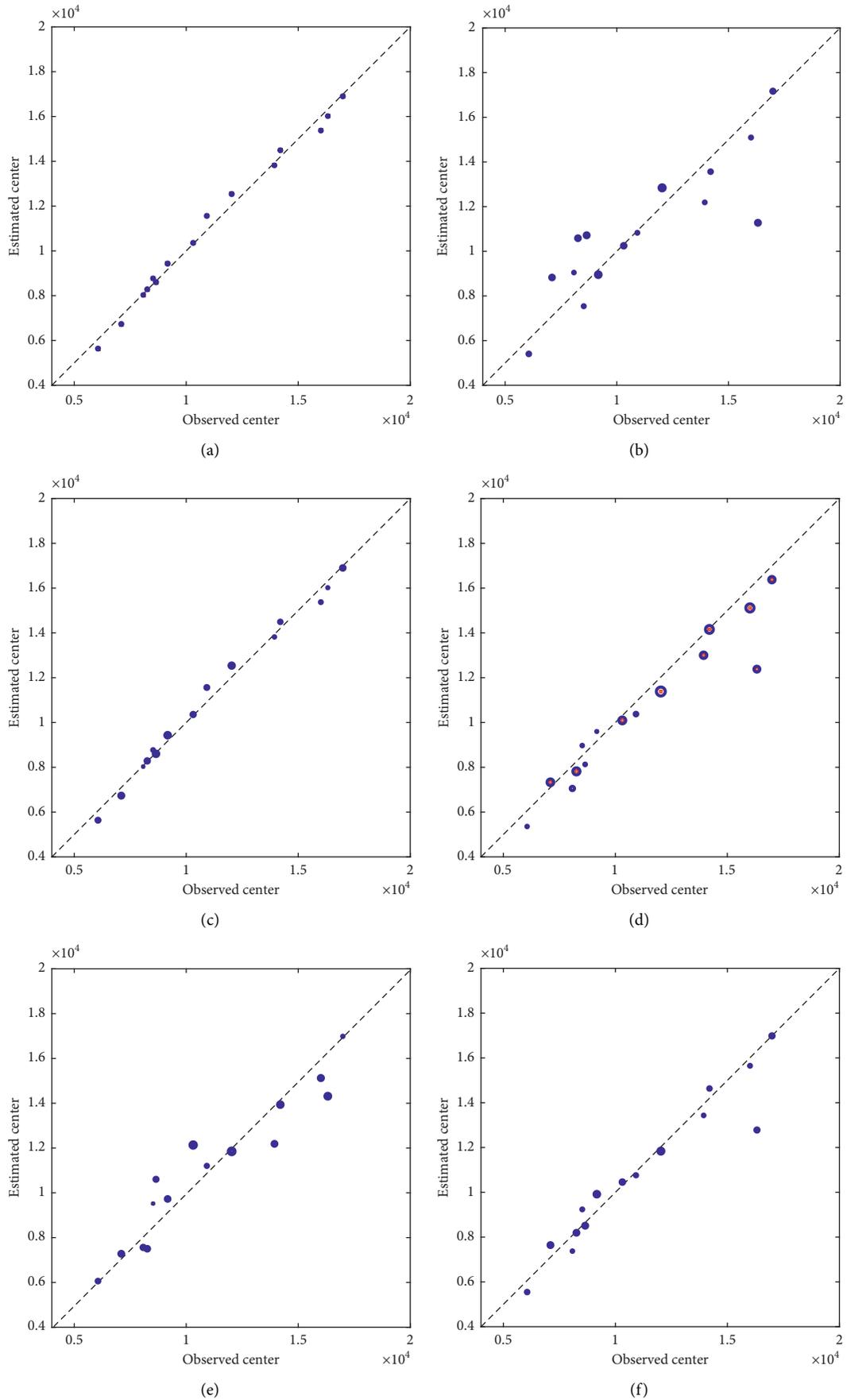


FIGURE 16: Continued.

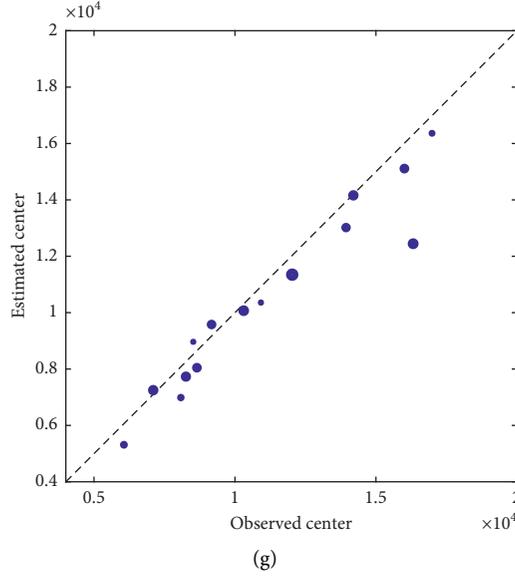


FIGURE 16: Fuzzy bubble plots in Example 9. (a) Proposed fuzzy bridge regression model with $\gamma = 2$, (b) the model in [71], (c) the model in [81], (d) the model in [76], (e) the model in [82], (f) the model in [83], and (g) the model in [84].

Appendix

Here, we sketch the proofs of theoretical results.

Proof (Lemma 1). If $(\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}})^+ = \bar{U}$, then $\bar{U} = 0$ or $\bar{U} > 0$. We have the following cases:

- (a) If $\bar{U} = 0$, then $(\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}})^+ = (\bar{\mathbf{G}}\bar{\mathbf{A}})^+ = 0$; thus, $\bar{\mathbf{G}}\bar{\mathbf{A}} \leq 0$ and $\bar{U}^\top (\bar{\mathbf{G}}\bar{\mathbf{A}}) = 0$
- (b) If $\bar{U} > 0$, then $(\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}})^+ = \bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}} = \bar{U}$; thus, $\bar{\mathbf{G}}\bar{\mathbf{A}} = 0$ and $\bar{U}^\top (\bar{\mathbf{G}}\bar{\mathbf{A}}) = 0$

The converse is straightforward. □

Proof (Theorem 3). Let $\mathbf{y}^* = (\bar{\mathbf{A}}^{*\top}, \bar{U}^{*\top})^\top$ be the equilibrium of (53) and (54). Then, $d\bar{\mathbf{A}}^*/dt = 0$ and $d\bar{U}^*/dt = 0$. It follows easily that

$$(\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}}) = \left((\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}})_1, (\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}})_2, \dots, (\bar{U} + \bar{\mathbf{G}}\bar{\mathbf{A}})_n, \underbrace{0, 0, \dots, 0}_{(4p+4)-n} \right)^\top. \tag{A.5}$$

With a simple calculation, it is clearly shown that

$$\nabla\phi(\mathbf{y}) = \left(-\left((\bar{\mathbf{Q}} + (\bar{\mathbf{G}}^*)^\top \bar{\mathbf{G}}^*) - (\bar{\mathbf{G}}^*)^\top \bar{\mathbf{G}}^* W_{(4p+4) \times (4p+4)} \right), \right) \tag{A.6}$$

$$\bar{\mathbf{Q}}\bar{\mathbf{A}}^* + \bar{\mathbf{B}} + \bar{\mathbf{G}}^\top (\bar{U}^* + \bar{\mathbf{G}}\bar{\mathbf{A}}^*)^+ = 0, \tag{A.1}$$

$$(\bar{U}^* + \bar{\mathbf{G}}\bar{\mathbf{A}}^*)^+ - \bar{U}^* = 0. \tag{A.2}$$

From Lemma 1, $(\bar{U}^* + \bar{\mathbf{G}}\bar{\mathbf{A}}^*)^+ = \bar{U}^*$ if and only if

$$\bar{U}^* \geq 0, \bar{\mathbf{G}}\bar{\mathbf{A}}^* \leq 0, \bar{U}^{*\top} (\bar{\mathbf{G}}\bar{\mathbf{A}}^*) = 0. \tag{A.3}$$

Moreover, substituting (A.2) into (A.1), we have

$$\bar{\mathbf{Q}}\bar{\mathbf{A}}^* + \bar{\mathbf{B}} + \bar{\mathbf{G}}^\top \bar{U}^* = 0. \tag{A.4}$$

From (A.2) and (A.4), it is seen that $\mathbf{y}^* = (\bar{\mathbf{A}}^{*\top}, \bar{U}^{*\top})^\top$ satisfies KKT conditions (51). The converse is straightforward. □

Proof (Lemma 2). Without loss of generality, assume that there exists $0 < n < (4p + 4)$ such that

where

$$\overline{\mathbf{G}}^* = \begin{pmatrix} \mathbf{R}_{n \times (4p+4)} \\ \mathbf{O}_{(4p+4-n) \times (4p+4)} \end{pmatrix} = \begin{pmatrix} \overline{\mathbf{G}}_1 \\ \overline{\mathbf{G}}_2 \\ \dots \\ \overline{\mathbf{G}}_n \\ \mathbf{O}_{1 \times (4p+4)} \\ \mathbf{O}_{1 \times (4p+4)} \\ \dots \\ \mathbf{O}_{1 \times (4p+4)} \end{pmatrix}, \quad (\text{A.7})$$

$$W_{(4p+4) \times (4p+4)} = \begin{pmatrix} \mathbf{O}_{n \times n} & \mathbf{O}_{n \times (4p+4-n)} \\ \mathbf{O}_{(4p+4-n) \times n} & -I_{(4p+4-n) \times (4p+4-n)} \end{pmatrix},$$

and \mathbf{O} indicates a zero matrix. From [86], we see that $(\overline{\mathbf{G}}^*)^\top \overline{\mathbf{G}}^*$ is a positive semidefinite matrix. Matrix $\overline{\mathbf{Q}}$ is also assumed to be positive semidefinite. Moreover, it is clear that matrix $W_{(4p+4) \times (4p+4)}$ is negative semidefinite. As a result, the Jacobian matrix $\nabla \phi(\mathbf{y})$ is a negative semidefinite matrix.

If $n = 4p + 4$, i.e., $(\overline{\mathbf{U}} + \overline{\mathbf{GA}})^+ = ((\overline{\mathbf{U}} + \overline{\mathbf{GA}})_1, (\overline{\mathbf{U}} + \overline{\mathbf{GA}})_2, \dots, (\overline{\mathbf{U}} + \overline{\mathbf{GA}})_{4p+4})^\top$, then

$$\nabla \phi(\mathbf{y}) = -(\overline{\mathbf{Q}} + \overline{\mathbf{G}}^\top \overline{\mathbf{G}}) - \overline{\mathbf{G}}^\top \overline{\mathbf{Q}}_{(4p+4) \times (4p+4)}. \quad (\text{A.8})$$

Similar to the previous case, it is easily proved that $\nabla \phi(\mathbf{y})$ is a negative semidefinite matrix.

Finally, if $n = 0$, i.e., $(\overline{\mathbf{U}} + \overline{\mathbf{GA}})^+ = (0, 0, \dots, 0_{4p+4})^\top$, then we have

$$\nabla \phi(\mathbf{y}) = (-\overline{\mathbf{Q}}_{(4p+4) \times (4p+4)} \mathbf{O}_{(4p+4) \times (4p+4)} - I_{(4p+4) \times (4p+4)}). \quad (\text{A.9})$$

In this case also, it is easy to verify that $\nabla \phi(\mathbf{y})$ is a negative semidefinite matrix. This completes the proof. \square

Proof (Lemma 3). Obviously, $\|(\overline{\mathbf{U}} + \overline{\mathbf{GA}})^+\|^2 = \sum_{k=1}^{4p+4} ((\overline{\mathbf{U}} + \overline{\mathbf{GA}})_k^+)^2$ and

$$((\overline{\mathbf{U}} + \overline{\mathbf{GA}})_k^+)^2 = \begin{cases} ((\overline{\mathbf{U}} + \overline{\mathbf{GA}})_k)^2, & \text{if } (\overline{\mathbf{U}} + \overline{\mathbf{GA}})_k \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad \text{for } k = 1, \dots, 4p + 4. \quad (\text{A.10})$$

Then, according to [87], the result is obtained from the differentiable convexity of $(\overline{\mathbf{GA}})_k$ ($k = 1, \dots, 4p + 4$). We also have

$$\nabla \left(\|(\overline{\mathbf{U}} + \overline{\mathbf{GA}})^+\|^2 \right) = [2\overline{\mathbf{G}}^\top (\overline{\mathbf{U}} + \overline{\mathbf{GA}})^+ 2(\overline{\mathbf{U}} + \overline{\mathbf{GA}})^+]. \quad (\text{A.11})$$

This completes the proof. \square

Proof (Theorem 4). Consider the Lyapunov function $E: \mathbb{R}^{8p+8} \rightarrow \mathbb{R}$ as follows:

$$E(\mathbf{y}) = E_1(\mathbf{y}) + E_2(\mathbf{y}), \quad (\text{A.12})$$

where $E_1(\mathbf{y}) = \|\phi(\mathbf{y})\|^2$ and $E_2(\mathbf{y}) = (1/2)\|\mathbf{y} - \mathbf{y}^*\|^2$. From optimization literature [88] and Lemma 3, we know that $E_1(\mathbf{y})$ is a differentiable function. From (55), it is seen that

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial \mathbf{y}} \frac{d\mathbf{y}}{dt} = \nabla \phi(\mathbf{y}) \phi(\mathbf{y}). \quad (\text{A.13})$$

Calculating the derivative of $E(t)$ along the solution $\mathbf{y}(t)$ of neural network (53) and (54), we have

$$\frac{dE(\mathbf{y}(t))}{dt} = \left(\frac{d\phi}{dt} \right)^\top \phi + \phi^\top \left(\frac{d\phi}{dt} \right) + (\mathbf{y} - \mathbf{y}^*)^\top \frac{d\mathbf{y}(t)}{dt} = \phi^\top (\nabla \phi(\mathbf{y})^\top + \nabla \phi(\mathbf{y})) \phi + (\mathbf{y} - \mathbf{y}^*)^\top \phi(\mathbf{y}). \quad (\text{A.14})$$

Employing Lemma 2, we attain

$$\phi^\top (\mathbf{y}) (\nabla \phi(\mathbf{y})^\top + \nabla \phi(\mathbf{y})) \phi(\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \neq \mathbf{y}^*. \quad (\text{A.15})$$

Moreover, from Definition 2.2 and Lemma 2.3 in [64], we have

$$(\mathbf{y} - \mathbf{y}^*)^\top (\phi(\mathbf{y}) - \phi(\mathbf{y}^*)) = (\mathbf{y} - \mathbf{y}^*)^\top \phi(\mathbf{y}) \leq 0, \quad \forall \mathbf{y} \neq \mathbf{y}^*. \quad (\text{A.16})$$

Thus,

$$\frac{dE(\mathbf{y}(t))}{dt} \leq 0. \quad (\text{A.17})$$

This means that the proposed neural network is stable in the sense of Lyapunov. \square

Proof (Lemma 4)

- (i) It is easy to verify that $\{\overline{\mathbf{Q}}\mathbf{A} + \overline{\mathbf{B}} + \overline{\mathbf{G}}^\top (\overline{\mathbf{U}} + \overline{\mathbf{G}}\mathbf{A})^+\}$ and $\{(\overline{\mathbf{U}} + \overline{\mathbf{G}}\mathbf{A})^+ - \overline{\mathbf{U}}\}$ are locally Lipschitz continuous on an open convex set $D \subseteq \mathbb{R}^{8p+8}$. According to the local existence of ordinary differential equations in [89], neural network (53) and (54) has a unique continuous solution $\mathbf{y}(t)$, $t \in [t_0, \eta]$ for some $\eta > t_0$.

By the proof of Theorem 5, we know that E is a nonincreasing function with respect to t , so

$$\frac{1}{2} \|\mathbf{y} - \mathbf{y}^*\|^2 \leq E(\mathbf{y}(t)) \leq E(\mathbf{y}(t_0)), \quad \forall t \geq t_0. \quad (\text{A.18})$$

This shows that the state trajectory of neural network (53) and (54) is bounded. Thus, $\eta \rightarrow +\infty$.

- (ii) For the given initial point $\mathbf{y}(t_0)$ with $\overline{\mathbf{U}}(t_0) \geq 0$, we have

$$\frac{d\overline{\mathbf{U}}}{dt} + \overline{\mathbf{U}} = (\overline{\mathbf{U}} + \overline{\mathbf{G}}\mathbf{A})^+, \int_{t_0}^t \left(\frac{d\overline{\mathbf{U}}}{dt} + \overline{\mathbf{U}} \right) e^s ds = \int_{t_0}^t e^s (\overline{\mathbf{U}} + \overline{\mathbf{G}}\mathbf{A})^+ ds. \quad (\text{A.19})$$

It follows

$$\overline{\mathbf{U}}(t) = e^{-(t-t_0)} \overline{\mathbf{U}}(t_0) + e^{-t} \int_{t_0}^t e^s (\overline{\mathbf{U}} + \overline{\mathbf{G}}\mathbf{A})^+ ds. \quad (\text{A.20})$$

Since $(\overline{\mathbf{U}} + \overline{\mathbf{G}}\mathbf{A})^+ \geq 0, \overline{\mathbf{U}}(t) \geq 0$ for any $t \geq t_0$. \square

Proof (Theorem 5). From the proof of Lemma 5, we have that the state trajectory

$$((\overline{\mathbf{A}}(t))^\top, \overline{\mathbf{U}}(t)^\top)^\top \quad (\text{A.21})$$

of neural network (53) and (54) is bounded. Therefore, there exists an increasing sequence $\{t_k\}$ with $t_k \rightarrow \infty$ as $k \rightarrow \infty$, and a limit point $(\widehat{\mathbf{A}}^\top, \widehat{\mathbf{U}}^\top)^\top$ such that

$$\lim_{k \rightarrow \infty} (\overline{\mathbf{A}}(t_k)^\top, \overline{\mathbf{U}}(t_k)^\top)^\top = (\widehat{\mathbf{A}}^\top, \widehat{\mathbf{U}}^\top)^\top. \quad (\text{A.22})$$

Using the LaSalle invariant set theorem [90], one has that $\{(\overline{\mathbf{A}}(t)^\top, \overline{\mathbf{U}}(t)^\top)^\top \rightarrow H\}$ as $t \rightarrow \infty$, where H is the largest invariant set in $K = \{(\overline{\mathbf{A}}(t)^\top, \overline{\mathbf{U}}(t)^\top)^\top | (dE(\mathbf{y}(t))/dt) = 0\}$. From (53), (54), and (A.17), it follows that $d\overline{\mathbf{A}}/dt = 0$, and $d\overline{\mathbf{U}}/dt = 0 \Leftrightarrow dE(\mathbf{y}(t))/dt = 0$. Thus, $(\widehat{\mathbf{A}}^\top, \widehat{\mathbf{U}}^\top)^\top \in D^*$ by $H \subseteq K \subseteq D^*$.

Secondly, we prove the state trajectory $(\overline{\mathbf{A}}(t)^\top, \overline{\mathbf{U}}(t)^\top)^\top$ globally converges to the equilibrium point $(\widehat{\mathbf{A}}^\top, \widehat{\mathbf{U}}^\top)^\top$. We define another Lyapunov function

$$\overline{E}(\mathbf{y}) = \|\phi(\mathbf{y})\|^2 + \frac{1}{2} \|\mathbf{y} - \overline{\mathbf{y}}\|^2, \quad (\text{A.23})$$

where we substitute $\overline{\mathbf{A}}^* = \widehat{\mathbf{A}}$ and $\overline{\mathbf{U}}^* = \widehat{\mathbf{U}}$ in (A.12). Then, $\overline{E}(\mathbf{y})$ is continuously differentiable and $\overline{E}(\overline{\mathbf{y}}) = 0$. Note that $\lim_{k \rightarrow \infty} (\overline{\mathbf{A}}(t_k)^\top, \overline{\mathbf{U}}(t_k)^\top)^\top = (\widehat{\mathbf{A}}^\top, \widehat{\mathbf{U}}^\top)^\top$. We, therefore, have $\lim_{k \rightarrow \infty} \overline{E}(\overline{\mathbf{A}}(t_k)^\top, \overline{\mathbf{U}}(t_k)^\top)^\top = \overline{E}(\widehat{\mathbf{A}}(h)^\top, \widehat{\mathbf{U}}^\top)^\top$. So, $\forall \varepsilon > 0$ there exists $q > 0$ such that for all $t \geq t_q$, we have $\overline{E}(\mathbf{y}(t)) < \varepsilon$. Similarly, we can obtain $(d\overline{E}(\mathbf{y}(t))/dt) \leq 0$. It follows that, for $t \geq t_q$,

$$\frac{1}{2} \|\mathbf{y}(t) - \overline{\mathbf{y}}\|^2 \leq \overline{E}(\mathbf{y}(t)) \leq \varepsilon. \quad (\text{A.24})$$

It follows that $\lim_{t \rightarrow \infty} \|\mathbf{y}(t) - \overline{\mathbf{y}}\| = 0$ and $\lim_{t \rightarrow \infty} \mathbf{y}(t) = \overline{\mathbf{y}}$. Therefore, the network in (53) and (54) is globally convergent to an equilibrium point $\overline{\mathbf{y}} = (\widehat{\mathbf{A}}^\top, \widehat{\mathbf{U}}^\top)^\top$, where $\widehat{\mathbf{A}}$ is the optimal solution of (26)–(28).

In particular, if $D^* = \{(\overline{\mathbf{A}}^{*\top}, \overline{\mathbf{U}}^{*\top})^\top\}$, then the state trajectory $(\overline{\mathbf{A}}(t)^\top, \overline{\mathbf{U}}(t)^\top)^\top$ with any initial point $(\overline{\mathbf{A}}(t_0)^\top, \overline{\mathbf{U}}(t_0)^\top)^\top$ is globally asymptotically stable to the unique equilibrium point $\mathbf{y}^* = (\overline{\mathbf{A}}^{*\top}, \overline{\mathbf{U}}^{*\top})^\top$. \square

Proof (Theorem 6). From (66) and (A.17), it can be seen that, for any $\tau > 0$ in (53) and (54),

$$\frac{dE(\mathbf{y})}{dt} \leq \tau \phi(\mathbf{y})^\top (\nabla \phi(\mathbf{y})^\top + \nabla \phi(\mathbf{y})) \phi(\mathbf{y}) \leq 0, \quad (\text{A.25})$$

$$E(\mathbf{y}(t)) \leq E(\mathbf{y}(t_0)) + \tau \int_{t_0}^t \phi(\mathbf{y}(s))^\top (\nabla \phi(\mathbf{y}(s))^\top + \nabla \phi(\mathbf{y}(s))) \phi(\mathbf{y}(s)) ds.$$

Since $E(\mathbf{y}) \geq (1/2) \|\mathbf{y} - \mathbf{y}^*\|^2$, where \mathbf{y}^* is a KKT point of (26) and (28), we get

$$\|\mathbf{y}(t) - \mathbf{y}^*\|^2 \leq 2E(\mathbf{y}(t_0)) + 2\tau \int_{t_0}^t \phi(\mathbf{y}(s))^\top (\nabla \phi(\mathbf{y}(s))^\top + \nabla \phi(\mathbf{y}(s))) \phi(\mathbf{y}(s)) ds. \quad (\text{A.26})$$

Therefore, the convergence rate of the trajectory $y(t)$ increases as τ increases. \square

Data Availability

The data for Example 1 are simulated, and the data for Example 2 are taken from [85].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [2] C. L. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1974.
- [3] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.
- [4] M. Roozbeh and M. Arashi, "Shrinkage ridge regression in partial linear models," *Communications in Statistics—Theory and Methods*, vol. 45, no. 20, pp. 6022–6044, 2016.
- [5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
- [6] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.
- [7] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.
- [8] T. Park and G. Casella, "The bayesian lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.
- [9] K. Knight and W. Fu, "Asymptotics for lasso-type estimators," *Annals of Statistics*, vol. 28, pp. 1356–1378, 2000.
- [10] H. Zou and M. Yuan, "The f-norm support vector machine," *Statistica Sinica*, vol. 18, pp. 379–398, 2008.
- [11] J. Huang, J. L. Horowitz, S. Ma et al., "Asymptotic properties of bridge estimators in sparse high-dimensional regression models," *The Annals of Statistics*, vol. 36, no. 2, pp. 587–613, 2008.
- [12] H. Zou and H. H. Zhang, "On the adaptive elastic-net with a diverging number of parameters," *The Annals of Statistics*, vol. 37, no. 4, p. 1733, 2009.
- [13] Z. Xu, H. Zhang, Y. Wang, X. Chang, and Y. Liang, "L 1/2 regularization," *Science China Information Sciences*, vol. 53, no. 6, pp. 1159–1169, 2010.
- [14] C. Park and Y. J. Yoon, "Bridge regression: adaptivity and group selection," *Journal of Statistical Planning and Inference*, vol. 141, no. 11, pp. 3506–3519, 2011.
- [15] N. G. Polson, J. G. Scott, and J. Windle, "The bayesian bridge," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 4, pp. 713–733, 2014.
- [16] H. Mallick and N. Yi, "Bayesian bridge regression," *Journal of Applied Statistics*, vol. 45, no. 6, pp. 988–1008, 2018.
- [17] Y. Tian and X. Song, "Bayesian bridge-randomized penalized quantile regression," *Computational Statistics & Data Analysis*, vol. 144, Article ID 106876, 2020.
- [18] H. Asai, S. Tanaka, and K. Uegima, "Linear regression analysis with fuzzy model," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 12, no. 6, pp. 903–907, 1982.
- [19] M. Khashei, S. Reza Hejazi, and M. Bijari, "A new hybrid artificial neural networks and fuzzy regression model for time series forecasting," *Fuzzy Sets and Systems*, vol. 159, no. 7, pp. 769–786, 2008.
- [20] A. Chaudhuri and K. De, "Time series forecasting using hybrid neuro-fuzzy regression model," in *Proceedings of the International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, pp. 369–381, Springer, Delhi, India, December 2009.
- [21] A. Azadeh, M. Saberi, and O. Seraj, "An integrated fuzzy regression algorithm for energy consumption estimation with non-stationary data: a case study of Iran," *Energy*, vol. 35, no. 6, pp. 2351–2366, 2010.
- [22] A. Azadeh, M. Saberi, and A. Gitiforouz, "An integrated simulation-based fuzzy regression-time series algorithm for electricity consumption estimation with non-stationary data," *Journal of the Chinese Institute of Engineers*, vol. 34, no. 8, pp. 1047–1066, 2011.
- [23] E. Ciavolino and A. Calcagni, "A generalized maximum entropy (GME) estimation approach to fuzzy regression model," *Applied Soft Computing*, vol. 38, pp. 51–63, 2016.
- [24] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular fuzzy regression domain adaptation in takagi-sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 847–858, 2017.
- [25] M. B. Ferraro, "On the generalization performance of a regression model with imprecise elements," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 5, pp. 723–740, 2017.
- [26] Z. S. Zolfaghari, M. Mohebbi, and M. Najariyan, "Application of fuzzy linear regression method for sensory evaluation of fried donut," *Applied Soft Computing*, vol. 22, pp. 417–423, 2014.
- [27] M. R. Rabiei, M. Arashi, and M. Farrokhi, "Fuzzy ridge regression with fuzzy input and output," *Soft Computing*, vol. 23, no. 23, pp. 12189–12198, 2019.
- [28] S. H. Choi, H.-Y. Jung, and H. Kim, "Ridge fuzzy regression model," *International Journal of Fuzzy Systems*, vol. 21, no. 7, pp. 2077–2090, 2019.
- [29] T. Wang, P. Shi, and G. Wang, "Solving fuzzy regression equation and its approximation for random fuzzy variable and their application," *Soft Computing*, vol. 24, no. 2, pp. 919–933, 2020.
- [30] L.-H. Chen and S.-H. Nien, "Mathematical programming approach to formulate intuitionistic fuzzy regression model based on least absolute deviations," *Fuzzy Optimization and Decision Making*, vol. 19, pp. 191–210, 2020.
- [31] Y. Gong, S. Yang, H. Ma, and M. Ge, "Fuzzy regression model based on incentre distance and application to employee performance evaluation," *International Journal of Fuzzy Systems*, vol. 20, no. 8, pp. 2632–2639, 2018.
- [32] M. Arefi, "Quantile fuzzy regression based on fuzzy outputs and fuzzy parameters," *Soft Computing*, vol. 24, no. 1, pp. 311–320, 2020.
- [33] T. Murata, H. Ishibuchi, and H. Tanaka, "Multi-objective genetic algorithm and its applications to flowshop scheduling," *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 957–968, 1996.
- [34] R. M. Dawes and B. Corrigan, "Linear models in decision making," *Psychological Bulletin*, vol. 81, no. 2, p. 95, 1974.
- [35] N. Gunantara, "A review of multi-objective optimization: methods and its applications," *Cogent Engineering*, vol. 5, no. 1, Article ID 1502242, 2018.

- [36] P.-T. Chang, "Fuzzy seasonality forecasting," *Fuzzy Sets and Systems*, vol. 90, no. 1, pp. 1–10, 1997.
- [37] F.-M. Tseng and G.-H. Tzeng, "A fuzzy seasonal arima model for forecasting," *Fuzzy Sets and Systems*, vol. 126, no. 3, pp. 367–376, 2002.
- [38] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fuzzy linear regression models with least square errors," *Applied Mathematics and Computation*, vol. 163, no. 2, pp. 977–989, 2005.
- [39] H. Ishibuchi, K. Kwon, and H. Tanaka, "A learning algorithm of fuzzy neural networks with triangular fuzzy weights," *Fuzzy Sets and Systems*, vol. 71, no. 3, pp. 277–293, 1995.
- [40] Y. Hayashi, J. J. Buckley, and E. Czogala, "Fuzzy neural network with fuzzy signals and weights," *International Journal of Intelligent Systems*, vol. 8, no. 4, pp. 527–537, 1993.
- [41] J. J. Buckley and E. Eslami, "Neural net solutions to fuzzy problems: the quadratic equation," *Fuzzy Sets and Systems*, vol. 86, no. 3, pp. 289–298, 1997.
- [42] S. Abbasbandy and M. Otadi, "Numerical solution of fuzzy polynomials by fuzzy neural network," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1084–1089, 2006.
- [43] S. Abbasbandy, M. Otadi, and M. Mosleh, "Numerical solution of a system of fuzzy polynomials by fuzzy neural network," *Information Sciences*, vol. 178, no. 8, pp. 1948–1960, 2008.
- [44] M. Mosleh, M. Otadi, and S. Abbasbandy, "Evaluation of fuzzy regression models by fuzzy neural network," *Journal of Computational and Applied Mathematics*, vol. 234, no. 3, pp. 825–834, 2010.
- [45] Y.-J. Lai and S. I. Chang, "A fuzzy approach for multiresponse optimization: an off-line quality engineering problem," *Fuzzy Sets and Systems*, vol. 63, no. 2, pp. 117–129, 1994.
- [46] E. Nasrabadi and S. M. Hashemi, "Robust fuzzy regression analysis using neural networks," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. 4, pp. 579–598, 2008.
- [47] M. Modarres, E. Nasrabadi, and M. M. Nasrabadi, "Fuzzy linear regression analysis from the point of view risk," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 12, no. 5, pp. 635–649, 2004.
- [48] R. Coppi, P. D'Urso, P. Giordani, and A. Santoro, "Least squares estimation of a linear regression model with lr fuzzy response," *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 267–286, 2006.
- [49] P. D'Urso and A. Santoro, "Fuzzy clusterwise linear regression analysis with symmetrical fuzzy output variable," *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 287–313, 2006.
- [50] P. D'Urso, R. Massari, and A. Santoro, "Robust fuzzy regression analysis," *Information Sciences*, vol. 181, no. 19, pp. 4154–4174, 2011.
- [51] M. Mosleh, M. Otadi, and S. Abbasbandy, "Fuzzy polynomial regression with fuzzy neural networks," *Applied Mathematical Modelling*, vol. 35, no. 11, pp. 5400–5412, 2011.
- [52] M. Mosleh, T. Allahviranloo, and M. Otadi, "Evaluation of fully fuzzy regression models by fuzzy neural network," *Neural Computing and Applications*, vol. 21, no. 1, pp. 105–112, 2012.
- [53] M. Otadi, "Fully fuzzy polynomial regression with fuzzy neural networks," *Neurocomputing*, vol. 142, pp. 486–493, 2014.
- [54] S.-B. Roh, T.-C. Ahn, and W. Pedrycz, "Fuzzy linear regression based on polynomial neural networks," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8909–8928, 2012.
- [55] Y.-L. He, X.-Z. Wang, and J. Z. Huang, "Fuzzy nonlinear regression analysis using a random weight network," *Information Sciences*, vol. 364–365, pp. 222–240, 2016.
- [56] Y.-L. He, C.-H. Wei, H. Long, R. A. Raza Ashfaq, and J. Z. Huang, "Random weight network-based fuzzy nonlinear regression for trapezoidal fuzzy number data," *Applied Soft Computing*, vol. 70, pp. 959–979, 2018.
- [57] N. Y. Pehlivan and A. Apaydin, "Fuzzy radial basis function network for fuzzy regression with fuzzy input and fuzzy output," *Complex & Intelligent Systems*, vol. 2, no. 1, pp. 61–73, 2016.
- [58] H.-t. Liu, J. Wang, Y.-L. He, and R. A. R. Ashfaq, "Extreme learning machine with fuzzy input and fuzzy output for fuzzy regression," *Neural Computing and Applications*, vol. 28, no. 11, pp. 3465–3476, 2017.
- [59] S. Yeylaghi, M. Otadi, and N. Imankhan, "A new fuzzy regression model based on interval-valued fuzzy neural network and its applications to management," *Beni-Suef University Journal of Basic and Applied Sciences*, vol. 6, no. 2, pp. 106–111, 2017.
- [60] N. Chukhrova and A. Johannssen, "Fuzzy regression analysis: systematic review and bibliography," *Applied Soft Computing*, vol. 84, Article ID 105708, 2019.
- [61] H. Fazlollahabadi and H. Gholizadeh, "Fuzzy possibility regression integrated with fuzzy adaptive neural network for predicting and optimizing electrical discharge machining parameters," *Computers & Industrial Engineering*, vol. 140, Article ID 106225, 2020.
- [62] D. Karbasi, A. Nazemi, and M. Rabiei, "A parametric recurrent neural network scheme for solving a class of fuzzy regression models with some real-world applications," *Soft Computing*, vol. 24, pp. 11159–11187, 2020.
- [63] J.-S. Chen, C.-H. Ko, and S. Pan, "A neural network based on the generalized Fischer-Burmeister function for nonlinear complementarity problems," *Information Sciences*, vol. 180, no. 5, pp. 697–711, 2010.
- [64] A. R. Nazemi, "A dynamic system model for solving convex nonlinear optimization problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 4, pp. 1696–1705, 2012.
- [65] A. Nazemi, "Solving general convex nonlinear optimization problems by an efficient neurodynamic model," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 685–696, 2013.
- [66] A. Nazemi and S. Effati, "An application of a merit function for solving convex programming problems," *Computers & Industrial Engineering*, vol. 66, no. 2, pp. 212–221, 2013.
- [67] A. Nazemi and M. Nazemi, "A gradient-based neural network method for solving strictly convex quadratic programming problems," *Cognitive Computation*, vol. 6, no. 3, pp. 484–495, 2014.
- [68] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [69] A. M. Jubril, "A nonlinear weights selection in weighted sum for convex multiobjective optimization," *Facta Universitatis*, vol. 27, no. 3, pp. 357–372, 2012.
- [70] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multi-objective Optimization*, Elsevier, Amsterdam, Netherlands, 1985.
- [71] G. Hesamian and M. G. Akbari, "Fuzzy lasso regression model with exact explanatory variables and fuzzy responses," *International Journal of Approximate Reasoning*, vol. 115, pp. 290–300, 2019.
- [72] M. Mohammadi, Y.-H. Tan, W. Hofman, and S. H. Mousavi, "A novel one-layer recurrent neural network for the 11-

- regularized least square problem,” *Neurocomputing*, vol. 315, pp. 135–144, 2018.
- [73] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2013.
- [74] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and their Application*, Vol. 1, Cambridge University Press, Cambridge, UK, 1997.
- [75] J. Lu and R. Wang, “An enhanced fuzzy linear regression model with more flexible spreads,” *Fuzzy Sets and Systems*, vol. 160, no. 17, pp. 2505–2523, 2009.
- [76] S. M. Taheri and M. Kelkinnama, “Fuzzy linear regression based on least absolute deviations,” *Iranian Journal of Fuzzy Systems*, vol. 9, no. 1, pp. 121–140, 2012.
- [77] K. E. Taylor, “Summarizing multiple aspects of model performance in a single diagram,” *Journal of Geophysical Research: Atmospheres*, vol. 106, no. D7, pp. 7183–7192, 2001.
- [78] R. Xu, “A linear regression model in fuzzy environment,” *Advance in Modelling Simulation*, vol. 27, pp. 31–40, 1991.
- [79] H. Hassanpour, H. R. Maleki, and M. Yaghoobi, “Fuzzy linear regression model with crisp coefficients: a goal programming approach,” *Iranian Journal of Fuzzy Systems*, vol. 7, no. 2, pp. 1–153, 2010.
- [80] L. Wasserman, *All of Nonparametric Statistics*, Springer Science & Business Media, Berlin, Germany, 2006.
- [81] W. Zeng, Q. Feng, and J. Li, “Fuzzy least absolute linear regression,” *Applied Soft Computing*, vol. 52, pp. 1009–1019, 2017.
- [82] S. H. Choi and J. H. Yoon, “General fuzzy regression using least squares method,” *International Journal of Systems Science*, vol. 41, no. 5, pp. 477–485, 2010.
- [83] S. H. Choi and J. J. Buckley, “Fuzzy regression using least absolute deviation estimators,” *Soft Computing*, vol. 12, no. 3, pp. 257–263, 2008.
- [84] K. Ş. Kula and A. Apaydin, “Fuzzy robust regression analysis based on the ranking of fuzzy sets,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. 5, pp. 663–681, 2008.
- [85] H. V. Henderson and P. F. Velleman, “Building multiple regression models interactively,” *Biometrics*, vol. 37, no. 2, pp. 391–411, 1981.
- [86] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Vol. 37, Springer Science & Business Media, Berlin, Germany, 2010.
- [87] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Courier Corporation, Chelmsford, MA, USA, 2003.
- [88] M. Fukushima, “Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems,” *Mathematical Programming*, vol. 53, no. 1–3, pp. 99–110, 1992.
- [89] R. Miller and A. N. Michel, *Ordinary Differential Equations*, Academic Press, Cambridge, MA, USA, 1982.
- [90] J. Hale, *Ordinary Differential Equations*, Michigan State University, East Lansing, MI, USA, 1969.